

Web Accessible Knowledge-Based Course and Classroom Scheduling Management

Pamilerin Vivian Aribisogan

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
May 2019
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Hadi Işık Aybay
Chair, Department of Computer
Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Zeki Bayram
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Zeki Bayram

2. Asst. Prof. Dr. Yiltan Bitirim

3. Asst. Prof. Dr. Shahin Mehdiphour Ataee

ABSTRACT

This thesis examines Course and Classroom scheduling problems. Course and Classroom Scheduling is a significant procedure in any educational institution. It is an open-ended problem in which courses must be assigned to classrooms and timeslots, meeting some set of constraints. Allocating classrooms and timeslots to courses employing a manual technique requires a lot time and human resource to deliver a good result that works absolutely without clashes.

We develop a web interface to an existing knowledge-based course scheduler system to unravel this drawback. Our method reads through a web interface information from the user regarding courses, classrooms and course opening request for a specific semester. This data is stored in an SQL database, which is used to generate input files for Flora-2. The course scheduling program is written in Flora-2 and the Flora-2 interpreter is accessed through its JAVA interface to run the scheduler. The result of the scheduler is obtained through the JAVA interface again, stored in the database and presented to the user.

Keywords: Scheduling, Timetable, Constraint, Frame-Logic, Knowledge Representation.

ÖZ

Bu tezde ders ve sınıfların zamanlaması problemi ele alınmaktadır. Ders ve sınıf zamanlaması, her eğitim kurumunda, belli kısıtları gözeterek derslerin zaman dilimlerine ve sınıflara atanmasını gerektiren önemli bir işlemdir. Elde yapılan bir teknikle tüm kısıtları sağlayan iyi bir neticeye varmak çok zaman ve insan kaynağı gerektirir.

Bu sorunla başa çıkmak için Flora-2 dilinde yazılmış, daha önceden varolan bir bilgi-tabanlı zamanlama sistemine bir web arayüzü geliştirdik. Yöntemimiz, web arayüzü vasıtası ile belli bir dönem için kullanıcıdan dersler, sınıf odaları ve ders açma istekleri bilgilerini alır. Bu veri SQL veritabanında depolanır, ve zamanlayıcının ihtiyacı olan girdi dosyalarının üretilmesinde kullanılır. Flora-2 yorumlayıcısına JAVA arayüzü vasıtası ile erişilip zamanlayıcı çalıştırılır. Zamanlayıcının neticesi yine JAVA arayüzü ile elde edilir, veri tabanında depolanır, ve kullanıcıya sunulur.

Anahtar Kelimeler: Zamanlama, Zaman Tablosu, Kısıt, Çerçeve Mantığı, Bilgi Temsiliyeti

DEDICATION

I dedicate this thesis to God, my wonderful Parents, Friends and Family, who has always been a constant source of support and encouragement to my life. Also to myself Pamilerin Vivian Aribisogan, thank you for not giving up, when you thought it was hard, thank you for daring to ride the storm.

ACKNOWLEDGEMENT

Special thanks to God for his Grace and giving me strength and ability to understand, learn and complete this thesis.

My sincere gratitude goes to my supervisor Assoc. Prof Dr. Zeki Bayram for his patience, help, motivation, supervision toward the finishing of this work; and special thanks to Sr. Instr. Cem Yağlı for his esteemed support towards the completion of this project.

Special appreciation goes to Mr John Olaifa and Mr Hussaini Habib, thank you for teaching, correcting and explaining, believing, pushing me to be a better person, making my whole Computer Engineering years worthwhile.

A big thank you to my family for their invaluable support and encouragement throughout my studies.

To my amazing friends that stood by me, May God Almighty bless you all.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ.....	iv
DEDICATION.....	v
ACKNOWLEDGEMENT	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
1 INTRODUCTION	1
1.1 Background to the Study.....	1
1.2 Aim and Objective of the Study	2
1.3 Problem with the Current System.....	2
1.4 Significance of the Study	2
1.5 Scope of the Study	2
1.6 Limitations of the Study.....	3
1.7 Requirements.....	3
1.7.1 Operating System	3
1.7.2 RAM	3
1.7.3 Disk Space	3
1.8 Thesis Overview.....	3
2 REVIEW OF RELATED WORKS	5
2.1 Timetable Solution Generation Algorithms.....	6
2.1.1 Linear Programming.....	6
2.1.2 Genetic Algorithms	6

2.1.3 Simulated Annealing	7
2.1.4 Tabu Search.....	8
2.1.5 Tiling Algorithms	8
2.1.6 Constraint Satisfaction Programming.....	8
2.1.7 Direct Heuristics.....	9
2.1.8 Logic Programming.....	9
2.1.9 Particle Swam Optimization	10
2.2 Summary.....	11
3 PROBLEM SPECIFICATION AND PROPOSED SOLUTION.....	12
3.1 Constraints	12
3.1.1 Hard Constraints.....	12
3.1.2 Soft Constraints	13
3.2 Features of the System.....	13
3.3 Timetable Generation Method	14
3.4 Web Based Front End and Manager to the Course Scheduler.....	14
4 ANALYSIS AND DESIGN	16
4.1 System Analysis	16
4.2 Work Breakdown Structure	16
4.3 Flow within the Project.....	17
4.4 Functional Requirements	17
4.4.1 Use-case	17
4.5 Non Functional Requirements	18
4.5.1 Usability Requirement.....	18
4.5.2 Efficiency Requirement	18
4.5.3 Reliability Requirement.....	18

4.5.4 Robustness	18
4.5.5 Flexibility	19
4.6 Proposed System	19
4.7 System Design.....	20
4.7.1 Class Diagram	20
4.7.2 Sequence Diagram.....	21
4.8 Database Design	25
4.8.1 E-R Diagram	25
4.8.2 Relational Schema	27
4.8.3 Normalized Tables.....	28
4.8.4 MYSQL	29
4.9 Output Design	29
4.10 Input Design.....	29
5 IMPLEMENTATION	31
5.1 Tools and Technologies Used	31
5.1.1 Flora 2.....	31
5.1.2 Java	31
5.1.3 NetBeans	32
5.2 Algorithm Implementation and Programming	32
5.3 User Manual.....	38
5.3.1 Login Section	39
5.3.2 Homepage	39
5.3.3 Course Page.....	40
5.3.4 Classroom Page	42
5.3.5 Course Opening Page	42

6 TESTING AND SAMPLE SCHEDULE GENERATION	46
6.1 Testing Strategy Employed.....	46
6.1.1 User Interface Testing.....	46
6.1.2 Interface testing mechanism.....	46
6.2 Quality Dimension of the System	47
6.3 Using the System for Schedule Generation	48
6.3.1 Sample Run	48
7 CONCLUSION.....	55
REFERENCES	56
APPENDICES	60
Appendix A: Source Code.....	61
Appendix B: Database.....	124
Appendix C: Sample generated Flora-2 files as input to the scheduler	132

LIST OF TABLES

Table 1: List of course opening requests for Fall 2019	49
Table 2: List of rooms available in computer engineering department	49
Table 3: List of Computer Engineering courses	50

LIST OF FIGURES

Figure 4.1: The Work Break Down Structure	16
Figure 4.2: The Flow within the project	17
Figure 4.3: Use-case Diagram of the Scheduling System.....	18
Figure 4.4: Three Tier Architecture of the System.....	20
Figure 4.5: The Class Diagram of the Timetable Generation System.	20
Figure 4.6: Sequence Diagram for the login Page.....	21
Figure 4.7: The Sequence Diagram for Adding a course opening for a specific semester.....	22
Figure 4.8: The Sequence Diagram for adding a course.....	22
Figure 4.9: The Sequence Diagram for Adding a Room.	23
Figure 4.10: The Sequence Diagram for editing a course opening for a specific semester.....	23
Figure 4.11: The Sequence Diagram for editing room.	24
Figure 4.12: The Sequence Diagram for Deleting.....	24
Figure 4.13: The Homepage of the System.....	40
Figure 4.14: The Add Course Page.....	41
Figure 4.15: The Add Classroom Page.	42
Figure 4.16: The Add Course Opening Page	45

Chapter 1

INTRODUCTION

The purpose of this study is to provide a web-based and user friendly front-end to a knowledge-based course scheduler, thus eliminating the need for manual timetabling in a university setting. The work contained in this thesis not only summarizes the literature but also contributes to the objective comparison of future research. This chapter begins with a discussion of the ideas and general analysis of the work done. Specifically, the usage and classification of the general timetable problem are highlighted. General trends during this timetable literature are mentioned.

1.1 Background to the Study

This research is aimed at developing a web accessible course and classroom scheduling management system centered on a knowledge-based course scheduler that lacks management features and a practical interface. Apart from knowing the number of courses, the number of students, the capacity of the classroom must be put into consideration. The system is designed to make scheduling of classroom and courses more convenient and efficient. Users can immediately check and publish a timetable. It is a web application, in which java servlet page was used as the front end, MYSQL was the database management system, and Flora-2 is the back end application implementation language used.

1.2 Aim and Objective of the Study

The aim and objective of this project are to generate an automatic timetable eliminating the use of manual effort in order to achieve an optimal and complete timetable with little or no conflict.

1.3 Problem with the Current System

Problems with the manual generation of timetables:

- Time allocation for a particular course is often repeated.
- Confusing time requirement often results in a lot of administrative error.
- Due to wrong timing or clashing results, the final generated timetable may not be optimal.
- It is highly demanding and generates lots of paperwork.
- It is not flexible.

1.4 Significance of the Study

The reason for this work is:

- To provide a user-friendly system for user.
- To improve the flexibility in timetable generation.
- Help save time and human resource.
- Improvement in productivity.
- Proper recording of class size, number of course offered, number and capacity of the available classroom.

1.5 Scope of the Study

This study will cover the allocation of classroom and course for lectures in any department of any university.

1.6 Limitations of the Study

These are some of the problems faced while executing this project.

Getting familiar with the Flora-2 system; since Flora-2 system was used as the backend application for this system, seeing Flora-2 is a new application and it is not commonly used, I had to learn from the basics.

1.7 Requirements

This section shows the minimum system requirement that will ensure that the software works efficiently. Therefore, the system requirement will be equivalent to the operating system the user uses.

1.7.1 Operating System

The project is expected to work on Windows XP and later versions, Red Hat Linux 10 distribution and above, also other distributions of Linux that have the same kernel version with that of Red Hat 10 and later distribution above 10. Mac OS 10 and above.

1.7.2 RAM

512 MB is enough to run most of the developmental software with little or no glitch.

1.7.3 Disk Space

1024 MB is enough for the storage of the files and the entire multimedia file that will be designed and used for the project.

1.8 Thesis Overview

The next chapter outlines the literature about timetable generation systems. This summary is rationalized and grouped by the solution algorithm. It starts with an introduction of major solution generation algorithms that have persisted within the literature. This chapter then concludes with the summary of the different algorithm

that was presented. In chapter 3, we outline the problem specification and proposed solution for the problem. In chapter 4, we have the system analysis and design, work breakdown structure, and we describe an overview of the front end we developed. In chapter 5, we have a detailed description of our implementation. In chapter 6, we outline the test strategy used in the system and also use our system for an actual classroom scheduling problem for the spring 2019 semester in the computer engineering department of Eastern Mediterranean University. Finally, in chapter 7, we summarize the work given in this thesis and make suggestions for future research directions.

Chapter 2

REVIEW OF RELATED WORKS

This section of the report gives an evaluation of the automatic timetable system literature. Various approaches and models have been projected for solving the timetabling problem. The problem varying from course timetabling to examination timetabling system.

Earlier the timetable activity was carried out using the manual system, and this system is not flexible. The education system has changed over the years; scheduling timetable has, however, become more complicated. As a result, the need for automatic timetable generation is increasing and it is, therefore, essential to developing a timetable generation system that generates valid solutions.

The techniques used to solve the scheduling problem were based on a reflection of human approach. One of the techniques includes direct heuristics [1] and [2]. These techniques were based on the concept of developing incomplete timetable by assigning the most constrained lectures first and then scheduling the other lectures until all lectures are allocated. Then techniques like integer and linear programming [3], graph coloring [4] were used to solve the scheduling problem.

De Werra [5] listed different problem of managing timetabling formally and gave diverse plans trying to unravel them. He likewise described the methodologies considered the most useful, focusing on the chart theoretic ones.

2.1 Timetable Solution Generation Algorithms

2.1.1 Linear Programming

Linear Programming and Integer Programming are the first set of techniques to be applied to a scheduling problem, which was developed from an extensive region of programming with mathematics [3]. Mathematical programming is suitable for problems with a larger range of variables that come across at interval boundaries [6]. The portion of mathematical programming involved with the distribution of source to recognized activities with the aim of meeting a favored object is called Linear Programming, while Integer Programming deals with the management of mathematical programming problems so that majority of the variables can expect a non-negative whole number [7]. Developing a linear programming model includes 3 steps [7]: The first step is to recognize the constraints, the second step is to identify proof of constraints which these constraints are written in linear equations. In the last step the work will be classified as a function of the decision variable.

2.1.2 Genetic Algorithms

Genetic Algorithms (GA) are a category of random search algorithms that employ the process of survival of the fittest as a selection mechanism [8]. This technique utilizes the genetic algorithm methodology by depending on bio-inspired operators like crossover, and mutation.

Scheduling problem using genetic algorithm is accurate compared to linear or integer programming. The genetic algorithm starts with the population of a randomly generated solution set, the fitness of the solution set is evaluated using weighted randomization as a perspective [9]. The best solution gets more duplicates, and the worst solutions die off.

2.1.3 Simulated Annealing

This is a randomized local search optimization procedure used to discover solutions for optimization problems. The methodology begins by generating a random initial solution. The fundamental procedure comprises of a loop that generates stochastically and for each iteration a neighbor of the present solution [10] [11].

Implementation of the Simulated Annealing (SA) algorithm is extremely needy of how basic components are characterized, such as the solution space, generating new solutions.

An example of this process is illustrated using SA to decrease the cost of a university timetable solution: [10].

While (Cost >0) and timetable not frozen repeat

- Repeat some constant number of times
- Choose a tuple T_n (class, roo, period)
- Choose a new field value for period called Period
- Evaluate the cost of removing this tuple from period
- Evaluate the cost of removing this tuple into period
- Compute change cost
- If (change in cost=0) or (change in cost is acceptable at this temperature)

- Then accept change and update cost
- Compute new temperature.

2.1.4 Tabu Search

Tabu search is a metaheuristic system that controls a local heuristics search strategy to explore the solution space beyond local optimal. Metaheuristics are said to be an expert method that guides and alters distinctive heuristics when searching for local optimal in an optimization problem [12]. This method has been proven reasonable to solve the difficult scheduling problem [13]. Applying tabu search to scheduling problems, the goal is to reduce the number of conflicts due to having lectures simultaneously [13].

2.1.5 Tiling Algorithms

This kind of algorithm gathers the classroom or courses to be allocated into groups called tiles. The tiles hold classes or courses that can work together, these tiles are now divided, utilizing distinctive search algorithm [14]. This algorithm can only be useful in secondary school where few students have a similar subject at the same time. These group of classes are grouped into tiles for scheduling, however, for a college, this hardly happens.

2.1.6 Constraint Satisfaction Programming

This approach has turned out to be beneficial for the scientist in the area of artificial intelligence, logic programming [15], and operational research [16]. Using these approach, constraints are being prioritized in a way that the more important one is satisfied first [16].

A scheduling problem can be defined by classifying the timeslot and classroom as a variable of the CSP, while the relationship between the lessons is called constraints

[16] [17]. Hence, scheduling can be figured out by choosing the variables, values, and constraints.

2.1.7 Direct Heuristics

This technique fills the timetable with one lecture at a given moment to the extent that no clashes arise, by interchanging the courses in order to allow different lectures to fit the schedule [2].

A direct heuristic example is the SCHOLA system in which the system depends on 3 strategies [18]

- Assigning the most necessary lecture to the most convenient period for the lecture
- When there is a free slot on the timetable, it allocates course to that space
- Move an already scheduled lecture to an empty slot, for the course we are trying to schedule.

A lecture is “necessary” when it has a tight constraint and a period is "convenient" when it has a soft constraint.

SCHOLA system schedules the courses by rotating between the first and second strategy and at the point where there are no more courses to be scheduled it uses the third strategy.

2.1.8 Logic Programming

A logic programming way of dealing with university scheduling problem was introduced in [19] and [15]. The language used for the implementation of the

scheduling system is PROLOG. The main advantage of this approach is the ability to express in a declarative way the constraints involved in the problem.

2.1.9 Particle Swam Optimization

Dr. Eberhart and Dr. Kennedy in 1995, proposed the population-based evolutionary computation called Particle Swarm Optimization [20]. It shares a lot of similarity with evolution computational approaches such as Genetic algorithms. However, unlike genetic algorithm, PSO has no crossover. The approach used in the research to solve scheduling problem was to randomly create a group of candidate solutions as particles, at that point the performance of the candidate solution was evaluated. The best position among every one of the particles was used.

This procedure shows how the movement of the timetable is being processed [21]:

Each particle (X_k^i) must be changed two slots at random by itself.

$$S_k^{i+1} = \text{rand} - \text{mutate} (X_k^i)$$

Copy a slot for a subject randomly from the local best (P_k^i) to particle (S_k^i).

$$W_k^{i+1} = \text{rand} - \text{change} (S_k^{i+1}, P_k^i)$$

Copy a slot for a subject randomly from the global best (G^i) to W_k^{i+1} at random.

$$X_k^{i+1} = \text{rand- change} (W_k^{i+1}, G^i)$$

This process reiterates until an optimal timetable has been generated or the maximum number of iteration has been met. Experimental outcomes show that PSO can be a productive technique for tackling the course scheduling problem with no clash.

2.2 Summary

The above completes the synopsis of the major techniques used in the timetabling problem. Based on the study exhibited above it shows that in future timetabling will move from the theoretical basis of the CSP [16] way to a more practical method.

Chapter 3

PROBLEM SPECIFICATION AND PROPOSED SOLUTION

In this chapter, we focus on developing a system that will generate and also manage timetables in a university setting. Furthermore, we focus our study on trying to combine the implementation of the algorithm with the main issue of application and management of the timetable solution. The algorithm employed in our study to generate solutions is later presented and the data needed for generation procedures are shown.

3.1 Constraints

Scheduling processes involve the arrangement of all courses into a weekly timetable. This would have designated period and a classroom according to a given set of constraint. Constraints are classified into two groups [17]: Hard Constraints and Soft Constraints. Both Hard and soft constraints are equally important in timetabling creation.

3.1.1 Hard Constraints

Hard Constraints are requirements that must be met in order for the timetabling to be attainable. The hard constraint in our work includes:

- No two course sections must be allocated the same room or timeslot combination (i.e. a classroom can host only one class at a time)

- The classroom must be large enough to hold the number of students requested for a course section.
- Courses belonging to the same year must not clash.
- For 4 hours courses, the time should be distributed as 2+2 on different days. For 3 hours courses, the time should be distributed as 2+1 or 1+2 on different days.

3.1.2 Soft Constraints

Soft constraints refer to different requirements that help to make a better timetable.

Soft constraints are usually not satisfiable when it comes to the process of scheduling [18]. However, it can be adjusted to meet the criteria of the system.

For example:

- The lecture of a course ought to be disseminated across the days stipulated.
- Period or specific classroom preferred by the lecturer.
- The lectures are not assigned to timeslots, which are in the instructor's forbidden time slots
- As far as possible, classes are scheduled in the instructor's preferred timeslots.
- Classes should be scheduled in their corresponding departments exclusive-use classrooms.

3.2 Features of the System

- It is not complicated, very easy to understand and navigate when using the system.
- It is fast, the timetables are generated in minimal time, and fewer efforts, money and time are also saved.

3.3 Timetable Generation Method

The scheduler implemented in Flora-2 [22] requires an input knowledge about courses, classrooms, and course opening request for a specific semester. Based on the constraint specification in section 3.1.1 it performs a depth-first search of the state-space consisting of course section allocations to room/timeslot combinations, returning the first solution that is found.

3.4 Web Based Front End and Manager to the Course Scheduler

The timetable management system is developed using a form method to get information from the user to be able to generate the timetable. Before performing any operation in the system the user must be logged in or he/she will create a new account. The login page validates your username, password and department with the information in the database, if validation is correct it logs you in, else, you will enter the correct information or create a new account if login details are not found.

Users are categorized by department. When logging in, the user's department is stored in a session, so every operation performed in the system is categorized based on the department. If a new user is created in same department, all data saved in the database relating with the department will be displayed.

To be able to generate a timetable you have to enter a course information, classroom information and the course that will be opening in a specific semester and year. This information is collected using a form method and data are stored into the SQL database.

The system allows you to manage the information you have added by clicking on the manage button in the system where you can edit, or delete your data. This “manage information” page gets data from the database and populates a tabular form so that the user can view data he/she created.

To generate a timetable, the user has to enter information of the year and semester he/she wants to generate timetable for. The system checks the database to see if year and semester are valid; if valid, the system sends the data in the SQL database to a knowledge-base which is a specific format that Flora-2 reads, the Flora-2 scheduler then generates the timetable and sends it to the system which is presented in a tabular format so the user can view easily. This generated timetable is also saved in the SQL database. The user can choose to save this generated timetable as PDF, print it out, or view later which is why the generated was saved in the database.

Chapter 4

ANALYSIS AND DESIGN

4.1 System Analysis

In this section, our aim is to review the existing system, how it works and also provides a better alternative for its improvement. The major task involves designing a user interface and the system is developed with the mind that the end user will find it easy to use.

4.2 Work Breakdown Structure

Below is an activity-based work breakdown structure. The diagram shows how we divided the whole process into simpler ones. Most of the steps are repeated again and again in the lifecycle of the project.

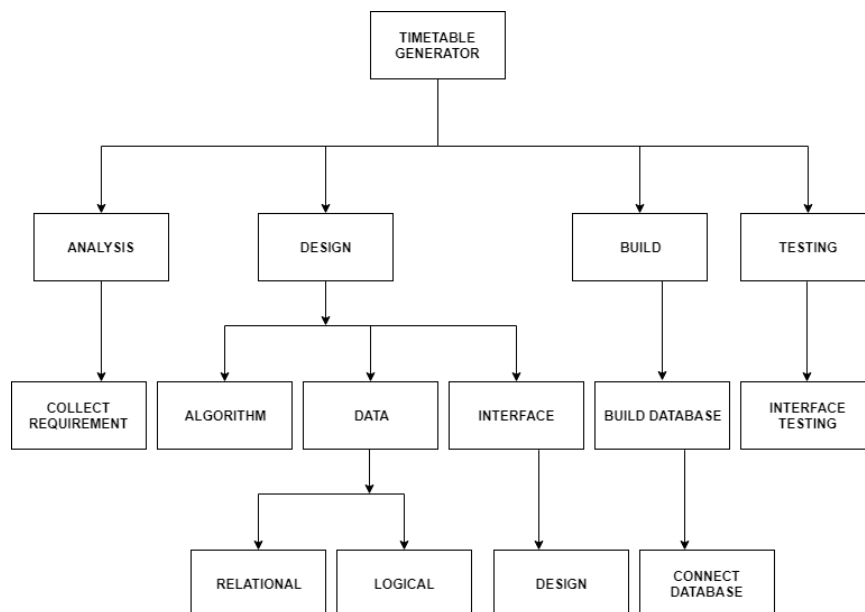


Figure 4.1: The Work Break Down Structure

4.3 Flow within the Project

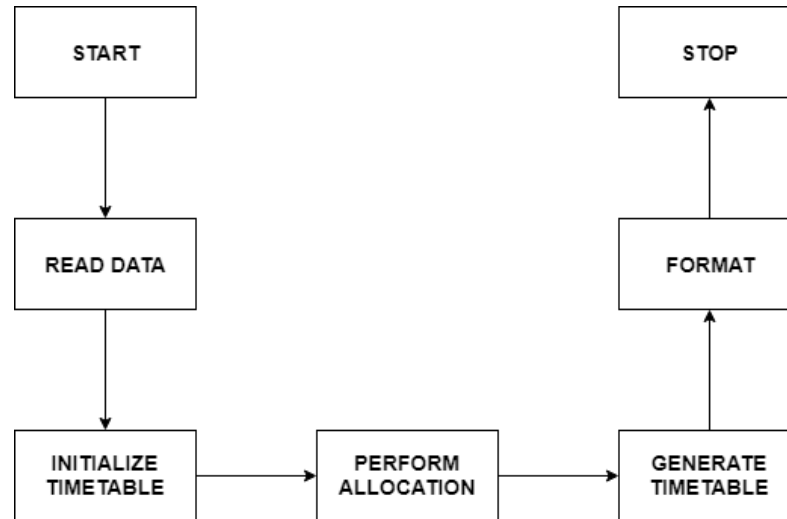


Figure 4.2: The Flow within the project

4.4 Functional Requirements

4.4.1 Use-case

Use-case diagram helps the external observer to understand what the system does.

Use-case diagram emphasized what the system does rather than focusing on how the system works.

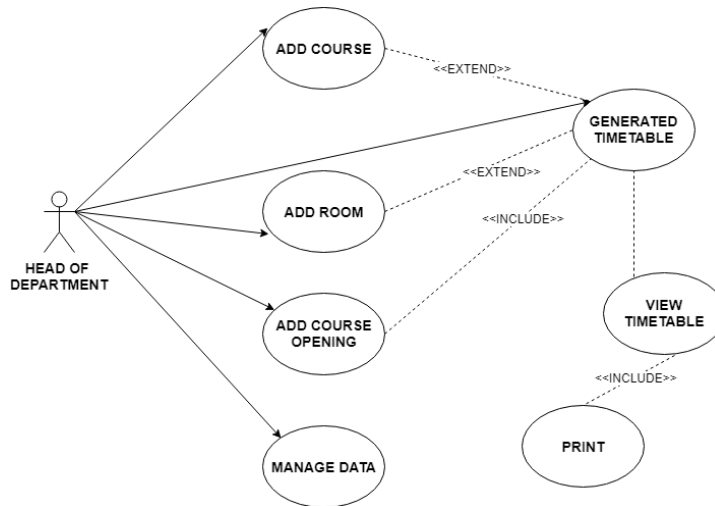


Figure 4.3: Use-case Diagram of the Scheduling System

4.5 Non Functional Requirements

Non-functional requirements describe principles use in judging the performance of the system which may include performance, security and more. NFR serves as restrictions on the system design during the implementation stage.

4.5.1 Usability Requirement

The system must be user-friendly, which ensures that the user will be satisfied with the level of efficiency of the system.

4.5.2 Efficiency Requirement

The efficiency of a system is based on the response and processing time, putting into consideration these factors the system should not make unnecessary use of system resources.

4.5.3 Reliability Requirement

- The system function must be accessible to the user most of the time.
- The system failure rate should be less than 0.5%

4.5.4 Robustness

- In case the system fails the recovery time should be short.

- The probability of data being corrupted should be low.

4.5.5 Flexibility

The system must be easily modified depending on users need, and modifications should not involve recreating a new software.

4.6 Proposed System

The aim of the proposed system is to provide a web-based interface with management capabilities to an existing course scheduler thus reducing the high cost and slow turnaround involved in generating timetables, a problem that the university faces every academic year.

Furthermore, the system has the capabilities for the input of vital information like various courses, classroom, departments, and specification of a few constraints from which the timetable will be constructed using forms, and the data collected are stored in a database. The timetable algorithm uses the form's input to generate the schedule.

The system is arranged into a three-level design. The information gathered from the user which serves as a knowledge base is placed at the center, in between the algorithm and the interface.

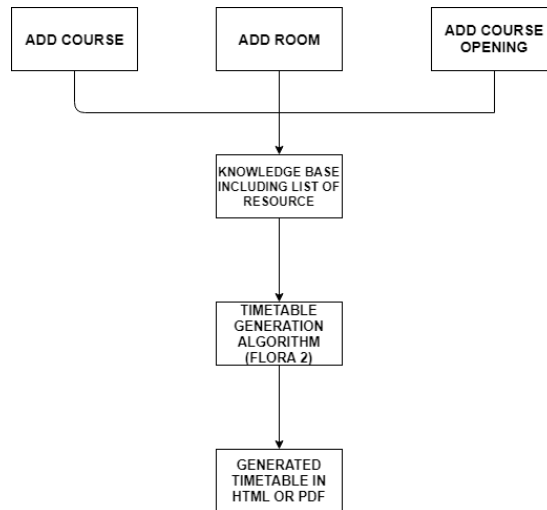


Figure 4.4: Three Tier Architecture of the System.

4.7 System Design

4.7.1 Class Diagram

This is an organization of related objects. It gives a graphical representation of a system by showing its classes and their relationships. The class diagram only displays what interact and not what happens during the interaction.

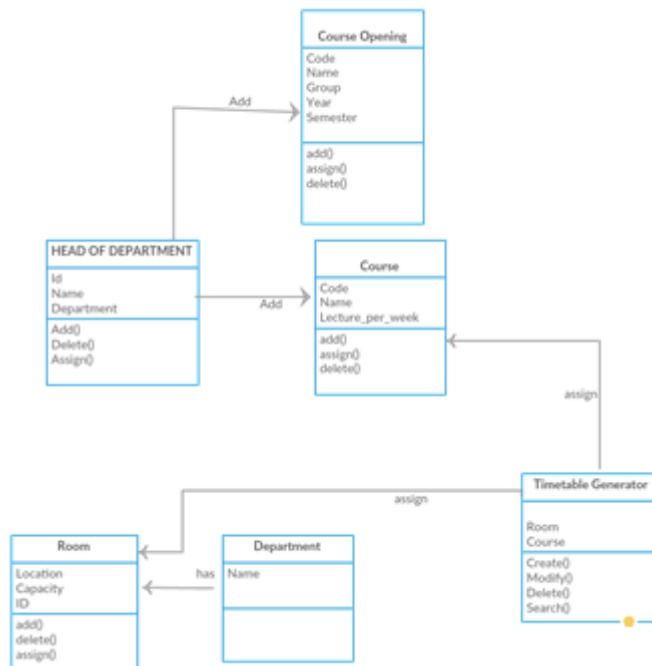


Figure 4.5: The Class Diagram of the Timetable Generation System.

4.7.2 Sequence Diagram

Sequence diagrams show a detailed flow for a specific use case. They show the calls and message flow between the objects in their sequence and also show the different calls towards different objects. Sequence diagrams must be on the detailed level showing the actors, user-interface objects and controller objects.

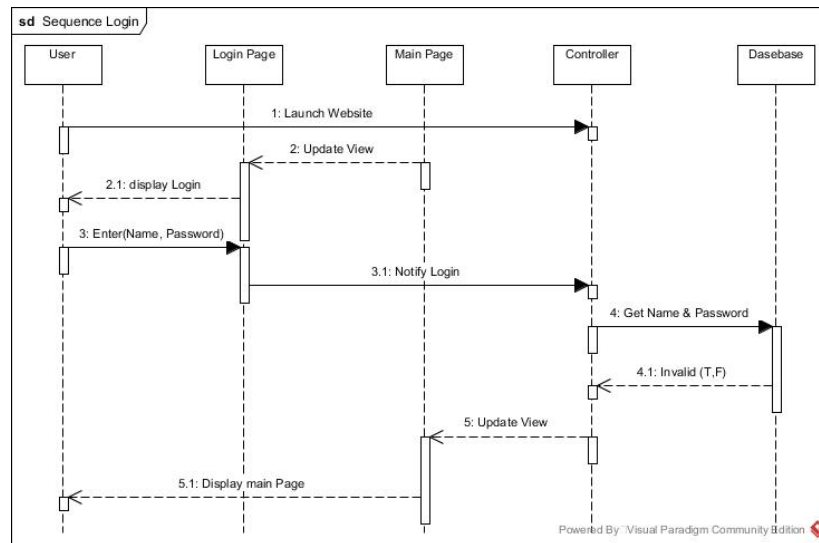


Figure 4.6: Sequence Diagram for the login Page.

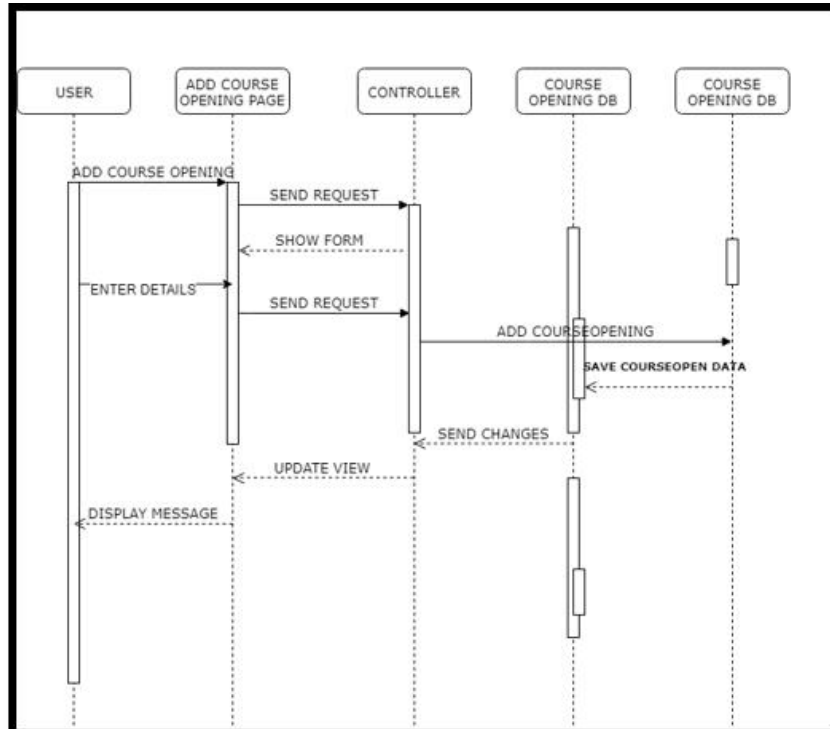


Figure 4.7: The Sequence Diagram for Adding a course opening for a specific semester.

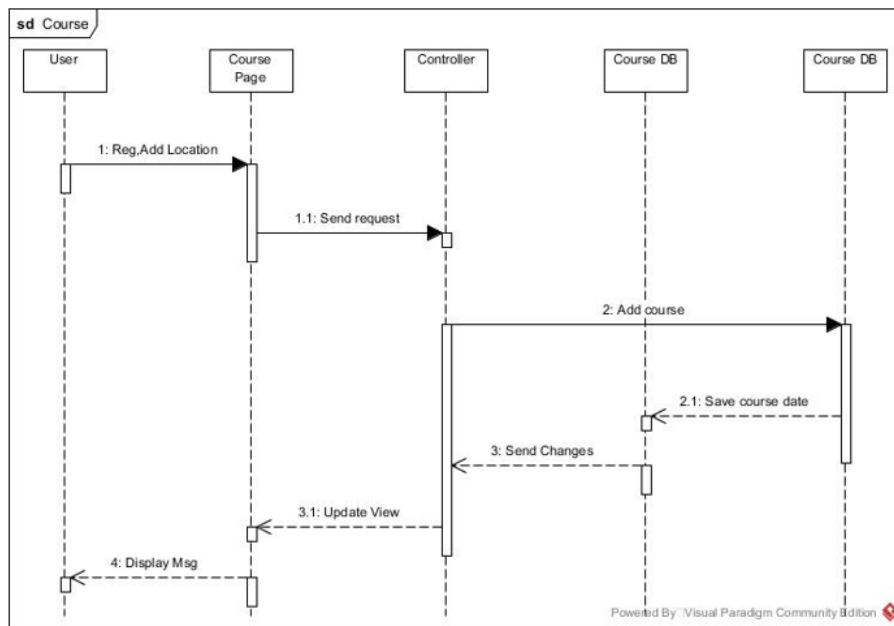


Figure 4.8: The Sequence Diagram for adding a course.

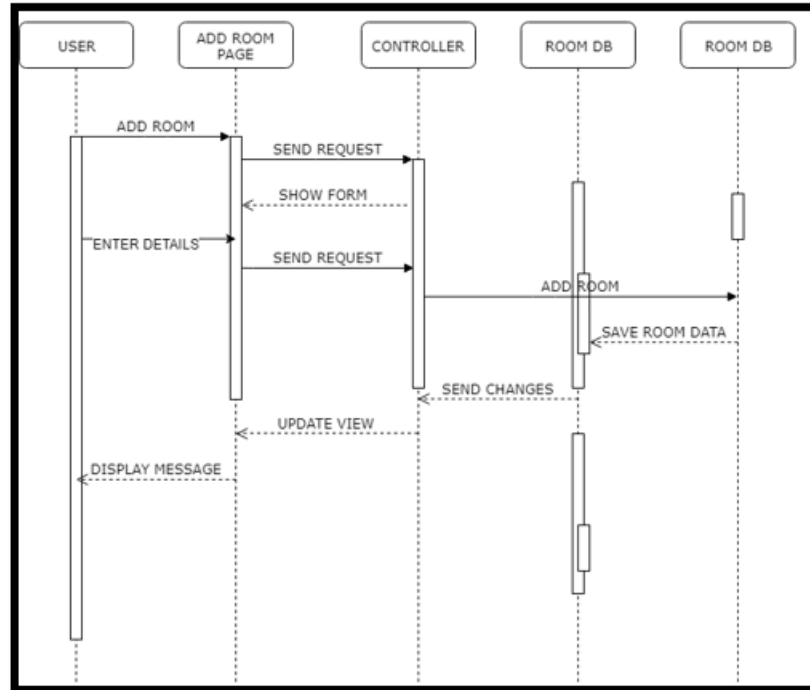


Figure 4.9: The Sequence Diagram for Adding a Room.

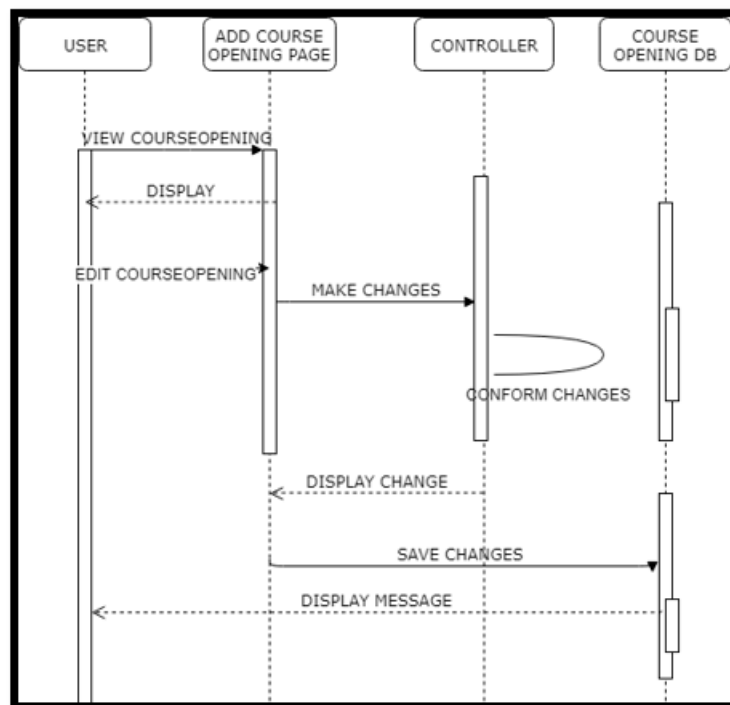


Figure 4.10: The Sequence Diagram for editing a course opening for a specific semester.

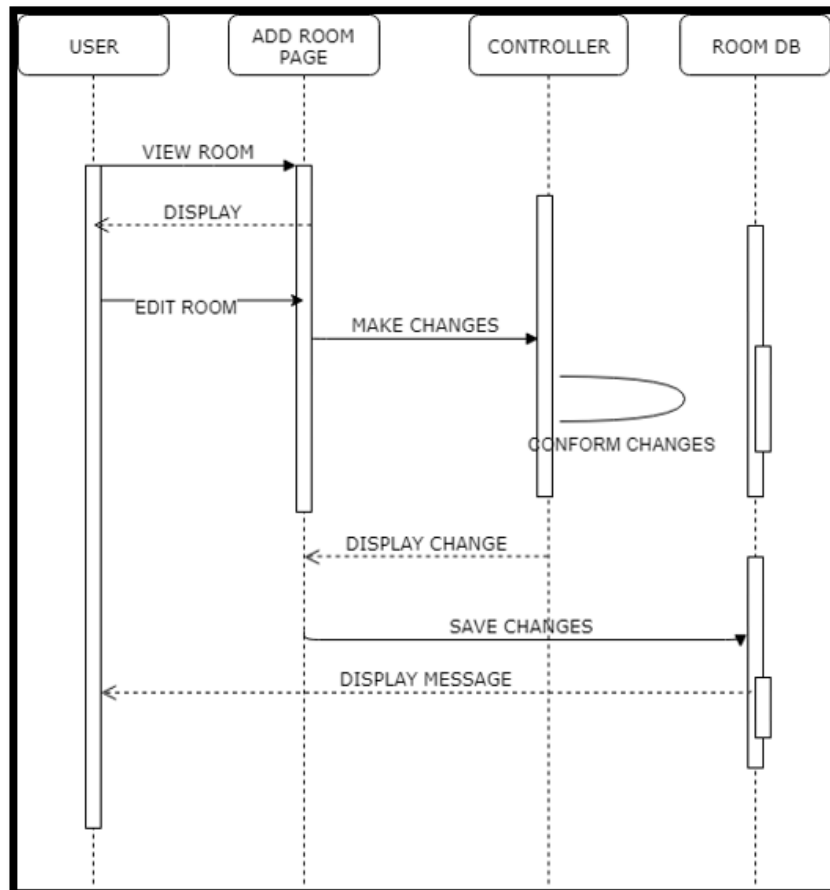


Figure 4.11: The Sequence Diagram for editing room.

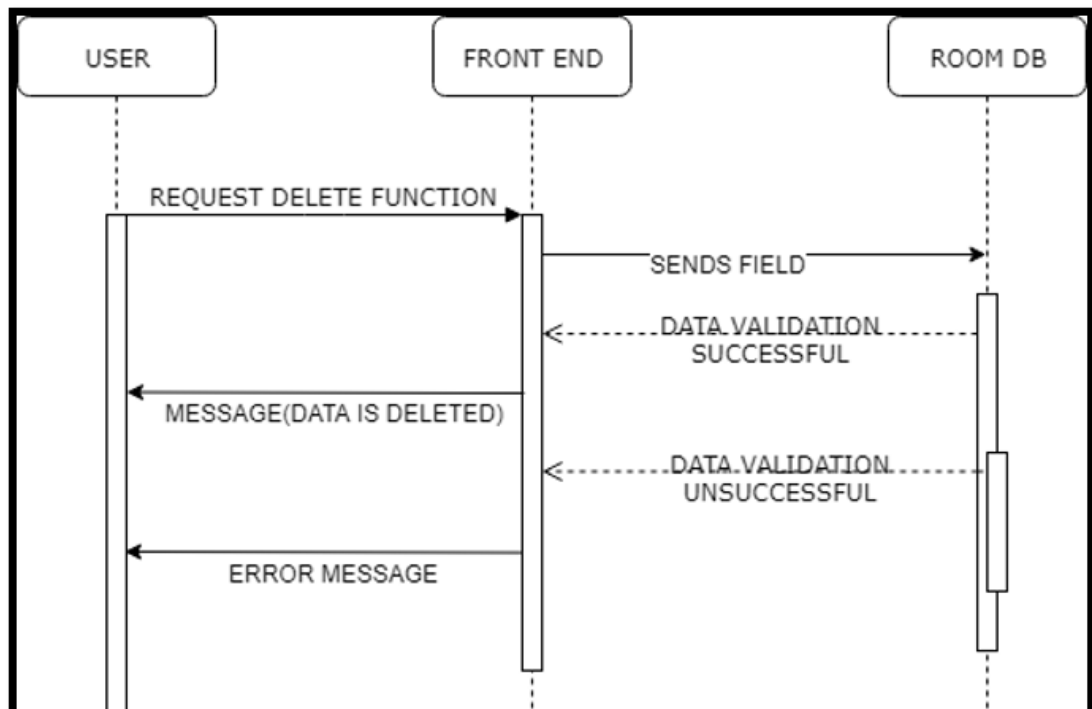


Figure 4.12: The Sequence Diagram for Deleting.

4.8 Database Design

A relational database was used in the database implementation of this system because it supports multiple tables that store each data once. More details of the database design are shown in appendix B.

4.8.1 E-R Diagram

This section contains the data modelling such as entities, significant objects, primary keys, relationships between entities and additional attributes for the entities.

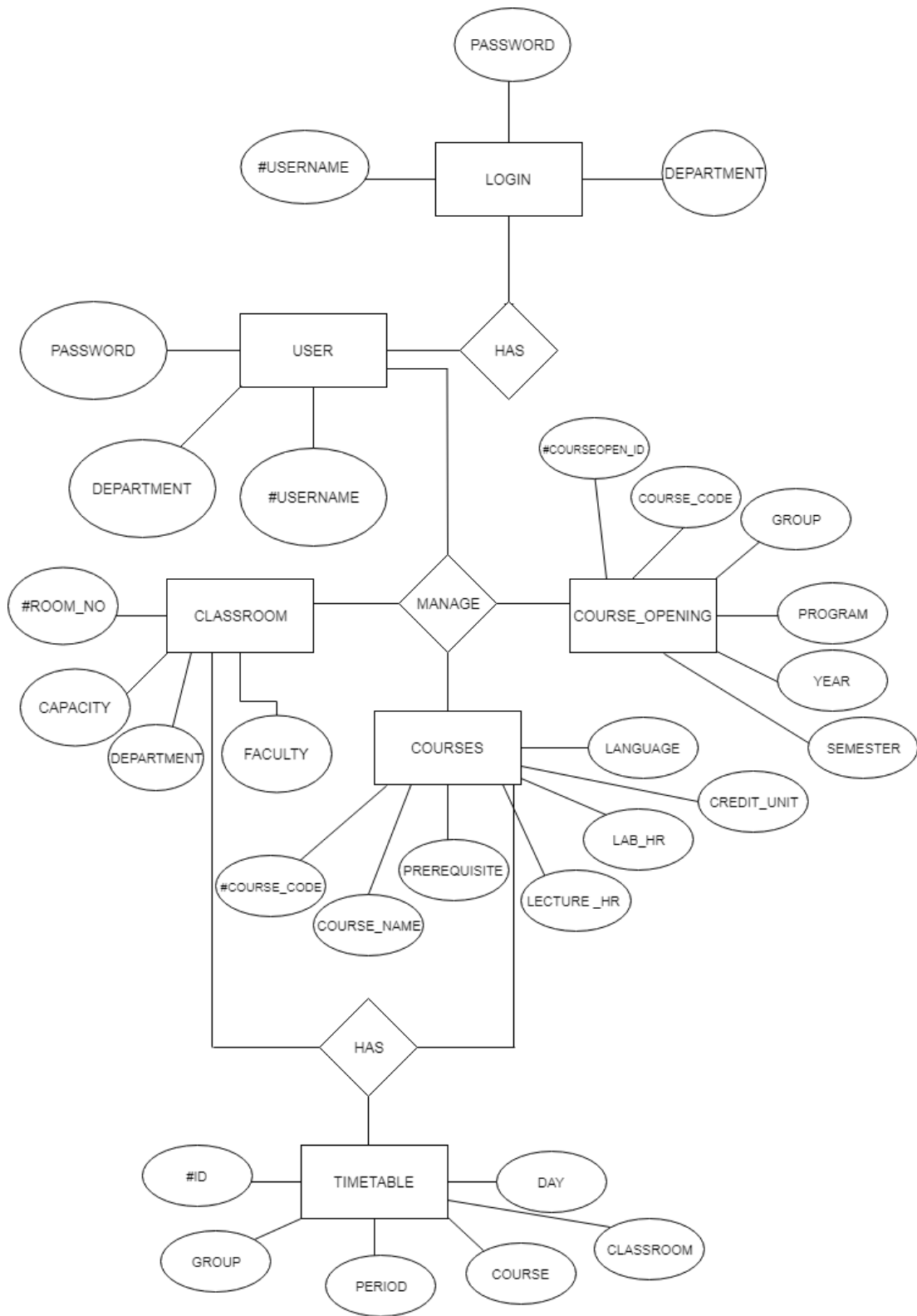


Figure 4.8.1 E-R Diagram of the Timetable generation system

4.8.2 Relational Schema

Courses (**course_id**, course_title, prerequisite, language, credit_unit, lec_hr, lab_hr, ects, dept, user)

Opencourse (**pk_openCourse**, courseCode, maxStudent, year, program, semester, groupNo, department, user, us_room, us_lab, instructor, external_room, external_lab)

Course_code: reference foreign key courses (course_id)

Exlab (course, lab_id)

course: reference foreign key opencourse (pk_openCourse)

Exroom (course, room_id)

course: reference foreign key opencourse (pk_openCourse)

lab_exdayperiod (course, day, period)

course: reference foreign key opencourse (pk_openCourse)

lab_fdayperiod (course, day, period)

course: reference foreign key opencourse (pk_openCourse)

lab_pdayperiod (course, day, period)

course: reference foreign key opencourse (pk_openCourse)

predayperiod (course, day, period)

course: reference foreign key opencourse (pk_openCourse)

pre_lab (course, lab_id)

course: reference foreign key opencourse (pk_openCourse)

pre_room (course, room)

course: reference foreign key opencourse (pk_openCourse)

user (**username**, password, department)

room (**room_id**, faculty, department, type, capacity, user)

lecturer(**staff_id**, firstName, lastName, type, department, faculty)

timetable (**timetable_id**, year, semester, course, period, groupNo, room, day, dept, user, instructor)

course: reference foreign key from course(course_id)

user: reference foreign key from user(username)

room: reference foreign key from room(room_id)

4.8.3 Normalized Tables

First Normalized Form

(course_id, room_id, username, pk_openCourse, course_title, prerequisite, language, credit_unit, lec_hr, lab_hr, ects, department, maxStudent, year, program, semester, groupNo, password, faculty, type, capacity, period, day, us_room, us_lab, pre_lab, pre_room, external_room, external_lab, staff_id, firstName, lastName, instructor_type)

Partial Dependancies

Second Normalized Form

course_id → course_title, prerequisite, language, credit_unit, lec_hr, lab_hr, ects, dept, user

pk_openCourse → course_code, maxStudent, year, program, semester, groupNo, department, user, us_room, us_lab, instructor, external_room, external_lab, exdayperiod, exlab, fdayperiod, lab_exdayperiod, lab_pdayperiod, pre_room, pre_lab

username → password, department

room_id → faculty, department, type, capacity, user

timetable_id → year, semester, course, period, groupNo, room, day, dept, user, instructor

Third Normalized Form

Courses (**course_id**, course_title, prerequisite, language, credit_unit, lec_hr, lab_hr, ects, dept, user)

Opencourse (**courseGroup**, course_code, maxStudent, year, program, semester, groupNo, department, user)

user (**username**, password, department)

room (**room_id**, faculty, department, type, capacity, user)

timetable (**timetable_id**, year, semester, course, period, groupNo, room, day, dept, user)

4.8.4 MYSQL

MySQL was used in the creation of the system to serve as the relational database management system (RDBMS) of the project.

MySQL is the world's second most widely used open source RDBMS, it's a great choice of database for use in web applications. MySQL is a robust, fast, open source, lowest cost, highly available, cross-platform and secure and its many features met the requirement of the project.

4.9 Output Design

The output design contains all the necessary details of the processed input by the system which is drawn from different tables in the database. In this case, the generated timetable is the output of the new system.

4.10 Input Design

The input design is the interaction point between the user and the system. There are three major input forms design:

- Course input design.
- Room input design.
- Course opening design.

Chapter 5

IMPLEMENTATION

This part shows how the system has been implemented, and also the activities involved in the system implementation and programming, and system testing.

5.1 Tools and Technologies Used

5.1.1 Flora 2

Flora-2 is an open source semantic rule-based system for knowledge representation and reasoning which is gotten from "F-logic HiLog and Transaction logic". Since it uses F-Logic and Hi-Log it means that the main element of the system is an object-oriented structure [25]. "Logic Programming with Defaults and Argumentation Theories" (LPDA) is supported by Flora-2. Applications include intelligent agents, Semantic Web, knowledge-bases networking, ontology management and more. Flora-2 is an incredible knowledge representation and reasoning system designed for developing knowledge-intensive applications [25].

5.1.2 Java

Java is a commonly used programming language which allows object-oriented syntax and it is particularly created to have less implementation dependency. It allows the user to "write once run once anywhere".

Java is a platform independent language. Platform independence simply means that programs written in the Java language must run similarly on any supported operating

system or hardware platform. Also one ought to be able to write a program once, compile it and run it anywhere.

5.1.3 NetBeans

NetBeans application was developed, using Java as a front end tool. Furthermore, the application has been coded to be platform independent running on Java Virtual Machine. NetBeans IDE has been chosen as its development environment because of the following features

- Using the Logger class, debugging can be easily done
- NetBeans database plugin allows for easy database access
- Simplified automated editor error detection.

5.2 Algorithm Implementation and Programming

The conversion of the design specification into a computer readable language is performed in this section. The modules were developed and compiled to produce a new system. Glassfish (java server), JSF (JSP framework), bootstrap (CSS framework) were used as frameworks and servers. The user interface was developed using JAVA, JSP, XML, the JSP was used to program the web pages using HTML form methods, bootstrap which is a cascade styling framework was used to beautify the web pages, XML was used to store, transport and encode data in a format that is both human and machine readable. After getting users' data collected from the form the JAVA code collates the data and sends it to the SQL database. In the "generate timetable" page after filling the form and you click on generate it calls a java page called "call". In this page the program gets the data based on the user's input from the database and sends it to a knowledge based which Flora-2 can only understand.

The main logic of our algorithm resides in the “time” class. The “time” class is called when the user clicks on the *generate* button after choosing the year and semester he or she wants to generate timetable for. In Listing 1, it gets the user’s department, year and semester from the session object. In Listing 2, we set the system properties for Java, and Flora-2 using the current directory; this is important for connecting Java with Flora-2, since Flora-2 is the back end application used for this project. In Listing 3, we create a new Flora session which indicates that you are starting a new entry. In Listing 4, we get the file which is “generate_meeting_time8c.flr” using its current directory. This file is the backbone of the project since it has the algorithm and imports all the data from the user necessary to generate a timetable satisfying the constraints. In line 16 the “%r” is the predicate used to import the files course.flr, room.flr and courseopening.flr which represent the knowledge base used to store data collected from the user using the form method in the web application. In Listing 6, we execute the flora object using the java function “Execute Query”, then we add the flora object as a vector string. The “%generateonebyone” predicate produces classroom and time-slot assignments to course groups one at a time, which are then stored in the database, and displayed to the user in a tabular form.

```
/*
 *
 * @author pamilerin
 */
Public class time extends HttpServlet {
Private preparedStatement pstmt;
Private ResultSet rs;
Private Connection conn= connect.DBconnect();
```

Listing 5.1: Setting private and public class.

```

1.      String user = (String) request.getSession().getAttribute("uname");
2.      String dept = (String) request.getSession().getAttribute("dept");
3.      String yearr = (String) request.getSession().getAttribute("year");
4.      String          semesterr          =          (String)
request.getSession().getAttribute("semester");

```

Listing 5.2: Getting resource from session.

```

5. System.setProperty("JAVA_BIN","C:\\ProgramFiles\\Java\\jdk1.8.0_191\\bin");
6. System.setProperty("PROLOGDIR","C:\\Users\\reslab\\Coherent\\ErgoAI\\XSB\\co
nfig\\x64-pc-windows\\bin");
7. System.setProperty("FLORADIR","C:\\Users\\reslab\\Coherent\\ErgoAI\\Ergo");

```

Listing 5.3: Setting Java and Flora-2 System Property.

```

8. FloraSession session = new FloraSession ();
9. String fileName=
"C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/
generate_meeting_times8c.flr";
10. if (session.loadFile (fileName, "main")) {
11.     out.println ("Example loaded successfully!");
12. } else {
13.     out.println ("Error loading the example!");
14. }

```

Listing 5.4: Creating a new Flora-2 session in java.

```

15. /* load data files */
16. String command = "%r.";
17. out.println("Query:" + command + user + dept);
18. Iterator<FloraObject> classroomObjs = session.ExecuteQuery(command);
19.     Vector<String> vars = new Vector<String>();

```

Listing 5.5: Load data Files.

```

Vector<String> vars = new Vector<String>();
vars.add("?Year");
vars.add("?Semester");
vars.add("?Course");
vars.add("?GN");
vars.add("?Room");
vars.add("?Day");
vars.add("?Period");
vars.add("?Instructor");

```

Listing 5.6: Adding data as a Vector variable.

```

String query = "?Year=" + yearr + ",?Semester=" + semesterr + "
%generateOneByOne(?Year,?Semester,?Course,?GN,?Room,?Day,?Period, ?Instructor).";

Iterator<HashMap<String, FloraObject>> allmatches
= session.ExecuteQuery(query, vars);
out.println(query);

```

Listing 5.7: Executing Query.

```

while (allmatches.hasNext()) {
HashMap<String, FloraObject> firstmatch = allmatches.next();
Object YearObj = firstmatch.get("?Year");
Object SemesterObj = firstmatch.get("?Semester");
Object CourseObj = firstmatch.get("?Course");
Object GNObj = firstmatch.get("?GN");
Object RoomObj = firstmatch.get("?Room");
Object DayObj = firstmatch.get("?Day");
Object PeriodObj = firstmatch.get("?Period");
Object InstructorObj = firstmatch.get("?Instructor");

```

Listing 5.8: Getting backtrack results.

```

out.println(" <tr>");

out.println("<td class='column1'>" + CourseObj + "</td>");
out.println("<td class='column2'>" + PeriodObj + "</td>");
out.println("<td class='column3'>" + GNObj + "</td>");
out.println("<td class='column3'>" + RoomObj + "</td>");
out.println("<td class='column4'>" + DayObj + "</td>");
out.println("<td class='column4'>" + InstructorObj + "</td>");
out.println(" </tr>");

```

Listing 5.9: Creating a JSP table Header..

```

String sql = "INSERT INTO `timetable`(`year`,`semester`,`course`,`room`,`day`,`period`,`user`,`department`,`groupNo`,`instructor`) VALUES (?,?,?,?,?,?,?,?,?)";

pstmt = conn.prepareStatement (sql);
pstmt.setString (1, years);
pstmt.setString (2, semesters);
pstmt.setString (3, courses);
pstmt.setString (4, rooms);
pstmt.setString (5, days);
pstmt.setString (6, periods);
pstmt.setString (7, user);
pstmt.setString (8, dept);
pstmt.setString (9, groups);
pstmt.setString (10, instructor);
pstmt.executeUpdate ();

```

Listing 5.10: Inserting generated timetable into database.

```

1. String year=request.getParameter("year");
2. String semester= request.getParameter("semester");
3. String uname=request.getParameter("username");
4. String dept= request.getParameter("department");
5. request.getSession().setAttribute("uname", uname);
6. request.getSession().setAttribute("dept", dept);
7. request.getSession().setAttribute("year", year);
8. request.getSession().setAttribute("semester", semester);

```

Listing 11: Getting request from session.

```

9. List c_data = new ArrayList();
10. List r_data = new ArrayList();
11. List cp_data = new ArrayList();
12. try {
13. Class.forName ("com.mysql.jdbc.Driver");
14. Connection conn = DriverManager.getConnection ("jdbc: mysql:
//localhost:3306/schedules?","root","");
15. Statement st = conn.createStatement ();

```

Listing 5.12: Setting data object as an array-list.


```

15. Statement st = conn.createStatement ();
16. ResultSet rs = st.executeQuery("Select * from course where
    department='"+dept+"'");
17. while (rs.next()) {
18. String id = rs.getString("course_id");
19. String name = rs.getString("course_title");
20. String prerequisite = rs.getString("course_prerequisite");
21. String language = rs.getString("course_language");
22. String creditUnit=rs.getString("course_credit_unit");
23. String lecturehr=rs.getString("lecture_hour");
24. String labhr= rs.getString("lab_hour");
25. String ects=rs.getString("ects");
26. c_data.add(id + ":Course" + " "+"["+" "+" "+"courseCode->" + ""+id+""
    +","+" "+"
27. "courseName->" + ""+name + ""+", "+" "+"hasPrerequisite->" + prerequisite
    +","+" "+" "instructionLanguage->" + language +","+" "+"credits->" +
    creditUnit +","+" "+" lecture_hours->" + lecturehr+","+" "+"lab_hours->"
    + labhr +","+" "+"ects->" + ects + " "+"].");
28. }

```

Listing 5.13: Getting course list from database.

```

21. rs = st.executeQuery("Select * from room where department='"+dept+"'");
22. while(rs.next())
23. {
24. String id=rs.getString("room_id");
25. //String name=rs.getString("room_name");
26. String faculty=rs.getString("faculty");
27. String department=rs.getString("department");
28. String capacity=rs.getString("capacity");
29. r_data.add(id + ":Classroom" + " "+"["+" "+" "+"roomNumber->" +id+","+" "+"
    + "location->" + ""+faculty+""+", "+" "+"inDepartment->" +
    ""+department+""+", "+" "+"capacity->" + capacity +""+"].");
30. }

```

Listing 5.14: Getting room list from classroom database.

```

39. rs=st.executeQuery("select * from opencourse where
    department='"+dept+"'");
40. while(rs.next())
41. {
42. String c_group=rs.getString("groupNo");
43. String c_id=rs.getString("courseCode");
44. String c_maxStd=rs.getString("maxStudent");
45. String c_year=rs.getString("year");
46. String c_program=rs.getString("program");
47. String c_semester=rs.getString("semester");
48. cp_data.add("request_"+c_id+"_"+c_group + ".CourseOpeningRequest" + "
    "+"["+ " " + " "+"groupNumber->" +c_group + "," + " "+
49. "ofCourse->" +c_id+"," + " "+"maxStudentSize->" + c_maxStd +"," + " "+
    "year->" + c_year +"," + " "+"forProgram ->" + c_program + " "+ " "+
    "+"semester->" + c_semester + " "+ "].");
50. }

```

Listing 5.15: Getting course opening list from database.

```

51. writeToFile(cp_data,
    "C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/
    course_opening_requests.flr");
52. writeToFile(r_data,
    "C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/
    classroom_instances.flr");
53. writeToFile(c_data,
    "C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/
    course_instances.flr");

```

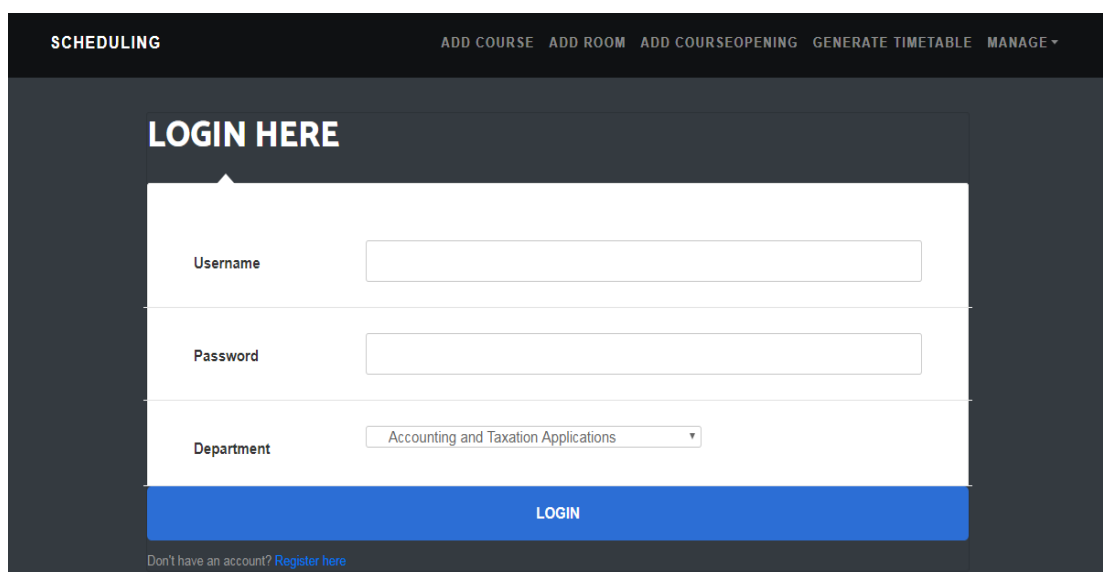
Listing 5.16: Writing File into Knowledge base.

5.3 User Manual

This is where the program that will run the sections identified in the control center is specified. This system consists of various program sections. Each section of the system is part of the entire system.

5.3.1 Login Section

The Login page prompts the user to enter a unique username and password or also register if the user is a guest. This username and password will be validated against the one already stored in the user database table. If the login information is valid, a new session will be created for the user and the dashboard page will be loaded, else an error message will be displayed and the login page will be reloaded.



The screenshot shows a web interface for a scheduling system. At the top, there is a dark navigation bar with the text 'SCHEDULING' on the left and a list of menu items: 'ADD COURSE', 'ADD ROOM', 'ADD COURSEOPENING', 'GENERATE TIMETABLE', and 'MANAGE' with a dropdown arrow. Below the navigation bar, the main content area has a dark background. In the center, there is a white box titled 'LOGIN HERE'. Inside this box, there are three input fields: 'Username', 'Password', and 'Department'. The 'Department' field is a dropdown menu currently showing 'Accounting and Taxation Applications'. Below these fields is a blue button labeled 'LOGIN'. At the bottom left of the white box, there is a link: 'Don't have an account? [Register here](#)'.

Figure 4.14: The Login Page of the System

5.3.2 Homepage

This interface is loaded if and only if a session was created for the user. This module manages subsystems like course page, room, and course opening page, adding new lecturer, departments, forbidden slots and timetable scheduling page.

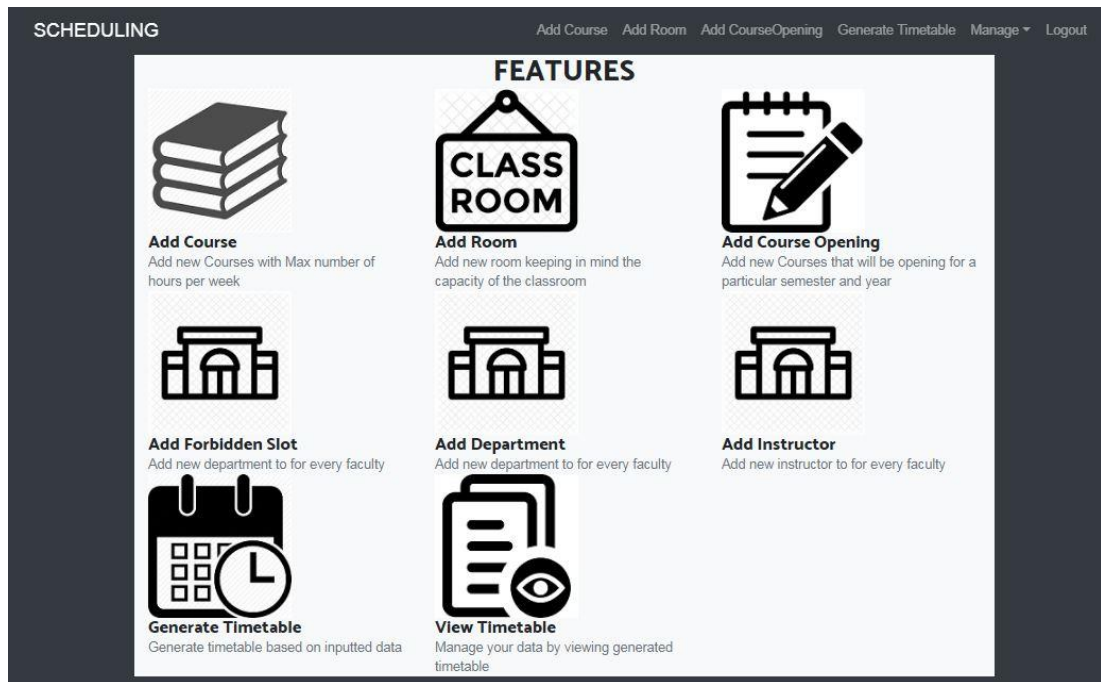


Figure 4.13: The Homepage of the System.

5.3.3 Course Page

The course module is the interface that manages the course available in the database. To create a new course, the user enters, course code, course title, group, credit unit. The user adds the program type, course type which allows user to choose if the course is a service course, area core, university elective, area elective e.t.c. then the user selects the program year, in this part the user choose if the course is a first year course or a graduate course.

SCHEDULING ADD COURSE ADD ROOM ADD COURSEOPENING GENERATE TIMETABLE MANAGE LOGOUT

ADD COURSES

Course Code	<input type="text"/>
Course Name	<input type="text"/>
Prerequisite	<input type="text" value="None"/>
Department	<input type="text" value="Computer Engineering"/>
Language	<input type="text" value="English"/>
Program Type	<input type="text" value="CMPE"/>
Course Type	<input type="text" value="AREA ELECTIVE"/>
Program Year	<input type="text" value="1"/>
Lecture Hour Per Week	<input type="text"/>
Lab Hour Per Week	<input type="text"/>
Credit Unit	<input type="text"/>
Ects	<input type="text"/>

ADD COURSE

Figure 4.14: The Add Course Page.

5.3.4 Classroom Page

The room module manages available rooms to the database. To create a new record, you enter room number, faculty, department, capacity, room type if the room is either a classroom or laboratory.

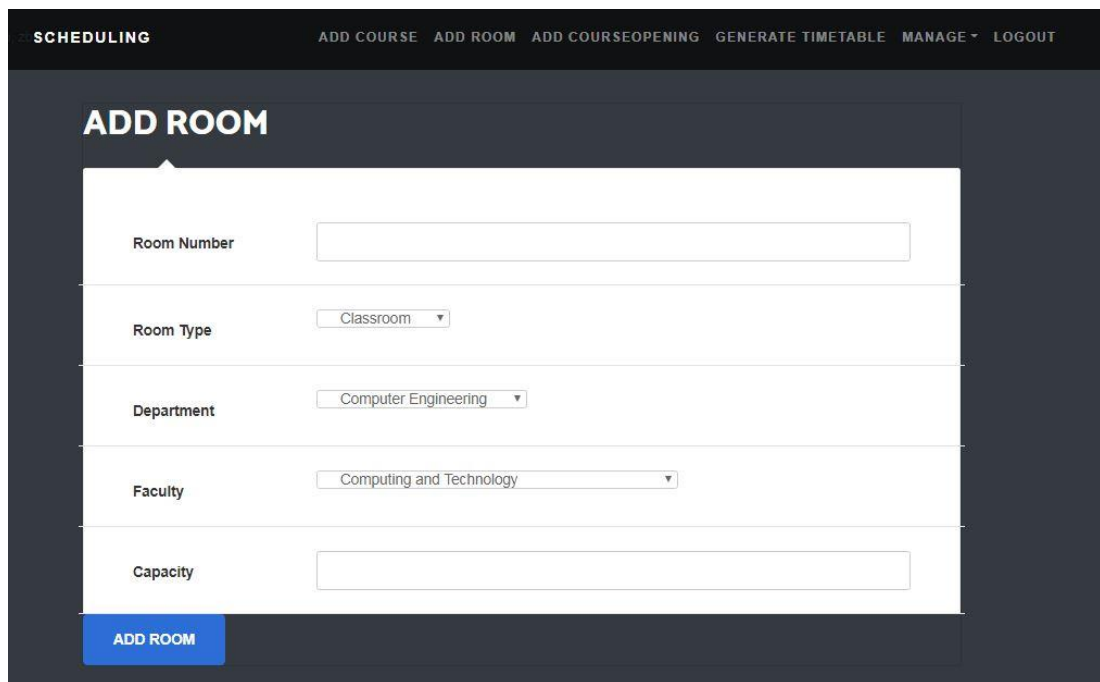


Figure 4.15: The Add Classroom Page.

5.3.5 Course Opening Page

The course opening page is where the user enters the course that will be opened in a particular semester to the database. It fetches the course from the database; the user enters, the year, semester, the maximum number of student to be registered to the course for that semester, the instructor for that particular course. If the course has an external room and lab from another the department, the user enters the classroom/lab information. User can enter a particular classroom or laboratory he/she wants the course to take place, the classroom details are gotten from database. The user can enter a classroom or laboratory he/she does not want in the exclude lab and room

field. The user can enter the preferred day and period he/she would like the course to take place. In the fixed day and period field the user enters the exact day and period he/she wants the course lecture to be fixed. In the exclude day period field the user enters the day or period he/she does not want the lecture to take place. The user enters lab fixed day and period, also enters the lab preferred day and period or the day and period he/she does not want.

The screenshot shows a web interface for adding a course opening. At the top, there is a navigation bar with the title 'SCHEDULING' and several menu items: 'ADD COURSE', 'ADD ROOM', 'ADD COURSEOPENING', 'GENERATE TIMETABLE', 'MANAGE', and 'LOGOUT'. Below the navigation bar is a dark header with the title 'Add Course Opening'. The main content area is a white form with the following fields:

- Course Code:** A dropdown menu with 'bio105' selected.
- Group:** A text input field.
- Max Student:** A text input field.
- Year:** A dropdown menu.
- Semester:** A dropdown menu.
- Instructor*:** A dropdown menu with 'None' selected.
- External Room:** A text input field.

External Lab	<input type="text"/>	
Use only classroom	<input type="text" value="None"/>	
Use only lab	<input type="text" value="None"/>	
Exclude Lab	Select Lab	<input type="text"/>
	<input type="text" value="None"/>	<input type="text"/>
Exclude Room	Select Room	<input type="text"/>
	<input type="text" value="None"/>	<input type="text"/>
Preferred Room	Select Room	<input type="text"/>
	<input type="text" value="None"/>	<input type="text"/>

Preferred Lab	Select Room	<input type="text"/>
	<input type="text" value="None"/>	<input type="text"/>
Exclude Day and Period	Select Day	Select Period
	<input type="text" value="None"/>	<input type="text" value="None"/>
Preferred Day and Period	Select Day	Select Period
	<input type="text" value="None"/>	<input type="text" value="None"/>
Fixed Day and Period	Select Day	Select Period
	<input type="text" value="None"/>	<input type="text" value="None"/>







Lab Fixed Day and Period	Select Day	Select Period	
	None ▾	None ▾	
Lab Preferred Day and Period	Select Day	Select Period	
	None ▾	None ▾	
Lab Exclude Day and Period	Select Day	Select Period	
	None ▾	None ▾	
ADD COURSE OPENING		VIEW COURSEOPENING	

Figure 4.16: The Add Course Opening Page

Chapter 6

TESTING AND SAMPLE SCHEDULE GENERATION

The new system was tested with the required constraints and the output was a feasible solution. System testing involves testing a complete and fully integrated software product. System Testing enables you to test, validate and verify both the Application architecture and requirements. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

6.1 Testing Strategy Employed

6.1.1 User Interface Testing

The interface features are tested to ensure that design rules, aesthetics and related visual content is available for the user without error. Individual interfaces are tested in a manner that is comparable to unit testing.

6.1.2 Interface testing mechanism

- Links: All links in the system was tested and they are all functional and accurate
- Forms: The forms on various webpages of the system were tested by giving sample data and submitting the forms.
- Authentication: The Login page user authentication was tested.

6.2 Quality Dimension of the System

- Content-> The content is evaluated both at syntactic and semantic level
 - Syntactic level: Spelling, punctuations and grammar are assessed.
 - Semantic level: correction of information presented, consistency across the entire content object and lack of ambiguity are all assessed.
- Function: is tested for correction, instability, and general conformance to appropriate implementation standard [Java Language standards].
- Structure: is assessed to ensure that it
 - Properly delivers web application content and function
 - Is extensible
 - Can be supported as new content
- Usability is tested to ensure that each category of user is supported by the interface and can learn and apply all required navigation syntax and semantic.
- Navigability: is tested to ensure that all navigation syntax and semantic are exercised to uncover any navigation errors (E.g. dead links, improper links, and erroneous links).
- Performance is tested under a variety of operating conditions, configurations and loading to ensure that
 - The system is responsive to user interaction
 - The system handles extreme loading without unacceptable operational degradation
- Interoperability is tested to ensure that the system properly interface with other applications and databases.

- Security is tested by assessing potential vulnerabilities and attempting to exploit the system information.

6.3 Using the System for Schedule Generation

This section shows a graphical example of a timetable for Eastern Mediterranean University 2019 spring semester. The user called “zb” wants to generate a timetable for computer engineering department spring 2019/2020 semester, the user first login into the system by entering his username, password and department in the login page. The system validates his information and logs him into the system. The user add new courses for computer engineering department, then he/she adds new rooms for computer engineering department, if data of courses and rooms were already in the database or the new data added are saved in the database, the user then proceeds to entering information of courses that he/ she wants to open for spring 2019/2020 semester, the user starts by selecting the course that were already in the database, this courses are displayed as a dropdown list, the user enters the group number for the course, the maximum number of student for that course or course group, then chooses the year and semester, and then submit. The user repeats this process until he/she has entered the courses that will be open that semester. Then the users click on “generate timetable” link, enter the year and semester then click on generate. The system runs the generation algorithm and displays the result to the user in a tabular format. The user can print or save the result displayed.

6.3.1 Sample Run

Based on the information given in section 6.3, the information needed to generate a timetable for user is displayed in the tables below. Table 1 highlights the courses that will be opened in spring 2019 that has been saved in the database, the classrooms

available in the computer engineering department database are being shown in table 2, and the computer engineering courses in the database is shown in table 3. The result of the generated timetable for fall 2019 semester is shown in Figure 6.1.

Table 1: List of course opening requests in fall 2019

Pk_open	courseCode	maxStudent	year	semester	group	Department	user	instructor
38	cmpe100	40	2019	FALL	1	Computer Engineering	zb	zeki bayram
39	cmpe108	30	2019	FALL	1	Computer Engineering	zb	dogu arifler
40	cmpe112	45	2019	FALL	1	Computer Engineering	zb	pamilerin v
41	cmpe110	40	2019	FALL	2	Computer Engineering	zb	cidem e
42	cmpe211	25	2019	FALL	1	Computer Engineering	zb	john olaifa
43	cmpe223	50	2019	FALL	1	Computer Engineering	zb	mehmet bodur

Table 2: List of rooms available in computer engineering department

Room	Faculty	Department	Capacity
CMPE126	ENGINEERING	Computer Engineering	70
CMPE127	ENGINEERING	Computer Engineering	70
CMPE128	ENGINEERING	Computer Engineering	70
CMPE129	ENGINEERING	Computer Engineering	70

CMPE25	ENGINEERING	Computer Engineering	70
CMPE26	ENGINEERING	Computer Engineering	70
CMPE27	ENGINEERING	Computer Engineering	70
CMPE23	ENGINEERING	Computer Engineering	70
CMPE33	ENGINEERING	Computer Engineering	70
CMPE36	ENGINEERING	Computer Engineering	70

Table 3: List of Computer Engineering courses

Code	Course name	Prerequisite	Language	Credit	Lec_ hr	Lab_ hr	ECTS
biol105	BiologicalBasisofBehavior	Biotin	english	3	3	1	6
biol124	IntroductiontoMolecularBiolo gy&Genetics	None	english	3	3	1	6
biol316	EnvironmentalManagement	None	english	3	3	1	6
chem101	GeneralChemistry	None	english	3	3	1	6
cmpe100	IntroductiontoComputerEngin eering	None	english	2	2	0	2
cmpe101	FoundationsofComputerEngin eering	None	english	3	3	1	6
cmpe108	AlgorithmsandProgramming	None	english	3	3	1	6
cmpe110	FundamentalsofProgramming	None	english	3	3	1	6
cmpe112	ProgrammingFundamentals	cmpe101	english	4	4	1	7
cmpe211	Object-OrientedProgramming	cmpe112	english	4	4	1	7
cmpe223	DigitalLogicDesign	math163	english	4	4	1	7
cmpe224	DigitalLogicSystem	cmpe223	english	4	4	1	7
cmpe226	ElectronicsforComputerEngin	math241	english	4	4	1	6

	eers						
cmpe231	DataStructures	cmpe112	english	4	4	1	7
cmpe242	OperatingSystems	cmpe112	english	4	4	1	6
cmpe318	PrinciplesofProgrammingLang uages	cmpe211	english	4	4	1	6
cmpe321	BasicsofSignalsandSystems	cmpe226	english	4	4	1	6
cmpe323	Microprocessors	cmpe224	english	4	4	1	7
cmpe324	ComputerArchitectureandOrga nization	cmpe224	english	4	4	1	7
cmpe343	SystemsProgramming	cmpe242	english	4	4	1	6
cmpe344	ComputerNetworks	cmpe343	english	4	4	1	7
cmpe354	DatabaseManagementSystems	cmpe231	english	4	4	1	6
cmpe371	AnalysisofAlgorithms	cmpe231	english	4	4	1	7
cmpe400	SummerTraining	None	english	1	2	0	1
cmpe405	GraduationProjectI	None	english	1	2	0	3
cmpe411	InformationSecurity	cmpe354	english	4	4	1	6
cmpe415	VisualProgramming	cmpe354	english	4	4	1	6
cmpe416	Object- OrientedProgramming&Graph icalUserInterfaces	cmpe354	english	4	4	1	6
cmpe418	InternetProgramming	cmpe354	english	4	4	1	6
cmpe423	EmbeddedSystems	cmpe354	english	4	4	1	6
cmpe466	ComputerGraphics	cmpe354	english	4	4	1	6
cmpe471	AutomataTheory	math163	english	4	4	1	6
cmpe538	Evolutionary multiobjective optimization	None	ENGLISH	3	3	2	4
cmpe553	Cryptography	None	english	3	2	4	
cmpe558	Data Mining	None	ENGLISH	3	3	3	4
cmpe562	Pattern recognition	None	ENGLISH	3	3	4	

cmpe576	Simulation	None	ENGLISH	3	3	2	4
engl191	CommunicationinEnglishI	None	english	3	3	1	4
engl192	CommunicationinEnglishII	engl191	english	3	1	4	4
engl201	Communicationskills	engl192	english	3	3	1	4
ghost101	ghost101 course	None	english	4	4	2	2
hist280	HistoryofTurkishReforms	None	turkish	2	2	0	3
math151	CalculusI	None	english	4	4	1	7
math152	CalculusII	math151	english	4	4	1	7
math163	DiscreteMathematics	None	english	3	3	1	5
math241	LinearAlgebraandOrdinaryDif fEquations	math151	english	4	4	1	6
math322	ProbabilityandStatisticalMetho ds	math151	english	3	3	1	4
phys101	PhysicsI	None	english	4	4	1	7
phys102	PhysicsII	phys101	english	4	4	1	6
tusl181	Turkishasasecondlanguage	None	english	2	2	0	3

SCHEDULING 2019 ADD COURSE ADD ROOM ADD COURSEOPENING GENERATE TIMETABLE MANAGE LOGOUT

Semester FALL

Search

Course Code	Period	Group	Room	Day	Instructor
cmpe112	4	1	cmpe129	thursday	pamilerin v
cmpe112	3	1	cmpe129	thursday	pamilerin v
cmpe112	4	1	cmpe218	thursday	pamilerin v
cmpe112	6	1	cmpe129	friday	pamilerin v
cmpe112	3	1	cmpe218	thursday	pamilerin v
cmpe112	5	1	cmpe129	friday	pamilerin v
cmpe110	5	2	AS108	thursday	cidem e
cmpe110	6	2	AS108	thursday	cidem e
cmpe110	7	2	AS108	friday	cidem e
cmpe110	5	2	IENG203	thursday	cidem e
cmpe110	6	2	IENG203	thursday	cidem e
cmpe108	7	1	cmpe127	thursday	dogu arifler
cmpe108	8	1	cmpe127	thursday	dogu arifler
cmpe108	8	1	cmpe127	friday	dogu arifler
cmpe108	7	1	cmpe230	thursday	dogu arifler
cmpe108	8	1	cmpe230	thursday	dogu arifler

Figure 6.1: Generated timetable for fall 2019

SCHEDULING ADD COURSE ADD ROOM ADD COURSEOPENING GENERATE TIMETABLE MANAGE LOGOUT

Year 2019

Semester FALL

Course cmpe112

Search

Course Code	Period	Group	Room	Day
cmpe112	4	1	cmpe129	thursday
cmpe112	3	1	cmpe129	thursday
cmpe112	4	1	cmpe218	thursday
cmpe112	6	1	cmpe129	friday
cmpe112	3	1	cmpe218	thursday
cmpe112	5	1	cmpe129	friday

Figure 6.2: View timetable per course.

SCHEDULING ADD COURSE ADD ROOM ADD COURSEOPENING GENERATE TIMETABLE MANAGE LOGOUT

Instructor:

Search

Course Code	Period	Group	Room	Day	Instructor
cmpe108	7	1	cmpe127	thursday	dogu arifler
cmpe108	8	1	cmpe127	thursday	dogu arifler
cmpe108	8	1	cmpe127	friday	dogu arifler
cmpe108	7	1	cmpe230	thursday	dogu arifler
cmpe108	8	1	cmpe230	thursday	dogu arifler

PRINT **DELETE TIMETABLE**

Figure 6.3: View timetable per Instructor.

Chapter 7

CONCLUSION

The objective of this work is to eliminate the use of manual timetabling in generating a course timetable, more specifically assigning meeting times and classrooms to course groups (sections) in a given semester, while observing inherent constraints. It provides a web based user interface backed by data storage for managing the whole schedule production process. Our system can not only handle hard constraints, but also soft constraints, such as instructor requests to teach their courses at specified periods or in specified rooms.

With the implemented system, users will be able to not only automatically generate timetables, rather than using the arduous conventional manual system, but also manage the whole process of timetable generation, including editing of input data, accessing previously generated timetables and incrementally generating timetables as new requirements arise at the beginning of each semester (e.g. new groups/section of course being opened, groups/sections of course being closed etc.).

Future work in this area includes making the system available to students so that they can see the courses that are going to be opened and plan their studies accordingly.

REFERENCES

- [1] Karmaker, D., Islam, M. R., Rahman, H., Bhowmik, A., & Imteaj, M. N. (2015). A Heuristic Approach to Course Scheduling Problem. *The Second International Conference on Education Technologies and Computers*.
- [2] Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc.
- [3] Lawrie, N. L. (1969). An integer linear programming model of a school timetabling problem. *The Computer Journal*, 12(4), 307-316.
- [4] Jain, Rahul & Pydimarri, Avinash. (2018). University Time Table Scheduling Using Graph Coloring Technique. 10.13140/RG.2.2.35925.52960.
- [5] de Werra, D. (1985). An introduction to timetabling. *European journal of operational research*, 19(2), 151-162.
- [6] Gunawan, A., Ng, K. M., & Poh, K. L. (2006, June). A Mathematical Programming Model for a Timetabling Problem. In *CSC* (pp. 42-47).
- [7] Daskalaki, S., & Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1), 106-120.

- [8] Sanchez, E., Shibata, T., & Zadeh, L. A. (1997). *Genetic algorithms and fuzzy logic systems: Soft computing perspectives* (Vol. 7). World Scientific.
- [9] Colorni, A., Dorigo, M., & Maniezzo, V. (1992). A genetic algorithm to solve the timetable problem. *Politecnico di Milano, Milan, Italy TR*, 90-060.
- [10] Abramson, D., & Dang, H. (1993). School timetables: A case study in simulated annealing. In *Applied simulated annealing* (pp. 103-124). Springer, Berlin, Heidelberg.
- [11] Henderson, D., Jacobson, S. H., & Johnson, A. W. (2003). The theory and practice of simulated annealing. In *Handbook of metaheuristics* (pp. 287-319). Springer, Boston, MA.
- [12] Lewis, R., Paechter, B., & Rossi-Doria, O. (2007). Metaheuristics for university course timetabling. In *Evolutionary scheduling* (pp. 237-272). Springer, Berlin, Heidelberg.
- [13] ZhipengLü Jin-KaoHaoa (2011) Adaptive Tabu Search for course timetabling.
- [14] Kingston, J. H. (2004, August). A tiling algorithm for high school timetabling. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 208-225). Springer, Berlin, Heidelberg.

- [15] Fahrion, R., & Dollansky, G. (1992). Construction of university faculty timetables using logic programming techniques. *Discrete Applied Mathematics*, 35(3), 221-236.
- [16] BraWEilsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European journal of operational research*, 119(3), 557-581.
- [17] Abdennadher, S., & Marte, M. (2000). University course timetabling using constraint handling rules. *Applied Artificial Intelligence*, 14(4), 311-325.
- [18] Junginger, W. (1986). Timetabling in Germany—a survey. *Interfaces*, 16(4), 66-74.
- [19] Kang, L., & White, G. M. (1992). A logic approach to the resolution of constraints in timetabling. *European Journal of Operational Research*, 61(3), 306-317.
- [20] Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Computing*, 22(2), 387-408.
- [21] Chu, S. C., Chen, Y. T., & Ho, J. H. (2006, August). Timetable scheduling using particle swarm optimization. In *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)* (Vol. 3, pp. 324-327). IEEE.

- [22] Toprak, Imren (2015). A Knowledge-based Intelligent Agent for Course Registration. *Master's Thesis, Computer Engineering Department, EMU.*
- [23] Yang, G., Kifer, M., & Zhao, C. (2003, November). \mathcal{F} lora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 671-688). Springer, Berlin, Heidelberg.
- [24] [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- [25] <https://en.wikipedia.org/wiki/NetBeans>
- [26] https://en.wikipedia.org/wiki/Functional_requirement
- [27] Larman, C. (2012). *Applying UML and patterns: an introduction to object oriented analysis and design and iterative development.* Pearson Education India.

APPENDICES

Appendix A: Source Code

A.1. ADD COURSE.JSP

```
<%--
  Document   : addcourse
  Created on : Jan 10, 2019, 10:38:54 PM
  Author    : Pamilerin
--%>

<% @page import="java.sql.PreparedStatement"%>
<% @page import="java.sql.DriverManager"%>
<% @page import="java.sql.ResultSet"%>
<% @page import="java.sql.Connection"%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Add Course</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom fonts for this template -->
    <link
      href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
      rel="stylesheet">
    <link
      href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
      rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>

  </head>
  <body id="page-top" class="index">

    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <!--Get session id so when user is not logged in you can't view the page -->
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              String dept = (String) session.getAttribute("department");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
      </div>
    </nav>
  </body>
</html>
```

```

        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
            <!--Navigation bar -->
            <li class="nav-item">
                <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>

            <li class="nav-item">
                <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
                <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
                <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
                <ul class="dropdown-menu" aria-labelledby="manage">
                    <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>

                    <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                    <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
                    <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
                </ul>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="logout"><span></span> Logout</a></li>
        </ul>
    </div>
</div>
</nav>
<!--end of navigation bar -->
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">ADD COURSES</h2>
            </div>
            <div class="card-body">
                <form method="POST" action="course">
                    <fieldset>
                        <%
                            String id = (String) session.getAttribute("username");
                            String deptt = (String) session.getAttribute("department");
                        %>
                        <input id="username" name="username" type="hidden" value="<%=id%>">
                    <div class="form-row">
                        <div class="name" for="subcode">Course Code</div>

```



```

        <option value="Turkish">Turkish</option>
    </select>
</div>
</div>

<div class="form-row">
    <div class="name" for="subdescription">Lecture Hour Per Week</div>
    <div class="value">
        <input id="lechr" name="lechr" type="text" placeholder="" class="input--
style-6" required="">
    </div>
</div>

<div class="form-row">
    <div class="name" for="subdescription">Lab Hour Per Week</div>
    <div class="value">
        <input id="labhr" name="labhr" type="text" placeholder="" class="input--
style-6" required="">
    </div>
</div>

<div class="form-row">
    <div class="name" for="subdescription">Credit Unit</div>
    <div class="value">
        <input id="cUnit" name="cUnit" type="cUnit" placeholder="" class="input--
style-6" required="">
    </div>
</div>

<div class="form-row">
    <div class="name" for="subdescription">Ects</div>
    <div class="value">
        <input id="ects" name="ects" type="text" placeholder="" class="input--
style-6" required="">
    </div>
</div>
</div>
</div>

<div class="footer" >
    <button class="btn btn--radius-2 btn--blue-2" type="submit">ADD
COURSE</button>
</div>
</form>

</div>
</div>
</div>
</body>
</html>

```

A.2. ADD DEPARTMENT.JSP

```
<%--
  Document : adddept
  Created on : Feb 14, 2019, 4:10:31 AM
  Author : Pamilerin
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Add department</title>

    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>

  </head>
  <body id="page-top" class="index">

    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              String dept = (String) session.getAttribute("department");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>

        <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
          <div class="container">

            <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
```

```

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
            <li class="nav-item">
                <a class="nav-link" href="addcourse.jsp"><span></span> Add
Course</a></li>
            <li class="nav-item">
                <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
                <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
                <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
                <ul class="dropdown-menu" aria-labelledby="manage">
                    <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
                    <li><a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                    <li><a class="dropdown-item" href="opencourselist.jsp"><span></span>
View CourseOpening</a></li>
                    <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
                </ul>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="logout"><span></span> Logout</a></li>
        </ul>
    </div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">ADD DEPARTMENT</h2>
            </div>
            <div class="card-body">
                <form method="POST" action="department">
                    <div class="form-row">
                        <div class="name" for="room">Department Name</div>
                        <div class="value">
                            <input id="deptname" name="deptname" type="text" placeholder=""
class="input--style-6" required="">
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name" for="room">Faculty </div>
                        <div class="value">
                            <input id="faculty" name="faculty" type="text" placeholder=""
class="input--style-6" required="">
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```
        </div>
      </div>
      <div class="footer" >
        <button class="btn btn--radius-2 btn--blue-2" type="submit">ADD
DEPARTMENT</button>
      </div>
    </form>
  </div>
</div>
</div>
</body>
</html>
```

A.3. ADD COURSE OPENING.JSP

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
    <script>
      $(document).ready(function () {
        document.getElementById("year").value = localStorage.getItem("item1");
        document.getElementById("semester").value = localStorage.getItem("item2");
      });
    </script>
    <script>
      $(window).on('beforeunload', function () {
        localStorage.setItem("item1", document.getElementById("year").value);
        localStorage.setItem("item2", document.getElementById("semester").value);
      });
    </script>
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
```



```

        <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
    <li class="nav-item">
        <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
    <li class="nav-item">
        <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
        <ul class="dropdown-menu" aria-labelledby="manage">
            <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
            <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
            <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
            <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
        </ul>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="logout"><span></span> Logout</a></li>
</ul>
</div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">Add CourseOpening</h2>
            </div>
            <div class="card-body">
                <form method="POST" action="open">
                    <fieldset>
                        <%
                            String id = (String) session.getAttribute("username");
                            String dept = (String) session.getAttribute("department");
                        %>
                        <input id="username" name="username" type="hidden" value="<%=id%>">
                        <input id="department" name="department" type="hidden"
value="<%=dept%>">
                        <div class="form-row">
                            <div class="name" for="course_id">Course Code</div>
                            <div class="value">
                                <select id="course_id" name="course_id" placeholder="select course"
class="input--style-6" required="">
                                    <%
                                        Connection con = null;
                                        ResultSet rs = null;
                                        try {
                                            Class.forName("com.mysql.jdbc.Driver");
                                            con
DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                                            HttpSession ss = request.getSession();
                                            PreparedStatement pst = con.prepareStatement("select * from course
where department='' + dept + ''");

```

```

        rs = pst.executeQuery();
        while (rs.next()) {
            String name = rs.getString("course_id");
%>
<option value="<%=name%>"><%=name%></option>
<%
    }
    } catch (Exception e) {
        out.print(e);
    }
%>
</select>
</select>
</div>
</div>
<div class="form-row">
    <div class="name" for="group">Group</div>
    <div class="value">
        <input id="group" name="group" type="text" placeholder="" class="input--
style-6" required="">
    </div>
</div>
<div class="form-row">
    <div class="name" for="student">Max Student</div>
    <div class="value">
        <input id="maxStd" name="maxStd" type="text" placeholder=""
class="input--style-6" required="">
    </div>
</div>
<div class="form-row">
    <div class="name" for="program">Program</div>
    <div class="value">
        <select id="program" name="program" placeholder="select program"
class="input--style-6" required="">
            <option value="Undergraduate">Undergraduate</option>
            <option value="Postgraduate">Postgraduate</option>
        </select>
    </div>
</div>
<div class="form-row">
    <div class="name" for="year">Year</div>
    <div class="value">
        <select id="year" name="year" placeholder="select year" class="input--
style-6" required="">
            <option value="2019">2019</option>
            <option value="2020">2020</option>
            <option value="2021">2021</option>
            <option value="2022">2022</option>
            <option value="2023">2023</option>
            <option value="2024">2024</option>
            <option value="2025">2025</option>
            <option value="2026">2026</option>
            <option value="2027">2027</option>
            <option value="2028">2028</option>
            <option value="2029">2029</option>
            <option value="2030">2030</option>
        </select>
    </div>
</div>
</div>

```

```

        <div class="form-row">
            <div class="name" for="semester">Semester</div>
            <div class="value">
                <select id="semester" name="semester" placeholder="select semester"
class="input--style-6" required="">
                    <option value="FALL">FALL</option>
                    <option value="SPRING">SPRING</option>
                    <option value="SUMMER">SUMMER</option>
                </select>
            </div>
        </div>
    </div>
    <div class="footer" >
        <button class="btn btn--radius-2 btn--blue-2" type="submit">ADD COURSE
OPENING</button>
        <a href=<%= "\opencourselist.jsp?year=" + (String)
request.getSession().getAttribute("year") + "&semester=" + (String)
request.getSession().getAttribute("semester") + "\"%> class="btn btn--radius-2 btn--blue-2">VIEW
COURSEOPENING</a>
    </div>
</form>
</div>
</div>
</div>
</body>
</html>

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              String dept = (String) session.getAttribute("department");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
              <ul class="dropdown-menu" aria-labelledby="manage">

```

```

        <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>

        <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>

        <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
        <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
    </ul>
</li>
<li class="nav-item">
    <a class="nav-link" href="logout"><span></span> Logout</a></li>
</ul>
</div>
</div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">ADD ROOM</h2>
            </div>
            <div class="card-body">
                <form method="POST" action="room">
                    <fieldset>
                        <%
                            String id = (String) session.getAttribute("username");
                            String deptt = (String) session.getAttribute("department");
                        %>
                        <input id="username" name="username" type="hidden" value="<%=id%>">
                        <div class="form-row">
                            <div class="name" for="room">Room Number</div>
                            <div class="value">
                                <input id="roomNo" name="roomNo" type="text" placeholder=""
class="input--style-6" required="">
                            </div>
                        </div>
                        <div class="form-row">
                            <div class="name" for="dept">Department</div>
                            <div class="value">
                                <select id="department" name="department" value="select your department"
class="input--style-6" >
                                    <%
                                        Connection con = null;
                                        ResultSet rs = null;

                                        try {
                                            Class.forName("com.mysql.jdbc.Driver");
                                            con
                                            DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                                            HttpSession ss = request.getSession();

                                            PreparedStatement pst = con.prepareStatement("select * from
department where name=" + deptt + "");

```

```

        rs = pst.executeQuery();
        while (rs.next()) {
            String name = rs.getString("name");
        %>
        <option value="<%=name%>"><%=name%></option>
        <%
        }
        } catch (Exception e) {
            out.print(e);
        }
        %>
    </select>
</div>
</div>
<div class="form-row">
    <div class="name" for="faculty">Faculty</div>
    <div class="value">
        <select id="faculty" name="faculty" placeholder="select faculty"
class="input--style-6" required="">
        <%
            try {
                con
                DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                HttpSession ss = request.getSession();
                PreparedStatement pst = con.prepareStatement("select DISTINCT
faculty from department");
                rs = pst.executeQuery();
                while (rs.next()) {
                    String name = rs.getString("faculty");
                %>
                <option value="<%=name%>"><%=name%></option>
                <%
                }
                } catch (Exception e) {
                    out.print(e);
                }
                %>
            </select>
        </div>
    </div>
    <div class="form-row">
        <div class="name" for="capacity">Capacity</div>
        <div class="value">
            <input id="capacity" name="capacity" type="text" placeholder=""
class="input--style-6" required="">
        </div>
    </div>
</div>
<div class="footer" >
    <button class="btn btn--radius-2 btn--blue-2" type="submit">ADD
ROOM</button>
    </div>
</form>
</div>
</div>
</div>
</body> </html>

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <link rel="stylesheet" href="table/style.css">
    <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
    <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/perfect-scrollbar/perfect-scrollbar.css">
    <link rel="stylesheet" type="text/css" href="css/util.css">
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              String dept = (String) session.getAttribute("department");
              out.print("Hello, ");
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>

```

```

        <li class="nav-item">
            <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
            <ul class="dropdown-menu" aria-labelledby="manage">
                <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
                <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
                <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
            </ul>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="logout"><span></span> Logout</a></li>
    </ul>
</div>
</div>
</nav>
<% @page import="java.sql.DriverManager;"%>
<% @page import="java.sql.ResultSet;"%>
<% @page import="java.sql.PreparedStatement;"%>
<% @page import="java.sql.Connection;"%>
<%
String id = request.getParameter("username");
String driverName = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://localhost:3306/";
String dbName = "schedules";
String userId = "root";
String password = "";
try {
    Class.forName(driverName);
} catch (Exception e) {
    out.println("Error type: " + e);
}

Connection connection = null;
PreparedStatement statement = null;
ResultSet resultSet = null;
%>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">

            <div class="card-body">

                <div class="limiter">
                    <div class="container-table100">
                        <div class="wrap-table100">
                            <div class="table100">
                                <table>

```



```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <link rel="stylesheet" href="table/style.css">
    <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
    <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/perfect-scrollbar/perfect-scrollbar.css">
    <link rel="stylesheet" type="text/css" href="css/util.css">
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
    <script>
      $(document).ready(function () {
        document.getElementById("year").value = localStorage.getItem("item1");
        document.getElementById("semester").value = localStorage.getItem("item2");
      });
    </script>
    <script>
      $(window).on('beforeunload', function () {
        localStorage.setItem("item1", document.getElementById("year").value);
        localStorage.setItem("item2", document.getElementById("semester").value);
      });
    </script>
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>

```

```

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
            <li class="nav-item">
                <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
                <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
                <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
                <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
                <ul class="dropdown-menu" aria-labelledby="manage">
                    <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
                    <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                    <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
                    <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
                </ul>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="logout"><span></span> Logout</a></li>
        </ul>
    </div>
</div>
</nav>
<% @page import="java.sql.DriverManager;"%>
<% @page import="java.sql.ResultSet;"%>
<% @page import="java.sql.PreparedStatement;"%>
<% @page import="java.sql.Connection;"%>
<%
String id = request.getParameter("username");
String driverName = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://localhost:3306/";
String dbName = "schedules";
String userId = "root";
String password = "";
try {
    Class.forName(driverName);
} catch (Exception e) {
    out.println("Error type: " + e);
}
Connection connection = null;
PreparedStatement statement = null;

```

```

    ResultSet resultSet = null;
%>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
  <div class="wrapper wrapper--w900">
    <div class="card card-6">
      <div class="card-body">
        <div class="limiter" id="outputDiv">
          <form action="" method="get" name="myform" enctype="multipart/form-data" >
            <div class="form-row">
              <div class="name" for="year">Year</div>
              <div class="value">
                <SELECT class="form-control" id="year" name="year">
                  <option disabled="" value="" selected="">Select Year</option>
                  <option value="ALL" selected="" >ALL</option>
                  <option value="2019" >2019</option>
                  <option value="2020" >2020</option>
                  <option value="2021" >2021</option>
                  <option value="2022" >2022</option>
                  <option value="2023" >2023</option>
                  <option value="2024" >2024</option>
                  <option value="2025" >2025</option>
                  <option value="2026" >2026</option>
                  <option value="2027" >2027</option>
                  <option value="2028" >2028</option>
                  <option value="2029" >2029</option>
                  <option value="2030" >2030</option>
                </select>
              </div>
            </div>
            <div class="form-row" >
              <div class="name" for="semester">Semester</div>
              <div class="value">
                <SELECT class="form-control" id="semester" name="semester">
                  <option disabled="" value="" selected="">Select Semester</option>
                  <option value="ALL" selected="" >ALL</option>
                  <option value="FALL" >FALL</option>
                  <option value="SPRING" >SPRING</option>
                  <option value="SUMMER" >SUMMER</option>
                </select>
              </div>
            </div>
            <div class="footer" >
              <button type="submit" class="btn btn--radius-2 btn--blue-2" type="submit"
>Search</button>
            </div>
          </form>
        <div class="container-table100">
          <div class="wrap-table100">
            <div class="table-wrapper-scroll-y">
              <table class="table table-bordered table-striped">
                <thead>
                  <tr class="table100-head">
                    <th class="column1">OpenCourseID</th>
                    <th class="column2">Code</th>
                    <th class="column3">GroupNo</th>
                    <th class="column4">Max Student</th>
                    <th class="column5">Program</th>
                    <th class="column6">Semester</th>
                    <th class="column6">Year</th>

```

```

        <th class="column7">Action</th>
    </tr>
</thead>
<%
    try {
        String name = (String) session.getAttribute("username");
        String dept = (String) session.getAttribute("department");
        connection = DriverManager.getConnection(connectionUrl +
dbName, userId, password);

        String year = request.getParameter("year");
        String semester = request.getParameter("semester");
        String sql;
        String error = "me";
        request.getSession().setAttribute("year", year);
        request.getSession().setAttribute("semester", semester);

        if ((year.equals("ALL")) || (year.equals("ALL"))) {
            sql = "SELECT * FROM `opencourse` WHERE department=" +
dept + """;
        } else if ((year != "ALL") && (semester != "ALL")) {
            sql = "SELECT * FROM `opencourse` WHERE year=" + year + ""
and semester=" + semester + "" and department=" + dept + """;
        } else {
            sql = "SELECT * FROM opencourse WHERE year=" + error +
""";
        }

        statement = connection.prepareStatement(sql);
        resultSet = statement.executeQuery();
        while (resultSet.next()) {
    %>
    <tr>
        <td class="column1"><%=resultSet.getString("courseGroup")%></td>
        <td class="column2"><%=resultSet.getString("courseCode")%></td>
        <td class="column3"><%=resultSet.getString("groupNo")%></td>
        <td class="column4"><%=resultSet.getString("maxStudent")%></td>
        <td class="column5"><%=resultSet.getString("program")%></td>
        <td class="column6"><%=resultSet.getString("semester")%></td>
        <td class="column6"><%=resultSet.getString("year")%></td>
        <td
                                class="column7"><a
href='edOpen.jsp?d=<%=resultSet.getString("courseGroup")%>' >EDIT</A> &nbsp; <a
href='delOpen?d=<%=resultSet.getString("courseGroup")%>'>DELETE</A></td>
    </tr>
    <%
        }
    } catch (Exception e) {
        out.println("No records found ");
    }
    %>
</tbody>
</table>
</div>
</div>
</div>
</div>
<a href="deleteOpen" class="btn btn--radius-2 btn--blue-2">DELETE ALL</a>

```

```
        <a href="opencourse.jsp" class="btn btn--radius-2 btn--blue-2">ADD NEW
COURSEOPENING</a>
    </div>
</div>
</div>
</div>
<script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<script src="vendor/bootstrap/js/popper.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<script src="vendor/select2/select2.min.js"></script>
<script src="js/main.js"></script>
</body>
</html>
```

A.7. EDIT COURSE

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>edit Course</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
  </head>
  <body id="page-top" class="index">

    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
              <ul class="dropdown-menu" aria-labelledby="manage">
```

```

        <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
        <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
        <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
        <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
    </ul>
</li>
<li class="nav-item">
    <a class="nav-link" href="logout"><span></span> Logout</a></li>
</ul>
</div>
</div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">EDIT COURSES</h2>
            </div>
            <div class="card-body">
                <%
                    if (request.getParameter("d") != null) {
                        String course_id = request.getParameter("d");
                        try {
                            Class.forName("com.mysql.jdbc.Driver");
                            Connection conn
                                DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                            PreparedStatement st = null;
                            st = conn.prepareStatement("select * from course where course_id=" +
                                course_id + "");
                            ResultSet rs = st.executeQuery();
                            while (rs.next()) {
                                %>
                                <form method= "POST" action="updatecourse">
                                    <div class="form-row">
                                        <div class="name" for="subcode">Course Code</div>
                                        <div class="value">
                                            <input id="subcode" name="subcode" type="text" placeholder="" class="input-
                                -style-6" value="<%=rs.getString("course_id")%>">
                                        </div>
                                    </div>
                                    <!-- Text input-->
                                    <div class="form-row">
                                        <div class="name" for="subdescription">Course Name</div>
                                        <div class="value">
                                            <input id="subdescription" name="subdescription" type="text" placeholder=""
                                class="input--style-6" value="<%=rs.getString("course_title")%>">
                                        </div>
                                    </div>
                                    <div class="form-row">
                                        <div class="name" for="subdescription">Prerequisite</div>
                                        <div class="value">
                                            <input id="prerequisite" name="prerequisite" type="text" placeholder=""
                                class="input--style-6" value="<%=rs.getString("course_prerequisite")%>">
                                        </div>
                                    </div>
                                </div>

```



```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
              <ul class="dropdown-menu" aria-labelledby="manage">

```

```

Room</a></li>
    <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Course</a></li>
    <li><a class="dropdown-item" href="courselist.jsp"><span></span> View
CourseOpening</a></li>
    <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
Timetable</a></li>
</ul>
</li>
<li class="nav-item">
    <a class="nav-link" href="logout"><span></span> Logout</a></li>
</ul>
</div>
</div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">Add CourseOpening</h2>
            </div>
            <div class="card-body">
                <%
String yearr = (String) request.getSession().getAttribute("year");
String semesterr = (String) request.getSession().getAttribute("semester");
if (request.getParameter("d") != null) {
String course_id = request.getParameter("d");
try {
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
PreparedStatement st = null;
st = conn.prepareStatement("select * from opencourse where courseGroup=" +
course_id + """);
ResultSet rs = st.executeQuery();
while (rs.next()) {

%>
<form method= "POST" action="updateOpen">
    <fieldset>
        <input id="courseGroup" name="courseGroup" type="hidden" placeholder=""
class="input--style-6" value="<%=rs.getString("courseGroup")%>">
        <div class="form-row">
            <div class="name" for="course_id">Course Code</div>
            <div class="value">
                <input id="course_id" name="course_id" type="text" placeholder=""
class="input--style-6" value="<%=rs.getString("courseCode")%>">
            </div>
        </div>
        <div class="form-row">
            <div class="name" for="group">Group</div>
            <div class="value">
                <input id="group" name="group" type="text" placeholder="" class="input--
style-6" value="<%=rs.getString("groupNo")%>">
            </div>

```

```

        </div>
        <div class="form-row">
            <div class="name" for="student">Max Student</div>
            <div class="value">
                <input id="maxStd" name="maxStd" type="text" placeholder=""
class="input--style-6" value="<%=rs.getString("maxStudent")%>">
            </div>
        </div>
        <div class="form-row">
            <div class="name" for="program">Program</div>
            <div class="value">
                <input id="program" name="program" type="text" placeholder=""
class="input--style-6" value="<%=rs.getString("program")%>">
            </div>
        </div>
        <div class="form-row">
            <div class="name" for="year">Year</div>
            <div class="value">
                <input id="year" name="year" type="text" placeholder="" class="input--
style-6" value="<%=rs.getString("year")%>">
            </div>
        </div>
        <div class="form-row">
            <div class="name" for="semester">Semester</div>
            <div class="value">
                <input id="semester" name="semester" type="text" placeholder=""
class="input--style-6" value="<%=rs.getString("semester")%>">
            </div>
        </div>
        </div>
        <div class="footer" >
            <button class="btn btn--radius-2 btn--blue-2" type="submit"
name="edit">UPDATE COURSE</button>
        </div>
    </form>
    <% }
        st.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
%>
</div>
</div>
</div>
</body>
</html>

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              out.print("Hello, " + name);
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
              <ul class="dropdown-menu" aria-labelledby="manage">

```

```

        <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
        <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
        <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
        <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
    </ul>
</li>
<li class="nav-item">
    <a class="nav-link" href="logout"><span></span> Logout</a></li>
</ul>
</div>
</div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="card card-6">
            <div class="card-heading">
                <h2 class="title">EDIT ROOM</h2>
            </div>
            <div class="card-body">
                <%
                    if (request.getParameter("d") != null) {
                        String course_id = request.getParameter("d");
                        try {
                            Class.forName("com.mysql.jdbc.Driver");
                            Connection conn
                                DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                            PreparedStatement st = null;
                            st = conn.prepareStatement("select * from room where room_id=" + course_id
                                + "");
                            ResultSet rs = st.executeQuery();
                            while (rs.next()) {
                                %>
                                <form method= "POST" action="updateRoom">
                                    <div class="form-row">
                                        <div class="name" for="room">Room Number</div>
                                        <div class="value">
                                            <input id="roomNo" name="roomNo" type="text" placeholder=""
class="input--style-6" value="<%=rs.getString("room_id")%>">
                                        </div>
                                    </div>
                                    <div class="form-row">
                                        <div class="name" for="dept">Department</div>
                                        <div class="value">
                                            <input id="dept" name="dept" type="text" placeholder="" class="input--style-
6" value="<%=rs.getString("department")%>">
                                        </div>
                                    </div>
                                    <div class="form-row">
                                        <div class="name" for="faculty">Faculty</div>
                                        <div class="value">
                                            <input id="faculty" name="faculty" type="text" placeholder="" class="input--
style-6" value="<%=rs.getString("faculty")%>">
                                        </div>
                                    </div>
                                <div class="form-row">

```

```

        <div class="name" for="capacity">Capacity</div>
        <div class="value">
            <input id="capacity" name="capacity" type="text" placeholder=""
class="input--style-6" value="<%=rs.getString("capacity")%>">
        </div>
        </div>
    </div>
    <div class="footer" >
        <button class="btn btn--radius-2 btn--blue-2" type="submit" name="edit">UPDATE
ROOM</button>
    </div>
</form>
<% }
    st.close();
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
%>
</div>
</div>
</div>
</body>
</html>

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
  </head>
  <body id="page-top" class="index">
    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
      <div class="container">
        <%
          if (session != null) {
            if (session.getAttribute("username") != null) {
              String name = (String) session.getAttribute("username");
              //out.print("Hello, " + name + " welcome");
            } else {
              response.sendRedirect("login.jsp");
            }
          }
        %>
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>

```



```

        <ul class="dropdown-menu" aria-labelledby="manage">
            <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
            <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
            <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
            <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
        </ul>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="logout"><span></span> Logout</a></li>
    </ul>
</div>
</div>
</nav>
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="contentArea">
            <section class="bg-light" id="portfolio">
                <div class="container">
                    <div class="row">
                        <div class="col-lg-12 text-center">
                            <h2 class="section-heading text-uppercase">Features</h2>
                        </div>
                    </div>
                    <div class="row">
                        <div class="col-md-4 col-sm-6 portfolio-item">
                            <a class="portfolio-link" href="addcourse.jsp">
                                <div class="portfolio-hover">
                                    <div class="portfolio-hover-content">
                                        <i class="fas fa-plus fa-3x"></i>
                                    </div>
                                </div>
                                
                            </a>
                            <div class="portfolio-caption">
                                <h4 >Add Course</h4>
                                <p class="text-muted">Add new Courses with Max number of hours per
week</p>
                            </div>
                        </div>
                        <div class="col-md-4 col-sm-6 portfolio-item">
                            <a class="portfolio-link" href="addroom.jsp">
                                <div class="portfolio-hover">
                                    <div class="portfolio-hover-content">
                                        <i class="fas fa-plus fa-3x"></i>
                                    </div>
                                </div>
                                
                            </a>
                            <div class="portfolio-caption">
                                <h4 >Add Room</h4>
                                <p class="text-muted">Add new room keeping in mind the capacity of the
classroom</p>
                            </div>
                        </div>
                    </div>
                </div>
            </section>
        </div>
    </div>
</div>
<div class="col-md-4 col-sm-6 portfolio-item">

```

```

<a class="portfolio-link" href="opencourse.jsp">
  <div class="portfolio-hover">
    <div class="portfolio-hover-content">
      <i class="fas fa-plus fa-3x"></i>
    </div>
  </div>
  
</a>
<div class="portfolio-caption">
  <h4 >Add Course Opening</h4>
  <p class="text-muted">Add new Courses that will be opening for a particular
semester and year</p>
</div>
</div>

<div class="col-md-4 col-sm-6 portfolio-item">
  <a href="adddept.jsp">
    <div class="portfolio-hover">
      <div class="portfolio-hover-content">
        <i class="fas fa-plus fa-3x"></i>
      </div>
    </div>
    
  </a>
  <div class="portfolio-caption">
    <h4 >Add Department</h4>
    <p class="text-muted">Add new department to for every faculty</p>
  </div>
</div>

<div class="col-md-4 col-sm-6 portfolio-item">
  <a class="portfolio-link" href="timetable.jsp">
    <div class="portfolio-hover">
      <div class="portfolio-hover-content">
        <i class="fas fa-plus fa-3x"></i>
      </div>
    </div>
    
  </a>
  <div class="portfolio-caption">
    <h4 >Generate Timetable</h4>
    <p class="text-muted">Generate timetable based on inputted data</p>
  </div>
</div>

<div class="col-md-4 col-sm-6 portfolio-item">
  <a class="portfolio-link" href="show.jsp">
    <div class="portfolio-hover">
      <div class="portfolio-hover-content">
        <i class="fas fa-plus fa-3x"></i>
      </div>
    </div>
    
  </a>
  <div class="portfolio-caption">
    <h4 >View Timetable</h4>
    <p class="text-muted">Manage your data by viewing generated timetable
</p>
  </div>
</div>

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Timetable index</title>
    <!-- Bootstrap core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>

  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
              <ul class="dropdown-menu" aria-labelledby="manage">
                <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
                <li><a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
                <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
              </ul>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </body>
</html>

```

```

        </li>
    </ul>
</div>
</div>
</nav>
<!-- Header -->
<div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
        <div class="contentArea">
            <h4 class="subheading"><font color="GOLD">Introduction &nbsp;</font></h4>

            <p class="text-muted"><font color="white">
                This is a free site to generate
                complete college schedule for which would be a tedious and
                error prone task to do by hand (on paper). We provide a very
                user friendly interface to work with and you will get your
                time table generated in seconds.</font></p>
        </div>
        <h4 class="subheading"><font color="GOLD">How to Proceed &nbsp;</font></h4>
        <p class="text-muted"><font color="white">You need to be logged in to make use
            of this free software. Once logged in, you will need to
            provide important details like batch names,subjects, faculties
            in college etc and you are ready to go.</font></p>
        <h4 class="subheading"><font color="GOLD">Quality of Timetable &nbsp;</font> </h4>
        <p class="text-muted"><font color="white">The software makes sure that
            generated time table is feasible at the least (there is no
            clash) ,students get required hours each week and teachers are
            given equal load. The user will be given more options to
            refine timetable using some soft constraints(like setting
            preference for a teacher,guest faculty, lab consideration etc)
            in a later version.</font></p>
        <a href="login.jsp" class="btn btn--radius-2 btn--blue-2">
            <div class="timeline-image">
                <h4>LOGIN</h4>
            </div>
        </a>
    </div>
</div>
</body>
</html>

```

A.12. LOGIN

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">

```

```

<link href="css/main.css" rel="stylesheet" media="all">
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
</head>
<body id="page-top" class="index">
  <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
    <div class="container">
      <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
          <li class="nav-item">
            <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
          <li class="nav-item">
            <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
          <li class="nav-item">
            <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
            <ul class="dropdown-menu" aria-labelledby="manage">
              <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
              <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
              <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
              <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
            </ul>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <div class="page-wrapper bg-dark p-t-100 p-b-50">
    <div class="wrapper wrapper--w900">
      <div class="card card-6">
        <div class="card-heading">
          <h2 class="title">LOGIN HERE</h2>
        </div>
        <div class="card-body">
          <form method="post" action="login">
            <div class="form-row">
              <div class="name" for="username">Username</div>
              <div class="value">
                <input id="username" name="username" type="text" placeholder=""
class="input--style-6" required="">
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
        <div class="form-row">
            <div class="name" for="password">Password</div>
            <div class="value">
                <input id="password" name="password" type="password" placeholder=""
class="input--style-6" required="">
            </div>
        </div>
        <div class="form-row">
            <div class="name" for="password">Department</div>
            <div class="value">
                <select name="department" name="department" value="select your
department" class="input--style-6" >
                    <%
                        Connection con = null;
                        ResultSet rs = null;
                        try {
                            Class.forName("com.mysql.jdbc.Driver");
                            con
                                =
                                DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                                HttpSession ss = request.getSession();
                                PreparedStatement pst = con.prepareStatement("select * from
                                department");

                                rs = pst.executeQuery();
                                while (rs.next()) {
                                    String name = rs.getString("name");
                                }
                                <option value="<%=name%>"><%=name%></option>
                                <%
                                    }
                                } catch (Exception e) {
                                    out.print(e);
                                }
                                <%
                                    </select>
                                </div>
                            </div>
                        </div>
                    <div class="footer" >
                        <input class="btn btn--radius-2 btn--blue-2" name="lgn" type="submit"
value="LOGIN">
                    </div>
                </form>
                <div class="label--desc">Don't have an account?</font> <a href="register.jsp">Register
here </a></div>
            </div>
        </div>
    </div>
</body>
</html>

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Custom fonts for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/one-page-wonder.min.css" rel="stylesheet">
    <link href="css/main.css" rel="stylesheet" media="all">
  </head>
  <body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
      <div class="container">
        <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
            <li class="nav-item">
              <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
            <li class="nav-item">
              <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
            <li class="nav-item">
              <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
              <ul class="dropdown-menu" aria-labelledby="manage">
                <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
                <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
                <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
              </ul>
            </li>
          </ul>
        </div>
      </div>
    </nav>

```

```

<div class="page-wrapper bg-dark p-t-100 p-b-50">
  <div class="wrapper wrapper--w900">
    <div class="card card-6">
      <div class="card-heading">
        <h2 class="title">ADD ACCOUNT</h2>
      </div>
      <div class="card-body">
        <form class="form-horizontal" method="post" action="reg">
          <div class="form-row">
            <div class="name" for="password">Username</div>
            <div class="value">
              <input id="username" name="username" type="text" placeholder=""
class="input--style-6" required="">
            </div>
          </div>
          <div class="form-row">
            <div class="name" for="password">Password</div>
            <div class="value">
              <input id="password" name="password" type="password" placeholder=""
class="input--style-6" required="">
            </div>
          </div>
          <div class="form-row">
            <div class="name" for="password">Department</div>
            <div class="value">
              <select name="department" name="department" value="select your
department" class="input--style-6" >
                <%
                  Connection con = null;
                  ResultSet rs = null;
                  try {
                    Class.forName("com.mysql.jdbc.Driver");
                    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                    HttpSession ss = request.getSession();

                    PreparedStatement pst = con.prepareStatement("select * from
department");

                    rs = pst.executeQuery();
                    while (rs.next()) {
                      String name = rs.getString("name");
                    }
                    <option value="<%=name%>"><%=name%></option>
                    <%
                      }
                    } catch (Exception e) {
                      out.print(e);
                    }
                    <%
                  </select>
                </div>
              </div>
            </div>
          <div class="footer" >
            <button class="btn btn--radius-2 btn--blue-2" type="submit">ADD</button>
          </div>
        </form>
      </div>
      <font color="grey">Already have an account?</font> <a href="index.jsp">Login here </a>

```



```

    </div>
</div>

</body>
</html>

```

A.14. VIEW PAGE

```

<%
    String username = request.getParameter("username");
    String driverName = "com.mysql.jdbc.Driver";
    String connectionUrl = "jdbc:mysql://localhost:3306/";
    String dbName = "schedules";
    String userId = "root";
    String password = "";
    try {
        Class.forName(driverName);
    } catch (Exception e) {
        out.println("Error type: " + e);
    }
    Connection connection = null;
    PreparedStatement statement = null;
    ResultSet resultSet = null;
%>

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
        <!-- Custom fonts for this template -->
        <link
href="https://fonts.googleapis.com/css?family=Catamaran:100,200,300,400,500,600,700,800,900"
rel="stylesheet">
        <link
href="https://fonts.googleapis.com/css?family=Lato:100,100i,300,300i,400,400i,700,700i,900,900i"
rel="stylesheet">
        <!-- Custom styles for this template -->
        <link href="css/one-page-wonder.min.css" rel="stylesheet">
        <link href="css/main.css" rel="stylesheet" media="all">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
        <!-- Include all compiled plugins (below), or include individual files as needed -->
        <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
        <!-- Include all compiled plugins (below), or include individual files as needed -->
        <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
        <script>
            $(document).ready(function () {
                document.getElementById("year").value = localStorage.getItem("item1");
                document.getElementById("semester").value = localStorage.getItem("item2");
            });
        </script>
        <script>
            $(window).on('beforeunload', function () {
                localStorage.setItem("item1", document.getElementById("year").value);

```

```

        localStorage.setItem("item2", document.getElementById("semester").value);
    });
</script>
</head>
<body id="page-top" class="index">
    <nav class="navbar navbar-expand-lg navbar-dark navbar-custom fixed-top">
        <div class="container">
            <%
                if (session != null) {
                    if (session.getAttribute("username") != null) {
                        String name = (String) session.getAttribute("username");
                        out.print("Hello, " + name);
                    } else {
                        response.sendRedirect("login.jsp");
                    }
                }
            %>

            <a class="navbar-brand" href="home.jsp">SCHEDULING</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="addcourse.jsp"><span></span> Add Course</a></li>
                    <li class="nav-item">
                        <a class="nav-link" href="addroom.jsp"><span></span> Add Room</a></li>
                    <li class="nav-item">
                        <a class="nav-link" href="opencourse.jsp"><span></span> Add
CourseOpening</a></li>
                    <li class="nav-item">
                        <a class="nav-link" href="timetable.jsp"><span></span> Generate
Timetable</a></li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" id="manage" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"><span></span> Manage</a>
                        <ul class="dropdown-menu" aria-labelledby="manage">
                            <li><a class="dropdown-item" href="roomlist.jsp"><span></span> View
Room</a></li>
                            <li> <a class="dropdown-item" href="courselist.jsp"><span></span> View
Course</a></li>
                            <li><a class="dropdown-item" href="opencourselist.jsp"><span></span> View
CourseOpening</a></li>
                            <li><a class="dropdown-item" href="view.jsp"><span></span> View
Timetable</a></li>
                        </ul>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="logout"><span></span> Logout</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="page-wrapper bg-dark p-t-100 p-b-50">
        <div class="wrapper wrapper--w900">
            <div class="card card-6">

```

```

<div class="card-body">
  <form action="" method="get">
    <div class="form-row">
      <div class="name" for="year">Year</div>
      <div class="value">
        <select class="form-control" id="year" name="year">
          <option disabled="" value="" selected="">Select Year</option>
          <option value="2019" >2019</option>
          <option value="2020" >2020</option>
          <option value="2021" >2021</option>
          <option value="2022" >2022</option>
          <option value="2023" >2023</option>
          <option value="2024" >2024</option>
          <option value="2025" >2025</option>
          <option value="2026" >2026</option>
          <option value="2027" >2027</option>
          <option value="2028" >2028</option>
          <option value="2029" >2029</option>
          <option value="2030" >2030</option>
        </select>
      </div>
    </div>
    <div class="form-row" >
      <div class="name" for="semester">Semester</div>
      <div class="value">
        <SELECT class="form-control" id="semester" name="semester">
          <option disabled="" value="" selected="">Select Semester</option>
          <option value="FALL" >FALL</option>
          <option value="SPRING" >SPRING</option>
          <option value="SUMMER" >SUMMER</option>
        </select>
      </div>
    </div>
    <div class="footer" >
      <button type="submit" class="btn btn--radius-2 btn--blue-2"
type="submit">Search</button>
    </div>
  </form>
  <div class="container-table100">
    <div class="wrap-table100">
      <div class="table-wrapper-scroll-y">
        <table class="table table-bordered table-striped">
          <thead>
            <tr class="table100-head">
              <th class="column1">Course Code</th>
              <th class="column2">Period</th>
              <th class="column3">Group</th>
              <th class="column4">Room</th>
              <th class="column5">Day</th>
            </tr>
          </thead>
          <tbody>
            <%
              try {
                String name = (String) session.getAttribute("username");
                String deptt = (String) session.getAttribute("department");
                connection = DriverManager.getConnection(connectionUrl +
dbName, userId, password);
                String year = request.getParameter("year");

```

```

String semester = request.getParameter("semester");
request.getSession().setAttribute("year", year);
request.getSession().setAttribute("semester", semester);
String error = "me";
String sql;
if ((year != null) && (semester != null)) {
    sql = "SELECT * FROM `timetable` WHERE department='" +
dept + "' and year='" + year + "' and semester='" + semester + "'";

    } else {
    sql = "SELECT * FROM `timetable` WHERE year='" + error + "'";
    }
statement = connection.prepareStatement(sql);
resultSet = statement.executeQuery();
while (resultSet.next()) {
%>
<tr>
    <td class="column1"><%=resultSet.getString("course")%></td>
    <td class="column2"><%=resultSet.getString("period")%></td>
    <td class="column3"><%=resultSet.getString("groupNo")%></td>
    <td class="column3"><%=resultSet.getString("room")%></td>
    <td class="column4"><%=resultSet.getString("day")%></td>
</tr>
<%
    }

    } catch (Exception e) {
        out.println("No records found");

    }
%>
</tbody>
</table>
</div>
</div>
</DIV>
</div>
</div>
<div class="footer" >
    <button type="submit" class="btn btn--radius-2 btn--blue-2" onClick="window.print()">
PRINT</button>
    <a href="delTime" class="btn btn--radius-2 btn--blue-2">DELETE TIMETABLE</a>
</div>
</div>
</div>
<script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<script src="vendor/bootstrap/js/popper.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<script src="vendor/select2/select2.min.js"></script>
<script src="js/main.js"></script>
</body>
</html>

```

```

public class call extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String year = request.getParameter("year");
            String semester = request.getParameter("semester");
            String uname = request.getParameter("username");
            String dept = request.getParameter("department");
            request.getSession().setAttribute("uname", uname);
            request.getSession().setAttribute("dept", dept);
            request.getSession().setAttribute("year", year);
            request.getSession().setAttribute("semester", semester);
            List c_data = new ArrayList();
            List r_data = new ArrayList();
            List cp_data = new ArrayList();
            try {
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/schedules?", "root", "");
                Statement st = conn.createStatement();
                ResultSet rs = st.executeQuery("Select * from course where department='" + dept + "'");
                while (rs.next()) {
                    String id = rs.getString("course_id");
                    String name = rs.getString("course_title");
                    String prerequisite = rs.getString("course_prerequisite");
                    String language = rs.getString("course_language");
                    String creditUnit = rs.getString("course_credit_unit");
                    String lecturehr = rs.getString("lecture_hour");
                    String labhr = rs.getString("lab_hour");
                    String ectsh = rs.getString("ects");
                    c_data.add(id + ":Course" + " " + "[" + " " + " " + "courseCode->" + "" + id + "" + "," + " "
+ "courseName->" + "" + name + "" + "," + " " + "hasPrerequisite->" + prerequisite
+ "," + " " + "instructionLanguage->" + language + "," + " " + "credits->" + creditUnit + "," + " " + "
lecture_hours->" + lecturehr + "," + " " + "lab_hours->" + labhr + "," + " " + "ects->" + ectsh + " " +
"].");
                }
                rs = st.executeQuery("Select * from room where department='" + dept + "'");
                while (rs.next()) {
                    String id = rs.getString("room_id");
                    //String name=rs.getString("room_name");

```

```

        String faculty = rs.getString("faculty");
        String department = rs.getString("department");
        String capacity = rs.getString("capacity");
        r_data.add(id + ":Classroom" + " " + "[" + " " + " " + "roomNumber->" + id + "," + " "
            + "location->" + " " + faculty + "" + "," + " " + "inDepartment->" + " " +
department + "" + "," + " " + "capacity->" + capacity + "" + "].");
    }
    rs = st.executeQuery("select * from opencourse where department='" + dept + "'");
    while (rs.next()) {
        String c_group = rs.getString("groupNo");
        String c_id = rs.getString("courseCode");
        String c_maxStd = rs.getString("maxStudent");
        String c_year = rs.getString("year");
        String c_program = rs.getString("program");
        String c_semester = rs.getString("semester");
        cp_data.add("request_" + c_id + "_" + c_group + ":CourseOpeningRequest" + " " + "[" +
" " + " " + "groupNumber->" + c_group + "," + " "
            + "ofCourse->" + c_id + "," + " " + "maxStudentSize->" + c_maxStd + "," + " " +
"year->" + c_year + "," + " " + "forProgram ->" + c_program + " " + "," + " " + "semester->" +
c_semester + " " + "].");
    }
    writeToFile(cp_data,
"C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/course_opening_request
s.flr");
    writeToFile(r_data,
"C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/classroom_instances.flr"
);
    writeToFile(c_data,
"C:/Users/reslab/Documents/NetBeansProjects/pamthesis/src/java/timetable1/course_instances.flr");
    String delete = "DELETE FROM `timetable` WHERE `year` = '" + year + "' and `year` = '"
+ semester + "' and `department` = '" + dept + "'";
    pstmt = conn.prepareStatement(delete);
    pstmt.executeUpdate();
    pstmt.close();
    rs.close();
    st.close();
    //out.println("<a href='time'> data saved click here to view timetable</a>");
    response.sendRedirect("delTime");

    } catch (Exception e) {
        out.println(e);
    }
}
}

private static void writeToFile(java.util.List list, String path) {
    BufferedWriter out = null;
    try {
        File file = new File(path);
        out = new BufferedWriter(new FileWriter(file, false));
        for (Object s : list) {
            out.write(s.toString());
            //reArrange(s.split(",");
            out.newLine();
        }
        out.close();
    } catch (IOException e) {
    }
}

```

}

```

public class course extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        boolean status = false;
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            List data = new ArrayList();
            String name = request.getParameter("username");
            //String dept=(String ) session.getAttribute("department");
            String c_id = request.getParameter("subcode");
            String c_title = request.getParameter("subdescription");
            String c_prerequisite = request.getParameter("prerequisite");
            String c_language = request.getParameter("language");
            int c_c_unit = Integer.parseInt(request.getParameter("cUnit"));
            String c_lecture_hour = request.getParameter("lechr");
            String c_lab_hour = request.getParameter("labhr");
            int c_ects = Integer.parseInt(request.getParameter("ects"));
            String c_dept = request.getParameter("department");
            //query
            String query = "SELECT * FROM `schedules`.`course` WHERE `course_id` = ?";
            try {
                pstmt = conn.prepareStatement(query);
                pstmt.setString(1, c_id);
                rs = pstmt.executeQuery();
                //check if the course has been registered before
                if (rs.next()) {
                    out.println("This course exists!");
                    rs.close();
                    pstmt.close();
                    return;
                } //if not then register it
                else if (!rs.next()) {

                    String sql = "INSERT INTO `course`(`course_id`, `course_title`,
                        + "`course_prerequisite`, `course_language`, `course_credit_unit`,
                        + "`lecture_hour`, `lab_hour`, `ects`, `department`, `user`)"
                        + " VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
                    pstmt = conn.prepareStatement(sql);
                    pstmt.setString(1, c_id);
                    pstmt.setString(2, c_title);
                    pstmt.setString(3, c_prerequisite);
                    pstmt.setString(4, c_language);
                    pstmt.setString(5, String.valueOf(c_c_unit));
                    pstmt.setString(6, c_lecture_hour);
                    pstmt.setString(7, c_lab_hour);
                    pstmt.setString(8, String.valueOf(c_ects));
                    pstmt.setString(9, c_dept);
                    pstmt.setString(10, name);
                    pstmt.executeUpdate();
                    data.add(c_id + ":Course" + " " + "[" + " " + " " + "courseCode->" + "" + c_id + "" + " "
+ " "

```



```

        + "courseName->" + "" + c_title + "" + "," + " " + "hasPrerequisite->" +
c_prerequisite + "," + " " + "instructionLanguage->" + c_language + "," + " " + "credits->" + c_c_unit
+ ""
        + "," + " " + "lecture_hours->" + c_lecture_hour + " " + "," + " " + "lab_hours->" +
c_lab_hour + " " + "," + " " + "ects->" + c_ects + " " + "].");
        writeToFile(data,
"C:/Users/Pamilerin/Downloads/pamthesis/src/java/timetable1/course_instances.flr");
        request.setAttribute("alertMsg", "inserted successfully");
        response.sendRedirect("courselist.jsp");

        } //if not registered prompt the user this error msg
        else {
            request.setAttribute("alertMsg", "invalid course id");

        }
    } catch (Exception e) {
        out.println("Error: " + e);
    }
}
}
private static void writeToFile(java.util.List list, String path) throws IOException, IOException {
    BufferedWriter out = null;
    try {
        File file = new File(path);
        out = new BufferedWriter(new FileWriter(file, true));
        for (Object s : list) {
            out.write(s.toString());
            out.newLine();
        }
        out.close();
    } catch (IOException ex) {
        Logger.getLogger(course.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
}

```

```
public class delCourse extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String course_id= request.getParameter("d");
            //out.println("course="+course_id);
            String delete = "DELETE FROM `course` WHERE `course_id` = '"+course_id+"'";
            try{
                pstmt=conn.prepareStatement(delete);
                int i= pstmt.executeUpdate();
                //check if the course has been registered before
                if (i>0) {
                    out.println( "This course has been deleted!");
                    request.setAttribute("alertMsg", "deleted successfully");
                    response.sendRedirect("courselist.jsp");
                    pstmt.close();
                    return;
                }
            }
            catch (Exception e) {
                //out.println("course="+course_id);
                out.println( "Error: " + e);
            }
        }
    }
}
```

```

public class delOpen extends HttpServlet {

    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String course_id = request.getParameter("d");
            String yearr = (String) request.getSession().getAttribute("year");
            String semesterr = (String) request.getSession().getAttribute("semester");
            //out.println("course="+course_id);

            String delete = "DELETE FROM `opencourse` WHERE `courseGroup` = '" + course_id + "'";
            try {
                pstmt = conn.prepareStatement(delete);
                int i = pstmt.executeUpdate();
                //check if the course has been registered before
                if (i > 0) {
                    out.println("This course has been deleted!");
                    request.setAttribute("alertMsg", "deleted successfully");
                    response.sendRedirect("opencourselist.jsp?year=" + yearr + "&semester=" + semesterr);
                    pstmt.close();
                    return;
                }
            } catch (Exception e) {
                //out.println("course="+course_id);
                out.println("Error: " + e);
            }
        }
    }
}

```

```
public class delRoom extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String course_id = request.getParameter("d");
            //out.println("course="+course_id);
            String delete = "DELETE FROM `room` WHERE `room_id` = " + course_id + "";
            try {
                pstmt = conn.prepareStatement(delete);
                int i = pstmt.executeUpdate();
                //check if the course has been registered before
                if (i > 0) {
                    out.println("This course has been deleted!");
                    request.setAttribute("alertMsg", "deleted successfully");
                    response.sendRedirect("roomlist.jsp");
                    pstmt.close();
                    return;
                }
            } catch (Exception e) {
                //out.println("course="+course_id);
                out.println("Error: " + e);
            }
        }
    }
}
```

```

public class delTime extends HttpServlet {

    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String yearr = (String) request.getSession().getAttribute("year");
            String semesterr = (String) request.getSession().getAttribute("semester");
            String dept = (String) request.getSession().getAttribute("dept");
            String delete = "DELETE FROM `timetable` WHERE year=" + yearr + " and semester =" +
                semesterr + " and department=" + dept + """;
            try {
                pstmt = conn.prepareStatement(delete);
                pstmt.executeUpdate();
                //check if the course has been registered before
                pstmt.close();
                response.sendRedirect("time");
            } catch (Exception e) {
                //out.println("course="+course_id);
                out.println("Error: " + e);
            }
        }
    }
}

```

```

public class department extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            List data = new ArrayList();
            //variables
            String c_id = request.getParameter("department");
            String c_group = request.getParameter("faculty");
            //query
            String query = "SELECT * FROM `schedules`.`department` WHERE `department` = ?";
            try {
                pstmt = conn.prepareStatement(query);
                pstmt.setString(1, c_id);
                rs = pstmt.executeQuery();
                //check if the course has been registered before
                if (rs.next()) {
                    JOptionPane.showMessageDialog(null, "This course exists!");
                    rs.close();
                    pstmt.close();
                    return;
                } //if not then register it
                else if (!rs.next()) {
                    String sql = "INSERT INTO `department` (`department`, "
                        + "`faculty`)"
                        + " VALUES (?,?)";
                    pstmt = conn.prepareStatement(sql);
                    pstmt.setString(1, c_group);
                    pstmt.setString(2, c_id);
                    pstmt.executeUpdate();
                    rs.close();
                    pstmt.close();
                    out.println("Inserted Successfully!");
                    request.setAttribute("alertMsg", "inserted successfully");
                    response.sendRedirect("deptlist.jsp");
                } //if not registered prompt the user this error msg
                else {
                    out.println("Not Inserted!");
                }
            } catch (HeadlessException | SQLException e) {
                out.println("Error: " + e);
            }
        }
    }
}

```

```

public class open extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            List data = new ArrayList();
            String id = request.getParameter("username");
            String dept = request.getParameter("department");
            String c_id = request.getParameter("course_id");
            String c_group = request.getParameter("group");
            String c_maxStd = request.getParameter("maxStd");
            String c_program = request.getParameter("program");
            String c_year = request.getParameter("year");
            String c_semester = request.getParameter("semester");
            String c_cgrou = request.getParameter("c_cgrou");
            request.getParameter("course_id").concat("G").concat(request.getParameter("group"));
            request.getSession().setAttribute("year", c_year);
            request.getSession().setAttribute("semester", c_semester);
            //query
            String query = "SELECT * FROM `schedules`.`opencourse` WHERE `courseCode` = ? and
            `groupNo` = ? and `year` = ? and `semester` = ? and `department` = ?";
            try {
                pstmt = conn.prepareStatement(query);
                pstmt.setString(1, c_id);
                pstmt.setString(2, c_group);
                pstmt.setString(3, c_year);
                pstmt.setString(4, c_semester);
                pstmt.setString(5, dept);
                rs = pstmt.executeQuery();
                //check if the course has been registered before
                if (rs.next()) {
                    out.println("This course exists!");
                    rs.close();
                    pstmt.close();
                    return;
                } //if not then register it
                else if (!rs.next()) {
                    String sql = "INSERT INTO `opencourse` (`groupNo`, "
                        + "`courseCode`, `maxStudent`, `year`, `program`,
                        `semester`, `department`, `user`, `courseGroup`)"
                        + " VALUES (?,?,?,?,?,?";
                    pstmt = conn.prepareStatement(sql);
                    pstmt.setString(1, c_group);
                    pstmt.setString(2, c_id);
                    pstmt.setString(3, c_maxStd);
                    pstmt.setString(4, c_year);
                    pstmt.setString(5, c_program);
                    pstmt.setString(6, c_semester);
                    pstmt.setString(7, dept);
                    pstmt.setString(8, id);
                    pstmt.setString(9, c_cgrou);
                    pstmt.executeUpdate();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        data.add("request_" + c_id + "_" + c_group + ":CourseOpeningRequest" + " " + "[" + " "
+ " " + "groupNumber->" + "" + c_group + "" + "," + " "
        + "ofCourse->" + "" + c_id + "" + "," + " " + "maxStudentSize->" + c_maxStd + ","
+ " " + "year->" + c_year + "," + " " + "forProgram ->" + c_program + " " + "," + " " + "semester->" +
c_semester + " " + "].");
        writeToFile(data,
"C:/Users/Pamilerin/Downloads/pamthesis/src/java/timetable1/course_opening_requests.flr");
        rs.close();
        pstmt.close();

        out.println("Inserted Successfully!");
        request.setAttribute("alertMsg", "inserted successfully");
        response.sendRedirect("opencourse.jsp");
    } //if not registered prompt the user this error msg
    else {
        out.println("Not Inserted!");
    }
} catch (HeadlessException | SQLException e) {
    out.println("Error: " + e);
}
}
}
private static void writeToFile(java.util.List list, String path) {
    BufferedWriter out = null;
    try {
        File file = new File(path);
        out = new BufferedWriter(new FileWriter(file, true));
        for (Object s : list) {
            out.write(s.toString());
            //reArrange(s.split(","));
            out.newLine();
        }
        out.close();
    } catch (IOException e) {
    }
}
}

```



```

public class reg extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String c_id = request.getParameter("username");
            String pwd = request.getParameter("password");
            String dept = request.getParameter("department");
            String query = "SELECT * FROM `schedules`.`users` WHERE `username` = ?";
            try {
                pstmt = conn.prepareStatement(query);
                pstmt.setString(1, c_id);
                rs = pstmt.executeQuery();
                //check if the course has been registered before
                if (rs.next()) {

                    JOptionPane.showMessageDialog(null, "This room exists!");
                    rs.close();
                    pstmt.close();
                    return;
                } //if not then register it
                else if (!rs.next()) {
                    String sql = "INSERT INTO `users`(`username`, `password`, `department`)"
                        + " VALUES (?, ?, ?)";
                    pstmt = conn.prepareStatement(sql);
                    pstmt.setString(1, c_id);
                    pstmt.setString(2, pwd);
                    pstmt.setString(3, dept);
                    pstmt.executeUpdate();
                    request.setAttribute("alertMsg", "inserted successfully");
                    response.sendRedirect("login.jsp");
                } //if not registered prompt the user this error msg
                else {
                    request.setAttribute("alertMsg", "invalid room id");
                }
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Error: " + e);
            }
        }
    }
}

```

```

public class room extends HttpServlet {
    private PreparedStatement pstmt;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            List data = new ArrayList();
            //variables
            String id = request.getParameter("username");
            String r_id = request.getParameter("roomNo");
            String Faculty = request.getParameter("faculty");
            String Department = request.getParameter("department");
            String Capacity = request.getParameter("capacity");
            //query
            String query = "SELECT * FROM `schedules`.`room` WHERE `room_id` = ?";
            try {
                pstmt = conn.prepareStatement(query);
                pstmt.setString(1, r_id);
                rs = pstmt.executeQuery();
                //check if the course has been registered before
                if (rs.next()) {

                    out.println("This room exists!");
                    rs.close();
                    pstmt.close();
                    return;
                } //if not then register it
                else if (!rs.next()) {
                    String sql = "INSERT INTO `room`(`room_id`, `faculty`, `department`, `capacity`,
`user`)
                    + " VALUES (?, ?, ?, ?)";
                    pstmt = conn.prepareStatement(sql);
                    pstmt.setString(1, r_id);
                    pstmt.setString(2, Faculty);
                    pstmt.setString(3, Department);
                    pstmt.setString(4, Capacity);
                    pstmt.setString(5, id);
                    pstmt.executeUpdate();
                    data.add(r_id + ":Classroom" + " " + "[" + " " + " " + "roomNumber->" + "" + r_id + ""
+ ", " + " "
                    + "location->" + "" + Faculty + "" + ", " + " " + "inDepartment->" + Department +
", " + " " + "capacity->" + Capacity + "" + ".");
                    writeToFile(data,
"C:/Users/Pamilerin/Downloads/pamthesis/src/java/timetable1/classroom_instances.flr");
                    request.setAttribute("alertMsg", "inserted successfully");
                    response.sendRedirect("roomlist.jsp");
                } //if not registered prompt the user this error msg
                else {
                    request.setAttribute("alertMsg", "invalid room id");
                }
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Error: " + e);
            }
        }
    }
}

```

```
    }  
  }  
  private static void writeToFile(java.util.List list, String path) {  
    BufferedWriter out = null;  
    try {  
      File file = new File(path);  
      out = new BufferedWriter(new FileWriter(file, true));  
      for (Object s : list) {  
        out.write(s.toString());  
        //reArrange(s.split(","));  
        out.newLine();  
      }  
      out.close();  
    } catch (IOException e) {  
    }  
  }  
}
```

```

public class time extends HttpServlet {
    private PreparedStatement pstm;
    private ResultSet rs;
    private Connection conn = connect.DBConnect();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException, ClassNotFoundException, SQLException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<html>");
            out.println("<head>");
            out.println("<meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>");
            out.println("<title>JSP Page</title>");
            out.println("<link href='css/one-page-wonder.min.css' rel='stylesheet'>");
            out.println("<link href='css/main.css' rel='stylesheet' media='all'>");
            out.println("<link rel='stylesheet' href='table/style.css'>");
            out.println("<link rel='icon' type='image/png' href='images/icons/favicon.ico'/>");
            out.println("<link
                rel='stylesheet'
                type='text/css'
href='vendor/bootstrap/css/bootstrap.min.css'>");
            out.println("<link rel='stylesheet' type='text/css' href='fonts/font-awesome-4.7.0/css/font-
awesome.min.css'>");
            out.println("<link rel='stylesheet' type='text/css' href='vendor/animate/animate.css'>");
            out.println("<link rel='stylesheet' type='text/css' href='vendor/select2/select2.min.css'>");
            out.println("<link rel='stylesheet' type='text/css' href='vendor/perfect-scrollbar/perfect-
scrollbar.css'>");
            out.println("<link rel='stylesheet' type='text/css' href='css/util.css'>");
            out.println("<link rel='stylesheet' type='text/css' href='css/main.css'>");
            out.println("<script
src='https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js'></script>");
            out.println("<script src='vendor/bootstrap/js/bootstrap.min.js'></script>");
            out.println("</head>");
            out.println("<body id='page-top' class='index'>");
            out.println("<nav class='navbar navbar-expand-lg navbar-dark navbar-custom fixed-top'>");
            out.println("<div class='container'>");

                out.println("<a class='navbar-brand' href='home.jsp'>SCHEDULING</a>");
                out.println("<button class='navbar-toggler' type='button' data-toggle='collapse' data-
target='#navbarResponsive' aria-controls='navbarResponsive' aria-expanded='false' aria-label='Toggle
navigation'>");
                out.println("<span class='navbar-toggler-icon'></span>");
                out.println("</button>");
                out.println("<div class='collapse navbar-collapse' id='navbarResponsive'>");
                out.println("<ul class='navbar-nav ml-auto'>");
                out.println("<li class='nav-item'>");
                out.println("<a class='nav-link' href='addcourse.jsp'><span></span> Add Course</a></li>");
                out.println("<li class='nav-item'>");
                out.println("<a class='nav-link' href='addroom.jsp'><span></span> Add Room</a></li>");
                out.println("<li class='nav-item'>");
                out.println("<a
                    class='nav-link'
                    href='opencourse.jsp'><span></span>
                Add
                CourseOpening</a></li>");
                out.println("<li class='nav-item'>");
                out.println("<a
                    class='nav-link'
                    href='timetable.jsp'><span></span>
                Generate
                TimeTable</a></li>");
                out.println("<li class='nav-item dropdown'>");
                out.println("<a class='nav-link dropdown-toggle' href='#' id='manage' data-
toggle='dropdown' aria-haspopup='true' aria-expanded='false'><span></span> Manage</a>");
                out.println("<ul class='dropdown-menu' aria-labelledby='manage'>");

```

```

        out.println("<li><a class='dropdown-item' href='roomlist.jsp'><span></span> View
Room</a></li>");
        out.println("<li> <a class='dropdown-item' href='courselist.jsp'><span></span> View
Course</a></li>");
        out.println(" <li><a class='dropdown-item' href='opencourselist.jsp'><span></span> View
Courseopening</a></li>");
        out.println("<li><a class='dropdown-item' href='view.jsp'><span></span> View
Timetable</a></li>");
        out.println(" </ul>");
        out.println("</li>");
        out.println("<li class='nav-item'>");
        out.println(" <a class='nav-link' href='logout'><span></span> Logout</a></li>");
        out.println("</ul>");
        out.println("</div>");
        out.println("</div>");
        out.println("</nav>");
        out.println("<div class='page-wrapper bg-dark p-t-100 p-b-50'>");
        out.println("<div class='wrapper wrapper--w900'>");
        out.println("<div class='card card-6'>");
        out.println("<div class='card-body'>");
        out.println("<div class='limiter'>");
        out.println("<div class='table-00'>");
        out.println("<table>");
        out.println("<thead>");
        out.println(" <tr class='table100-head'>");
        out.println(" <th class='column1'>Course Code</th>");
        out.println(" <th class='column2'>Period</th>");
        out.println(" <th class='column3'>Group</th>");
        out.println(" <th class='column4'>Room</th>");
        out.println(" <th class='column5'>Day</th>");
        out.println(" </tr>");
        out.println("</thead>");
        out.println("<tbody>");
        String user = (String) request.getSession().getAttribute("uname");
        String dept = (String) request.getSession().getAttribute("dept");
        String year = request.getParameter("year");
        String semester = request.getParameter("semester");
        String year = (String) request.getSession().getAttribute("year");
        String semesterr = (String) request.getSession().getAttribute("semester");

        System.setProperty("JAVA_BIN", "C:\\Program Files\\Java\\jdk1.8.0_201\\bin");
        System.setProperty("PROLOGDIR", "C:\\Users\\reslab\\Flora-2\\XSB\\config\\x64-pc-
windows\\bin");
        System.setProperty("FLORADIR", "C:\\Users\\reslab\\Flora-2\\flora2");
        FloraSession session = new FloraSession();

        String fileName = "C:/Users/reslab/Desktop/pamthesis-20190616T154348Z-
001/pamthesis/src/java/timetable1/generate_meeting_times8e.flr";
        //HttpSession session = request.getSession(true);
        // load the program into module main

        if (session.loadFile(fileName, "main")) {
            out.println("Example loaded successfully!");
        } else {
            out.println("Error loading the example!");
        }

        /* load data files */
        String command = "%r.";

```

```

out.println("Query:" + command + user + dept);
Iterator<FloraObject> classroomObjs = session.ExecuteQuery(command);

Vector<String> vars = new Vector<String>();
vars.add("?Year");
vars.add("?Semester");
vars.add("?Course");
vars.add("?GN");
vars.add("?Room");
vars.add("?Day");
vars.add("?Period");
vars.add("?Instructor");

String query = "?Year=" + yearr + ",?Semester=" + semesterr + ",
%generateOneByOne(?Year,?Semester,?Course,?GN,?Room,?Day,?Period, ?Instructor).";

Iterator<HashMap<String, FloraObject>> allmatches
    = session.ExecuteQuery(query, vars);
out.println(query);

while (allmatches.hasNext()) {
    HashMap<String, FloraObject> firstmatch = allmatches.next();
    Object YearObj = firstmatch.get("?Year");
    Object SemesterObj = firstmatch.get("?Semester");
    Object CourseObj = firstmatch.get("?Course");
    Object GNObj = firstmatch.get("?GN");
    Object RoomObj = firstmatch.get("?Room");
    Object DayObj = firstmatch.get("?Day");
    Object PeriodObj = firstmatch.get("?Period");
    Object InstructorObj = firstmatch.get("?Instructor");

    /*out.println("Year:" + YearObj + " SEMESTER: " + SemesterObj
    + "Course:" + CourseObj + "Group:" + GNObj
    + "room:" + RoomObj + "Day:" + DayObj + "period:" + PeriodObj);

    */
    out.println(" <tr>");

    out.println("<td class='column1'>" + CourseObj + "</td>");
    out.println("<td class='column2'>" + PeriodObj + "</td>");
    out.println("<td class='column3'>" + GNObj + "</td>");
    out.println("<td class='column3'>" + RoomObj + "</td>");
    out.println("<td class='column4'>" + DayObj + "</td>");
    out.println("<td class='column4'>" + InstructorObj + "</td>");
    out.println("</tr>");

    String courses = CourseObj.toString ();
    String rooms = RoomObj.toString();
    String days = DayObj.toString ();
    String periods = PeriodObj.toString ();
    String years = YearObj.toString ();
    String semesters = SemesterObj.toString ();
    String groups = GNObj.toString ();
    String instructor = InstructorObj.toString ();

```

```

String sql = "INSERT INTO `timetable`(`year`,`semester`,`course`,`room`,
`day`,`period`,`user`,`department`,`groupNo`,`instructor`) VALUES (?,?,?,?,?,?,?,?,?)";

    pstmt = conn.prepareStatement (sql);
    pstmt.setString (1, years);
    pstmt.setString (2, semesters);
    pstmt.setString (3, courses);
    pstmt.setString (4, rooms);
    pstmt.setString (5, days);
    pstmt.setString (6, periods);
    pstmt.setString (7, user);
    pstmt.setString (8, dept);
    pstmt.setString (9, groups);
    pstmt.setString (10, instructor);
    pstmt.executeUpdate ();

//      //add each item to the array list
//      timeTableAdapter.add(new TimeTableAdapter((String)CourseObj,
//      (String)RoomObj, (String)PeriodObj, (String)DayObj));
//      //switch case use
    pstmt.close();

}

    out.println("</tbody>");
    out.println("</table>");
    out.println("</div>");
    out.println("</div>");
    out.println("</div>");
    out.println("</div>");
    out.println("<div class='footer' >");
    out.println("<button      type='submit'      class='btn      btn--radius-2      btn--blue-2'
onClick='window.print()'> PRINT</button>");

    out.println("</div>");
    out.println("</div>");
    out.println("</div>");
} }

```

Appendix B: Database

Database schema

Table structure for table course

Column	Type	Null	Default
<i>course_id</i>	varchar(255)	No	
course_title	varchar(255)	Yes	NULL
course_prerequisite	varchar(255)	Yes	NULL
course_language	varchar(255)	Yes	NULL
course_credit_unit	int(11)	Yes	NULL
lecture_hour	varchar(255)	Yes	NULL
lab_hour	varchar(255)	Yes	NULL
Ects	varchar(255)	Yes	NULL
Department	varchar(255)	No	
User	varchar(255)	No	

Sample data in the table “course”

cmpe100	IntroductiontoComputerEngi neering	none	english	2	2	0	2	Computer Engineering	pamelyt
cmpe101	FoundationsofComputerEngi neering	none	english	3	3	1	6	Computer Engineering	pamelyt
cmpe108	AlgorithmsandProgramming	none	english	3	3	1	6	Computer	pamelyt

								Engineering	
cmpe110	Fundamentals of Programming	none	english	3	3	1	6	Computer Engineering	pamelyt
cmpe112	Programming Fundamentals	cmpe101	english	4	4	1	7	Computer Engineering	pamelyt
cmpe211	Object-Oriented Programming	cmpe112	english	4	4	1	7	Computer Engineering	pamelyt
cmpe223	Digital Logic Design	math163	english	4	4	1	7	Computer Engineering	pamelyt
cmpe224	Digital Logic System	cmpe223	english	4	4	1	7	Computer Engineering	pamelyt
cmpe226	Electronics for Computer Engineers	math241	english	4	4	1	6	Computer Engineering	pamelyt
cmpe231	Data Structures	cmpe112	english	4	4	1	7	Computer Engineering	pamelyt
cmpe242	Operating Systems	cmpe112	english	4	4	1	6	Computer Engineering	pamelyt
cmpe318	Principles of Programming Languages	cmpe211	english	4	4	1	6	Computer Engineering	pamelyt

Table structure for table department

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	varchar(255)	No	
faculty	varchar(255)	No	

Sample data in the table “department”

1	Accounting and Taxation Applications	Computing and Technology
3	Banking and Insurance	Computing and Technology
8	Information Technology	Computing and Technology
41	Civil Engineering	Engineering
46	Computer Engineering	Engineering
48	Economics	Economics
49	Information Systems Engineering	Engineering
50	Mechatronics Engineering	Engineering
51	Electrical and Electronic Engineering	Engineering
52	Electronics and Communications Engineering	Engineering
56	English Language Teaching	Foreign Language Education
61	Human Resources Management	Business and Finance

Table structure for table opencourse

Column	Type	Null	Default
courseCode	varchar(255)	No	
maxStudent	varchar(255)	Yes	NULL
year	varchar(255)	Yes	NULL
program	varchar(255)	Yes	NULL
semester	varchar(255)	Yes	NULL
groupNo	varchar(255)	Yes	NULL
courseGroup	varchar(255)	Yes	NULL
department	varchar(255)	No	
user	varchar(255)	No	

Sample data in the table “opencourse”

cmpe108	30	2019	Undergraduate	SPRING	2	cmpe108G 2	Computer Engineering	pamelyt
cmpe110	40	2019	Undergraduate	SPRING	1	cmpe110G 1	Computer Engineering	pamelyt
cmpe110	40	2019	Undergraduate	SPRING	2	cmpe110G 2	Computer Engineering	pamelyt
cmpe231	40	2019	Undergraduate	SPRING	1	cmpe231G 1	Computer Engineering	pamelyt
cmpe231	40	2019	Undergraduate	SPRING	2	cmpe231G 2	Computer Engineering	pamelyt
cmpe318	40	2019	Undergraduate	SPRING	1	cmpe318G 1	Computer Engineering	pamelyt
cmpe321	40	2019	Undergraduate	SPRING	1	cmpe321G 1	Computer Engineering	pamelyt
cmpe323	40	2019	Undergraduate	SPRING	1	cmpe323G 1	Computer Engineering	pamelyt

cmpe324	40	2019	Undergraduate	SPRING	1	cmpe324G 1	Computer Engineering	pamelyt
cmpe343	40	2019	Undergraduate	SPRING	1	cmpe343G 1	Computer Engineering	pamelyt
cmpe354	40	2019	Undergraduate	SPRING	1	cmpe354G 1	Computer Engineering	pamelyt
cmpe371	40	2019	Undergraduate	SPRING	1	cmpe371G 1	Computer Engineering	pamelyt
cmpe371	40	2019	Undergraduate	SPRING	2	cmpe371G 2	Computer Engineering	pamelyt
cmpe423	40	2019	Undergraduate	SPRING	1	cmpe423G 1	Computer Engineering	pamelyt
cmpe471	40	2019	Undergraduate	SPRING	1	cmpe471G 1	Computer Engineering	pamelyt

Table structure for table users

Column	Type	Null	Default
<i>username</i>	varchar(255)	No	
password	varchar(255)	No	
department	varchar(255)	No	

Sample data in the table “user”

cmpe	cmpe	Computer Engineering
econ	econ	Economics
pamelyt	vivian	Computer Engineering
zb	zb	Computer Engineering

Table structure for table room

Column	Type	Null	Default
<i>room_id</i>	varchar(255)	No	
faculty	varchar(255)	Yes	NULL
department	varchar(255)	Yes	NULL
type	varchar(255)	Yes	NULL
capacity	varchar(255)	Yes	NULL
user	varchar(255)	No	

Sample data in the table “classroom”

CMPE126	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE127	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE128	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE129	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE25	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE26	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE27	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE28	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt
CMPE33	ENGINEERING	Computer Engineering	CLASSROOM	70	pamelyt

Table structure for table timetable

Column	Type	Null	Default
year	varchar(255)	Yes	NULL
semester	varchar(255)	Yes	NULL
course	varchar(255)	Yes	NULL
Period	varchar(255)	Yes	NULL
groupNo	varchar(255)	Yes	NULL
Room	varchar(255)	Yes	NULL
Day	varchar(255)	Yes	NULL
department	varchar(255)	Yes	NULL
User	varchar(255)	Yes	NULL

Sample data in the table “timetable”

2021	FALL	asc105	1	1	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc105	2	1	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc105	5	1	asg111	friday	Actuarial Science	vivianpam
2021	FALL	asc102	3	1	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc102	4	1	asg111	thursday	Actuarial Science	vivianpam

2021	FALL	asc102	6	1	asg111	friday	Actuarial Science	vivianpam
2021	FALL	asc101	5	2	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc101	6	2	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc101	7	2	asg111	friday	Actuarial Science	vivianpam
2021	FALL	asc101	7	1	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc101	8	1	asg111	thursday	Actuarial Science	vivianpam
2021	FALL	asc101	8	1	asg111	friday	Actuarial Science	vivianpam

Appendix C: Sample generated Flora-2 files as input to the scheduler

A. Classroom_instances.flr

```
cmpe126:Classroom [ roomNumber->cmpe126, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe127:Classroom [ roomNumber->cmpe127, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe128:Classroom [ roomNumber->cmpe128, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe129:Classroom [ roomNumber->cmpe129, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe25:Classroom [ roomNumber->cmpe25, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe26:Classroom [ roomNumber->cmpe26, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe27:Classroom [ roomNumber->cmpe27, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe28:Classroom [ roomNumber->cmpe28, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe33:Classroom [ roomNumber->cmpe33, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
cmpe36:Classroom [ roomNumber->cmpe36, location-> 'engineering', inDepartment->
'computer engineering', capacity->70].
```

B. Course_instance.flr

```
biol105:Course [ courseCode->'biol105', courseName->'BiologicalBasisofBehavior',
hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3,
lab_hours->1, ects->6 ].
biol105:program_course_type_year(OUTSIDE,'SCIENCE',2).
biol124:Course [ courseCode->'biol124', courseName->'IntroductiontoMolecularBiology&Genetics',
hasPrerequisite->None, instructionLanguage->english, credits->3, lecture_hours->3, lab_hours->1, ects->6 ].
biol124:program_course_type_year(OUTSIDE,'SCIENCE',2).
biol316:Course [ courseCode->'biol316', courseName->'EnvironmentalManagement',
hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3,
lab_hours->1, ects->6 ].
biol316:program_course_type_year(OUTSIDE,'SCIENCE',3).
chem101:Course [ courseCode->'chem101', courseName->'GeneralChemistry',
hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3,
lab_hours->1, ects->6 ].
chem101:program_course_type_year(OUTSIDE,'SCIENCE',2).
cmpe100:Course [ courseCode->'cmpe100', courseName->'IntroductiontoComputerEngineering',
hasPrerequisite->none, instructionLanguage->english, credits->2, lecture_hours->0, lab_hours->0, ects->2 ].
cmpe100:program_course_type_year(CMPE,'AREA CORE',4).
```



```

cmpe101:Course [ courseCode->'cmpe101', courseName->'FoundationsofComputerEngineering', hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3, lab_hours->1, ects->6 ].
cmpe101:program_course_type_year(CMPE,'AREA CORE',1).
cmpe108:Course [ courseCode->'cmpe108', courseName->'AlgorithmsandProgramming', hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3, lab_hours->1, ects->6 ].
cmpe108:program_course_type_year(CMPE,'AREA CORE',1).
cmpe110:Course [ courseCode->'cmpe110', courseName->'FundamentalsofProgramming', hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3, lab_hours->1, ects->6 ].
cmpe110:program_course_type_year(CMPE,'AREA CORE',1).
cmpe112:Course [ courseCode->'cmpe112', courseName->'ProgrammingFundamentals', hasPrerequisite->cmpe101, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe112:program_course_type_year(CMPE,'AREA CORE',1).
cmpe211:Course [ courseCode->'cmpe211', courseName->'Object-OrientedProgramming', hasPrerequisite->cmpe112, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe211:program_course_type_year(CMPE,'AREA CORE',2).
cmpe223:Course [ courseCode->'cmpe223', courseName->'DigitalLogicDesign', hasPrerequisite->math163, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe223:program_course_type_year(CMPE,'AREA CORE',2).
cmpe224:Course [ courseCode->'cmpe224', courseName->'DigitalLogicSystem', hasPrerequisite->cmpe223, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe224:program_course_type_year(CMPE,'AREA CORE',2).
cmpe226:Course [ courseCode->'cmpe226', courseName->'ElectronicsforComputerEngineers', hasPrerequisite->math241, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->6 ].
cmpe226:program_course_type_year(CMPE,'AREA CORE',2).
cmpe231:Course [ courseCode->'cmpe231', courseName->'DataStructures', hasPrerequisite->cmpe112, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe231:program_course_type_year(CMPE,'AREA CORE',2).
cmpe242:Course [ courseCode->'cmpe242', courseName->'OperatingSystems', hasPrerequisite->cmpe112, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->6 ].
cmpe242:program_course_type_year(CMPE,'AREA CORE',2).
cmpe318:Course [ courseCode->'cmpe318', courseName->'PrinciplesofProgrammingLanguages', hasPrerequisite->cmpe211, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->6 ].
cmpe318:program_course_type_year(CMPE,'AREA CORE',3).
cmpe321:Course [ courseCode->'cmpe321', courseName->'BasicsofSignalsandSystems', hasPrerequisite->cmpe226, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->6 ].
cmpe321:program_course_type_year(CMPE,'AREA CORE',3).
cmpe323:Course [ courseCode->'cmpe323', courseName->'Microprocessors', hasPrerequisite->cmpe224, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe323:program_course_type_year(CMPE,'AREA CORE',3).
cmpe324:Course [ courseCode->'cmpe324', courseName->'ComputerArchitectureandOrganization', hasPrerequisite->cmpe224, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7 ].
cmpe324:program_course_type_year(CMPE,'AREA CORE',3).

```

cmpe343:Course [courseCode->'cmpe343', courseName->'SystemsProgramming',
 hasPrerequisite->cmpe242, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe343:program_course_type_year(CMPE,'AREA CORE',3).
 cmpe344:Course [courseCode->'cmpe344', courseName->'ComputerNetworks',
 hasPrerequisite->cmpe343, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->7].
 cmpe344:program_course_type_year(CMPE,'AREA CORE',3).
 cmpe354:Course [courseCode->'cmpe354', courseName->'DatabaseManagementSystems',
 hasPrerequisite->cmpe231, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe354:program_course_type_year(CMPE,'AREA CORE',3).
 cmpe371:Course [courseCode->'cmpe371', courseName->'AnalysisofAlgorithms',
 hasPrerequisite->cmpe231, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->7].
 cmpe371:program_course_type_year(CMPE,'AREA CORE',3).
 cmpe400:Course [courseCode->'cmpe400', courseName->'SummerTraining',
 hasPrerequisite->none, instructionLanguage->english, credits->0, lecture_hours->0,
 lab_hours->0, ects->1].
 cmpe400:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe405:Course [courseCode->'cmpe405', courseName->'GraduationProjectI',
 hasPrerequisite->none, instructionLanguage->english, credits->0, lecture_hours->1,
 lab_hours->0, ects->3].
 cmpe405:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe411:Course [courseCode->'cmpe411', courseName->'InformationSecurity',
 hasPrerequisite->cmpe354, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe411:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe415:Course [courseCode->'cmpe415', courseName->'VisualProgramming',
 hasPrerequisite->cmpe354, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe415:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe416:Course [courseCode->'cmpe416', courseName->'Object-
 OrientedProgramming&GraphicalUserInterfaces',
 hasPrerequisite->cmpe354,
 instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->6].
 cmpe416:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe418:Course [courseCode->'cmpe418', courseName->'InternetProgramming',
 hasPrerequisite->cmpe354, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe418:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe423:Course [courseCode->'cmpe423', courseName->'EmbeddedSystems',
 hasPrerequisite->cmpe354, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe423:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe466:Course [courseCode->'cmpe466', courseName->'ComputerGraphics',
 hasPrerequisite->cmpe354, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe466:program_course_type_year(CMPE,'AREA CORE',4).
 cmpe471:Course [courseCode->'cmpe471', courseName->'AutomataTheory',
 hasPrerequisite->math163, instructionLanguage->english, credits->4, lecture_hours->4,
 lab_hours->1, ects->6].
 cmpe471:program_course_type_year(CMPE,'AREA CORE',4).
 engl191:Course [courseCode->'engl191', courseName->'CommunicationinEnglishI',
 hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3,
 lab_hours->1, ects->4].
 engl191:program_course_type_year(OUTSIDE,'ENGLISH',1).

engl192:Course [courseCode->'engl192', courseName->'CommunicationinEnglishII',
 hasPrerequisite->engl191, instructionLanguage->english, credits->3, lecture_hours->3,
 lab_hours->1, ects->4].
 engl192:program_course_type_year(OUTSIDE,'ENGLISH',1).
 engl201:Course [courseCode->'engl201', courseName->'Communicationskills',
 hasPrerequisite->engl192, instructionLanguage->english, credits->3, lecture_hours->3,
 lab_hours->1, ects->4].
 engl201:program_course_type_year(OUTSIDE,'ENGLISH',3).
 hist280:Course [courseCode->'hist280', courseName->'HistoryofTurkishReforms',
 hasPrerequisite->none, instructionLanguage->turkish, credits->2, lecture_hours->2,
 lab_hours->0, ects->3].
 hist280:program_course_type_year(OUTSIDE,'HISTORY',1).
 math151:Course [courseCode->'math151', courseName->'CalculusI', hasPrerequisite->none,
 instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7].
 math151:program_course_type_year(OUTSIDE,'MATH',1).
 math152:Course [courseCode->'math152', courseName->'CalculusII', hasPrerequisite-
 >math151, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects-
 >7].
 math152:program_course_type_year(OUTSIDE,'MATH',1).
 math163:Course [courseCode->'math163', courseName->'DiscreteMathematics',
 hasPrerequisite->none, instructionLanguage->english, credits->3, lecture_hours->3,
 lab_hours->1, ects->5].
 math163:program_course_type_year(OUTSIDE,'MATH',1).
 math241:Course [courseCode->'math241', courseName->'LinearAlgebraandOrdinaryDiff.Equations',
 hasPrerequisite->math151, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->6].
 math241:program_course_type_year(OUTSIDE,'MATH',2).
 math322:Course [courseCode->'math322', courseName->'ProbabilityandStatisticalMethods',
 hasPrerequisite->math151, instructionLanguage->english, credits->3, lecture_hours->3, lab_hours->1, ects->5].
 math322:program_course_type_year(OUTSIDE,'MATH',3).
 math373:Course [courseCode->'math373', courseName->'NumericalAnalysisforEngineers',
 hasPrerequisite->math241, instructionLanguage->english, credits->3, lecture_hours->3,
 lab_hours->1, ects->6].
 math373:program_course_type_year(OUTSIDE,'MATH',3).
 phys101:Course [courseCode->'phys101', courseName->'PhysicsI', hasPrerequisite->none,
 instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects->7].
 phys101:program_course_type_year(OUTSIDE,'PHYSICS',1).
 phys102:Course [courseCode->'phys102', courseName->'PhysicsII', hasPrerequisite-
 >phys101, instructionLanguage->english, credits->4, lecture_hours->4, lab_hours->1, ects-
 >6].
 phys102:program_course_type_year(OUTSIDE,'PHYSICS',1).
 tusl181:Course [courseCode->'tusl181', courseName->'Turkishasasecondlanguage',
 hasPrerequisite->none, instructionLanguage->english, credits->2, lecture_hours->2,
 lab_hours->0, ects->3].
 tusl181:program_course_type_year(OUTSIDE,'TURKISH',1).

C. Course_opening_request.flr

request_cmpe100_1:CourseOpeningRequest [groupNumber->1, ofCourse->cmpe100,
 maxStudentSize->40, calendarYear->2019, semester->FALL, instructor->'zeki bayram',
 external_room->AS106,
 external_lab->IENG203,
 use_only_room->cmpe126,
 use_only_lab->cmpe218,
 preferred_day_period -> (tuesday,3),
 fixed_day_period -> (tuesday,3),

```

exclude_day_period -> (monday,1),
exclude_room -> cmpe25,
preferred_room -> cmpe129,
preferred_lab -> cmpe215,
exclude_lab -> cmpe218,
dummy -> dummy ].
request_cmpe108_1:CourseOpeningRequest [   groupName->1, ofCourse->cmpe108,
maxStudentSize->30, calendarYear->2019, semester->FALL , instructor->'dogu arifler',
use_only_room->cmpe127,
use_only_lab->cmpe230,
dummy -> dummy ].
request_cmpe112_1:CourseOpeningRequest [   groupName->1, ofCourse->cmpe112,
maxStudentSize->45, calendarYear->2019, semester->FALL , instructor->'pamilerin v',
use_only_room->cmpe129,
use_only_lab->cmpe218,
preferred_day_period -> (tuesday,3),
preferred_day_period -> (tuesday,4),
fixed_day_period -> (tuesday,3),
fixed_day_period -> (tuesday,4),
exclude_day_period -> (monday,2),
exclude_room -> cmpe26,
preferred_room -> cmpe128,
preferred_lab -> cmpe30,
exclude_lab -> cmpe215,
dummy -> dummy ].
request_cmpe110_2:CourseOpeningRequest [   groupName->2, ofCourse->cmpe110,
maxStudentSize->40, calendarYear->2019, semester->FALL , instructor->'cidem e',
external_room->AS106,
external_lab->IENG203,
use_only_lab->cmpe215,
preferred_day_period -> (wednesday,4),
fixed_day_period -> (wednesday,4),
exclude_day_period -> (monday,3),
exclude_room -> cmpe127,
preferred_room -> cmpe25,
preferred_lab -> cmpe215,
exclude_lab -> cmpe220,
lab_exclude_day_period -> (tuesday,3),
lab_fixed_day_period -> (thursday,6),
lab_preferred_day_period -> (thursday,6),
dummy -> dummy ].
request_cmpe211_1:CourseOpeningRequest [   groupName->1, ofCourse->cmpe211,
maxStudentSize->25, calendarYear->2019, semester->FALL , instructor->'john olaifa',
external_room->cla11,
external_lab->cla12,
dummy -> dummy ].
request_cmpe223_1:CourseOpeningRequest [   groupName->1, ofCourse->cmpe223,
maxStudentSize->50, calendarYear->2019, semester->FALL , instructor->'mehmet bodur',
external_room->CIVIL34,
external_lab->CIVIL203,
fixed_day_period -> (monday,1),
fixed_day_period -> (monday,2),
dummy -> dummy ].

```

D. Instructor.flr

```
zekibayramInstructor [ id->1, name-> 'zeki', lastName-> 'bayram', works_in->'Computer Engineering' ].
ekremvaInstructor [ id->5, name-> 'ekrem', lastName-> 'va', works_in->'Computer Engineering' ].
yiltanbitirimInstructor [ id->6, name-> 'yiltan', lastName-> 'bitirim', works_in->'Computer Engineering' ].
onsentoygarInstructor [ id->7, name-> 'onsen', lastName-> 'toygar', works_in->'Computer Engineering' ].
ahmetuvInstructor [ id->8, name-> 'ahmet', lastName-> 'uv', works_in->'Computer Engineering' ].
doguariflerInstructor [ id->9, name-> 'dogu', lastName-> 'arifler', works_in->'Computer Engineering' ].
johnolaifaInstructor [ id->10, name-> 'john', lastName-> 'olaifa', works_in->'Computer Engineering' ].
selinbitirimInstructor [ id->11, name-> 'selin', lastName-> 'bitirim', works_in->'Computer Engineering' ].
pamilerinvInstructor [ id->12, name-> 'pamilerin', lastName-> 'v', works_in->'Computer Engineering' ].
alexchefranovInstructor [ id->13, name-> 'alex', lastName-> 'chefranov', works_in->'Computer Engineering' ].
mustafailkanInstructor [ id->14, name-> 'mustafa', lastName-> 'ilkan', works_in->'Computer Engineering' ].
mehmetbodurInstructor [ id->15, name-> 'mehmet', lastName-> 'bodur', works_in->'Computer Engineering' ].
mehmettopaInstructor [ id->16, name-> 'mehmet', lastName-> 'topa', works_in->'Computer Engineering' ].
cidemeInstructor [ id->17, name-> 'cidem', lastName-> 'e', works_in->'Computer Engineering' ].
```

E. Lab_instances.flr

```
cmpe215:Lab [ roomNumber->cmpe215, location-> 'engineering', inDepartment->'computer engineering', capacity->50].
cmpe218:Lab [ roomNumber->cmpe218, location-> 'engineering', inDepartment->'computer engineering', capacity->50].
cmpe220:Lab [ roomNumber->cmpe220, location-> 'engineering', inDepartment->'computer engineering', capacity->60].
cmpe223:Lab [ roomNumber->cmpe223, location-> 'engineering', inDepartment->'computer engineering', capacity->60].
cmpe230:Lab [ roomNumber->cmpe230, location-> 'engineering', inDepartment->'computer engineering', capacity->60].
cmpe30:Lab [ roomNumber->cmpe30, location-> 'engineering', inDepartment->'computer engineering', capacity->60].
test556:Lab [ roomNumber->test556, location-> 'engineering', inDepartment->'computer engineering', capacity->70].
```