

**Tribhuvan University**

**Faculty of Humanities and Social Science**

**CLASS SCHEDULE MANAGEMENT SYSTEM**

**A PROJECT REPORT**

**Submitted to**

**Department of Computer Application**

**Padmakanya Multiple Campus**

**Bagbazar, Kathmandu**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted by

Saraswoti Shrestha (18380153)

Sija Kumari Gupta (18380156)

September, 2021

Under the Supervision of

**Mr. Dharma Raj Poudel**



**Tribhuvan University**

**Faculty of Humanities and Social Science**

**Padmakanya Multiple Campus**

# SUPERVISOR’S RECOMMENDATION

I hereby recommend that this project prepared under our supervision by Mr. Dharma Raj Poudel entitled “**CLASS SCHEDULE MANAGEMENT SYSTEM”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Dharma Raj Poudel

Project Supervisor

The British College, Kathmandu



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Padmakanya Multiple Campus**

# LETTER OF APPROVAL

This is to certify that this report is prepared by Saraswoti Shrestha and Sija Kumari Gupta entitled **“CLASS SCHEDULE MANAGEMENT SYSTEM**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **SIGNATURE of Supervisor**  Mr. Dharma Raj Poudel  Supervisor  Padmakanya Multiple Campus, BCA | **SIGNATURE of HOD/Coordinator**  Mr. Kumar Parsun  Director  Padmakanya Multiple campus, BCA |
| Mr. Jayram Gautam  Internal Examiner | Mr. Ramesh Singh Saud  External Examiner |

# ABSTRACT

Bachelor in Computer Application (BCA) offered by the Faculty of Humanities and Social Sciences, Tribhuvan University (TU) in Padmakanya Multiple Campus at Bagbazar, Kathmandu has been running from the academic year 2074/75 and has had phenomenal growth during past years. With this growth, however, came complex operational planning problems of which class scheduling is one.

The department now uses notice boards as a display means, this is not very reliable as one has to be physically available in the campus to view the schedule. Also taking into consideration this is not just other department, but the department of computer application. So, Class Schedule Management System is developed to generate complete and conflict-free schedules and reduced the time required to construct official schedules for the BCA department including all the semesters. The system is reliable and can be used by administrator, faculty members and students of the department from anywhere if have access to an internet-enabled device. The whole system is under the control of the administrator. The administrator has the full privileges to create, add, update, delete timeslot, semester, and course, approve users, assign faculty, approve faculty time preference and view the schedules after successful login. Students can read and view their respective schedule by providing user name and password for secure login and in case of new student the registration is available and should be approved by admin. Likewise, faculty members can login to the system and choose timeslot for the course assigned in respective semester of respective courses and view the respective schedule. In case of new faculty, the faculty should register in the system. Thus, the system helps administrator to manage the class schedules for the students and faculty members in a simple and easy way.

**Keyword: Schedule Management, web application.**

# ACKNOWLEDGMENT

We would like to express our special thanks of gratitude to our supervisor **Mr. Dharma Raj Poudel** who gave us the golden opportunity, guidance, inspiration and constructive suggestion to do this wonderful project on the topic “Class Schedule Management System”, which also helped us in doing a lot of research and we came to know about so many tools and technologies.

We would like to express our special thanks of gratitude to our lecturer **Mr. Jayram Gautam** for his expert advice and encouragement to do this project.

We would like to express a deep sense of gratitude to **Mr. Kumar Prasun**, for his exemplary guidance, monitoring and constant encouragement throughout which helped us in completing this task through various stages.

We appreciate all the personal and professional advice given by our teachers **Mr. Basanta Chapagain**, **Er. Shree Krishna Maharjan**, **and Mr. Sonik Shrestha** during our working tenure.

Finally, we would also like to thank our family and friends who encouraged and helped us a lot in finalizing this project within the limited time frame.

Yours sincerely,

Saraswoti Shrestha

Sija Kumari Gupta

TABLE OF CONTENTS

[**SUPERVISOR’S RECOMMENDATION i**](#_Toc84330412)

[**LETTER OF APPROVAL ii**](#_Toc84330413)

[**ABSTRACT iii**](#_Toc84330414)

[**ACKNOWLEDGMENT iv**](#_Toc84330415)

[**TABLE OF CONTENTS v**](#_Toc84330416)

[**ABBREVIATIONS vii**](#_Toc84330417)

[**LIST OF FIGURES viii**](#_Toc84330418)

[**LIST OF TABLES ix**](#_Toc84330419)

[**CHAPTER 1 1**](#_Toc84330420)

[**INTRODUCTION 1**](#_Toc84330421)

[1.1 Introduction 1](#_Toc84330422)

[1.2 Problem Statement 2](#_Toc84330423)

[1.3 Objectives 2](#_Toc84330424)

[1.4 Scope and Limitations 2](#_Toc84330425)

[1.4.1 Scope 2](#_Toc84330426)

[1.4.2 Limitations 3](#_Toc84330427)

[1.5 Report Organization 3](#_Toc84330428)

[**CHAPTER 2 5**](#_Toc84330429)

[**BACKGROUND STUDY AND LITERATURE REVIEW 5**](#_Toc84330430)

[2.1 Background Study 5](#_Toc84330431)

[2.2 Literature Review 5](#_Toc84330432)

[**CHAPTER 3 7**](#_Toc84330433)

[**SYSTEM ANALYSIS AND DESIGN 7**](#_Toc84330434)

[3.1 System Analysis 7](#_Toc84330435)

[3.1.1 Requirement Analysis 8](#_Toc84330436)

[3.1.2 Feasibility Analysis 10](#_Toc84330437)

[3.1.3 Data Modelling 12](#_Toc84330438)

[3.1.4 Process Modelling 13](#_Toc84330439)

[3.2 System Design 15](#_Toc84330440)

[3.2.1 Architectural Design 15](#_Toc84330441)

[3.2.2 System Flowchart 16](#_Toc84330442)

[3.2.3 Database Schema Design 19](#_Toc84330443)

[3.2.4 Interface Design 20](#_Toc84330444)

[3.2.5 Physical DFD 22](#_Toc84330445)

[**CHAPTER 4 24**](#_Toc84330446)

[**IMPLEMENTATION AND TESTING 24**](#_Toc84330447)

[4.1 Implementation 24](#_Toc84330448)

[4.1.1 Tools Used 24](#_Toc84330449)

[4.1.2 Implementations Details of Modules 25](#_Toc84330450)

[4.2 Testing 27](#_Toc84330451)

[4.2.1 Test Cases for Unit Testing 27](#_Toc84330452)

[4.2.2 Test Cases for system testing 33](#_Toc84330453)

[**CHAPTER 5 35**](#_Toc84330454)

[**CONCLUSION AND FUTURE RECOMMENDATIONS 35**](#_Toc84330455)

[5.1 Lesson Learnt/Outcome 35](#_Toc84330456)

[5.2 Conclusion 35](#_Toc84330457)

[5.3 Future Recommendations 35](#_Toc84330458)

[**REFRENCES 37**](#_Toc84330459)

[**APPENDIX: SYSTEM SCREENSHOTS 38**](#_Toc84330460)

# ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| AJAX  BCA  CASE  CSMS  CSS  DFD  ERD  HTML  JS  PHP  UI  XAMPP | :  :  :  :  :  :  :  :  :  :  : | Asynchronous JavaScript and XML  Bachelor In Computer Application  Computer Aided Software Engineering  Class Schedule Management System  Cascade Style Sheet  Data Flow Diagram  Entity Relationship Diagram  Hyper Text Markup Language  JavaScript  Hypertext Pre-Processor  User Interface  **X (cross platform), Apache, MySQL, PHP, Perl** |

# LIST OF FIGURES

[Figure 3.1. 1: Iterative Waterfall Model for Class Schedule Management System 7](#_Toc84323812)

[Figure 3.1.1. 1: Use Case Diagram of Class Schedule Management System 9](#_Toc84323817)

[Figure 3.1.3. 1: Entity Relationship Diagram for Class Schedule Management System 12](#_Toc84323822)

[Figure 3.1.4. 1: Level 0 for Class Schedule Management System 13](#_Toc84323827)

[Figure 3.1.4. 2: Level 1 DFD for Class Schedule Management System 14](#_Toc84323828)

[Figure 3.2.1. 1: Architectural Design for Class Schedule Management System 15](#_Toc84323833)

[Figure 3.2.2. 1: System Flow Chart for Administrator 16](#_Toc84323838)

[Figure 3.2.2. 2: System Flow Chart for Faculty 17](#_Toc84323839)

[Figure 3.2.2. 3: System Flow Chart for Student 18](#_Toc84323840)

[Figure 3.2.3. 1: Physical Schema Design of Class Schedule Management System 19](#_Toc84323843)

[Figure 3.2.4. 1: Login page of Class Schedule Management System 20](#_Toc84323849)

[Figure 3.2.4. 2: Registration page of Class Schedule Management System 20](#_Toc84323850)

[Figure 3.2.4. 3: Administrator dashboard of Class Schedule Management System 21](#_Toc84323851)

[Figure 3.2.4. 4: Faculty dashboard of Class Schedule Management System 21](#_Toc84323852)

[Figure 3.2.4. 5: Student dashboard of Class Schedule Management System 21](#_Toc84323853)

[Figure 3.2.5. 1: Physical DFD for Administrator 22](#_Toc84323855)

[Figure 3.2.5. 2: Physical DFD for Faculty 23](#_Toc84323856)

[Figure 3.2.5. 3: Physical DFD for Student 23](#_Toc84323857)

# LIST OF TABLES

[Table 3.1.2. 1: Project Schedule 11](#_Toc84323421)

[Table 4.2.1. 1: Test case for User Registration of Class Schedule Management System 28](#_Toc84323894)

[Table 4.2.1. 2: Test case for User Login of Class Schedule Management System 30](#_Toc84323895)

[Table 4.2.1. 3: Test case for Addition of semester, course, and timeslot 31](#_Toc84323896)

[Table 4.2.1. 4: Test case for Assigning Faculty 32](#_Toc84323897)

[Table 4.2.1. 5: Test case of Choose Preferences for Faculty 32](#_Toc84323898)

[Table 4.2.2. 1: Test case for Administrator module 33](#_Toc84323899)

[Table 4.2.2. 2: Test case for Faculty module 34](#_Toc84323900)

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Time is one of the most necessary and limited resource while carrying out our daily duties, but we often waste it or use it ineffectively. Therefore, scheduling helps us think about what we want to achieve in a day, week or month, and it keeps us on track to accomplish our goal. But improper scheduling results in failures leading to inefficient workflows and loss of productivity. Effective Scheduling is therefore far much necessary for the proper use of available time in carrying out several activities.

Padmakanya Multiple Campus located at Bagbazar, Kathmandu has several faculties of humanities and social sciences, science and technology, and management. The faculties comprise departments with each department responsible for preparing its class schedule. Currently, the administrator of that particular department creates class schedule manually, it is then printed and then placed on various notice boards. Also, the schedules keep changing with time. Hence their management get even more difficult due to change in parameters value. Furthermore, these schedules might contain clashing of courses, faculty being allocated to more than one course at the same time and one room being assigned to more than one course at the same time.

The Class Schedule Management System is a solution that replaces the manual method of preparing class schedules. This system automates the existing manual system for handling course studying in semesters. Hence the system will help the department to loosen the burden and increase efficiency. Furthermore, the system creates clash-free time schedule, as only the system administrator is able to manage semester, course, timeslots in the system. Faculty members and students are able to login to the system and they can view schedules easily for respective courses and semester.

## 1.2 Problem Statement

The main problem we have faced is due to the use of the manual system. The manual system of preparing class schedules by department is very time consuming and usually ends up with various class clashes either at same time or with same lecturer. The college departments experience many issues with the existing system while scheduling the classes manually. These are just due to common human errors which are difficult to prevent and they end up making poor schedules. This bitter compliment and delay in preparing of the schedules has been the major concern.

Some of the problems encountered in the traditional manual generation of class schedule are:

* Assigning of the different courses in a same room at the same time.
* A faculty allocated in more than a class at a same time.
* Repeated time allocations for a particular course.
* It generates a lot of paperwork and is tedious and more effort consuming task for the department administrator.

Class Schedule Management System helps department to create and maintain the schedules in more effective and efficient way. The system therefore, tries to solve these problems ensuring the class schedule created do not have a clash of course and faculty members.

## 1.3 Objectives

The proposed software system is expected to meet the following objectives:

* To eliminate the use of manual scheduling and manage class schedule in more efficient and easy way.
* To maintain record of courses, semesters, faculty members, and students.

## 1.4 Scope and Limitations

### **1.4.1 Scope**

The system is used to handle the class schedule of a department having semester system such as BCA department of Padmakanya Multiple Campus. There are many additional features, which are planned to be incorporated during the future enhancement of this project. This system is developed to automate the manual process of class scheduling of BCA department. Therefore, the system is concerned with the management of class schedule for the BCA department of Padmakanya Multiple Campus.

The system can be used by department’s administrator to manage schedules for students and faculty members. The administrator who has authority to access the system is being able to manage schedules, semesters, courses, timeslots, faculty members and their time preferences, and students after successful login. To get the authority as a faculty and student, the user needs to be register first before login as a faculty or student. Once the faculty login successfully, the system allows the faculty to choose time preference for the assigned course. After faculty choose their preference, the timetable based on their choices will be display after approval from administrator whereas student can only view their respective timetable. The system is focused on generating clash free schedules for faculty members and students.

### **1.4.2 Limitations**

The system is finally completed and well-developed as planned. However, there are several limitations in our system. The system is only developed for managing the class schedules of semester system. The system cannot be used for making schedules of activities other than class routine. In the system, the timeslots are fixed and once the faculty choose timeslot it cannot be altered.

## 1.5 Report Organization

**Introduction**

This chapter deals with the introduction, problem statement, objectives, scopes and limitations of the system.

**Background Study and Literature Review**

This chapter includes the brief description of the work that has been carried out in making class schedules and also describes the features about some existing applications related to Class Schedule Management System.

**System Analysis and Design**

This chapter deals with the requirement analysis of the system, which describes about the functional, non-functional, and feasibility analysis that includes analysis in economic, technical, operational, and schedule factors. Further, this chapter focuses on the Entity Relational diagram, Data Flow Diagram, design of the system with system architecture, database schema, and interface design.

**Implementation and Testing**

This chapter includes the description of tools that are used in system development. Further, this chapter also includes the result of testing performed in the system development.

**Conclusion and Further Recommendation**

This chapter includes the brief summary of lesson learnt, outcome and conclusion of the system build and explain what have been done and what further improvements could be done.

# CHAPTER 2

# BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1 Background Study

A class schedule is an organized list, usually set out in tabular form, providing information about a series of arranged events to take place in a particular time. The time scheduling problem is a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year. The problem involves the scheduling of courses, students, faculty members, and rooms with fixed number of time-slots and a certain number of constrains.

Currently, the entire process of creating timetable is done manually by the department administrator. This is tedious job and has a higher chance of having many clashes. Furthermore, the traditional system keeps records of courses, semesters, faculty members and students in registers or MS Word and it is costly for long term implementation and ultimately causes data loss. Thus, department needs a genuine software that can handle the data more efficiently and can be explore accordingly.

Class Schedule Management System aims in scheduling classes accurately without having clashing of course and time allocated. The system reduces amount of time taken by the manual process in creating schedules. Further, the system is applicable for providing the information about the courses, faculties, semesters and their schedules according to the time preferences. The system is implemented for providing facilitates to the faculty members and students in order to view their respective schedules easily. More specifically Class Schedule Management System focuses on generating clash-free time schedule that shows course, time allotment, days, semester, and name of faculties.

## 2.2 Literature Review

We reviewed some of the existing timetable management system and we came across their features and working mechanisms. There are some software related to Class Schedule Management System which works as like this application.

According to resource by P.V. Abdullah, Class Scheduling System is a software that can provide a database for storing records and information. It allows the end user to add, edit, delete, save and update records or information if some changes occur. It can generate reports for example class schedule, class list, instructors list, hall list, department list and school year with different semesters. The system is a class management system for a university for handling the course studying in which semester, managing user profiles and allowing authority for users to let them retrieve and export information of course details. It is a better solution with many flexible and convenient features, allowing class administrators and instructors to maximize efficiency while reducing time wastage. [1]

According to Timetable Management System Web Application Development, A web application was developed for timetable management. It is a system that is developed to provide better support to staff in a college. The timetable module acts as a robust platform to forecast. The time table management system provides a function to create and view the time table for the specific staff and student. This system also contains a database that stores staff and student allotted subject details and notifications before time. [2]

According to Online Class Scheduling Management System, the main purpose of developing this system is to help students in organizing their extra classes by choosing their own preference. This system used rule-based algorithm that often used to store and manipulate the knowledge to translate the information in a useful way. The main modules which are students and administrator will be included in this system. This system hopes that all students are able to attend all the extra classes after the well arrangement. [3]

By reviewing above systems features and working mechanisms, we came to know that they do not provide overall package of facilities so we tried to develop the system including extra feature for faculty to choose time preference so that generating class schedule is effective and efficient.

# CHAPTER 3

# SYSTEM ANALYSIS AND DESIGN

## 3.1 System Analysis

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives.

This system is designed with the series of processes starting with requirement gathering and analysis, design, coding and testing and maintenance. During requirement analysis, all the functional and nonfunctional requirement are analyzed and system is developed according to the requirement then designing of the system is carried out. After the design process, coding and development part is started then after unit testing followed by integrating the system and testing of the system. If the testing is positive then system is implemented otherwise some maintenance is done and again testing is done until system come in operation.

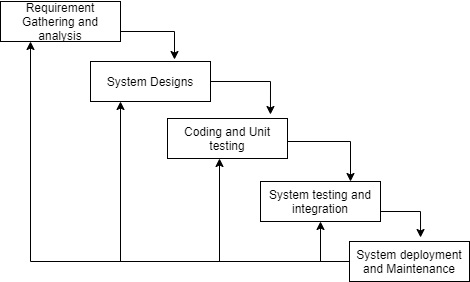


Figure 3.1. 1: Iterative Waterfall Model for Class Schedule Management System

### **3.1.1 Requirement Analysis**

The requirements are to be collected and analyze before starting system design. To design and develop system, the two major functional and non-functional requirements of the system has been studied.

#### **i. Functional Requirement**

The functional requirements identified are listed below:

* **Register**

The system allows new student and faculty member to register into the system.

* **Login**

The system allows the system administrator, faculty member and student to login to the system.

* **Add, Update and Delete**

The system allows administrator to create, add, update, and delete semester, course and timeslot.

* **Approve users**

The system allows administrator to approve registration request of faculty member and student. When the administrator approves new faculty member then the system generates faculty id for the faculty.

* **Assign Faculty**

The system allows administrator to assign faculty to a course in respective semester.

* **Choose time preferences**

The system allows faculty member to choose timeslot for assigned course.

* **Approve faculty preferences**

The system allows administrator to approve the timeslot preferred by faculty for respective course.

* **View Schedule**

The system allows administrator to view schedules of faculty member as well as schedules of semester. The student is able to view the schedule on the basis of semester in which they are enrolled. Likewise, faculty member is able to view their respective schedules.

**USECASE DIAGRAM**

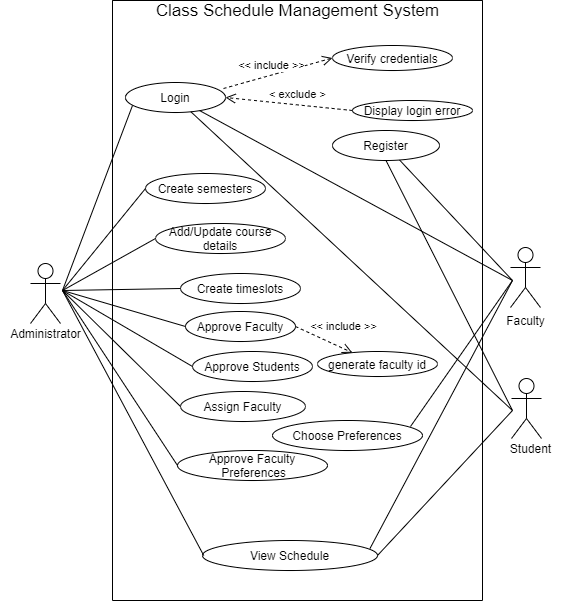


Figure 3.1.1. 1: Use Case Diagram of Class Schedule Management System

In Class Schedule Management System, there are three actors such as administrator, faculty and student. The administrator is a primary actor who can login to the system and create semester, course, and timeslot, approve users, assign faculty, approve faculty preference and view schedules. When the administrator approve faculty, the system will generate faculty id for the particular faculty member. Likewise, faculty can choose timeslot for the course assign in respective semester and view their respective schedules. Similarly, students can login to the system and view schedules of respective semester.

#### **ii. Non-Functional Requirement**

The different non-functional requirements identified are listed below:

* **Security**

Only the authorized users are allowed to access the data of the system. New student and faculty member must register fist and they can login to the system only after the approval of administrator. The system uses server-side validation and hence do not allow invalid data to enter during registration.

* **User friendly interface**

The interface of the system is user friendly and easy to understand and no need of extra training to use the system.

### **3.1.2 Feasibility Analysis**

Class Schedule Management System is able to be implemented as it is feasible with factors such as economic, technical, operational, and scheduling.

1. **Technical** **Feasibility Study**

The system is web-based system. It uses HTML, CSS, and JavaScript for front end and MySQL and PHP for backend. All of the technology required by the system are available and can be accessed freely, hence it is determined technically feasible.

#### **Operational Feasibility Study**

The system provides a simple interface for the users so that they can operate without having prior knowledge. It can be accessed from anywhere if the internet is available. Hence, without having any technical background, user can use this application. So, this project is operationally feasible.

#### **Economic Feasibility Study**

The system is designed for BCA department. After the completion of the system, the department didn’t need to install any new hardware and software for the deployment of the application. And, as all the tools and resources required are either open sources or free, so with limited budget our system can be easily developed and implemented. Hence, the system is economically feasible.

#### **Schedule Feasibility Study**

We can complete the project with limited time span and date mentioned accordingly in the Gantt chart given below. Hence the system is schedule feasible.

Table 3.1.2. 1: Project Schedule

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Months | July | | | | August | | | | | September | |
| Project Activities | Weeks | | | | | | | | | | |
| 1st week | 2nd week | 3rd week | 4th week | 5th  week | 6th week | 7th  week | 8th week | 9th  week | 10th  week | 11th  week |
| System Analysis |  |  |  |  |  |  |  |  |  |  |  |
| System Design |  |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |  |
| Testing/  Debugging |  |  |  |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |

### **3.1.3 Data Modelling**

In Entity-Relationship Diagram, the entities are user, faculty, administrator, student, course, semester, timeslot and schedule connected with each other through a relation. The user entity is a super class and the entities faculty, administrator and students are subclasses that inherit attributes and relationship of super class user. The faculty entity has a relation ‘assign\_for’ with schedule entity, again the schedule has a relation with semester and course entity. Further, faculty entity has a relation ‘prefer\_for’ with timeslot and semester as faculty can choose time slot for the semester in which the faculty is assigned. The administrator has relation with timeslot, semester, course, and schedule as administrator has the great role in generating schedule. The student entity has a relation with semester as student enrolls in semester.

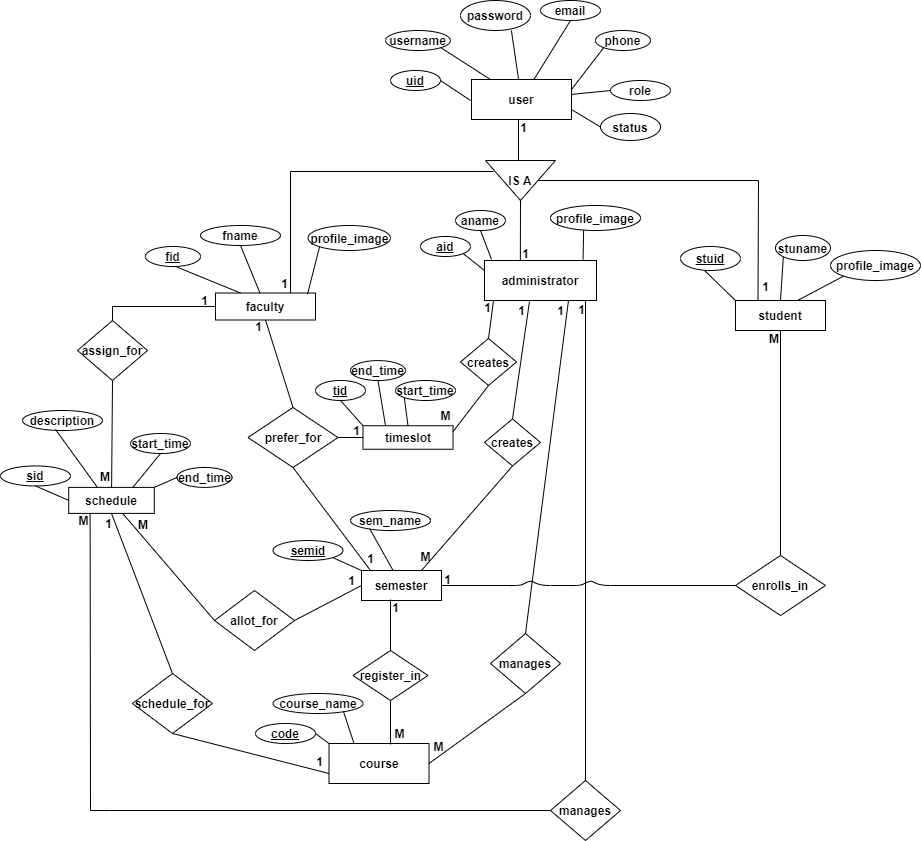


Figure 3.1.3. 1: Entity Relationship Diagram for Class Schedule Management System

### **3.1.4 Process Modelling**

The Data Flow Diagram of Class Schedule Management System consists of two levels of DFD, level zero DFD (or context diagram) and level one DFD.

In level zero DFD diagram, the Class Schedule Management System is at the center of the diagram. There are three external entities which are Student, Faculty and Administrator that interact with the system. Administrator request for login, creates semester, timeslot, course, approve other users, assign faculty members, approve faculty preferences, and finally view schedules. Similarly, Faculty register, login, choose time preferences, and view schedules. And, Students also register and login into the system and view schedule of respective semester.

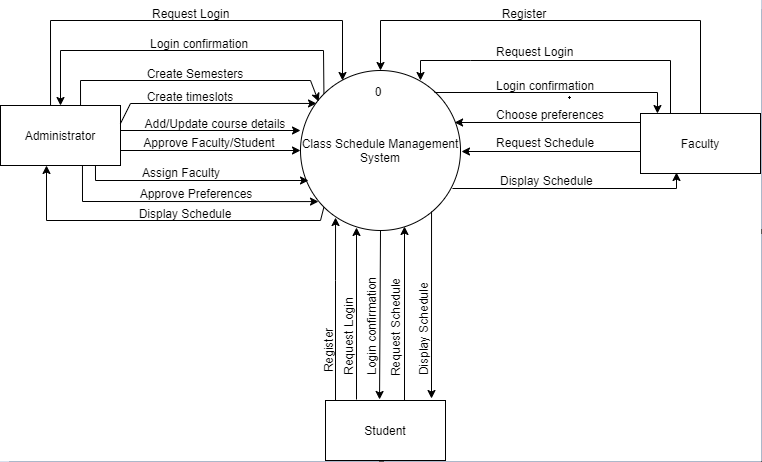


Figure 3.1.4. 1: Level 0 for Class Schedule Management System

In level 1 DFD, there are nine processes in which all the entities interact with login process in order to login into the system. Further, administrator interacts with processes such as create semester, manage course, create timeslots, approve Faculty/Student, assign faculty, approve faculty preferences and view schedule. Similarly, faculty interacts with processes such as choose time preference and view schedule. And, student interacts with process view schedule. There data required to input into the processes and the outcome of data from those processes are stored in respective data stores.

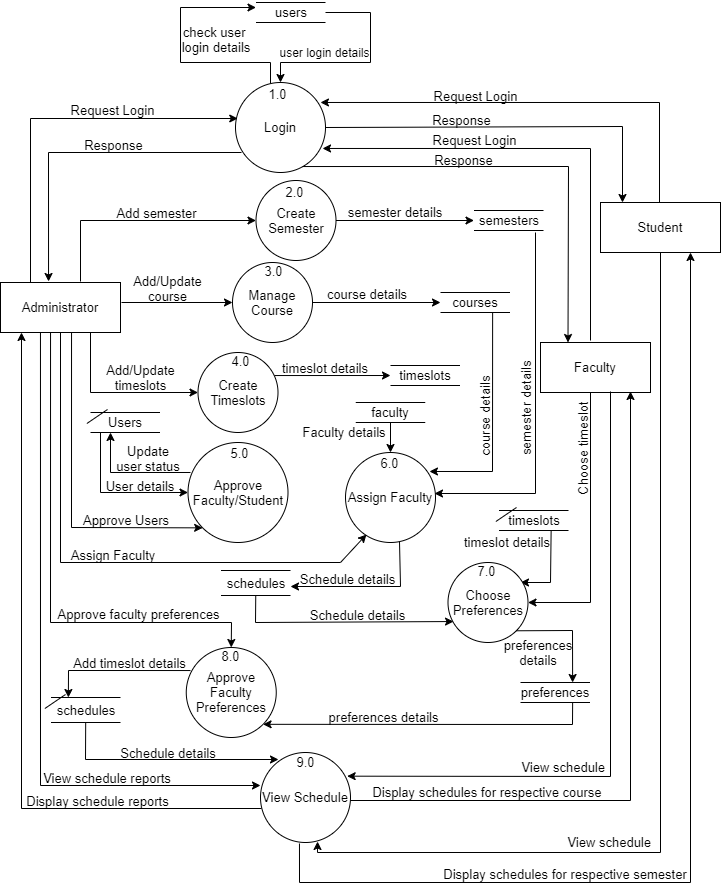


Figure 3.1.4. 2: Level 1 DFD for Class Schedule Management System

## 3.2 System Design

The different elements of the Class Schedule Management System are designed in order to realize different functional requirement of the system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. The system is designed using various schema design, CASE tools and is described using following designs.

### **3.2.1 Architectural Design**

The Class Schedule Management System has used the three-tier architecture which includes client tier, web tier, and data tier. The basic structure of architectural design of the system is shown below.

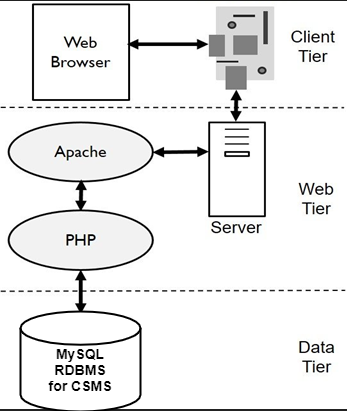


Figure 3.2.1. 1: Architectural Design for Class Schedule Management System

### **3.2.2 System Flowchart**

The figures shown below are the flowchart of Class Schedule Management System. Here, all the user successfully login to the system if their respective credentials match during login process. If faculty and student are not register then they need to register first then they are approved by the administrator. After successful login, system redirects users to respective dashboard. After successful login, administrator can create semesters, timeslots, and courses, assign faculty, approve faculty preferences, view schedules, and logout from the system. Likewise, after successful login, faculty can choose timeslot for respective course in respective semester, view respective schedule, and logout from the system. Similarly, after successful login, student can view schedules of respective semester and logout from the system successfully.

**For Administrator**

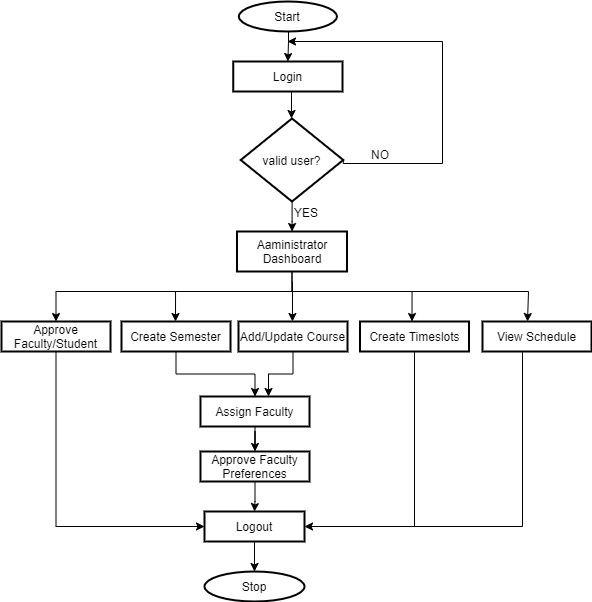


Figure 3.2.2. 1: System Flow Chart for Administrator

**For Faculty**



Figure 3.2.2. 2: System Flow Chart for Faculty

**For Student**

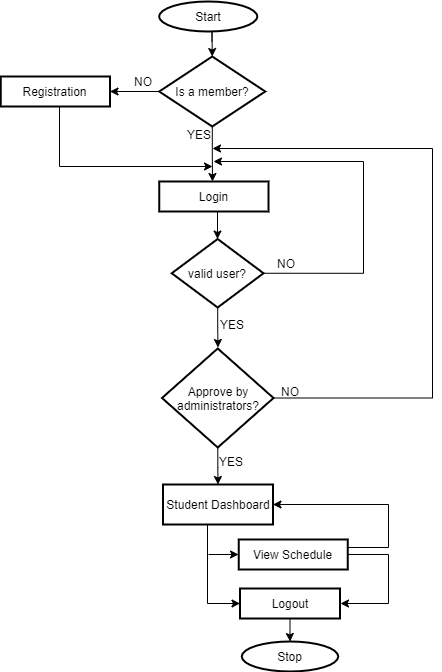


Figure 3.2.2. 3: System Flow Chart for Student

### **3.2.3 Database Schema Design**

The figure below is the database schema design of Class Schedule Management System. Database schema design organizes the data into separate entities, determines how to create relationships between organized entities, and how to apply the constraints on the data. In the figure below, there are nine tables in the databases each of them has their own fields and identified with their respective primary key which when used in another table, becomes a foreign key. The fields in each entity table have data type according to the data it stores. The cardinality is shown by the arrows as shown in the diagram.

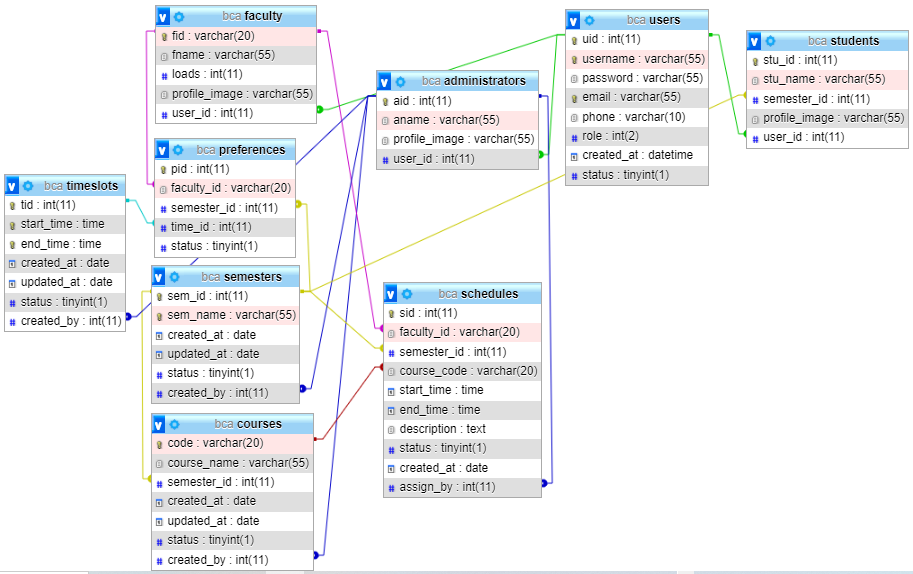


Figure 3.2.3. 1: Physical Schema Design of Class Schedule Management System

### **3.2.4 Interface Design**

User interface is the front-end application view to which user interacts in order to use the software. It is used to build interfaces in system focusing on looks and styles. Here we tried to design user friendly interfaces so that users find easy to use and pleasurable. The following are the UI design of Login page, Registration page, Administrator dashboard, Faculty dashboard, and Student dashboard.

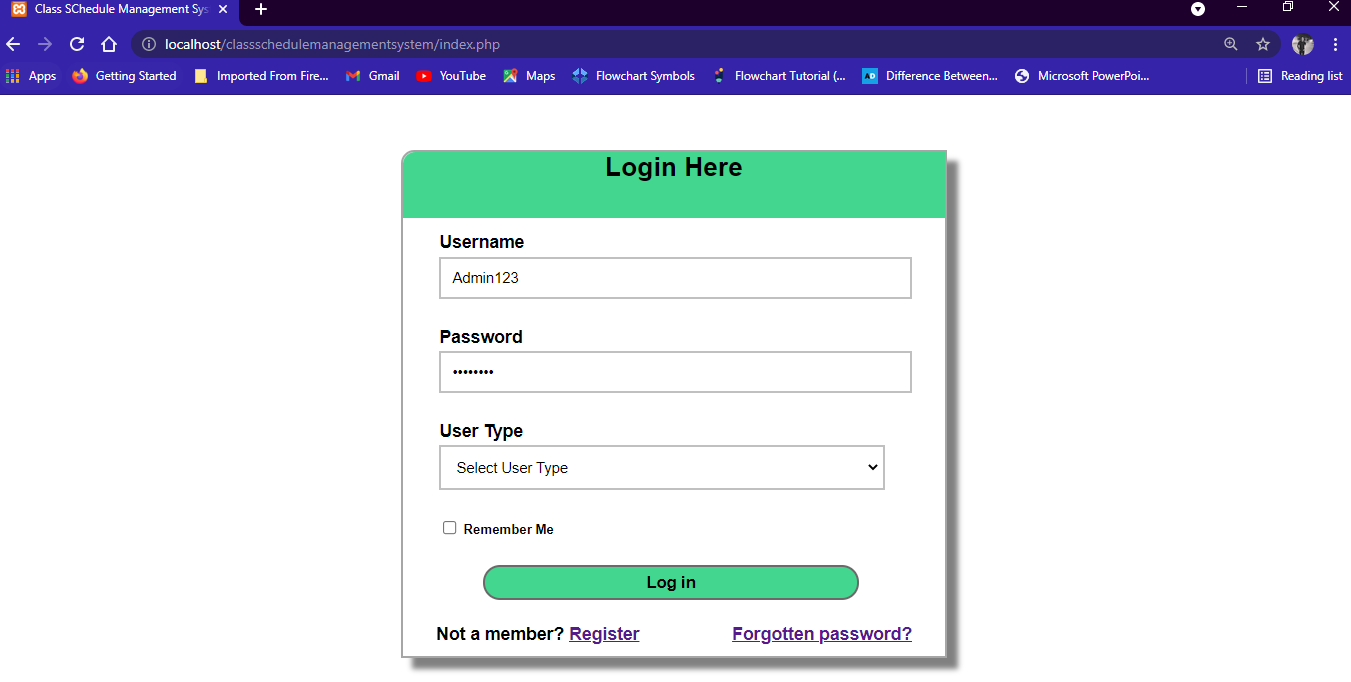


Figure 3.2.4. 1: Login page of Class Schedule Management System

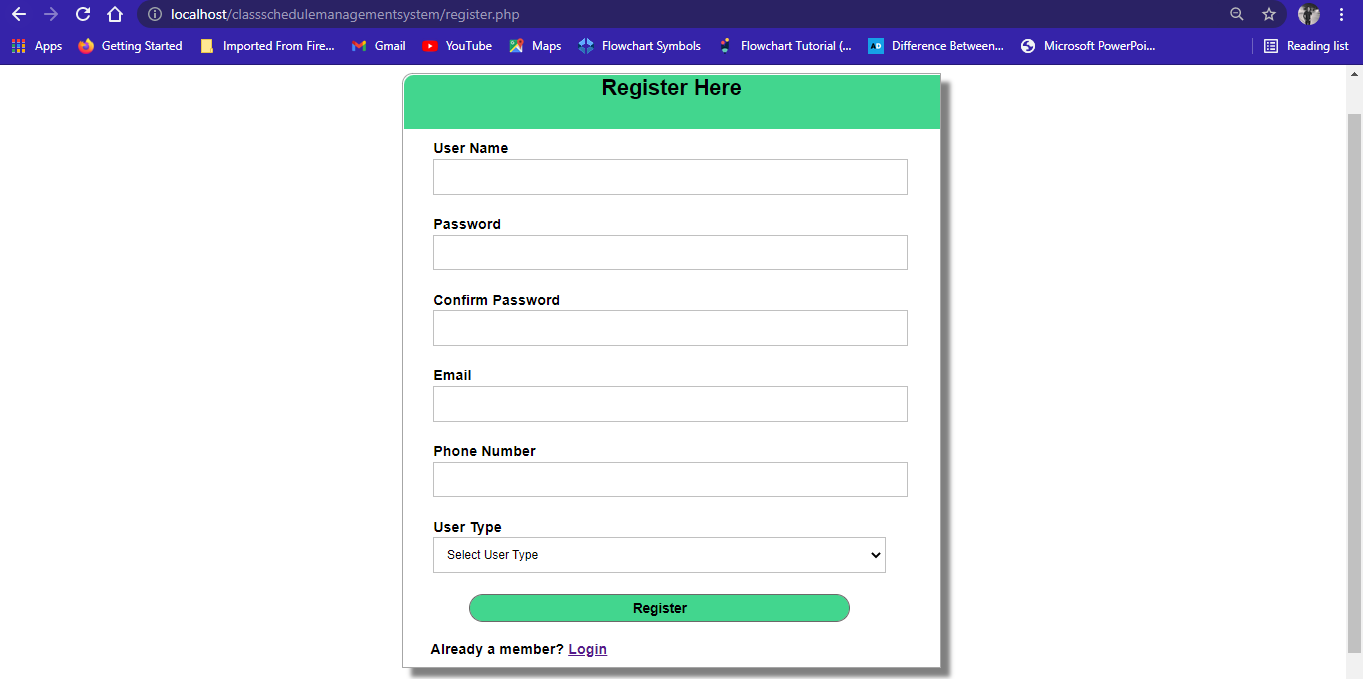


Figure 3.2.4. 2: Registration page of Class Schedule Management System

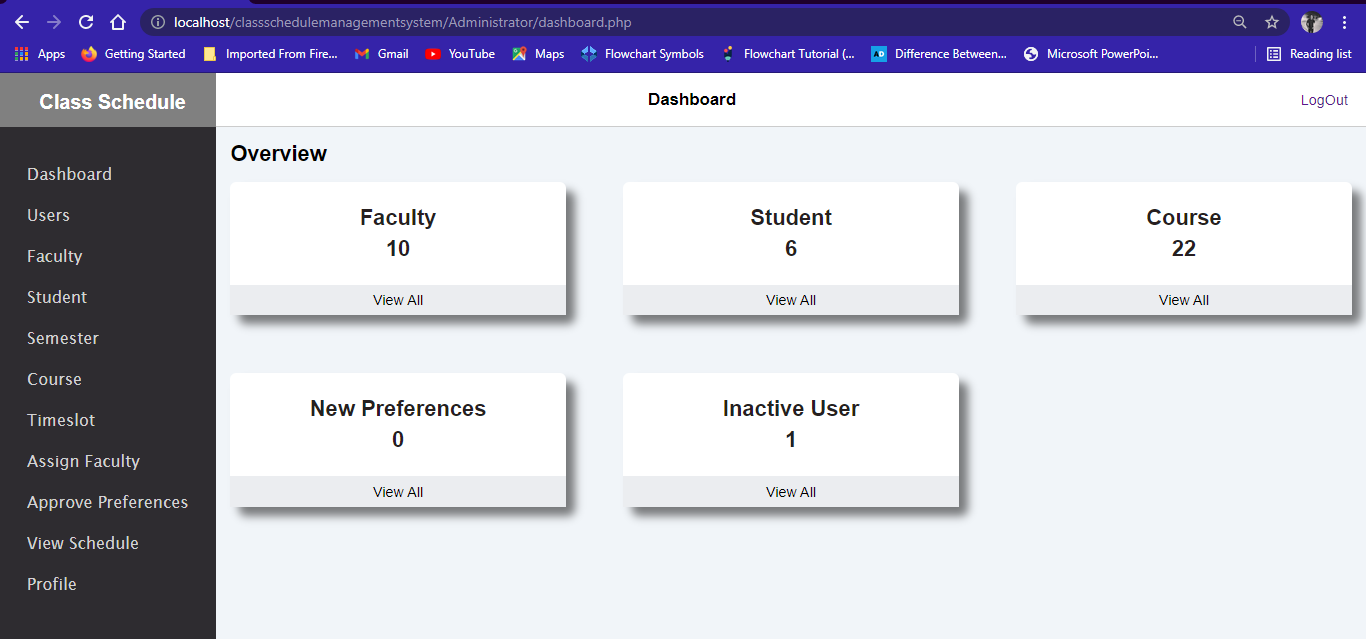


Figure 3.2.4. 3: Administrator dashboard of Class Schedule Management System

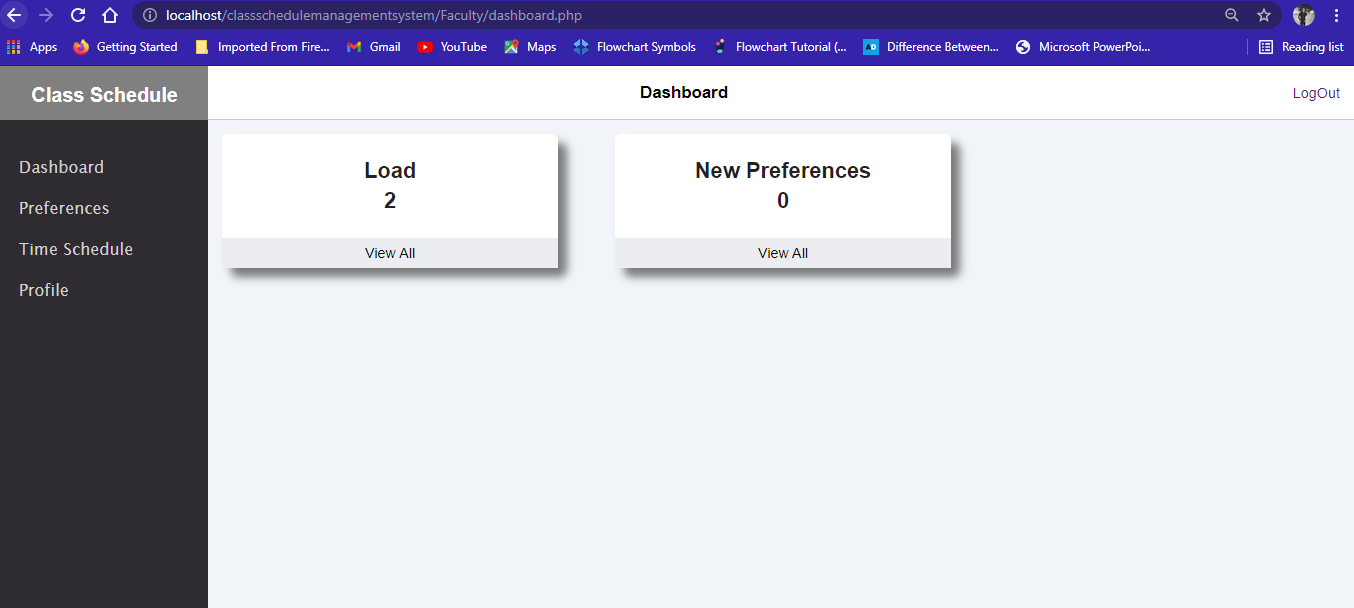


Figure 3.2.4. 4: Faculty dashboard of Class Schedule Management System

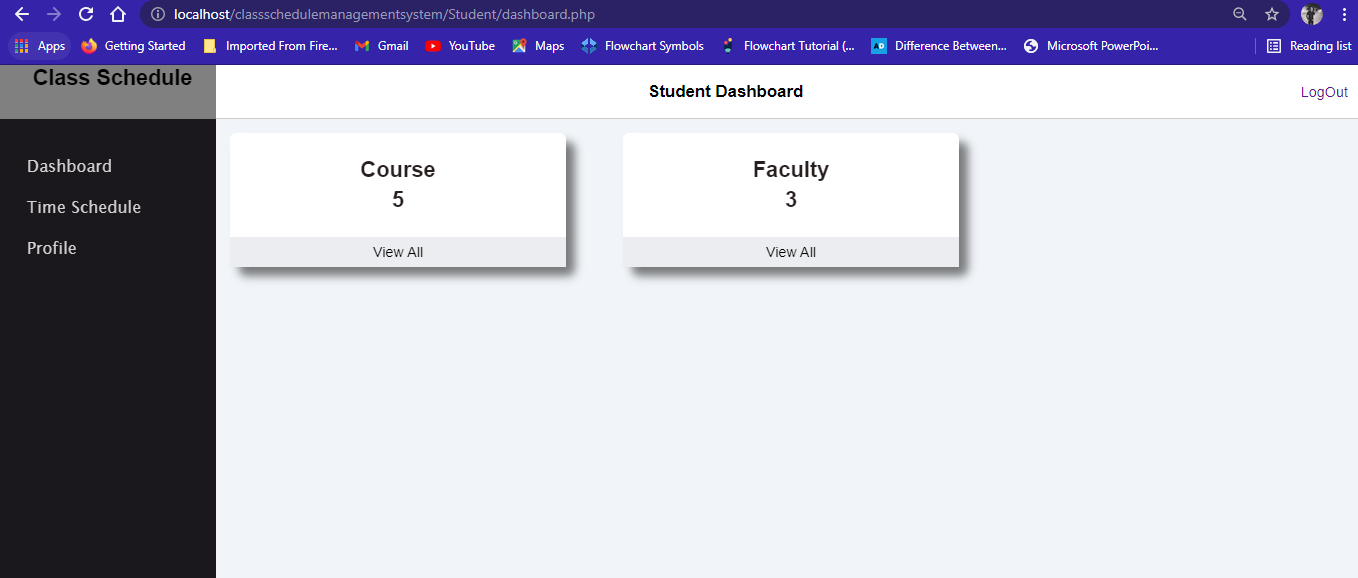


Figure 3.2.4. 5: Student dashboard of Class Schedule Management System

### **3.2.5 Physical DFD**

Here, all the users can successfully login to the system if the given credentials match with the information stored in database. Further, administrator can create semesters, courses, and timeslots, approve students/faculty, assign faculty, approve faculty preferences, and view schedules. Similarly, faculty can choose time preferences and view their own schedules. And, student can view schedules of respective semester. The data required in the different processes are stored and retrieved from respective data tables.

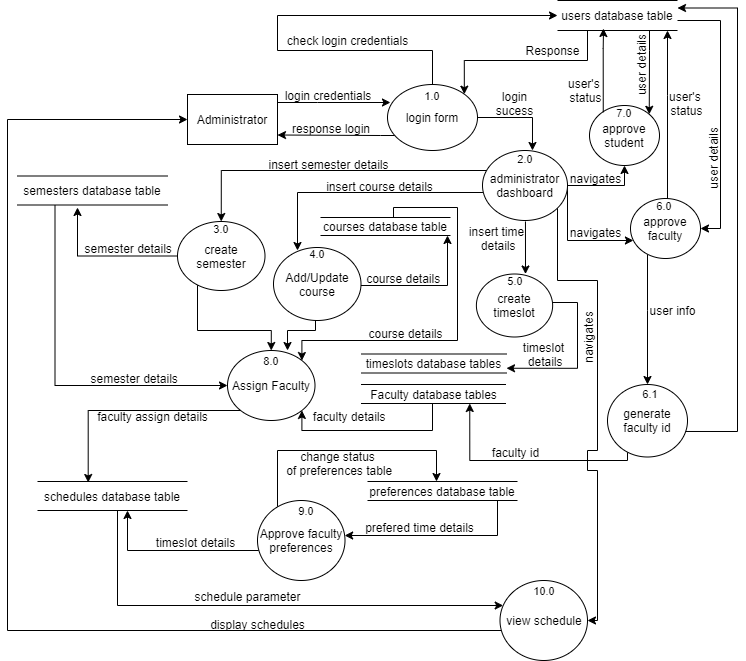


Figure 3.2.5. 1: Physical DFD for Administrator

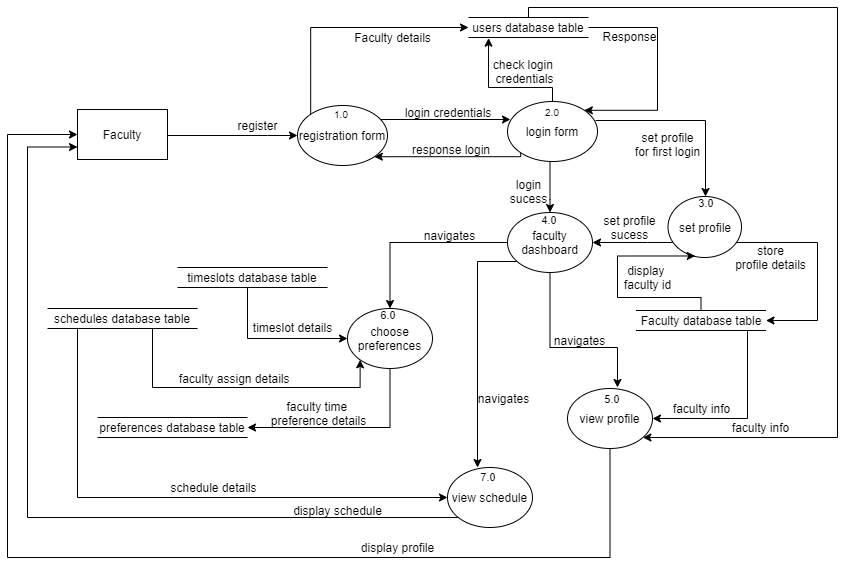


Figure 3.2.5. 2: Physical DFD for Faculty

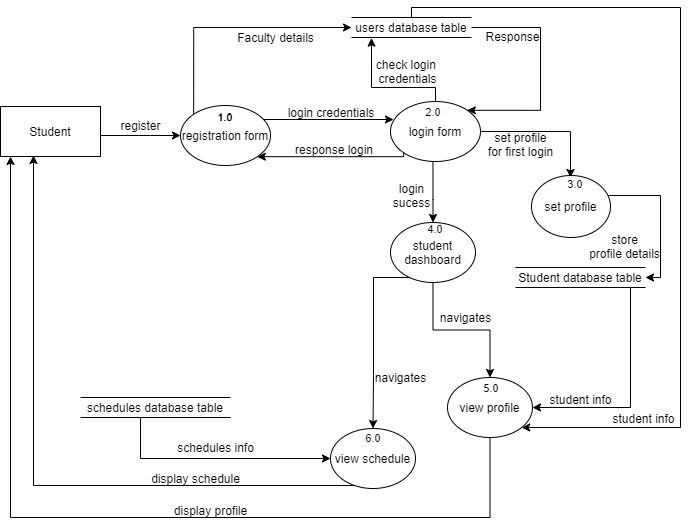


Figure 3.2.5. 3: Physical DFD for Student

# CHAPTER 4

# IMPLEMENTATION AND TESTING

## 4.1 Implementation

## 4.1.1 Tools Used

Following are the different types of case tools used in the project:

1. **Front-End Tools**

* **HTML**

In this system, html is used for creating different webpages and sites. It is used to create and structure tables, forms, sections, headings, links, paragraphs, images and so on using various tags and elements.

* **CSS**

In this system, CSS is used for styling different tags of html using different types of selectors such as id, class, and tag name. The different types of CSS are used such as inline CSS, internal CSS, and external CSS for styling the system’s web pages. With the use of CSS, we control the text color, font style, margins, paddings, the spacing between paragraphs, sizing of tables, layout designs, and many more.

* **JavaScript**

In this system, JavaScript is used to make dynamic, interactive and responsive web pages and for handling events such as onclick, submit, hover and so on.

* **AJAX**

In this system ajax is used to make dependent select box. During assigning faculty, according to the semester selected in drop-down box, dependent courses are retrieved from courses database table and displayed in next drop-down box without reloading the page. This is possible because of AJAX.

* **jQuery**

In this system, DataTables is used for creating table listings and adding interaction features to the tables.

1. **Backend Tool**

* **PHP**

PHP is a server-side scripting language which is a powerful tool for making dynamic and interactive web pages. It is used for backend purpose to create connectivity with database and also used in form validation, storing data in database tables, retrieving data form database tables, and other manipulation works. Further it is used for implementing sessions and cookies.

* **Server**

XAMPP is a free and open-source cross-platform which provides the Apache web server, MySQL database (actually MariaDB), Php and Perl (as command-line executables and Apache modules) all in one package. Hence, in Class Schedule Management System, Apache server is used to run php files and creating fast and dynamic web pages.

1. **Documentation Tool**

* **MS word**

Microsoft word is a word processing and documentation creation platform in which this document is created.

1. **Diagram Tool**

* **Draw.io**

This is used to generate diagrams such as use case, ER, context, DFD for system analysis and design of Class Schedule Management System. Diagrams were created using this tool in order to save time since all components are available with drag and drop functions.

### **4.1.2 Implementations Details of Modules**

The different modules of the Class Schedule Management System are described as below:

**Administrator Module**

The administrator can login to the system by entering respective valid credentials. After login administrator can control further sub modules given below.

* **Semester Module**

In this module, administrator can add, update, view and delete the semester details in the existing system. The administrator can add semester by clicking on add button after filling the details in add semester form. Then the page displays all the list of added semesters.

* **Course Module**

In this module, administrator can add, update, view and delete the course details in the existing system. The administrator can add course by clicking on add button after filling the details in add course form. Then the page displays all the list of added courses.

* **Timeslot Module**

In this module, administrator can add, update, view and delete the timeslot details in the existing system. The administrator can add timeslots by clicking on add button after filling the details in add timeslot form. Then the page displays all the list of added timeslots.

* **Users Module**

In this module, administrator can view the list of users registered in the system and can approve them.

* **Assign Faculty Module**

In this module, administrator can assign faculty to a course of respective semester and view the list.

* **Preference Module**

In this module, administrator can approve time preferred by the faculty for the assign course of respective semester.

* **View Schedule Module**

In this module, administrator can view class schedule semester wise and faculty wise.

**Faculty Module**

Faculty members first register into the system by entering all the details such as username, password, email, phone number, and user type. Then they can login to the system after the approval of administrator. After login, they can choose time preference for the assigned course in particular semester and can view their respective schedules.

**Student Module**

Student first register into the system by entering all the details such as username, password, email, phone number, and user type. Then they can login to the system after the approval of administrator. After login, they can view schedules.

**Login Module**

In this module, administrator, faculty member and student login to the system using their valid username and password.

**Register Module**

In this module, administrator, faculty member and student register into the system by entering all the details such as username, password, email, phone number, and user type.

## 4.2 Testing

Testing is a very important phase for any types of software. A software should go through different testing process to ensure that the website is working in the manner in which it was intended to. During the phase of the development of the system, our system is tested time and again. The series of testing conducted are as follow:

### **4.2.1 Test Cases for Unit Testing**

Generally, a software constitutes of several different modules and so does this project. A modules or unit can refer to a function, individual program or even a procedure. The register and login forms are tested so that they do not accept invalid inputs. Here is the list of test cases.

**User Registration**

Table 4.2.1. 1: Test case for User Registration of Class Schedule Management System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Open Application | <http://localhost/classschedulemanagementsystem/register.php> | Open registration page | Open registration page | Pass |
| 2 | Register with invalid password | User Name: Faculty1  Password: 123  Confirm Password: 123  Email: [faculty1@gmail.com](mailto:faculty1@gmail.com)  Phone: 9898971000  User Type: Faculty | Display message “  \*at least one uppercase, \*at least one lowercase, \*at least one digits \*may include special character \_ # @ % \* $ - \*minimum 8 character \*maximum 25 characters  ” | Registration Fail | Pass |
| 3 | Register with valid  password | User Name: Faculty1  Password: Faculty1  Confirm Password: Faculty1  Email: [faculty1@gmail.com](mailto:faculty1@gmail.com)  Phone: 9898971000  User Type: Faculty | Display message “  Faculty1 registered successfully” | Registration successful | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | Register with invalid name | User Name: prakash  Email: [prakash@gmail.com](mailto:prakash@gmail.com)  Password: Prakash123  Confirm Password: Prakash123  User Type: student | Display message” Username must be greater than or equal to 8 characters” | Registration Fail | Pass |
| 5 | Register with valid username, email, password and user type. | User Name: prakash123  Email: [prakash@gmail.com](mailto:prakash@gmail.com)  Password: Prakash123  Confirm Password: Prakash123  User Type: student | Display message”  prakash123 is registered successfully” | Registration  successful | Pass |
| 6 | Register without user type | User Name: prakash123  Email: [prakash@gmail.com](mailto:prakash@gmail.com)  Password: Prakash123  Confirm Password: Prakash123  User Type: | Display message” Role is required” | Registration  fail | Pass |
| **7** | Login without filling any fields | User Name:  Email:  Password:  Confirm Password:  User Type: | Display messages”  Username is required.  Password is required.  Confirm your password.  Email is required.  Phone no is required.  Usertype is required.” | Registration  fail | Pass |

**User Login**

Table 4.2.1. 2: Test case for User Login of Class Schedule Management System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test  Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Open Application | <http://localhost/classschedulemanagementsystem/index.php> | Open registration page | Open registration page | Pass |
| 2 | Login with correct username, password and user type. | User Name: Admin123  Password: Admin123  User Type: Administrator | Administrator  Dashboard | Login successful | Pass |
| 3 | Login with wrong password | User Name: admin123  Password: admin123  User Type: Administrator | Display messages “Credential does not match” | Login fail | Pass |
| 4 | Login with wrong username | User Name:Admin123  Password: Admin123  User Type: admin | Display messages “Credential does not match” | Login fail | Pass |
| 5 | Login with wrong user type | User Name: admin123  Password: Admin123  User Type: Faculty | Display message”  Please choose correct user type” | Login fail | Pass |
| 6 | Login without filling any fields. | User Name:  Password:  User Type: | Display message”  User Name is required.  Password is required.  Role is required.” | Login fail | Pass |

Table 4.2.1. 3: Test case for Addition of semester, course, and timeslot

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Adding semester | Semester: Seventh Semester Status: checked | Semester Added Successfully | Semester Added Successfully | Pass |
| 2 | Adding course | Course Name: Software Engineering  Course Code: CACS253  Semester: Fourth semester  Course status: checked | Display message “  CACS253 added successfully” | Course added successfully | Pass |
| 3 | Adding timeslot | Start time: 10:00 am  End time: 11:00 am  Timeslot Status: checked | Display message “  time added successfully” | Timeslot added | Pass |
|  |  |  |  |  |  |

Table 4.2.1. 4: Test case for Assigning Faculty

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Assigning Faculty in fourth semester | Faculty: Faculty10[F96]  Semester: Fourth Semester  Course: Numerical Methods[CACS252] | Faculty assign successfully | Faculty assign successfully | Pass |
| 2 | Again, assigning Faculty in the fourth semester | Semester: Seventh Semester Status: checked | Display message “faculty is already assigned in the semester” | Fail to assign faculty | Pass |

Table 4.2.1. 5: Test case of Choose Preferences for Faculty

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Choosing  timeslot for a single course | Checked timeslot 10:00 am – 11:00 am | Choose Successfully | Choose Successfully | Pass |
| 2 | Choosing same  timeslots for two courses | Checked timeslots “11:00 am – 12:00 am” for one course and  11:00 am – 12:00 am  for another course | Display message” Please choose different time for different semester” | Fail to Choose | Pass |

### **4.2.2 Test Cases for system testing**

In system testing, evaluation of how the various components of an application interact together in the full, integrated system is tested. System testing verifies that an application performs tasks as designed.

Table 4.2.2. 1: Test case for Administrator module

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Admin login | User Name: Admin123  Password: Admin123  User Type: administrator | Login successful | Admin dashboard | Pass |
| 2 | Adding semester | Semester: Seventh Semester Status: checked | Semester Added Successfully | Semester Added Successfully | Pass |
| 3 | Adding course | Course Name: Software Engineering  Course Code: CACS253  Semester: Fourth semester  Course status: checked | Display message “  CACS253 added successfully” | Course added successfully | Pass |
| 4 | Adding timeslot | Start time: 10:00 am  End time: 11:00 am  Timeslot Status: checked | Display message “  time added successfully” | Timeslot added | Pass |
| 5 | Assigning Faculty in fourth semester | Faculty: Faculty10[F96]  Semester: Fourth Semester  Course: Numerical Methods[CACS252] | Faculty assign successfully | Faculty assign successfully | Pass |
|  |  |  |  |  |  |

Table 4.2.2. 2: Test case for Faculty module

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Scenario | Input | Actual Result | Expected Result | Test Result |
| 1 | Faculty Registration | User Name: Faculty1  Password: Faculty1  Confirm Password: Faculty1  Email: [faculty1@gmail.com](mailto:faculty1@gmail.com)  Phone: 9898971000  User Type: Faculty | Display message “  Faculty1 registered successfully” | Registration Successful | Pass |
| 2 | Faculty login | User Name: Faculty1  Password: Faculty1  Confirm Password: Faculty1  Email: [faculty1@gmail.com](mailto:faculty1@gmail.com)  Phone: 9898971000  User Type: Faculty | Display message “  Faculty1 registered successfully” | Registration successful | Pass |
| 3 | Choosing  timeslot for assigned course | Checked timeslot 10:00 am – 11:00 am | Choose Successfully | Choose Successfully | Pass |

# CHAPTER 5

# CONCLUSION AND FUTURE RECOMMENDATIONS

## 5.1 Lesson Learnt/Outcome

The development of Class Schedule Management System made us learn and gain the knowledge in different aspects. It helped us to apply the knowledge that we have acquired in our classes. Project development taught us how to turning ideas into reality. We learned lots of problem-solving skills, proper planning, communication, team work, proper use of guidelines, time management, writing skills and management of team in order to achieve goals.

After development of the project there are various lesson that we learnt

➢ Learn how to do system analysis and feasibility study with proper planning.

➢ Learn full cycle of project development.

➢ Learn to integrate the system UI with database.

## 5.2 Conclusion

Class Schedule Management System has been developed successfully. The system is mainly designed to reduce the manual work of creating schedule. It is a system develop for BCA department which makes viewing class schedule experience amazing and quick for the end users with user friendly interface. The system can provide a database for storing records and information of enrolled courses, faculty members, students, and semesters. Faculty and Student can register and login to the system and view respective schedule. Further, faculty can choose timeslot for respective course in respective semester. Administrator can create semesters, timeslots, and courses, assign faculty member, approve faculty preferences and view schedules. The system displays schedule on faculty basis as well as semester basis. Hence, the system will help the department to loosen the burden and increase efficiency.

## 5.3 Future Recommendations

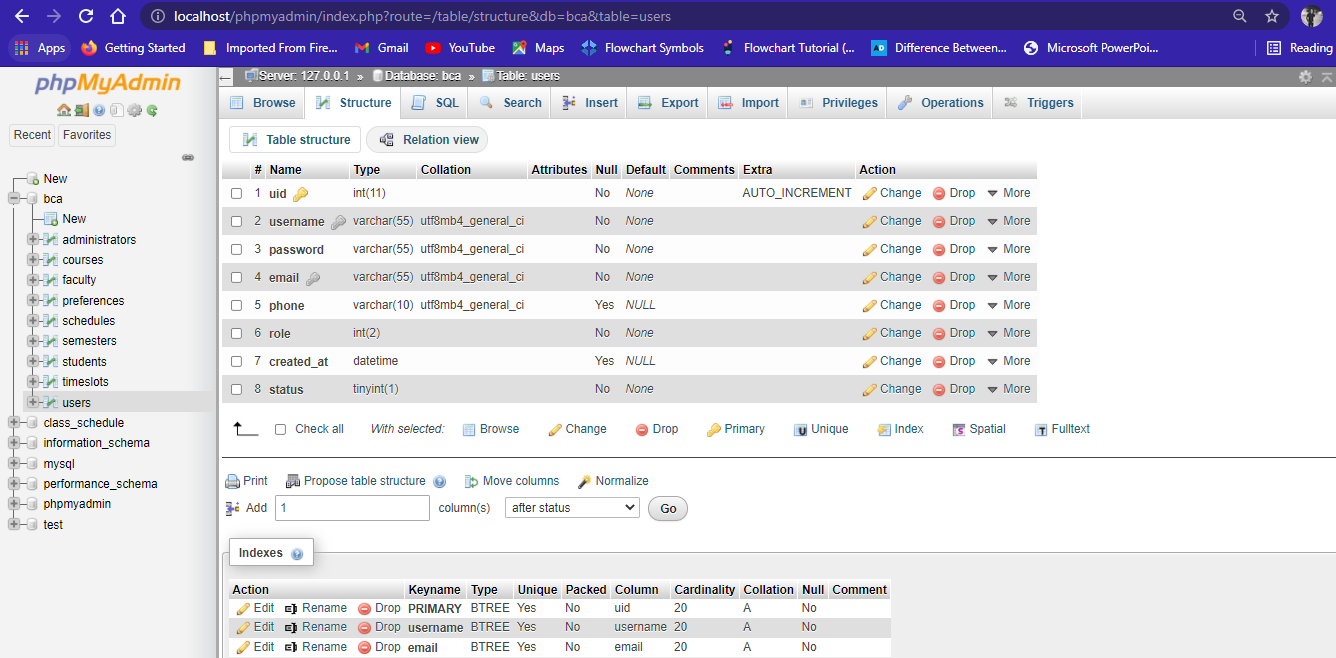
There is always a room for improvement in any software or system, however good and efficient it may be. But the important thing is that the system should be flexible enough for further modifications. Considering this important factor, the system is designed in such a way so that it can be further enhanced. The scope of the system could be increase to other departments having multiple sections as well, and the security of the database could be enhanced further, and so on. More research is needed to complete our interactive, automatic class scheduling system as there is always space for improving in any system.

# REFRENCES

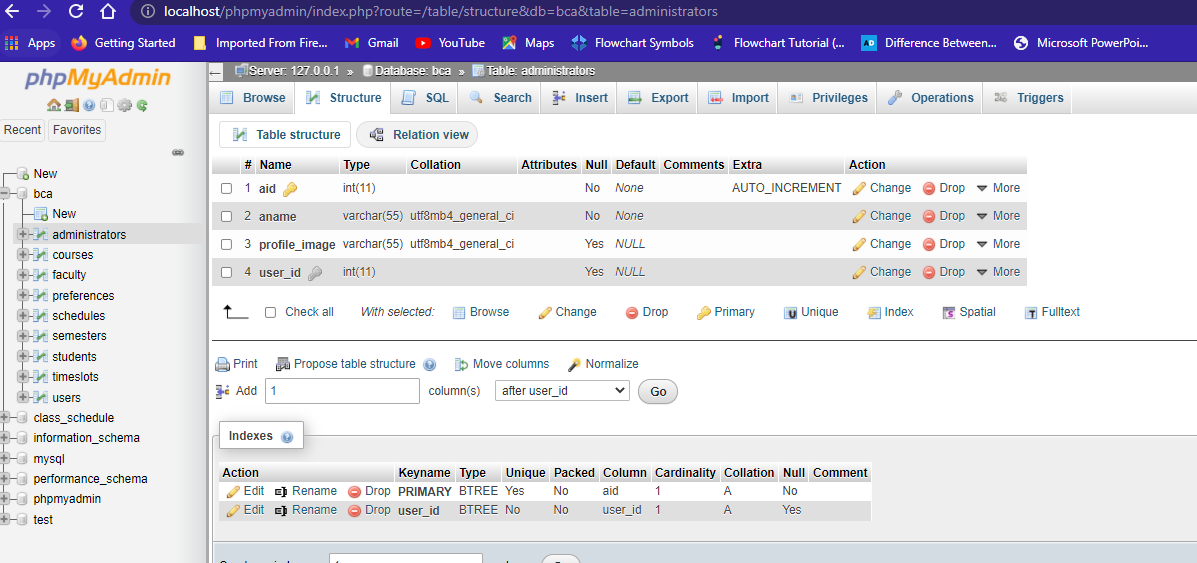
|  |  |
| --- | --- |
| [1] | P. Y. Abdullah, "Class Schedule System," p. 8, 08 September 2019. |
| [2] | "Timetable Management System Web Application Development," *Journal of Software Engineering & Software Testing,* vol. 5, no. 2, p. 17, 2020. |
| [3] | E. S. B. A. AZIZ, "ONLINE CLASS SCHEDULING MANAGEMENT," 2017. |

# APPENDIX: SYSTEM SCREENSHOTS

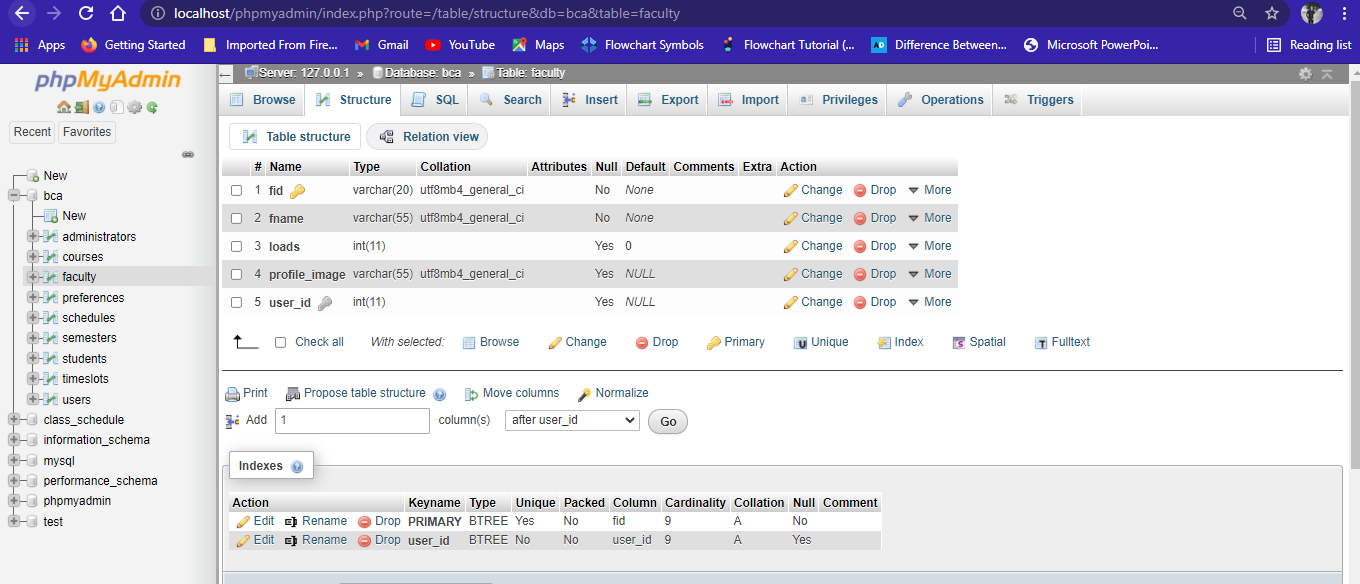
* **Database Overviews**
* **Users Database**



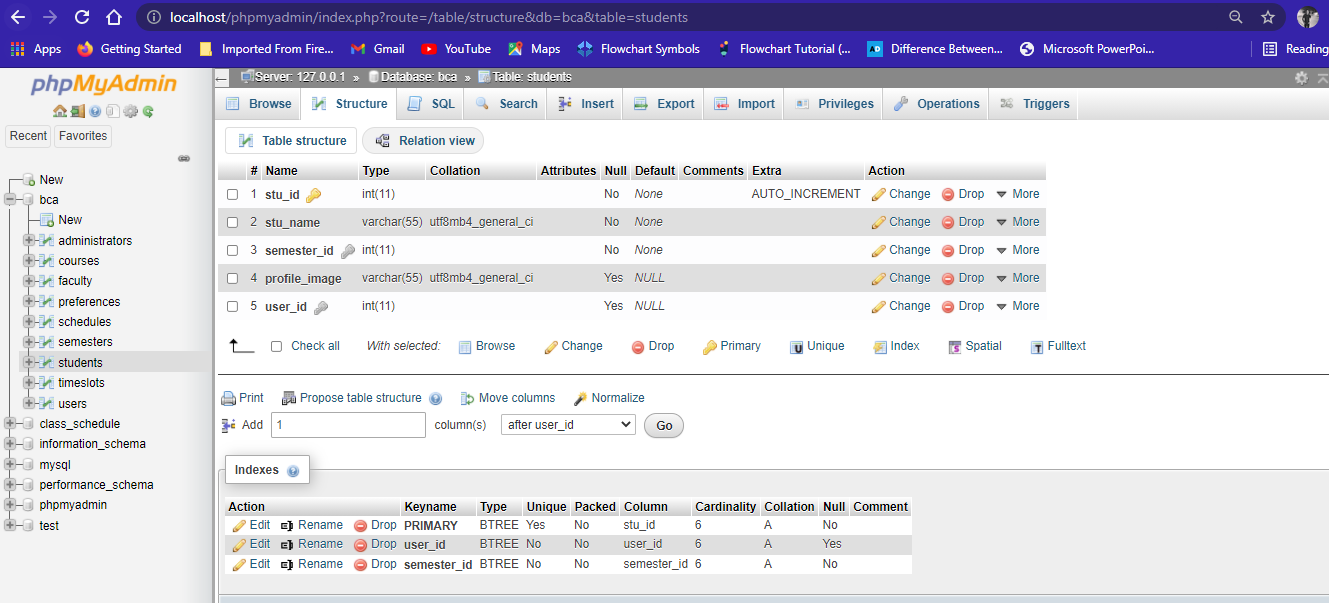
* **Administrators Database**



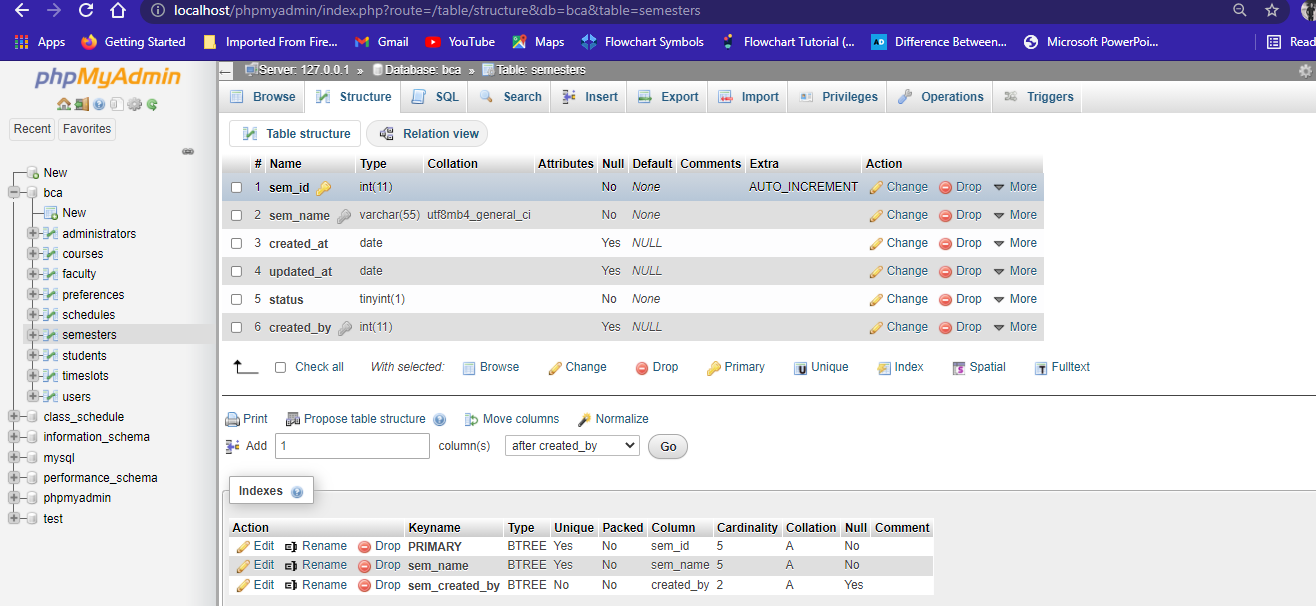
* **Faculty Dashboard**



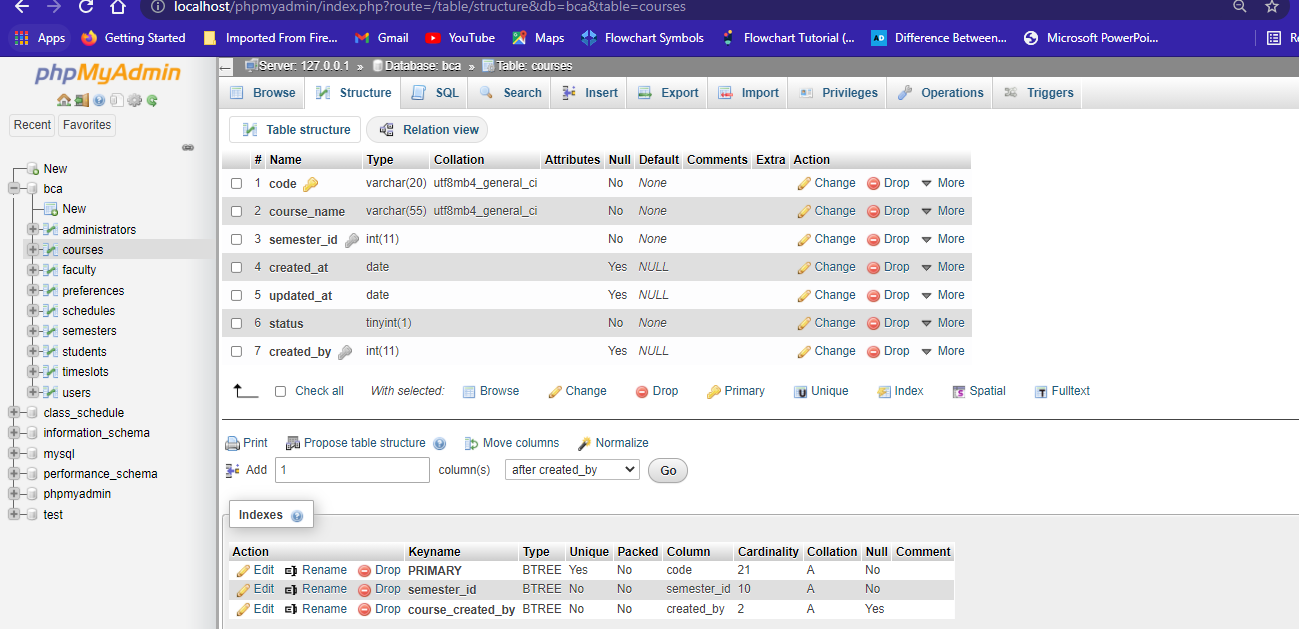
* **Students Database**



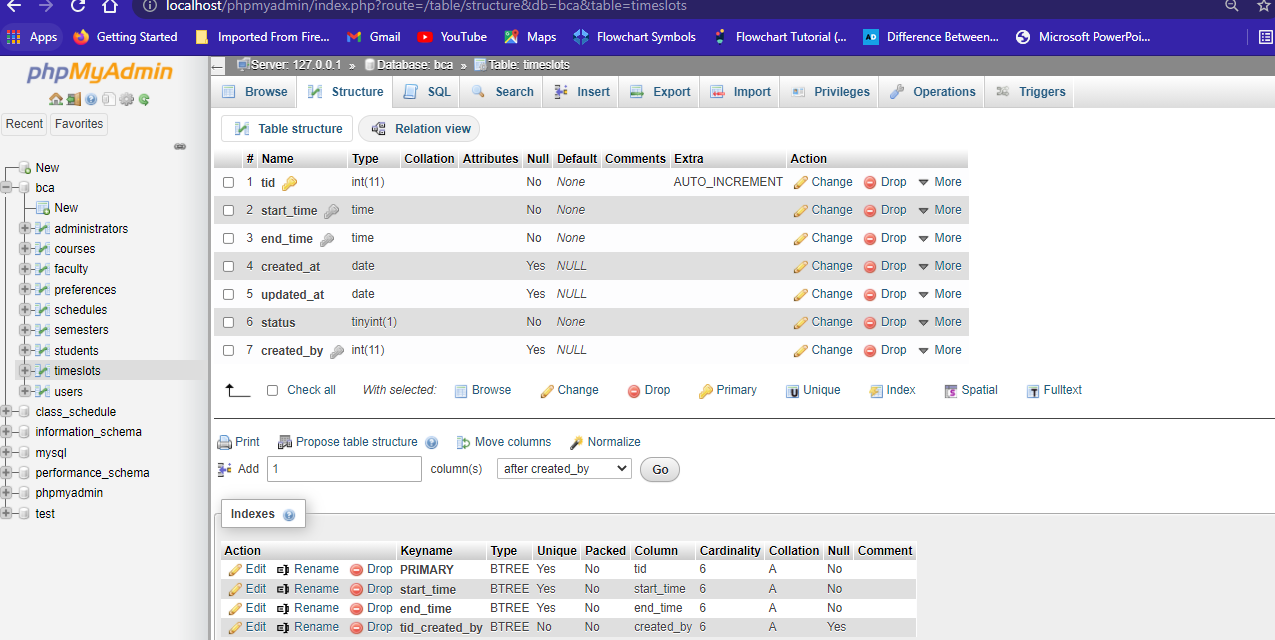
* **Semesters Database**



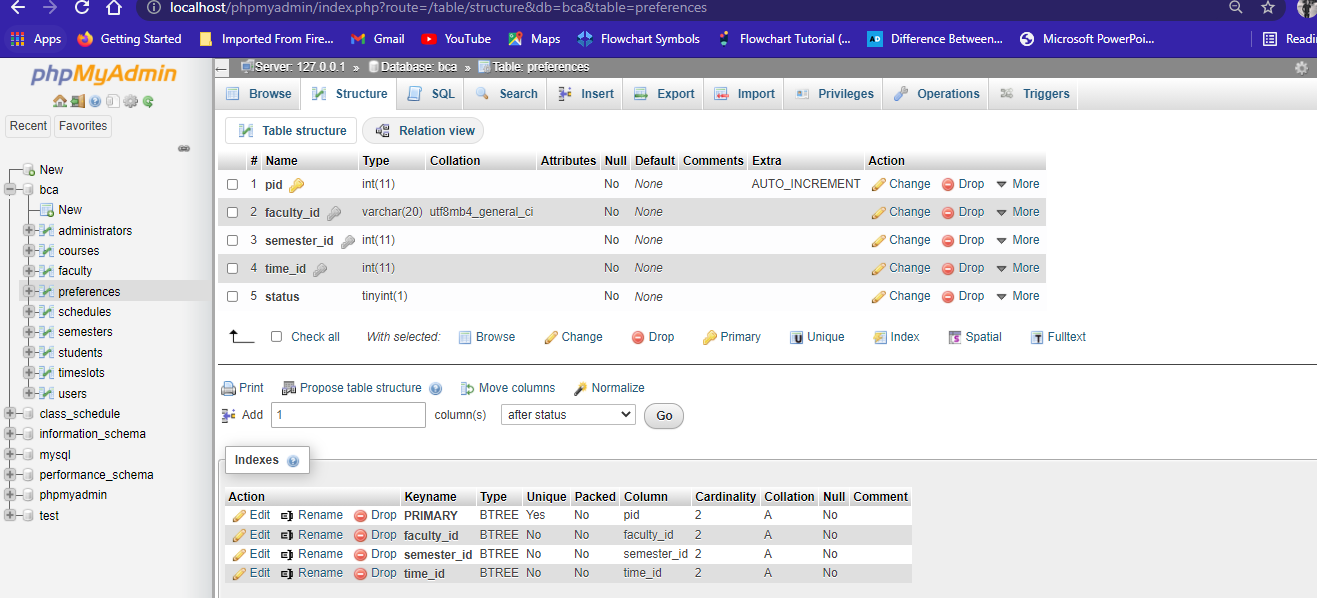
* **Courses Database**



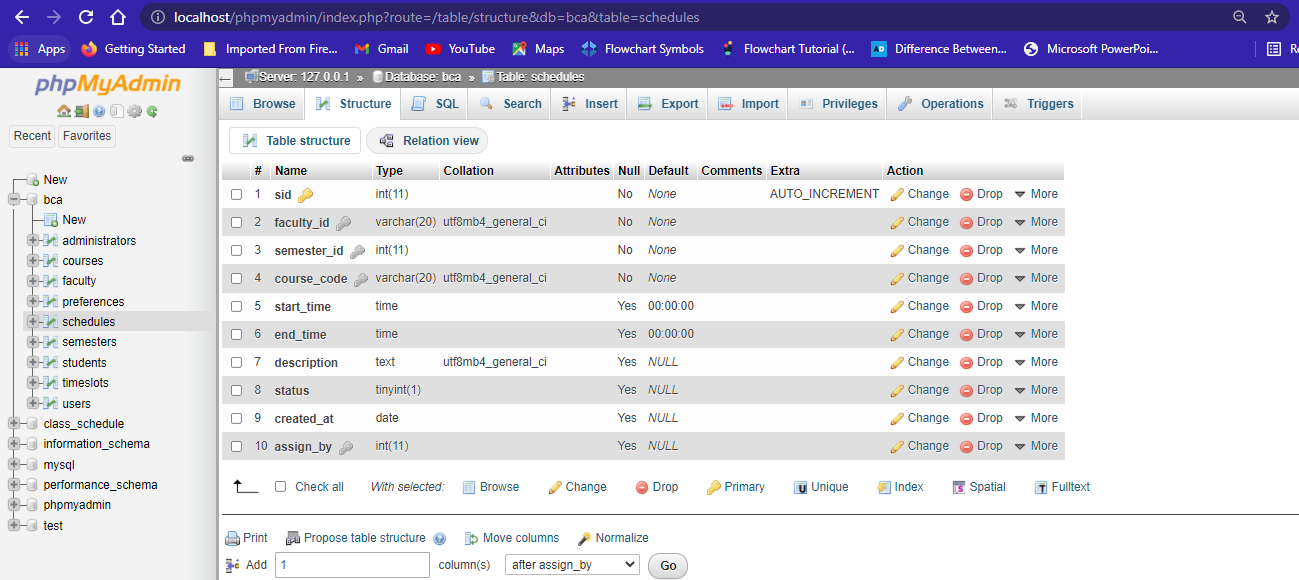
* **Timeslots Database**



* **Preferences Database**



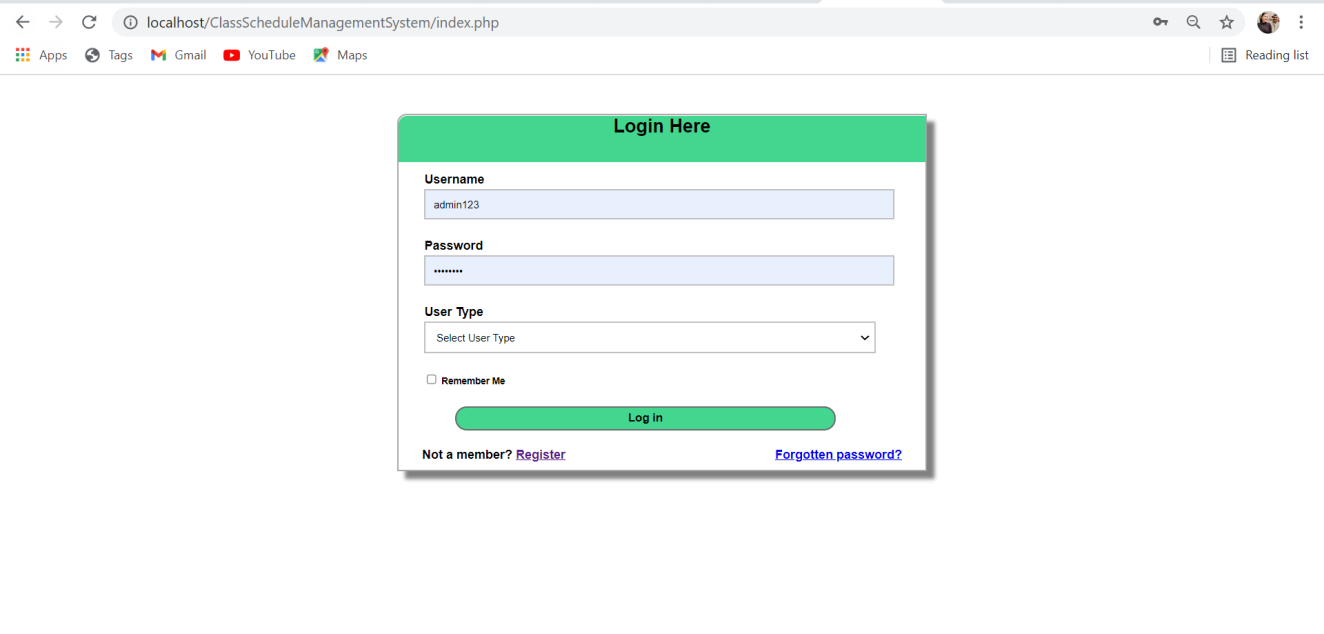
* **Schedules Database**



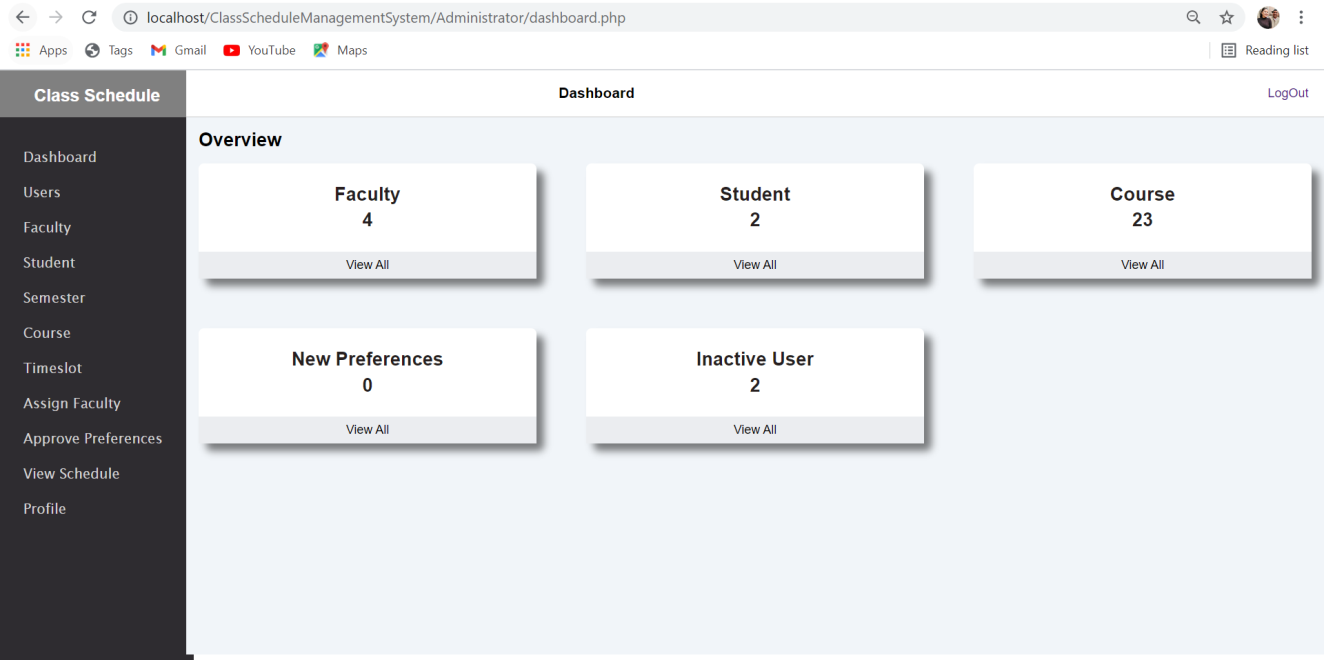
* **Front End Overview**
* **Register page**

****

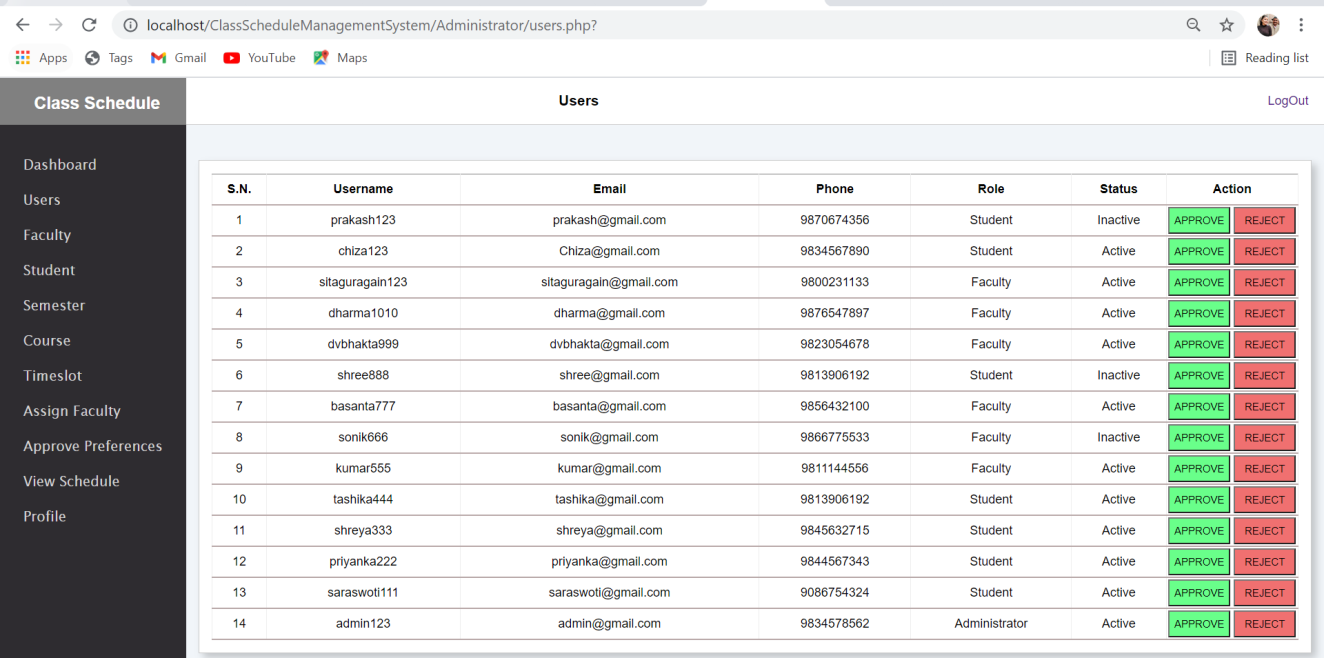
* **Login page**

****

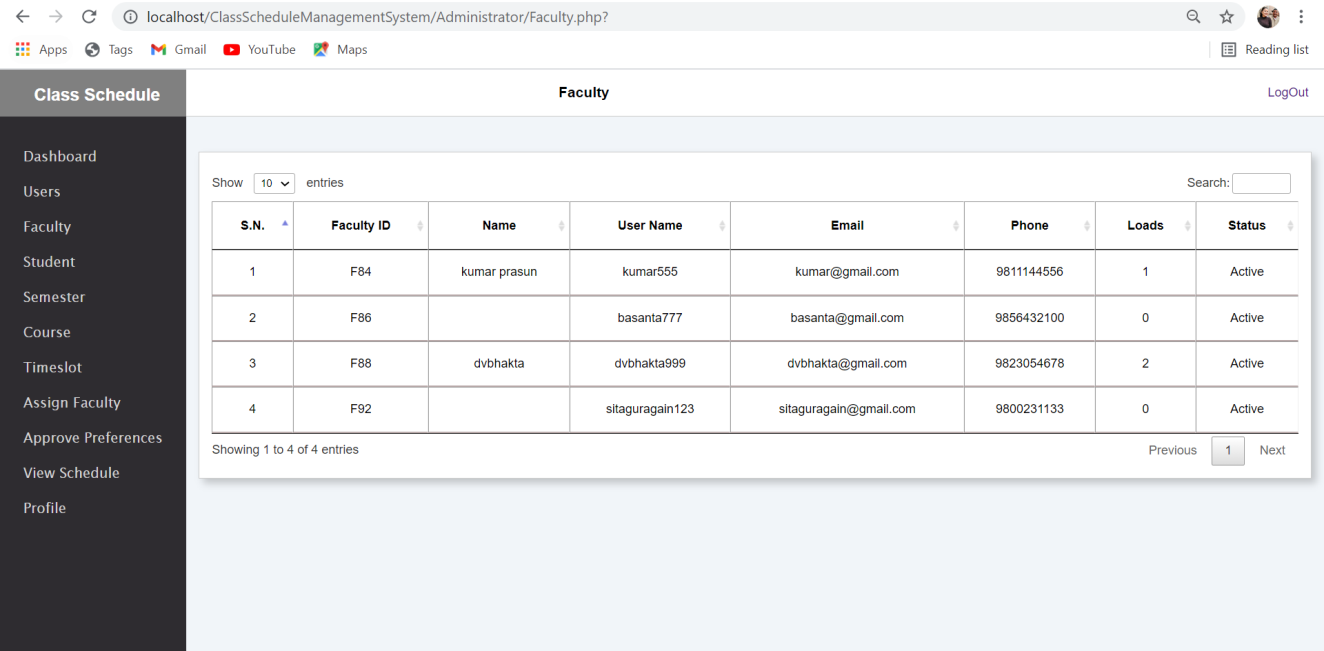
* **Admin Dashboard page**

****

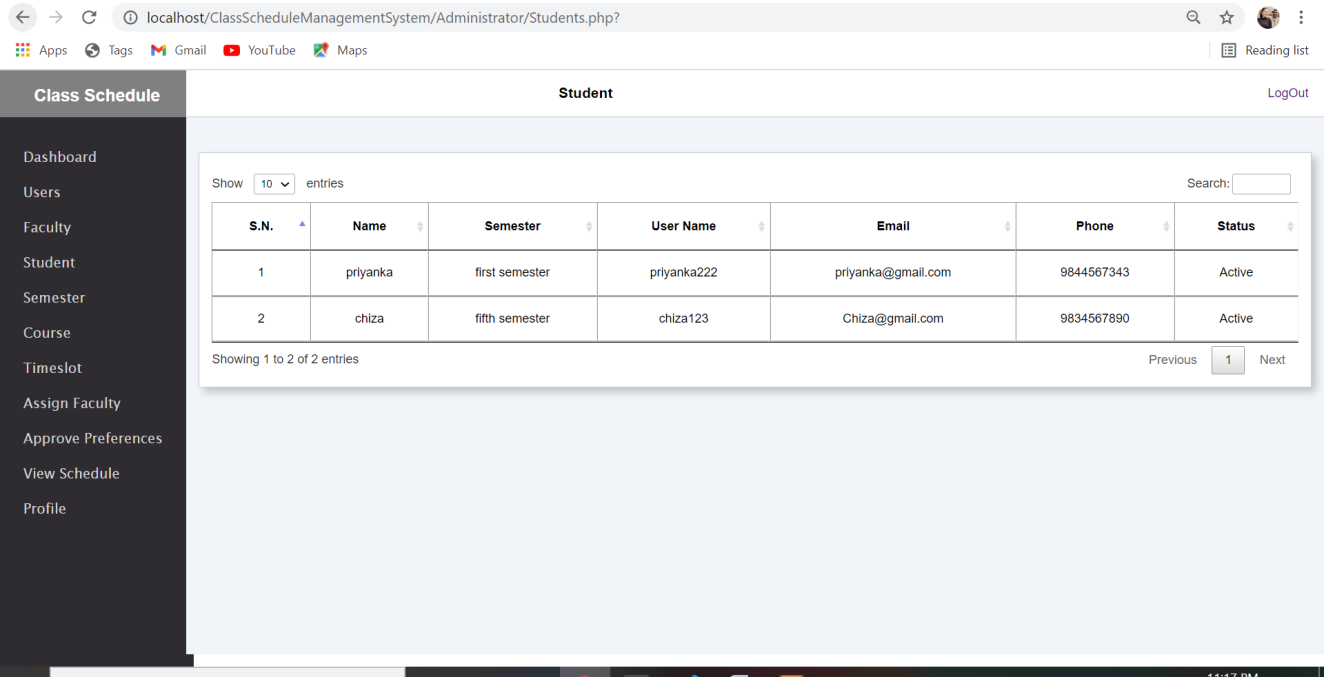
* **Admin Dashboard/users/users page**

****

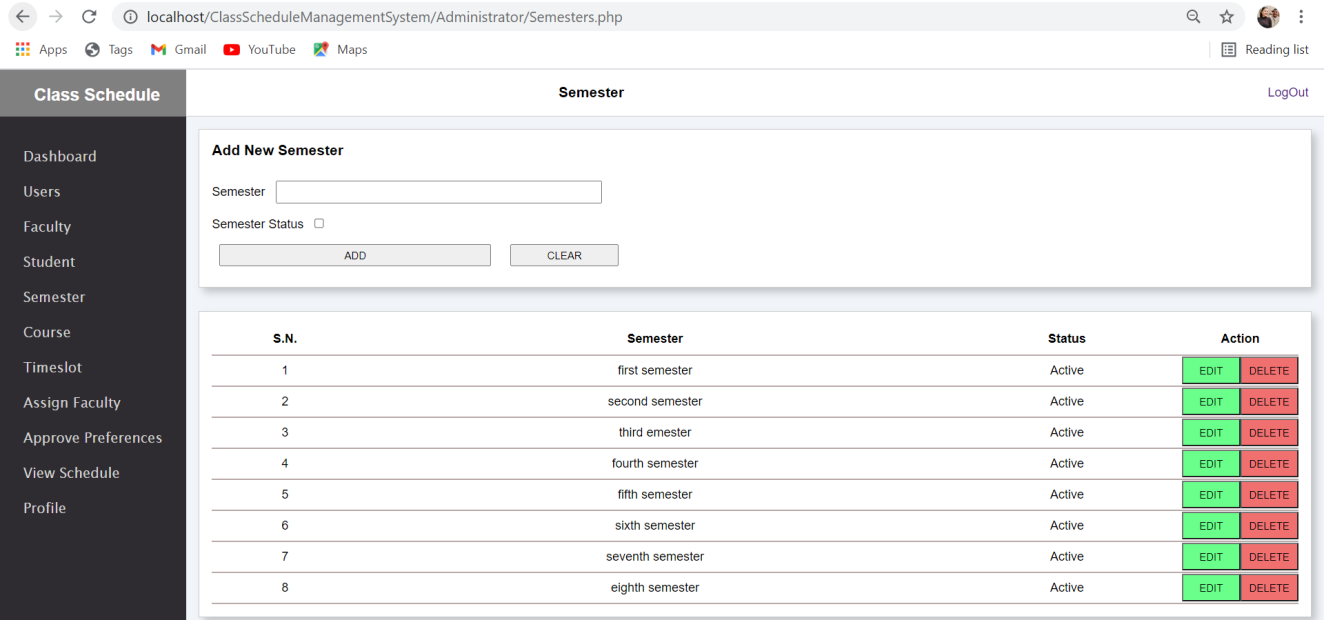
* **Admin Dashboard/Faculty page**

****

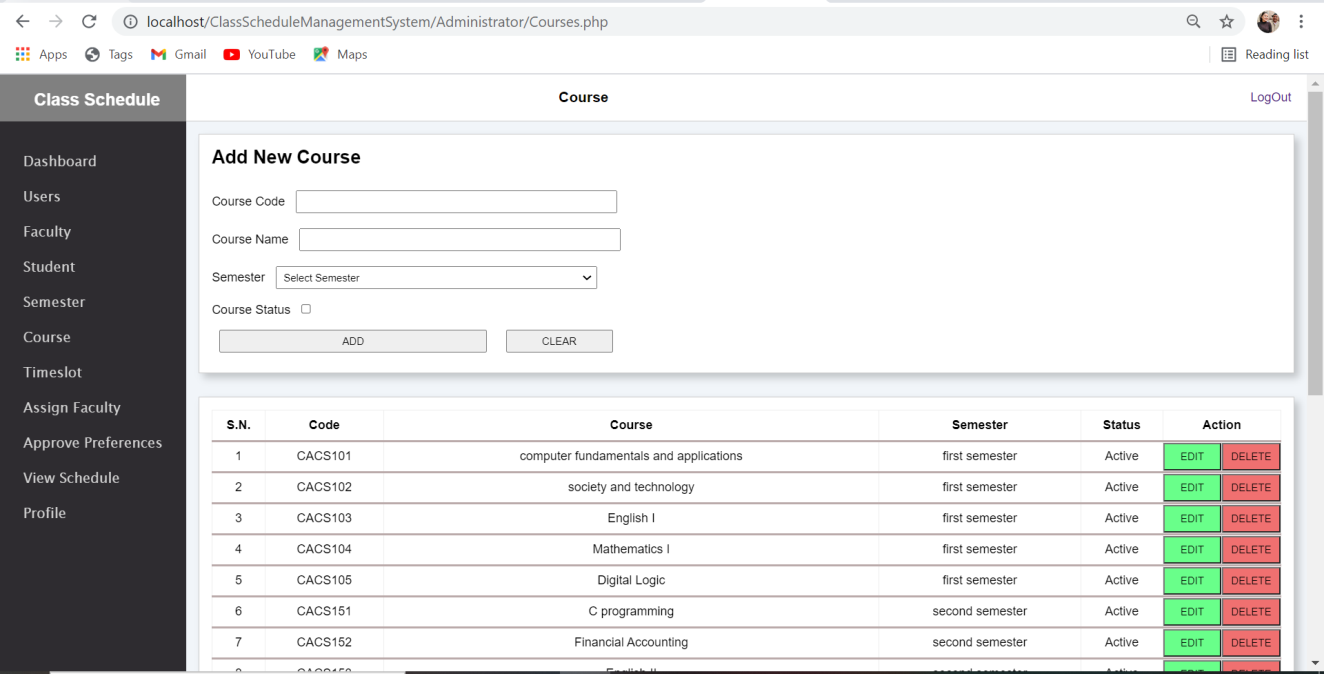
* **Admin Dashboard/Student page**

****

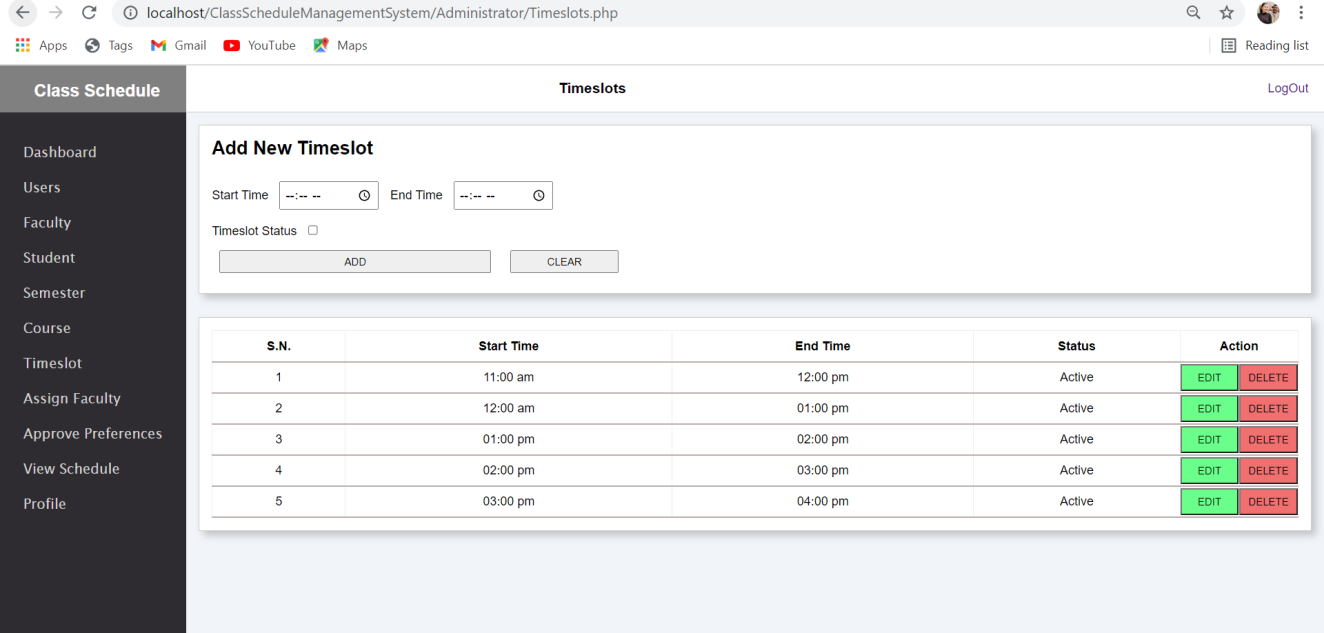
* **Admin Dashboard/semester page**

****

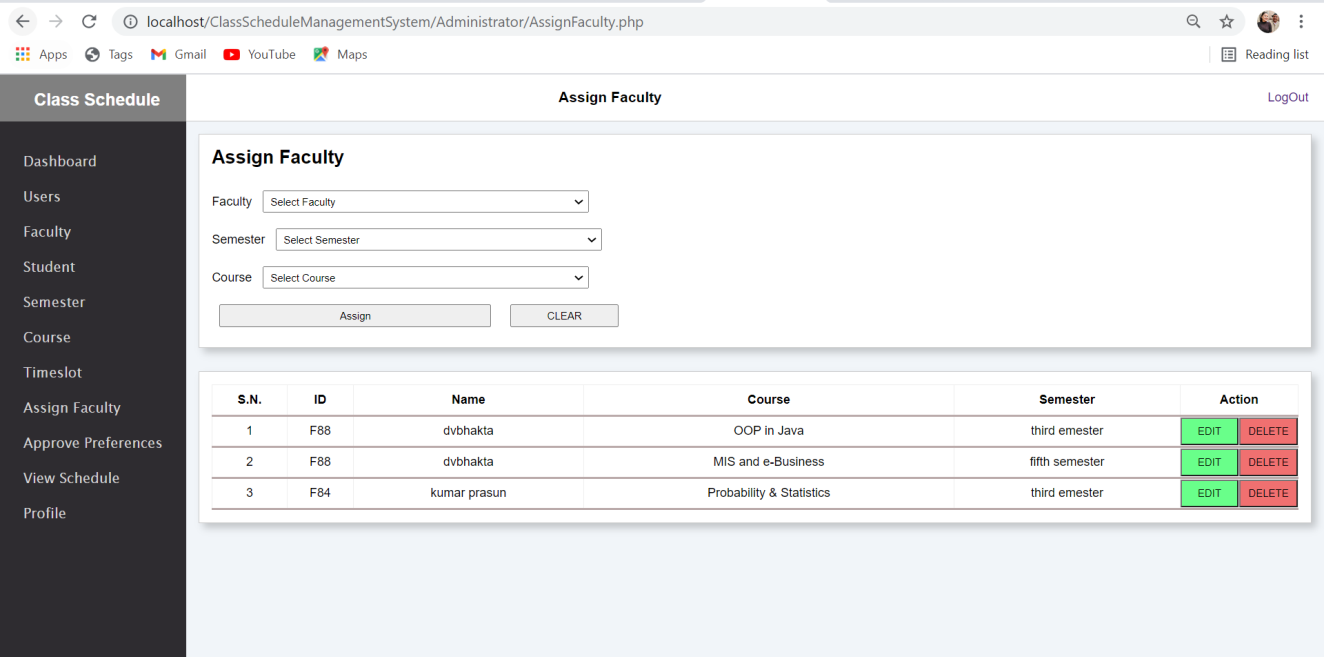
* **Admin Dashboard/Course page**

****

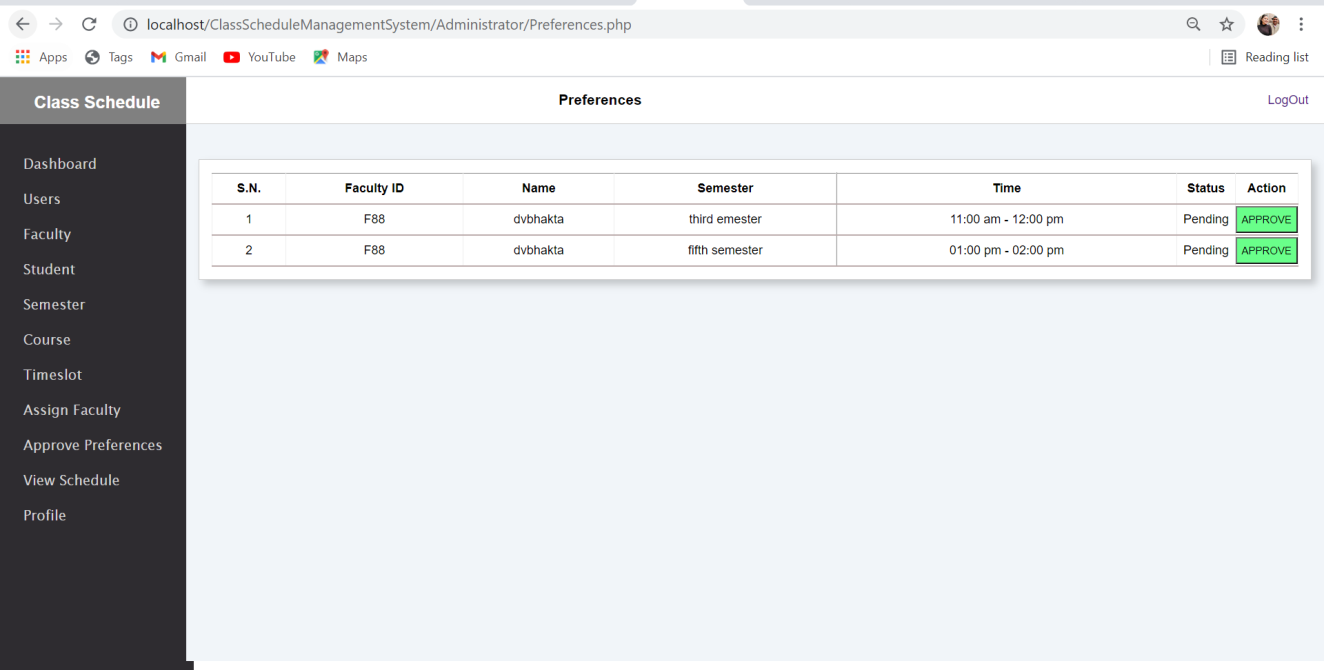
* **Admin Dashboard/Timeslot page**

****

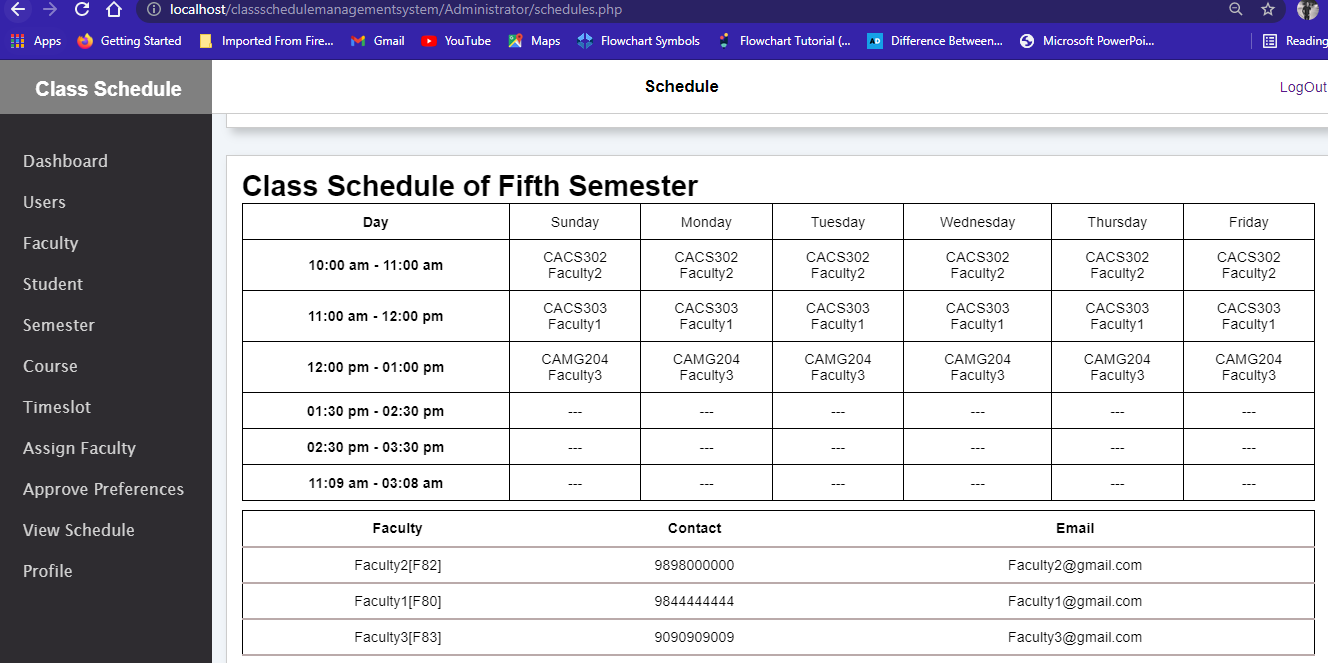
* **Admin Dashboard/Assign Faculty page**

****

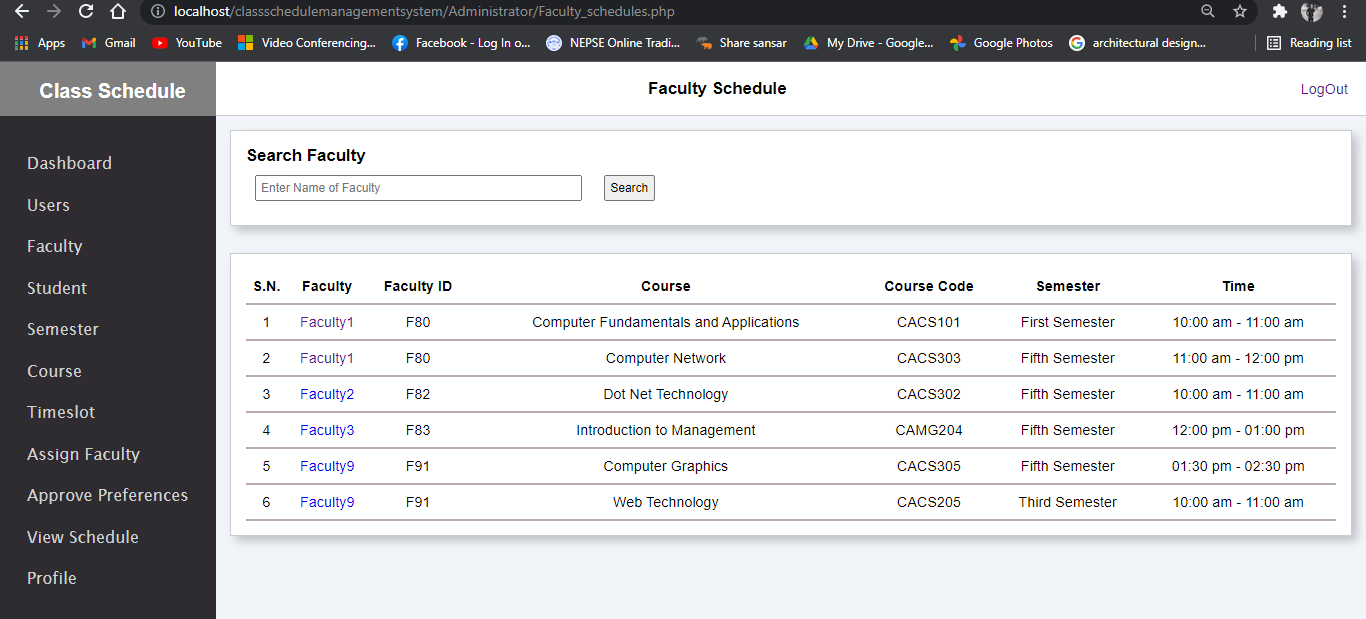
* **Admin Dashboard/Approve Preferences page**

****

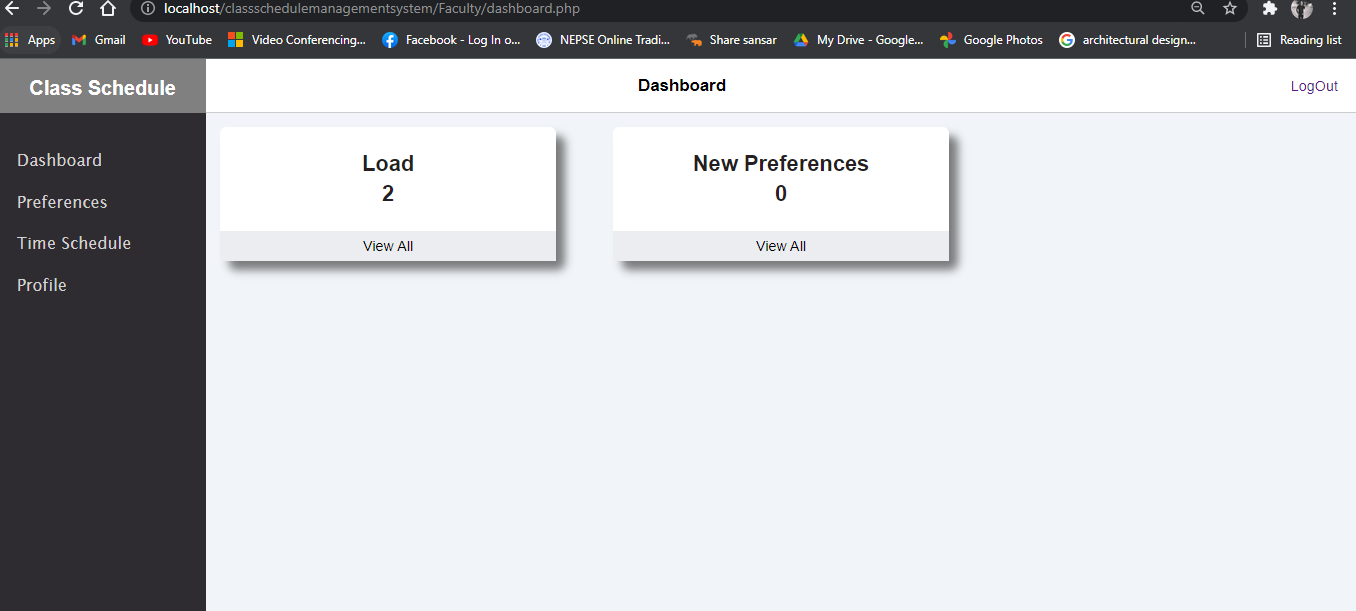
* **Admin Dashboard/View Schedule /semester page**



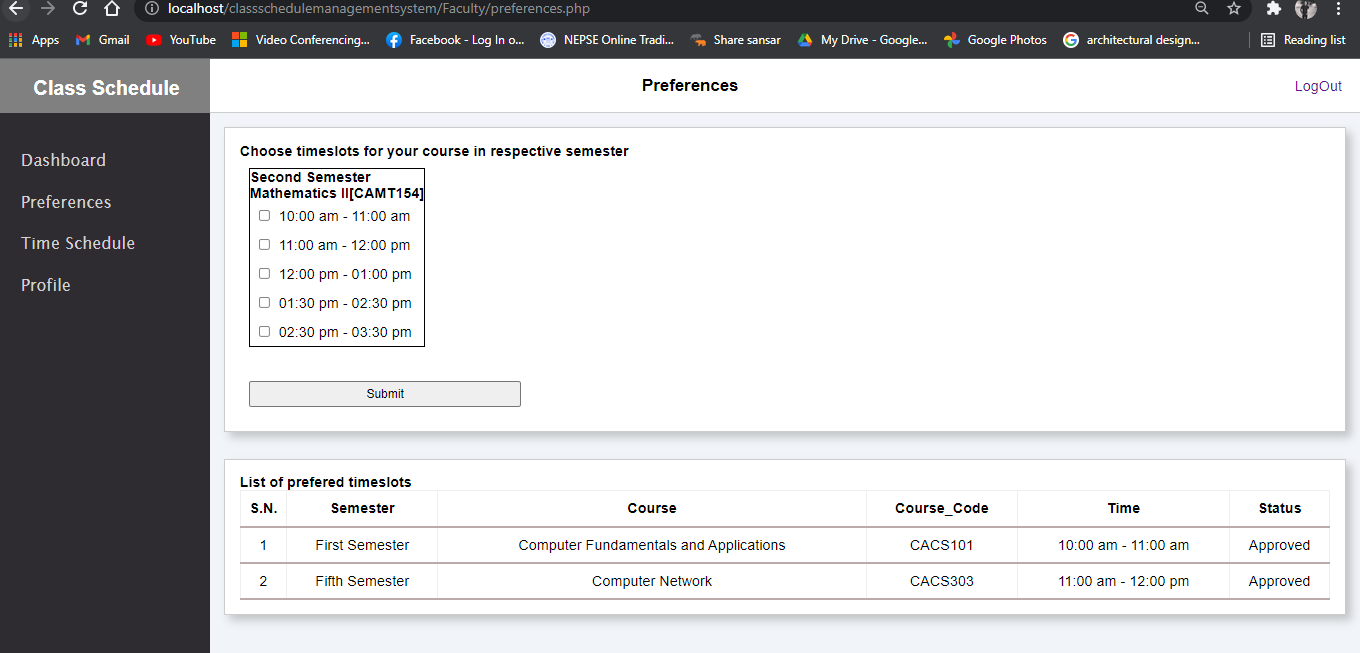
* **Admin Dashboard/View Schedule /faculty page**



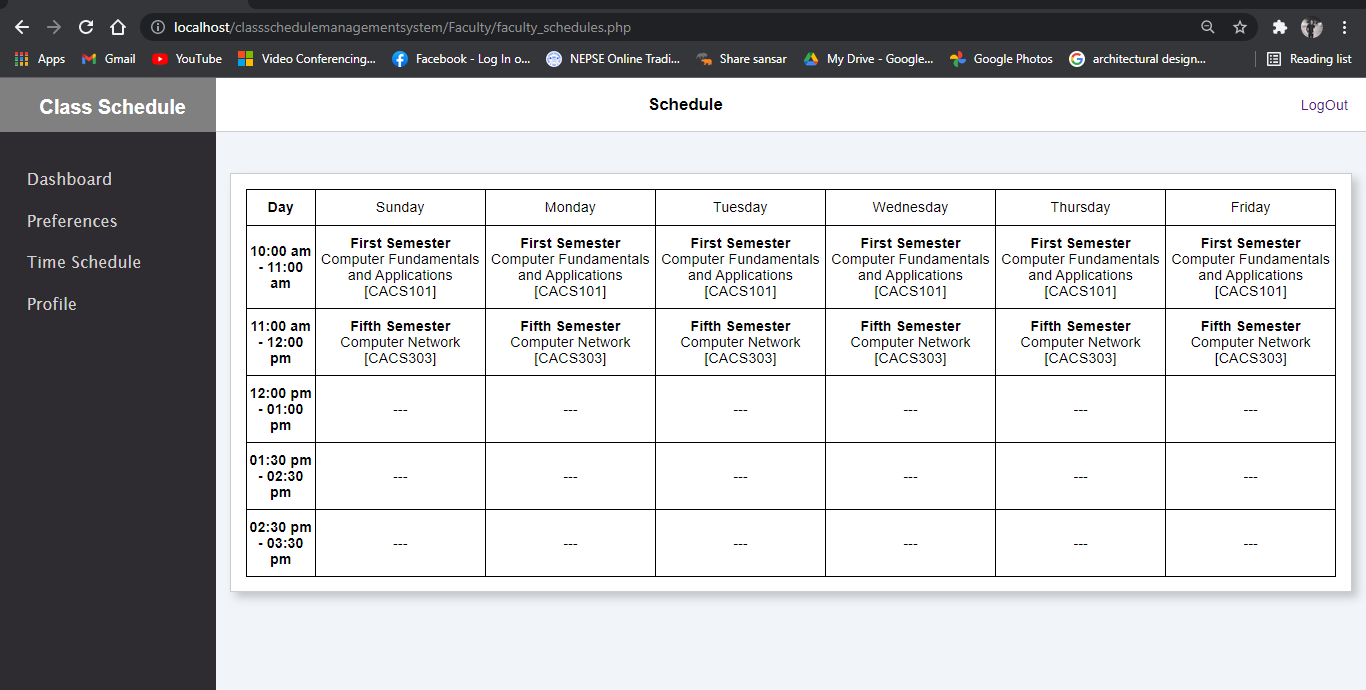
* **Faculty Dashboard page**



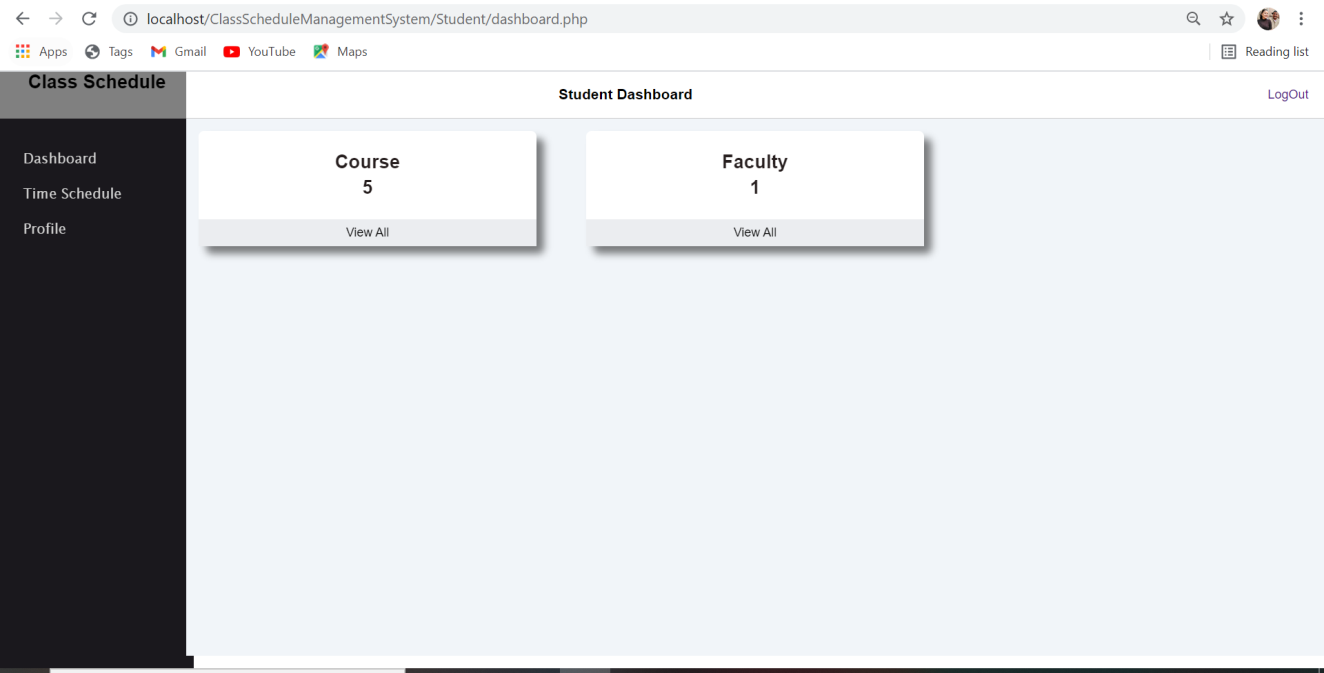
* **Faculty Dashboard/Preferences page**



* **Faculty Dashboard/Time Schedule page**



* **Student Dashboard page**

****

* **Student Dashboard/Time Schedule page**

