**STUDENT DECLARATION**

This is to certify that I have completed the summer project entitled “**Exam Seat Management System**” under the guidance of “Mr. Kushal Niraula” in partial fulfillment of the requirement for degree of **Bachelor of Information Management** at Faculty of Management, Tribhuvan University. This is my original work and I have not submitted it earlier elsewhere.

Date: August, 2022

Name: Kailash Yadav

Signature:

**CERTIFICATE FROM THE SUPERVISOR**

This is to certify that the summer project entitled “**Exam Seat Management System**” is an academic work done by “Kailash Yadav” submitted in the partial fulfillment of the requirement for the degree of **Bachelor of Information Management** at Faculty of Management, Tribhuvan University under my guidance and supervision. To the best of my knowledge, the information presented by him in the summer project report has not been submitted earlier.

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

Signature of the Supervisor

Name: Mr. Kushal Niraula

Designation:  Project Coordinator

Date: August, 2022

**ACKNOWLEDGEMENT**

I express my heartiest thanks to my supervisor Mr. Kushal Niraula for kind co-operation and encouragement which help in completion of this project entitled “**Exam Seat Management System**” for Himalaya Darshan College. I would like to acknowledge all respected teachers, friends who provided with all the knowledge, help and inspiration needed for the preparation of the summer project, and also being the guide whenever needed. I also like to thanks the University for including such course in our curriculum.

I am very glad that I got this wonderful opportunity to do this project. At last I am thankful to all the persons who supported me throughout this project directly or indirectly throughout my project.

Kailash Yadav (9327/18)

BIM 6th Semester

Himalaya Darshan College

**EXECUTIVE SUMMARY**

The Exam Seat Management System is a product which solves one of the big issues of centralized examination system by offering a series of algorithms. This will reduce a huge number of workload that have to be given by the employees before exam to prepare an exam seating arrangement plan. Also this will reduce the risk of mismanagement during exam like not having seats for all students, overlapping of multiple students in the same seat. It is very effective for large amount of students and seats. This gives good result while the number of students and seats are equal. In future we will be able to enhance the system with more features.

This system solves the problem of manual assignment of exam seat to the students. It makes the activities of the college smooth as it helps the admin as well as the system user to do their work without any delay.

**TABLE OF CONTENTS**

[STUDENT DECLARATION ii](#_Toc111994934)

[CERTIFICATE FROM THE SUPERVISOR iii](#_Toc111994935)

[ACKNOWLEDGEMENT iv](#_Toc111994936)

[EXECUTIVE SUMMARY v](#_Toc111994937)

[LIST OF FIGURES viii](#_Toc111994938)

[LIST OF TABLES ix](#_Toc111994939)

[LIST OF ABBREVIATIONS x](#_Toc111994940)

[CHAPTER I INTRODUCTION 1](#_Toc111994941)

[1.1 Background of Study 1](#_Toc111994942)

[1.2 Introduction of organization 1](#_Toc111994943)

[1.3 Current Situation of organization 1](#_Toc111994944)

[1.4 Literature Review 1](#_Toc111994945)

[1.5 Objective of study 2](#_Toc111994946)

[1.6 Methodology 2](#_Toc111994947)

[1.6.1 Data and Information 2](#_Toc111994948)

[1.6.2 Project Framework 2](#_Toc111994949)

[1.6.3 Tools used 3](#_Toc111994950)

[1.6.3.1 Programming Tool 3](#_Toc111994951)

[1.6.4 Techniques of project report analysis 4](#_Toc111994952)

[CHAPTER II TASK AND ACTIVITIES PERFROMED 5](#_Toc111994953)

[2.1 Analysis of tasks 5](#_Toc111994954)

[2.2 Analysis of possible solution 5](#_Toc111994955)

[2.2.1 Requirement Analysis 5](#_Toc111994956)

[2.2.2 Functional Requirement 5](#_Toc111994957)

[2.2.3 Non-Functional Requirement 10](#_Toc111994958)

[2.2.4 Software Requirement 11](#_Toc111994959)

[2.2.5 Entity Relationship Diagram 11](#_Toc111994960)

[2.2.6 Solution Design 12](#_Toc111994961)

[2.2.7 Activity Diagram 14](#_Toc111994962)

[2.2.8 Design Diagram 15](#_Toc111994963)

[2.2.9 Testing 15](#_Toc111994964)

[2.3 Findings 21](#_Toc111994965)

[CHAPTER III DISCUSSION AND CONCLUSION 22](#_Toc111994966)

[3.1 Discussion 22](#_Toc111994967)

[3.2 Conclusion 22](#_Toc111994968)

[REFERENCES 23](#_Toc111994969)

[APPENDICES 23](#_Toc111994970)

**LIST OF FIGURES**

Figure 1.1 Incremental Model…………….. …………...…………………………..…. 2

Figure 2.1 Use Case Diagram …………………………………………………….…... 6

Figure 2.2 ER Diagram ……….………………………………………………..……. 11

Figure 2.3 Schema Diagram ………………………………………………………… 13

Figure 2.4 Activity Diagram ……………………………………………..………….. 14

Figure 2.5 Sequence Diagram………………………………………………………… 15

**LIST OF TABLES**

Table 2.1 Login into the system ………………………...……………………………. 6

Table 2.2Create Account………………………………..………………….……….... 7

Table 2.3 Manage User ………………………...………..…………………………… 7

Table 2.4 Manage Room ………………………………..……………………………. 7

Table 2.5 Manage Seat ………………………………..…………………………….... 8

Table 2.6 Manage Teacher ………………………………..………………………….. 8

Table 2.7 Manage Invigilator………………………………..………………………... 8

Table 2.8 Manage Student ………………………………..…………………………... 9

Table 2.9 Manage Faculty ………………………………..…………………………... 9

Table 2.10 Manage Semester ………………………………..……………….………. 9

Table 2.11 Change Password………………………………..……………………….... 10

Table 2.12 Logout………………………………..…………………………….…….... 10

Table 2.13 Test Case (Login) ………………………………..……………………...... 16

Table 2.14 Test Case (User register) …………………………………..…………...... 17

Table 2.15 Test Case (Add new student) ………………………………..……..…...... 18

Table 2.16 Test Case (Navbar) ………………………………..…………………........ 19

**LIST OF ABBREVIATIONS**

BIM: Bachelor Information Management

ER: Entity Relationship

ESMS: Exam Seat Management System

IDE: Integrated Development Environment

PHP: Hypertext pre-processor

SDLC: Systems Development Life Cycle

TC: Test Case

TU : Tribhuvan University

UC: Use Case

**CHAPTER** **I INTRODUCTION**

**1.1 Background of Study**

Result Management System is a web-based application. Students will be able to view their result online and access result in pdf format. The system will allow the teachers to grade the students even from home, then automatically perform the grades calculation.

**1.2 Introduction of organization**

Himalaya Darshan College at Biratnagar-14, established in 2070 B.S. provides innovative opportunities in highly academic environment. The College has been established with an objective to promote value based quality education at the graduate level. The college fosters personal and professional growth of the students through its experienced and distinguished faculties, experts and professional from the national level.

**Programs: Bsc.CSIT, BIM, BCA, BHM and BBS.**

**1.3 Current Situation of organization**

* There was no such way to maintain exam seat plan.
* The seat plan were being maintained manually.

**1.4 Literature Review**

The examination management is a huge task of manually allocating the seats for conducting the examination. A software application is necessary to decrease our manual work time. The system must be user-friendly for the fast retrieval and storing of data. It has to be maintained efficiently with the graphical user interface and effective database design.

All the data required for generating the exam plan has to be entered into the computer and reports can be generated automatically so that work will be very easy because there is no need to keep data more on papers. User can generate the report and printed if necessary at the time of conducting the exams.

Examination management system is developed for the colleges to simplify the work of allocating exam halls and generating seat for the students during exams. It also facilitates to access the examination details of a particular student in a particular Faculty.

Reports like seating arrangements can be easily generated so that user can generate the report as per the requirement.

**1.5 Objective of study**

Following are the objectives of the project:

* To automate the examination hall allotment and seating arrangement.
* To generate the master seat-plan.
* To manage the details of Faculty, Student, Room, Invigilator and Examination.

**1.6 Methodology**

**1.6.1 Data and Information**

For the collection of data, here primary as well as secondary method of data collection method are used. For primary method, the queries were asked to the exam coordinator by visiting the college. For the secondary method use of internet to know about the organization and the system.

**1.6.2 Project Framework**

Here for the proposed system, Incremental Development model is used. This model is also known as iterative model or an evolutionary model. This type of process model is based on initial development of the software product, providing it for a review for users, and making the necessary changes to finally make the product stable. It is the approach where the software is developed mostly for E-commerce approach, personal and business-based application.

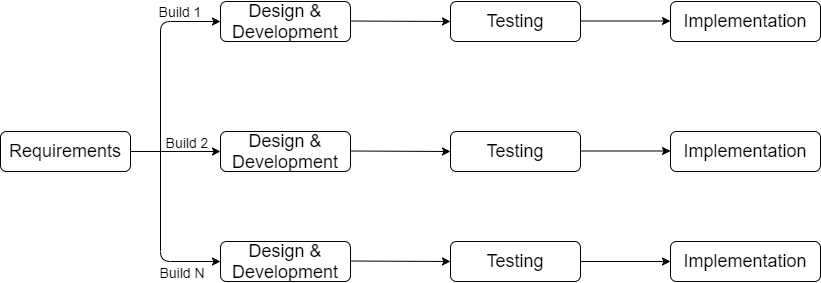


Figure 1.1 Incremental Model

The various phases of incremental model are as follows:

1. **Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
2. **Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
3. **Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.
4. **Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

The reason behind considering this methodology are as follows:

* Flexible and less expensive to change requirements and scope.
* Lower cost of change.

**1.6.3 Tools used**

**1.6.3.1 Programming Tool**

**Visual Studio Code**

Visual Studio Code is an innovative Integrated Development Environment (IDE) engineered by Microsoft for web developers. The IDE provides smart code completion, syntax highlighting, extended code formatting configuration, on-the-fly error checking, code folding, supports language mixtures and more. Automated refactoring treats your code with care, helping to make global project settings easy and safe.

**XAAMP**

XAAMP is a free open-source cross platform web server which is used by developer to create a local web server for testing and developing purposes. Here, it is used to connect to local server Apache and connect to MySQL.

**Draw.io**

Draw.io is a free diagramming application that allows users to create and share diagrams within a web browser. Here it is used to make use case diagram, ER-model, sequence diagram, schema diagram, activity diagram.

**MS-Word**

MS Word is a word processor developed by Microsoft. It is one of the most widely used programs of Microsoft Office suite. Used to make professional-quality documents, letters, reports, etc. It has advanced features which allow you to format and edit your files and documents in the best possible way.

**GitHub**

GitHub is a web-based interface that uses Git, the open-source version control software that lets multiple people make separate changes to web pages at the same time.  It offers the distributed version control and source code management functionality of Git, plus its own features.

**1.6.4 Techniques of project report analysis**

**1.6.4.1 Problem analysis**

The main problem of the organization is that they didn’t have proper computerized system to manage exam seat digitally.

**1.6.4.2 Feasibility analysis**

The analysis of the project has leads to conclusion that the project is feasible with time and cost. The tool used for the development is almost Open Source which involve less cost and maintenance. Our alternative system need only a computer which is already available to everyone now a days, So, all the alternative system are technically feasible.

**1.6.4.3 Economic feasibility**

This analysis involves the Cost-benefit analysis. It helps to determine the positive or negative effect created by the software to the organization; where the project has created positive effect.

**1.6.4.4 Operational feasibility**

Operational feasibility is the analysis where the system is analyzed on how well the purpose system solves the problem and work in real environment and how it satisfied the requirement analysis phase of the system development.

**CHAPTER II** **TASK AND ACTIVITIES PERFROMED**

**2.1 Analysis of tasks**

Before analyzing the task, the organization was visited for the requirement gathering. After the requirement is gather it was major task to break down the requirement in the proper procedure and building the working framework. The major objective of the task analysis were to find and define precisely the core problem that need to be address and thereby determine the scope of new system. During the time in the organization, various aspect of the organization were analyzed and find out the problem that the proposed system can solve during the summer project. 2.2 Analysis of possible solution.

**2.2 Analysis of possible solution**

**2.2.1 Requirement Analysis**

Requirement analysis contains requirements related to the users and the system are collected and analyzed so the information can be filtered and defined properly. At the end of this activity the requirements are ready to be specified. Requirement analysis consists of following steps/process:

* Requirement Discovery
* Requirement Classification & Organization
* Requirement Prioritization & Negotiations
* Requirement Specifications

**2.2.2 Functional Requirement**

Functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs.

* Admin can register a new account of User.
* Login function where admin can log in to the system.
* Admin can add/view/edit/delete records.
* Admin can activate or deactivate system user.
* Admin or system user can generate exam seat plan

It can be briefly described by the help of Use-Case Diagram which is as follows:

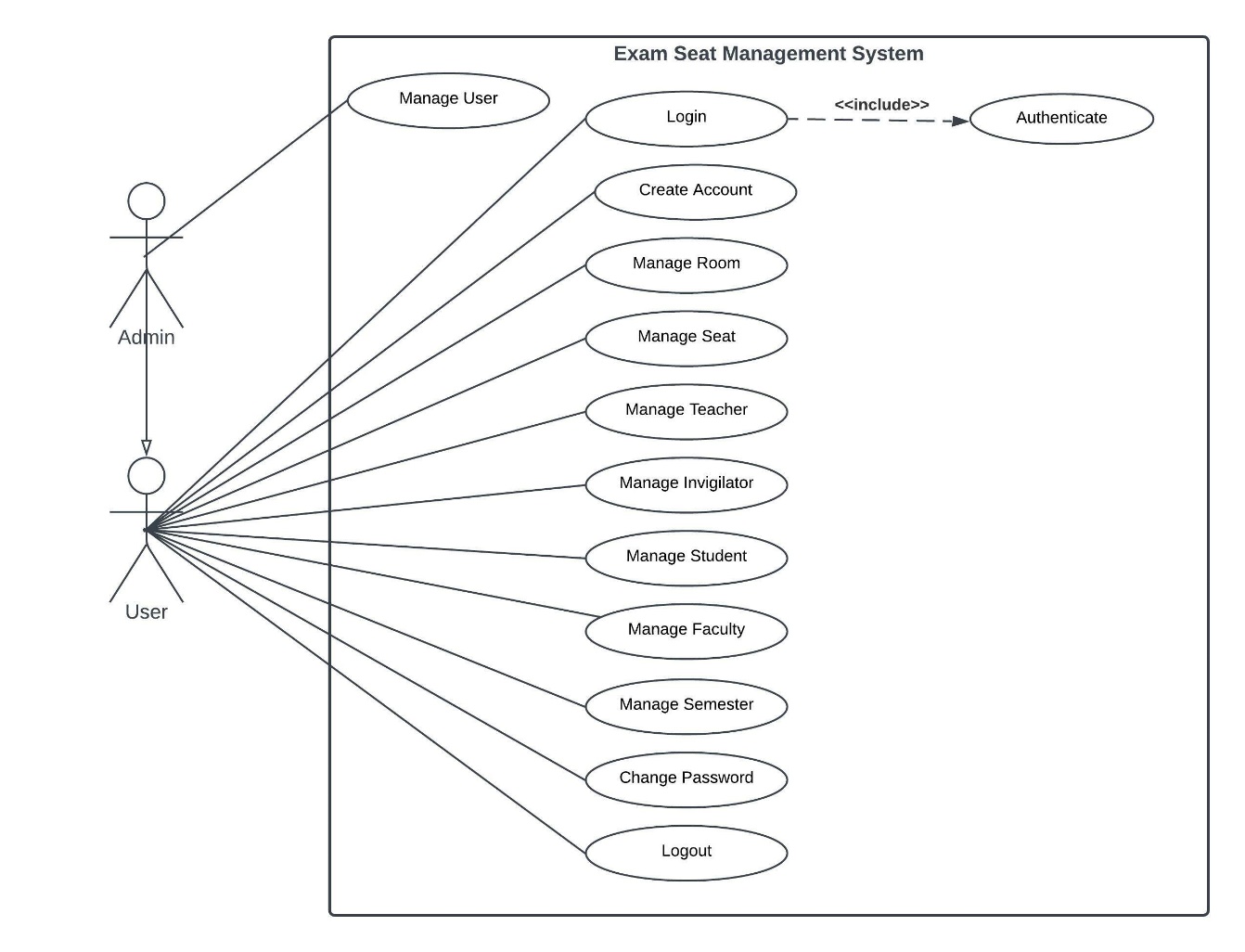


Figure 2.1: Use Case Diagram

Table 2.1 Login into the system

|  |  |
| --- | --- |
| Use Case ID | UC-01 |
| Use Case Name | Login into the system |
| Actors | Admin, System user |
| Description | Allow administrator to login into the system through allocated username and password. |
| Pre-condition | Administrator have to know the username and password assigned to them. |
| Post Condition | Admin can perform the task in the system according. |
| Success Scenario | Login success and display system dashboard. |
| Failure Scenario | Login failed message should be displayed. |

Table 2.2:Create Account

|  |  |
| --- | --- |
| Use Case ID | UC-02 |
| Use Case Name | Create new user into the system |
| Actors | Admin, System user |
| Description | Allow administrator to create new account into the system |
| Pre-condition | Administrator have to know the username and password assigned to them. |
| Post Condition | Admin can perform the task in the system according. |
| Success Scenario | New user creation is success. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.3: Manage User

|  |  |
| --- | --- |
| Use-case Identifier | UC-03: Manage User |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can add, edit, delete, activate or deactivate the Users. |
| Pre-condition | The admin should first login into the system. |
| Post-condition | The database must be updated after action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.4: Manage Room

|  |  |
| --- | --- |
| Use-case Identifier | UC-04: Manage Room |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can add, edit and delete any room in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.5: Manage Seat

|  |  |
| --- | --- |
| Use-case Identifier | UC-05: Manage Seat |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can generate and print seat plan. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be connected to perform this action. |
| Success Scenario | The seat plan must be displayed in printing format. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.6: Manage Teacher

|  |  |
| --- | --- |
| Use-case Identifier | UC-06: Manage Teacher |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can add, edit and delete any teacher in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after add action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.7: Manage Invigilator

|  |  |
| --- | --- |
| Use-case Identifier | UC-07: Manage Invigilator |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can add, edit, delete, activate and deactivate any invigilator in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.8: Manage Student

|  |  |
| --- | --- |
| Use-case Identifier | UC-08: Manage Students |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can add, edit and delete any students in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.9: Manage Faculty

|  |  |
| --- | --- |
| Use-case Identifier | UC-09: Manage Faculty |
| Primary Actor | Admin, system user |
| Secondary Actor | None |
| Description | The admin can add, edit and delete any faculty in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.10: Manage Semester

|  |  |
| --- | --- |
| Use-case Identifier | UC-09: Manage Semester |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can upgrade semester of any students in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after action is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.11: Change Password

|  |  |
| --- | --- |
| Use-case Identifier | UC-11: Change Password |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can change their password in the database. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The database must be updated after actions is performed. |
| Success Scenario | The success message is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

Table 2.12: Logout

|  |  |
| --- | --- |
| Use-case Identifier | UC-12: Logout |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | The admin can logout from the system. |
| Pre-condition | The admin should first login into the system. . |
| Post-condition | The session must destroy after action is performed. |
| Success Scenario | The index page is displayed. |
| Failure Scenario | The database is not connected or misconfigured. |

**2.2.3 Non-Functional Requirement**

Various non-functional requirement of the system are:

* The system shall be easier to use with better GUI.
* The system shall be secured enough to operate.
* The system should be affordable.
* The system shall be fast enough to process the data.

**2.2.4 Software Requirement**

The system requirements specify about the services or features of the system that are required so that the services are delivered properly. The system requirements of the project are:

* Software requirement:
  1. Operating System: Windows XP or higher
  2. Database tool: MYSQL
  3. Browser
* Hardware Requirement:

1. Processor Intel or Pentium 4.22 GHz
2. RAM 512MB
3. Hard Disk 80MB available in hard disk.

**2.2.5 Entity Relationship Diagram**

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, object, places, concepts, or events within that system.

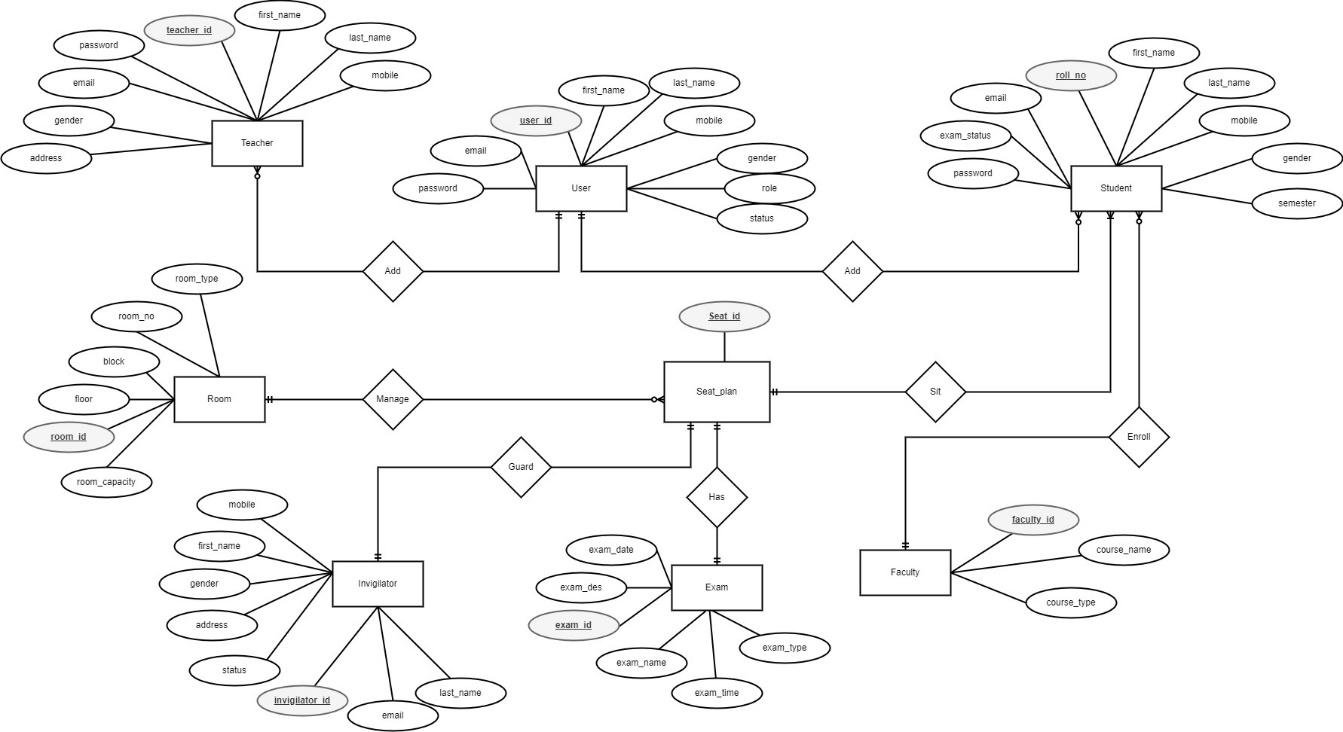


Figure 2.2: ER-Diagram

Here, there are eight entities and their attributes: User {**user\_id**, first\_name, last\_name, mobile, gender, email, password, role, status}, Teacher {**teacher\_id**, first\_name, last\_name, mobile, gender, email, password, address},Student {**roll\_no**,first\_name, last\_name, mobile, gender, semester, email,password, exam\_status,}, Invigilator {**invigilator\_id**, first\_name, last\_name, mobile, gender, email, address, status }, Faculty {**faculty\_id**, course\_name, course\_type}, Room {**room\_id,** room\_no, room\_type, block, floor, room\_capacity},Exam {**exam\_id**, exam\_name, exam\_type, exam\_date, exam\_time, exam\_des },and Seat\_plan {**seat\_id**}

There is one to many relationships between User and Student where User can adds as many as Students. There is one to many relationships between User and Teacher where User can adds as many as Teachers.T here is one to many relationships between Room and Seat\_plan where one room manage many seat plan. There in one to one relationship between Invigilator and Seat\_plan where one invigilator can only guard in one room of a seat plan. There is many to one relationship between Student and Faculty where many students can enrolls in one faculty. There is one to many relationships between Exam and Seat\_plan where one room can manage many Seat plan of different exam. There is many to one relationships between Student and Seat\_ plan where many students can be able sit in one seat plan of exam.

**2.2.6 Solution Design**

Software design is a process to conceptualize the software requirements into software implementation. It is initial phase of physical deployment of any solution, which shows the process of system functioning.

**Schema Diagram**

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. It provides conceptual tools to design the database schema of the relational database.

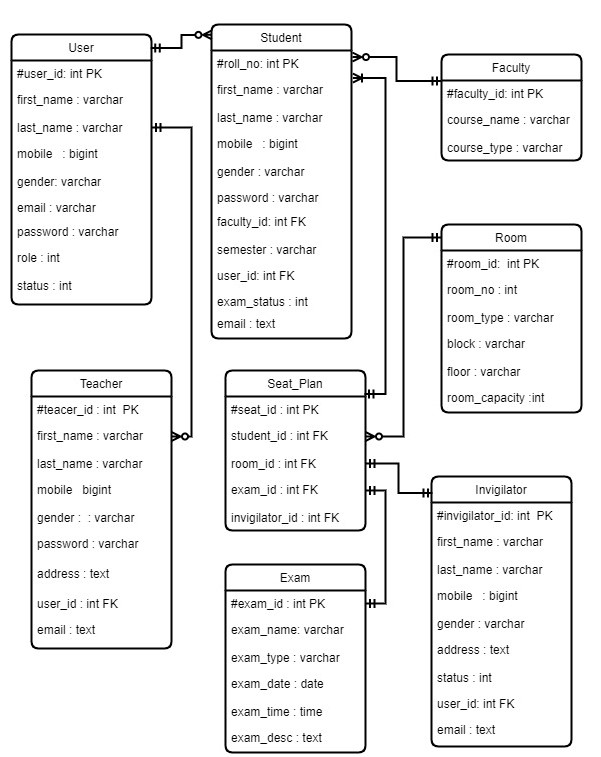


Figure 2.3: Schema-Diagram

**2.2.7 Activity Diagram**

 An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. It is an important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. Following diagram shows some of the activity of the system:

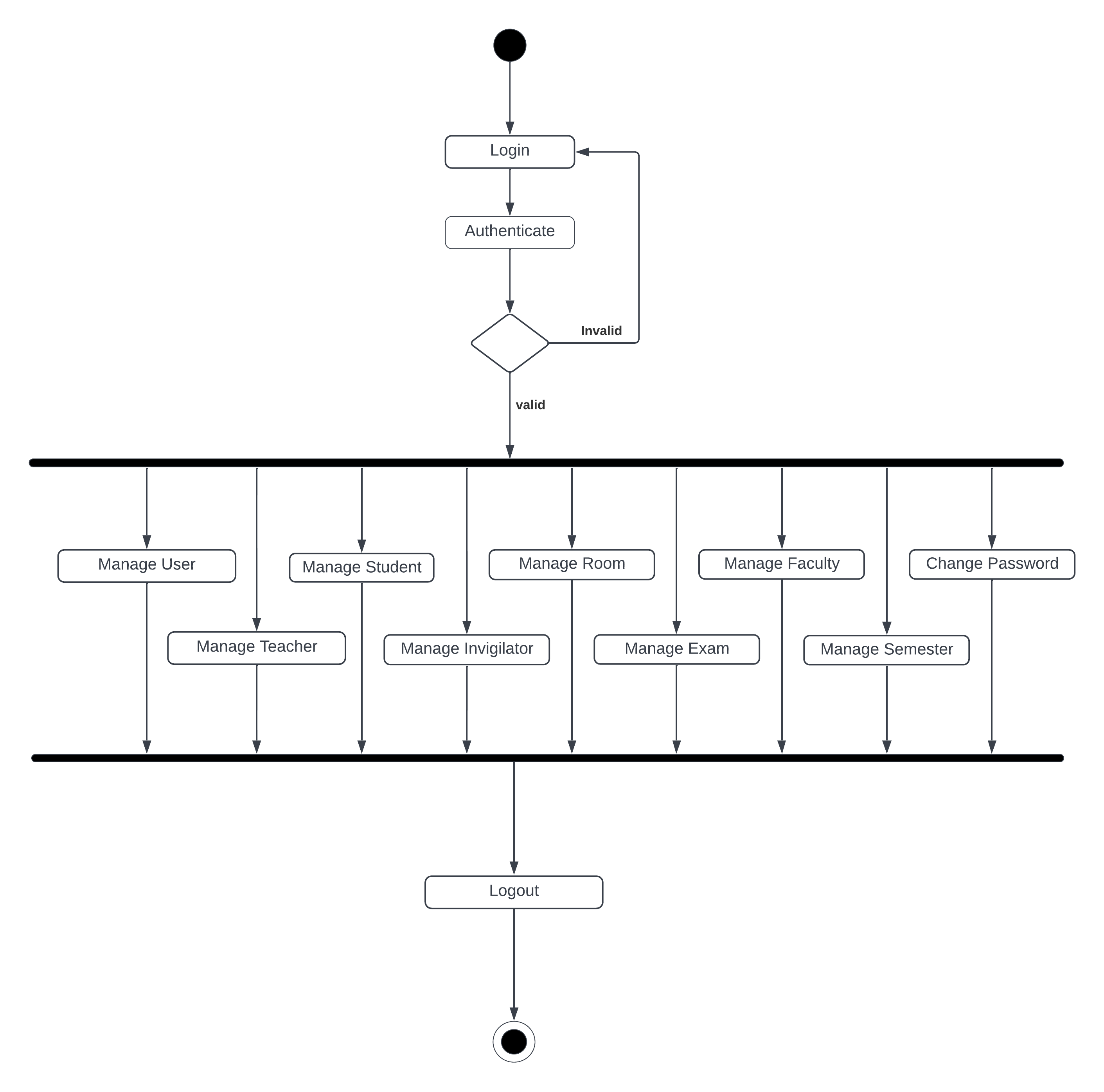


Figure 2.4: Activity Diagram

**2.2.8 Design Diagram**

**Sequence Diagram**

A sequence diagram shows the sequence of the interactions that take place during a particular use case or use case instance. It shows the interactions between actors and the system and between system components. Sequence diagram is shown below:

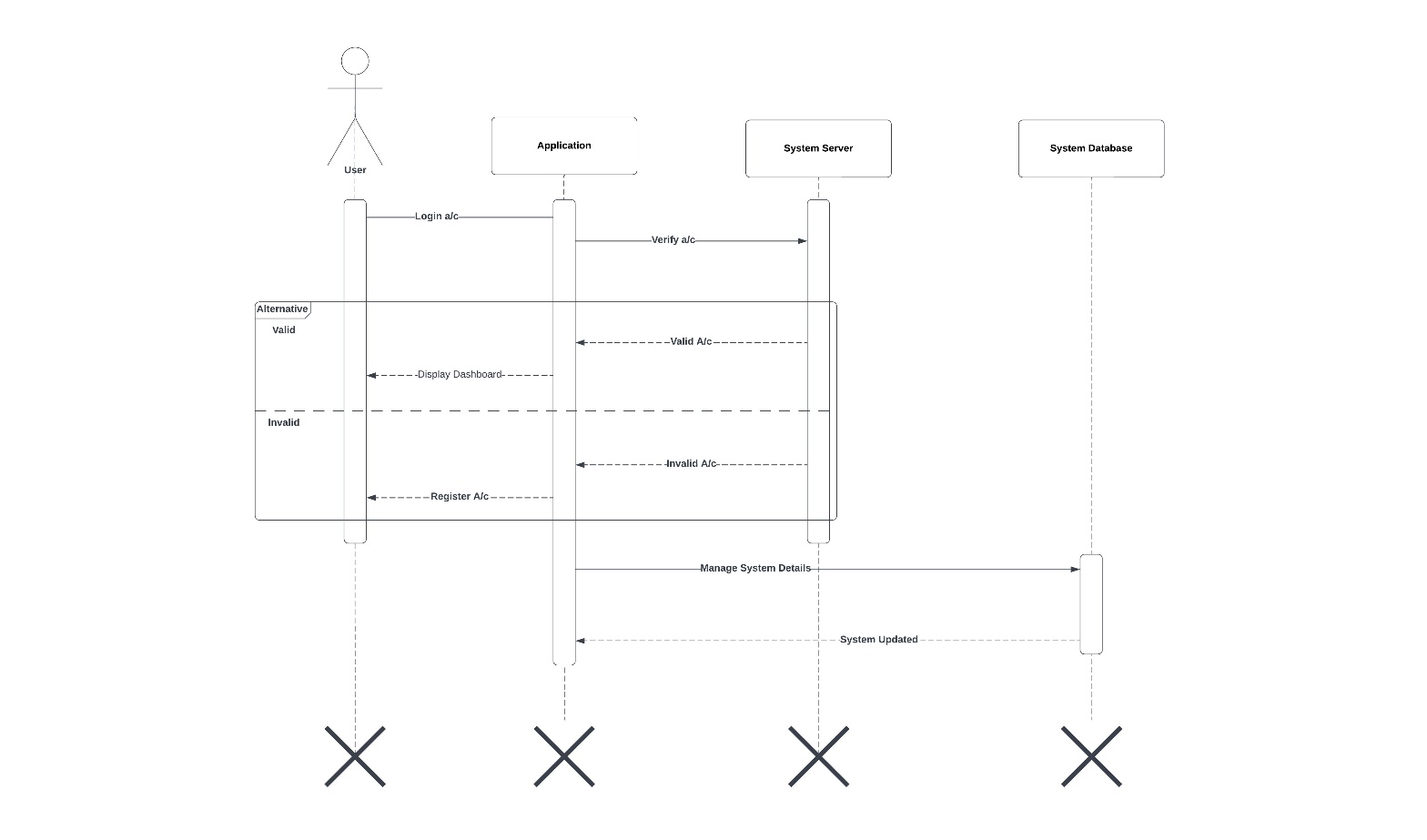


Figure 2.5: Sequence Diagram

Here, a user login enters the username, password and role type. The system then verify whether the entered details are valid or not. If the details provided by user is valid with the details of the database the user can view the dashboard and manage the system details and system get updated in the database according to user’s activities. Otherwise, user is not registered. So, user need to register s/he account and attempt login again.

**2.2.9 Testing**

Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Table 2.13 Test Case (login)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name: Exam Seat Management System (ESMS) | | | | | | | |
| Test Case | | | | | | | |
| Test Case ID: TC\_01 | | | | | Test Designed by: Kailash Yadav | | |
| Test Priority (Low/Medium/High): Medium | | | | | Test Designed date: 2022-07-05 | | |
| Module Name: Login | | | | | Test Executed by: Kailash Yadav | | |
| Test Title: Login | | | | | Test Execution date: 2022-07-05 | | |
| Description: Authentication of the user is checked. | | | | | | | |
| Pre-conditions: User has login username and password | | | | | | | |
| Dependencies: | | | | | | | |
| Steps | Test Steps | Test Data | Expected Result | Actual Result | | Status | comment |
| 1. | Valid username and password | [Username=admin@gmail.com](mailto:Username=admin@gmail.com)  Password=\*\*\*\*\* | Displayed Dashboard | As expected i.e. Displayed Dashboard | | Pass | Dashboard is displayed |
| 2. | Invalid username and password | [Username=x4yz@gmail.com](mailto:Username=x4yz@gmail.com)  Password=\*\*\*\*\* | Invalid credentials | As expected i.e. Invalid credentials | | Pass | Page directed to login |
| Post conditions: User successfully able to login in the system and do available activities | | | | | | | |

Table 2.14 Test Case (User Registration)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name: Exam Seat Management System (ESMS) | | | | | | | |
| Test Case | | | | | | | |
| Test Case ID: TC\_02 | | | | | Test Designed By: Kailash Yadav | | |
| Test Priority: Medium | | | | | Test Designed Date: 2022-07-06 | | |
| Module Name: Create User | | | | | Test Executed By: Kailash Yadav | | |
| Test Title: Register User | | | | | Test Execution Date: 2022-07-06 | | |
| Description: Create a new user in the system. | | | | | | | |
| Pre conditions: Database should be connected. | | | | | | | |
| Dependencies: | | | | | | | |
| Steps | Test Steps | Test Data | Expected Result | Actual Result | | Status (Pass/Fail) | comment |
| 1. | Click on Register Here Link | Link is clicked by mouse or keyboard | Register user form is displayed | As expected. | | Pass | User registrationform is displayed |
| 2. | Testing with valid data | First Name: Hari  Last Name: Subedi  Mobile: 9816859685  Email:[hari@gmail.com](mailto:hari@gmail.com)  Username:[hari@gmail.com](mailto:hari@gmail.com)  Password: \*\*\*\*\*  Confirm Password: \*\*\*\*\*  Select Gender: Male  User Role: System User | User Creation successfuldisplayed and data are saved in database. | As expected. | | Pass | New User is created |
| 3. | Testing with invalid data | First Name: H@455ri  Last Name: S05$#$i  Mobile: 981685  Email: [hari@gmail.com](mailto:hari@gmail.com)  Username: [hari@gmail.com](mailto:hari@gmail.com)  Password: \*\*\*\*  Confirm Password: \*\*\*\*\*  Select Gender: Male  User Role: System User | Invalid data are displayed for respective fields. | As expected. | | Pass | New user is not created. |
| Post conditions: New user is successfully created in the system and the data is saved in the database. | | | | | | | |

Table 2.15 Test Case (Add new Student)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Name: Exam Seat Management System (ESMS) | | | | | | |
| Test Case | | | | | | |
| Test Case ID: TC\_03 | | | | Test Designed By: Kailash Yadav | | |
| Test Priority: Medium | | | | Test Designed Date: 2022-07-05 | | |
| Module Name: Add New Student | | | | Test Executed By: Kailash Yadav | | |
| Test Title: Add student | | | | Test Execution Date: 2022-07-05 | | |
| Description: Create a new student in the system. | | | | | | |
| Pre conditions: User login with valid username and password. | | | | | | |
| Dependencies: | | | | | | |
| Steps | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | comment |
| 1. | Open add student form | Click on the add student button | Add student Form is displayed | As expected. | Pass | Add student form is displayed |
| 2. | Open add student form | All invalid data or any one of invalid data are filled | Creation failed displayed | As expected. | Pass | Creation is failed |
| 3. | Fill the form | Roll No:75408  First Name:Kailash  Last Name:Yadav  Mobile:9810101010  Email:kailash@gmail.com  Password:\*\*\*\*\*\*  Gender:Male  Faculty:BIM  Semester:Sixth | Student should be created and stored in database. | As expected. | Pass | Student is created successfully |
| Post conditions: User successfully create new Student in the system and the data is saved in the database. | | | | | | |

Table 2.16 Test Case (Navbar)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Name: Exam Seat Management System (ESMS) | | | | | | |
| Test Case | | | | | | |
| Test Case ID: TC\_04 | | | | Test Designed By: Kailash Yadav | | |
| Test Priority: Medium | | | | Test Designed Date: 2022-07-06 | | |
| Module Name: Navbar | | | | Test Executed By: Kailash Yadav | | |
| Test Title: Navbar | | | | Test Execution Date: 2022-07-06 | | |
| Description: Navbar is toggled and links are displayed | | | | | | |
| Pre conditions: User login with valid username and password. | | | | | | |
| Dependencies: | | | | | | |
| Steps | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | comment |
| 1. | Tesing Navbar | Click on the navbar icon | All list of links are displayed. | As expected. | Pass | List of the page links are displayed |
| 2. | Testing dashboard link | Click on the dashboard link | Dashboard is displayed again. | As expected. | Pass | Dashboard is displayed again. |
| 3. | Testing student management link | Click on the student management link | Displayed the student page. | As expected. | Pass | Displayed the student page. |
| 4. | Testing teacher management link | Click on the teacher management link | Displayed the teacher page. | As expected. | Pass | Displayed the teacher page. |
| 5. | Testing invigilator management link | Click on the invigilator management link | Displayed the invigilator page. | As expected. | Pass | Displayed the invigilator page. |
| 6. | Testing room management link | Click on the room management link | Displayed the room page. | As expected. | Pass | Displayed the room page. |
| 7. | Testing exam management link | Click on the exam management link | Displayed the exam page. | As expected. | Pass | Displayed the exam page. |
| 8. | Testing faculty management link | Click on the faculty management link | Displayed the faculty page. | As expected. | Pass | Displayed the faculty page. |
| 9. | Testing manage semester link | Click on the manage semester link | Displayed the semester page. | As expected. | Pass | Displayed the semester page. |
| 10. | Testing change password | Click on the change password | Displayed the change password page. | As expected. | Pass | Displayed the change password page. |
| 11. | Testing logout link | Click on the logout link | Displayed the login page. | As expected. | Pass | Displayed the login page. |
| Post conditions: User successfully able to use the navbar and go to the page that user needed. | | | | | | |

**2.3 Findings**

Now-a-days, all schools and colleges are finding complex in the matter of planning exam seat plans. In the same way, this system has been designed to meet requirement of Himalaya Darshan College by using this web based application which helps the college to bring efficiency by exam seat plan digitally. The college can provide exam seat plan details to their students without limitation of college time and staff requirement. It can helps to management by providing an easy way of planning their exam seat plan.

**CHAPTER III** **DISCUSSION AND CONCLUSION**

**3.1 Discussion**

Before implementation of the system, while conducting the project and after conduction of the project, several facts and figures were discussed to ensure smooth running of applications at the college screen. Major concerns like what problem to address, which programming languages to use, which organization to target, the feasibility of the project, the time limitation and the availability of the required resource were discussed. The discussions provided us a guide to develop the application.

**3.2 Conclusion**

Exam Seat Management System (ESMS) is web-based application in which exam seat plans are stored and available for printing. With all the learning and knowledge, the system-built project has been successfully completed. This system has fulfilled the necessary requirements as required by the Himalaya Darshan College.

This system is cost effective so it can be used by most college staff from anywhere at any time. It is ensured that all the programs are working properly in “Exam Seat Management System.” This system has been developed in PHP to improve more user interactivity. Time-to-time information updates is needed in this application. This system has been implemented and tested.

**REFERENCES**

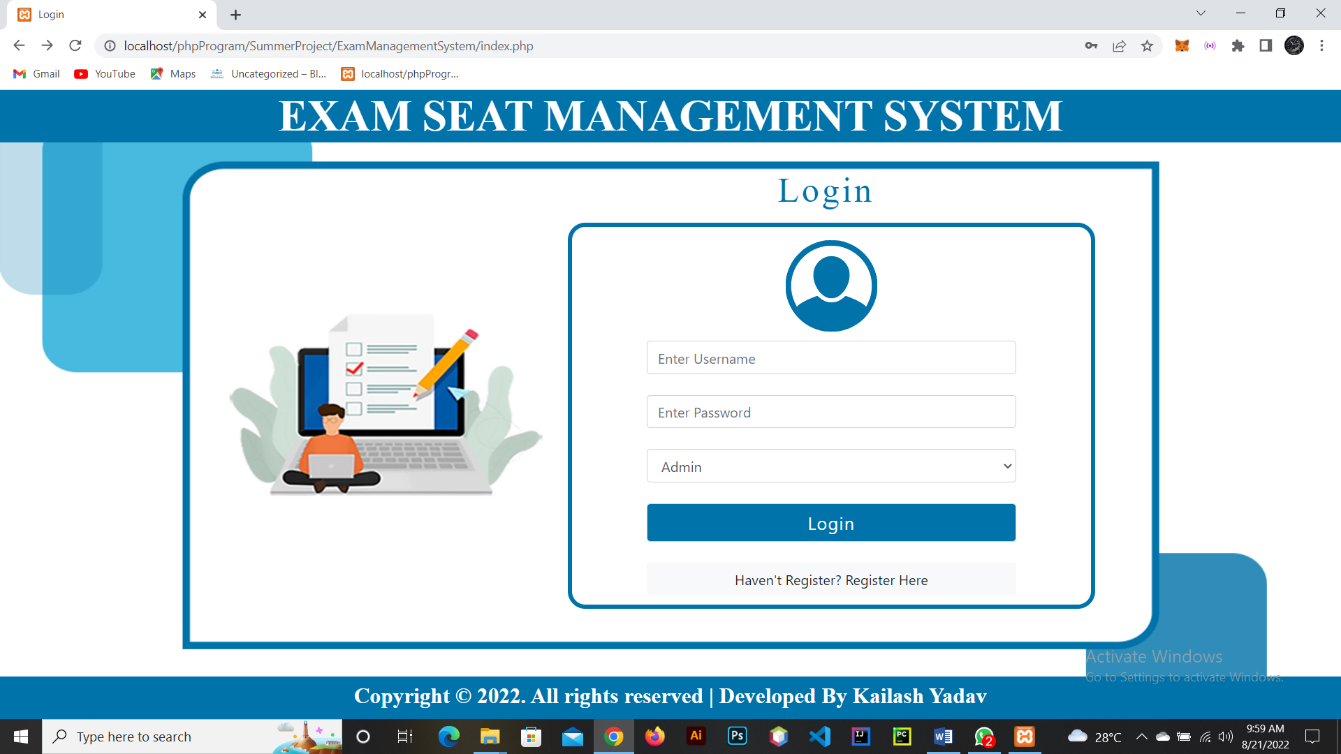
Gokila, R. and Dass, A.R. (2018) *Examination Hall and Seating Arrangement Application using PHP*. International Journal of Engineering Science and Computing. Available at: https://ijesc.org/upload/81007e44d9a4b5cf6db7d5748e1019fa.Examination%20Hall%20and%20Seating%20Arrangement%20Application%20using%20PHP.pdf (Accessed: July 20, 2022).

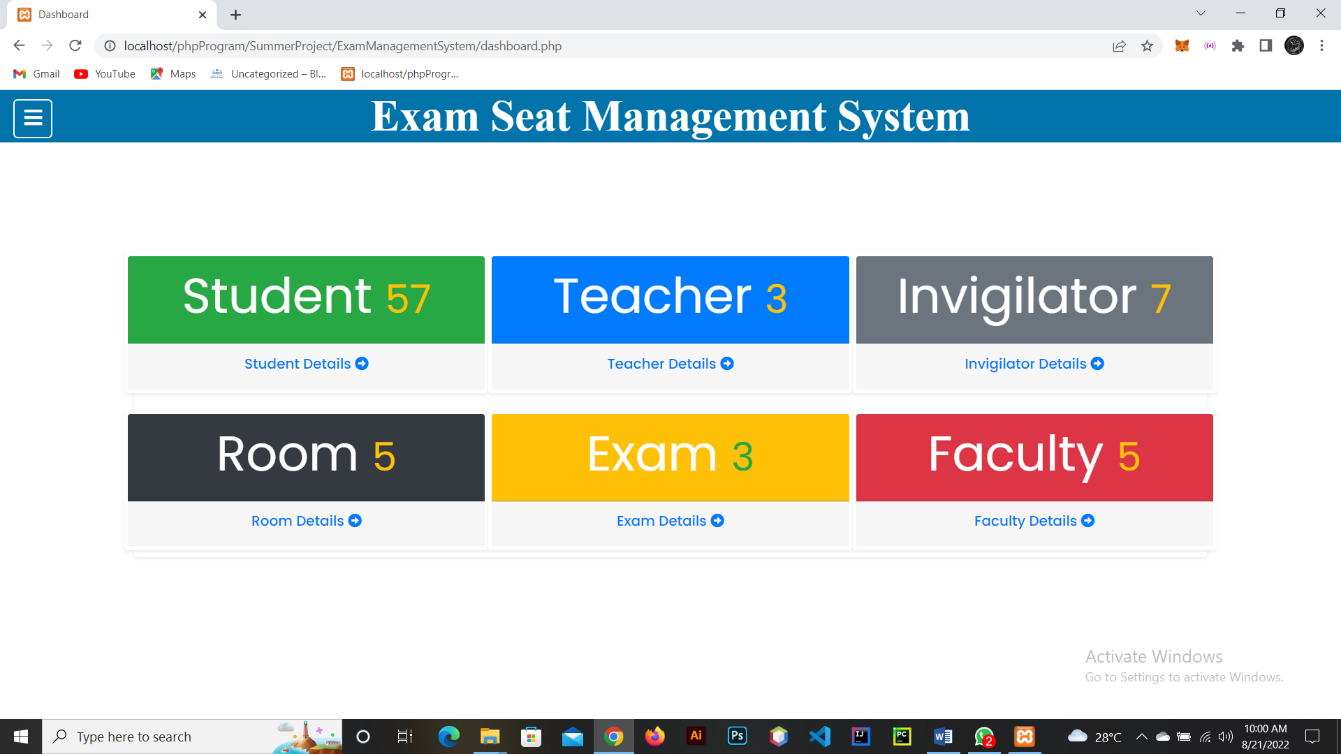
Freeprojectz (2016) *Exam Seating Arrangement System*, *Freeprojectz*. Freeproject. Available at: https://www.freeprojectz.com/paid-projects/exam-seating-arrangement-system (Accessed: July 20, 2022).

Anjum, S., Chodey, M.D. and C, M.A. (2021) *Automation of Exam Hall Allotment and Seating Arrangement*, *ijert*. IJERT. Available at: https://www.ijert.org/automation-of-exam-hall-allotment-and-seating-arrangement (Accessed: August 3, 2022).

.

**APPENDICES**

1. Login Page

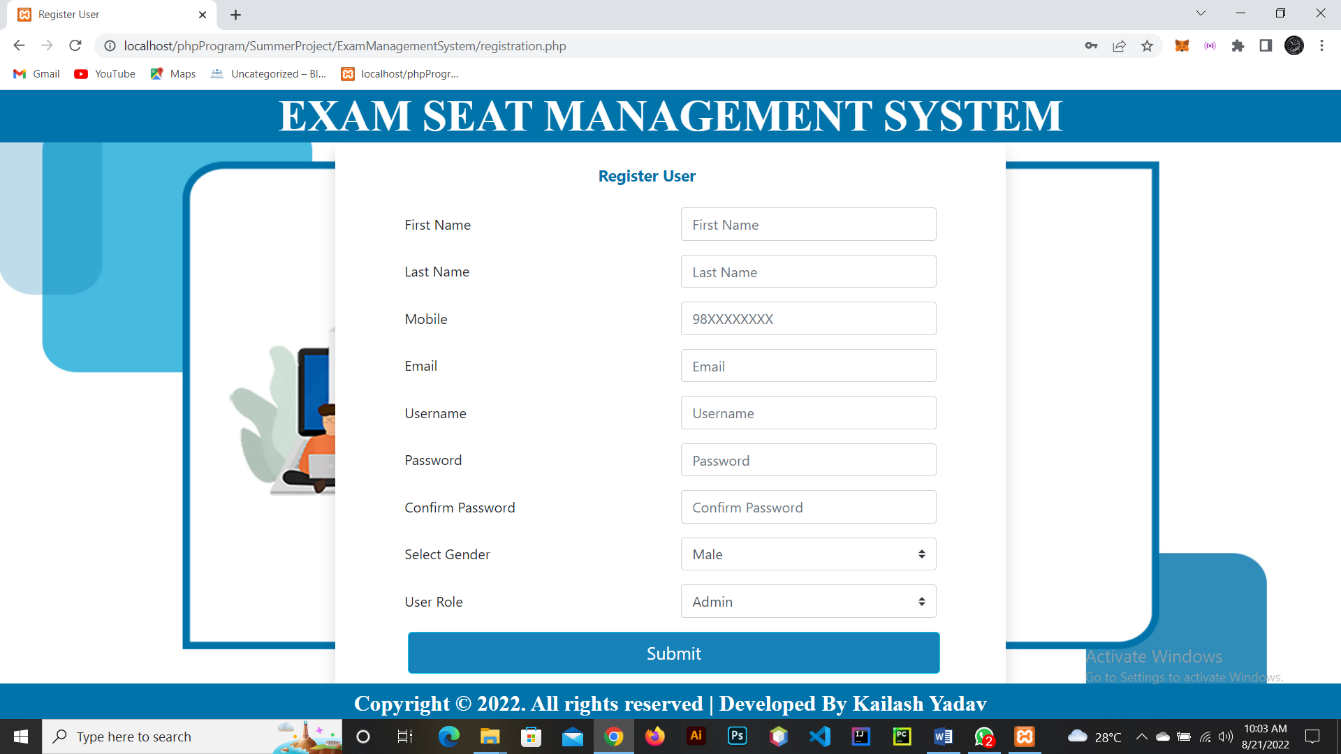
2. Dashboard With Closed Sidebar 

3. Dashboard With Open Sidebar

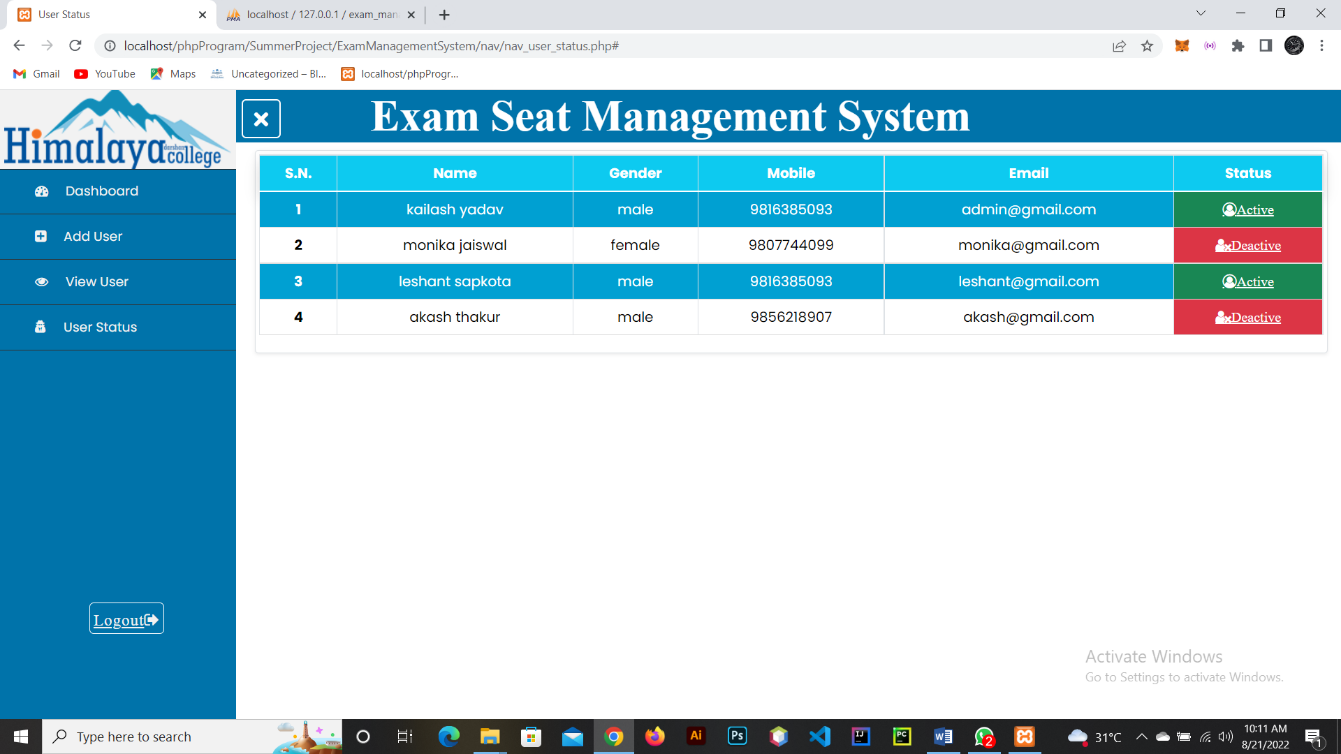


4. User

4.1 Register User

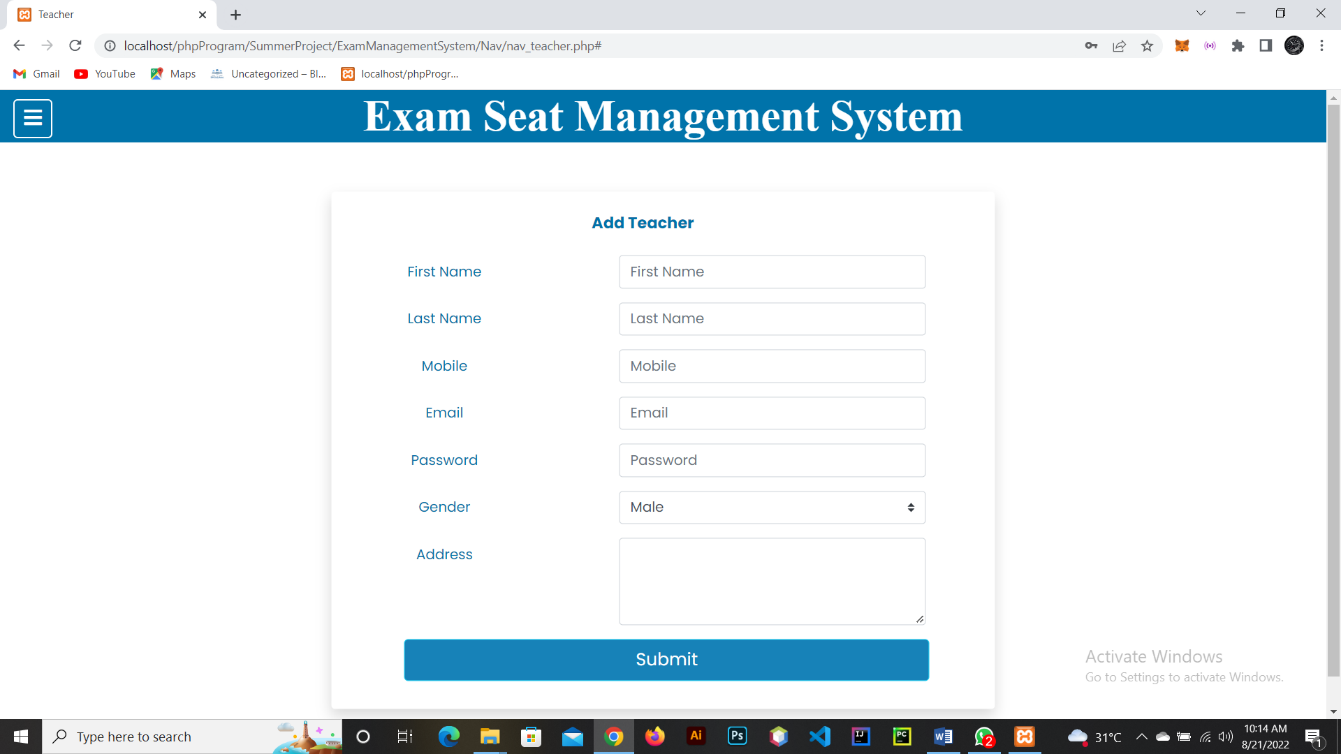


4.2.User Status

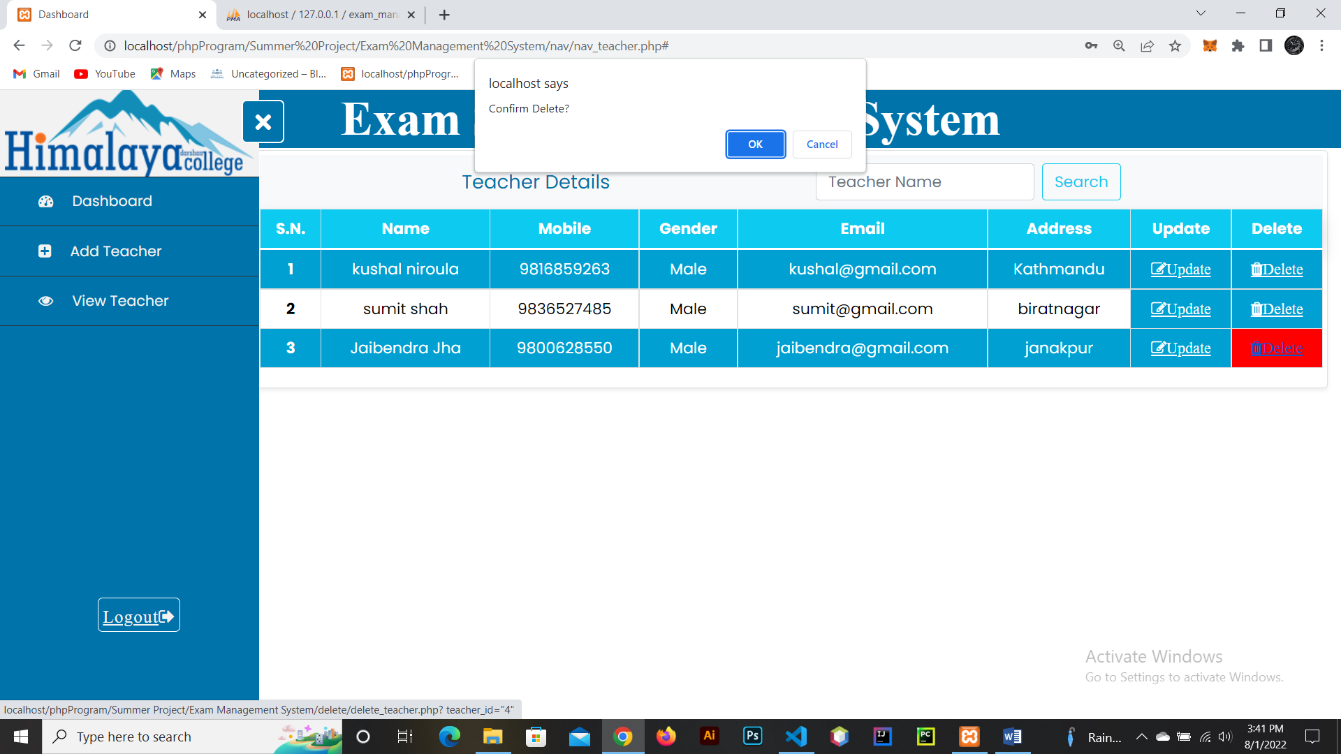


5. Teacher Management

5.1. Add Teacher

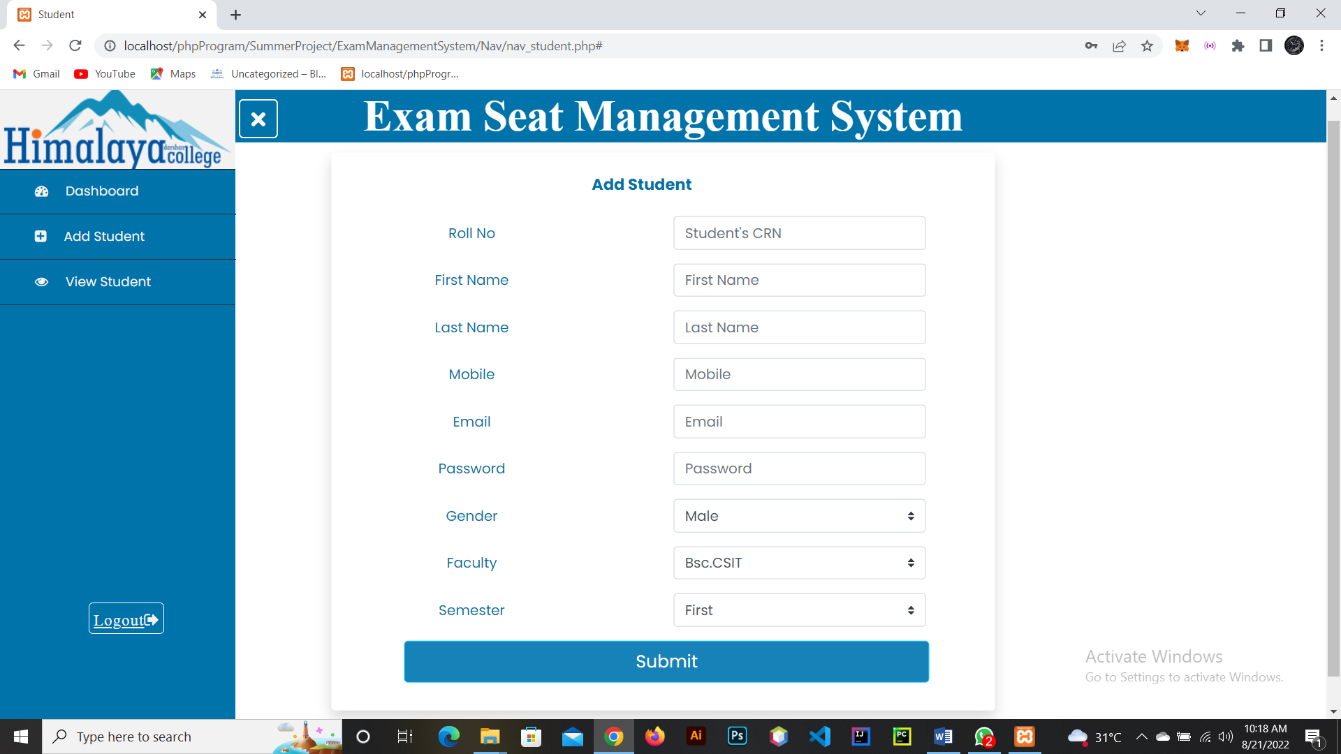


5.2.View Teacher

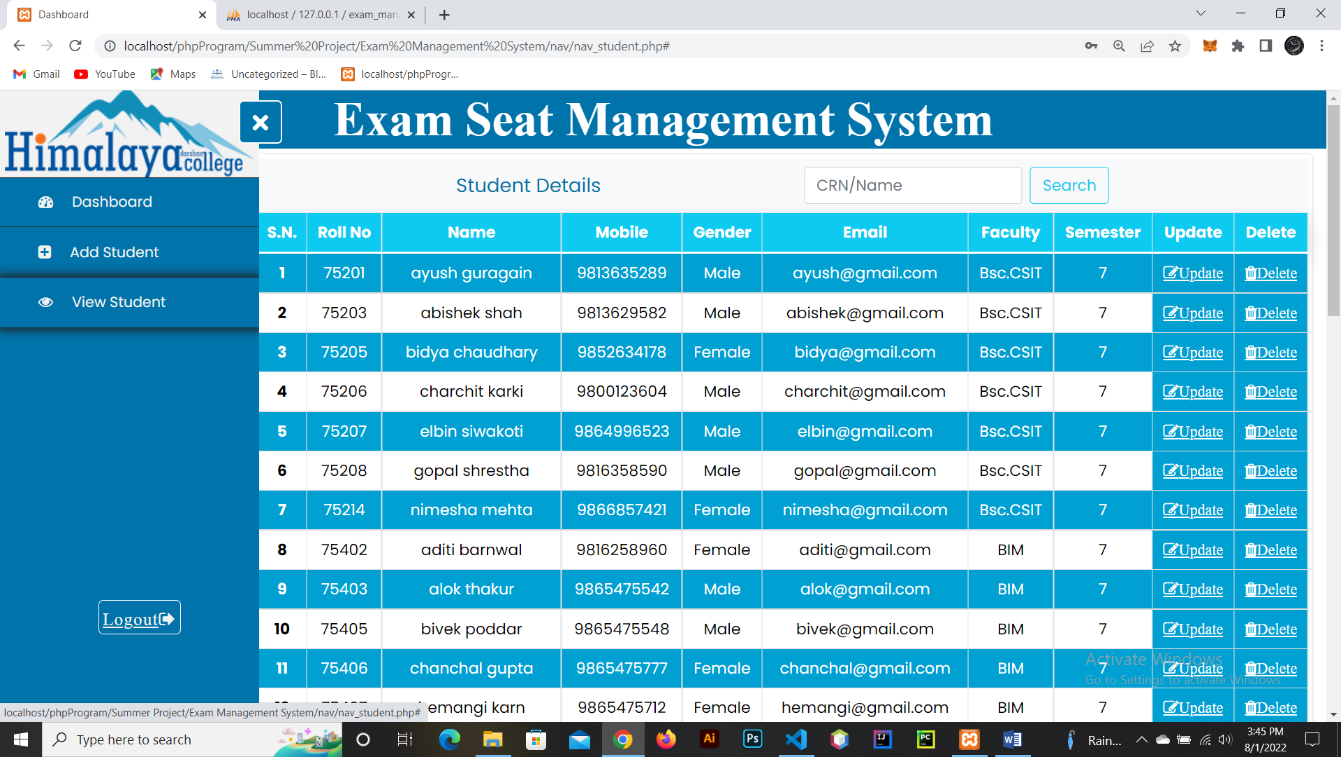


6. Student Management

6.1. Add Student

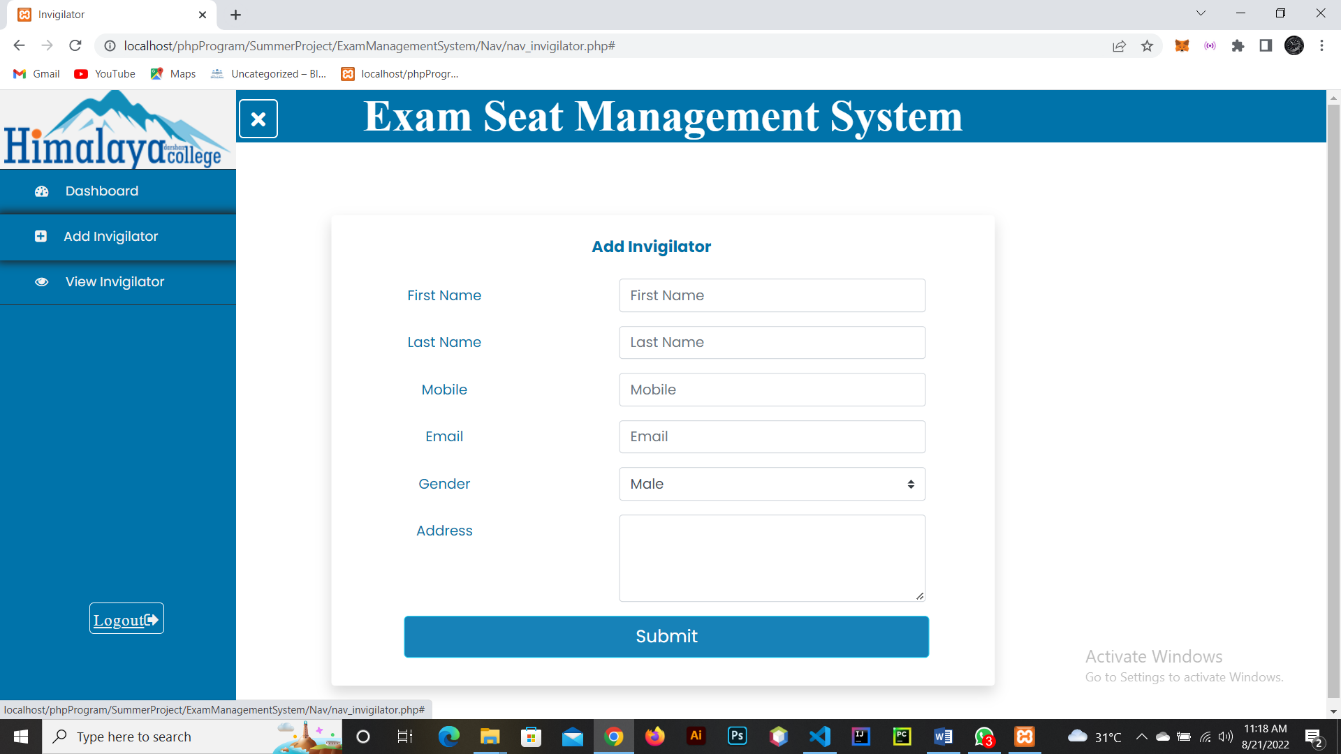


6.2.View Student

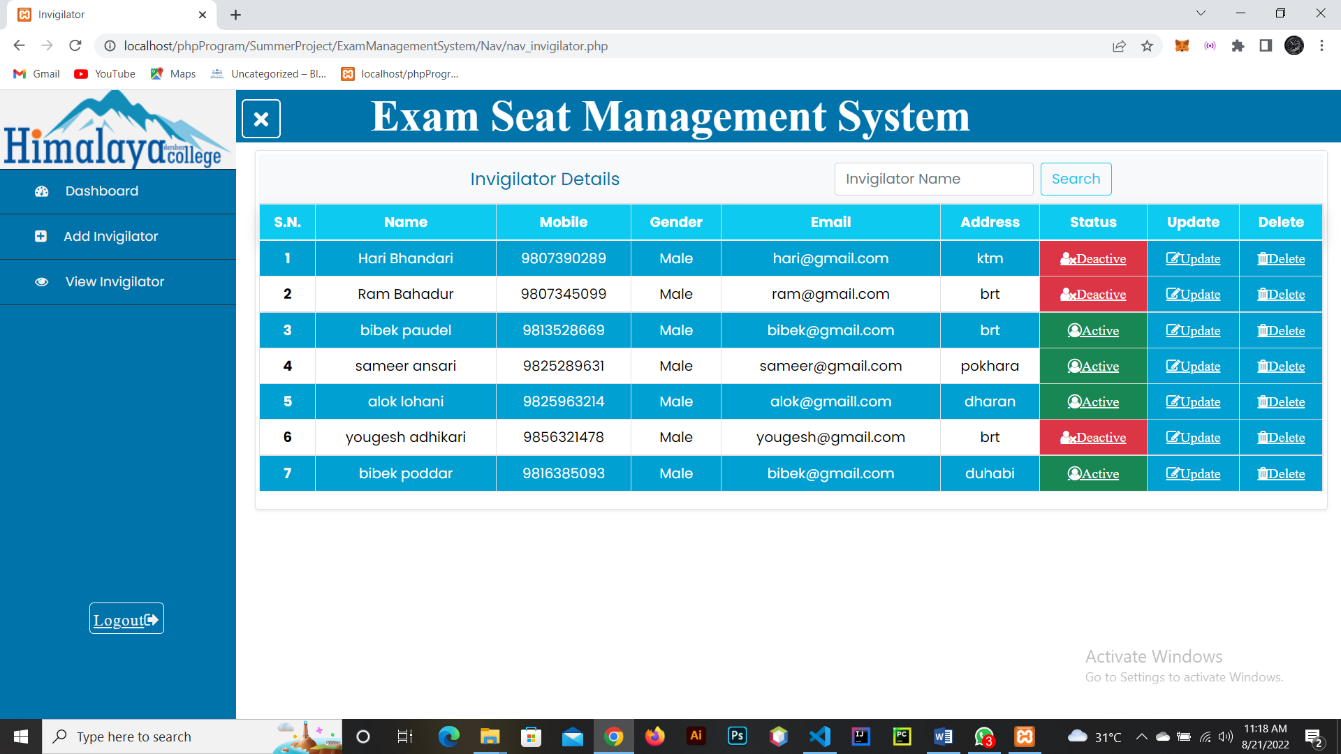


7. Invigilator Management

7.1 Add Invigilator

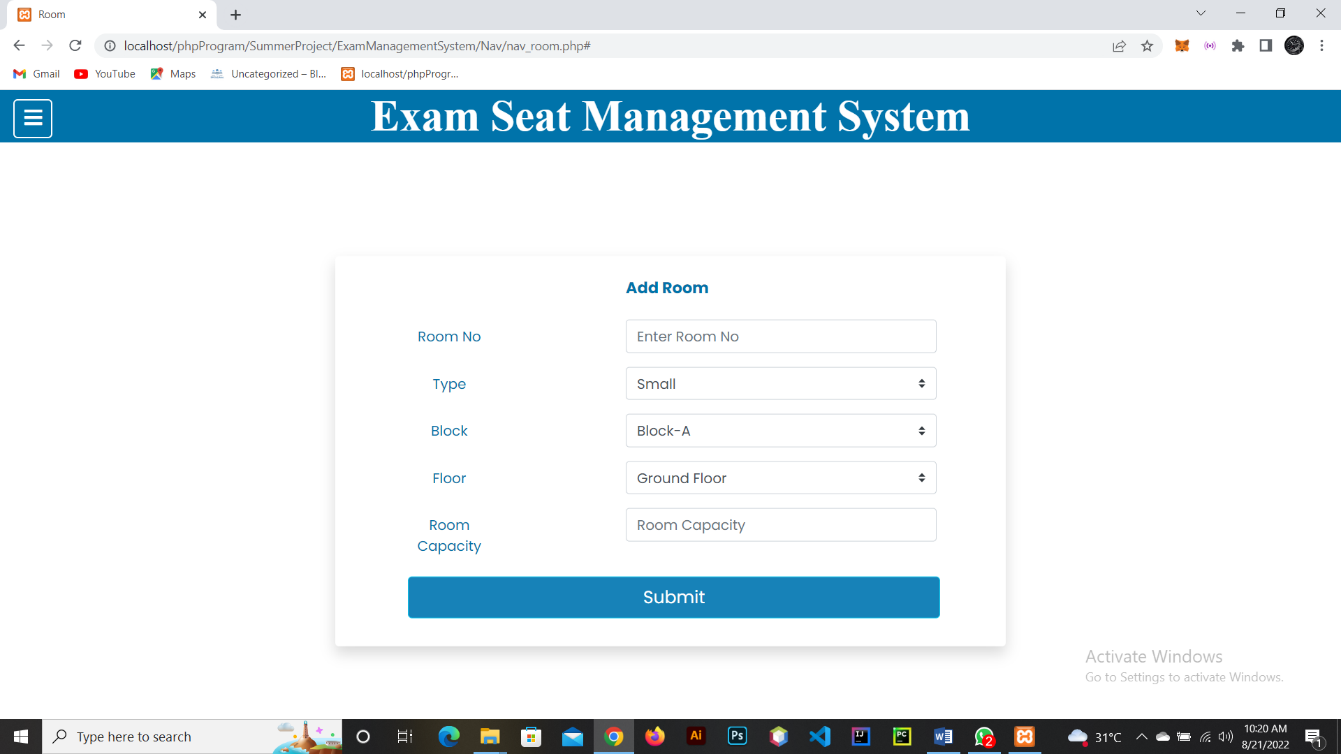


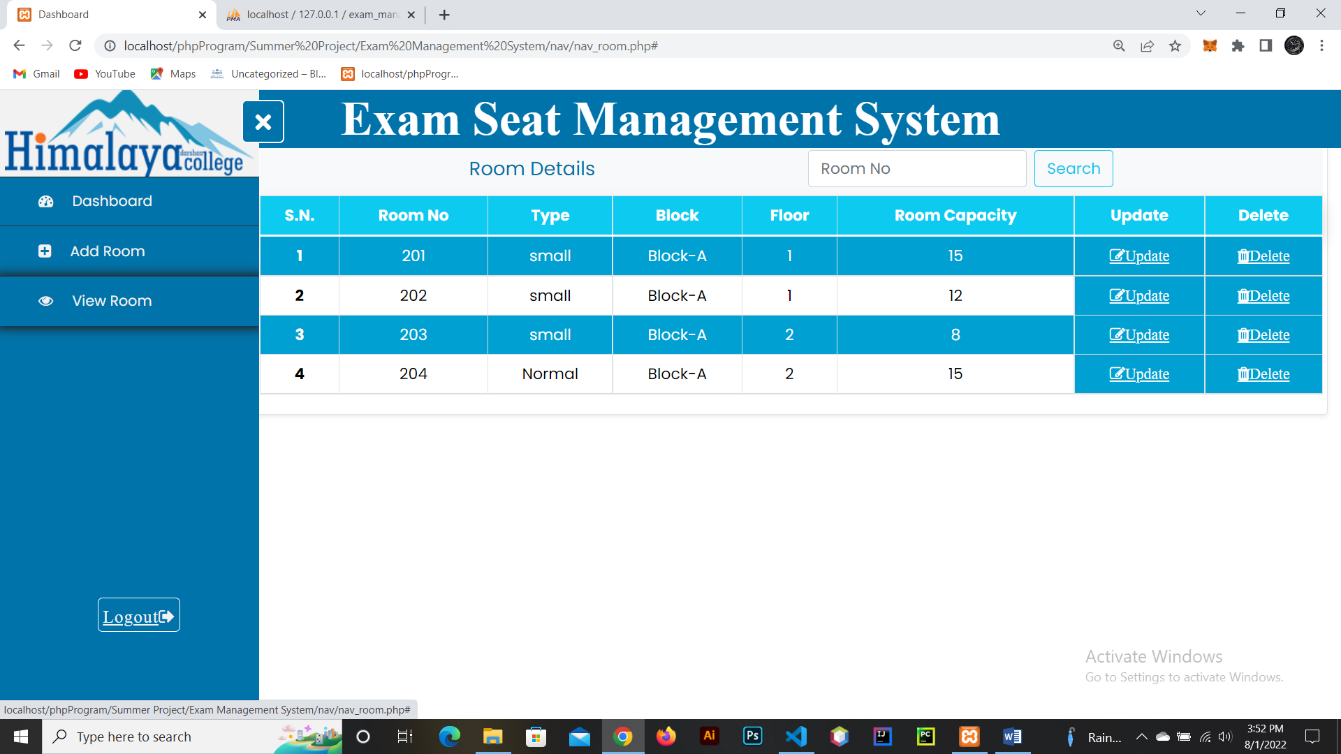
7.2 View Invigilator



8. Room Management

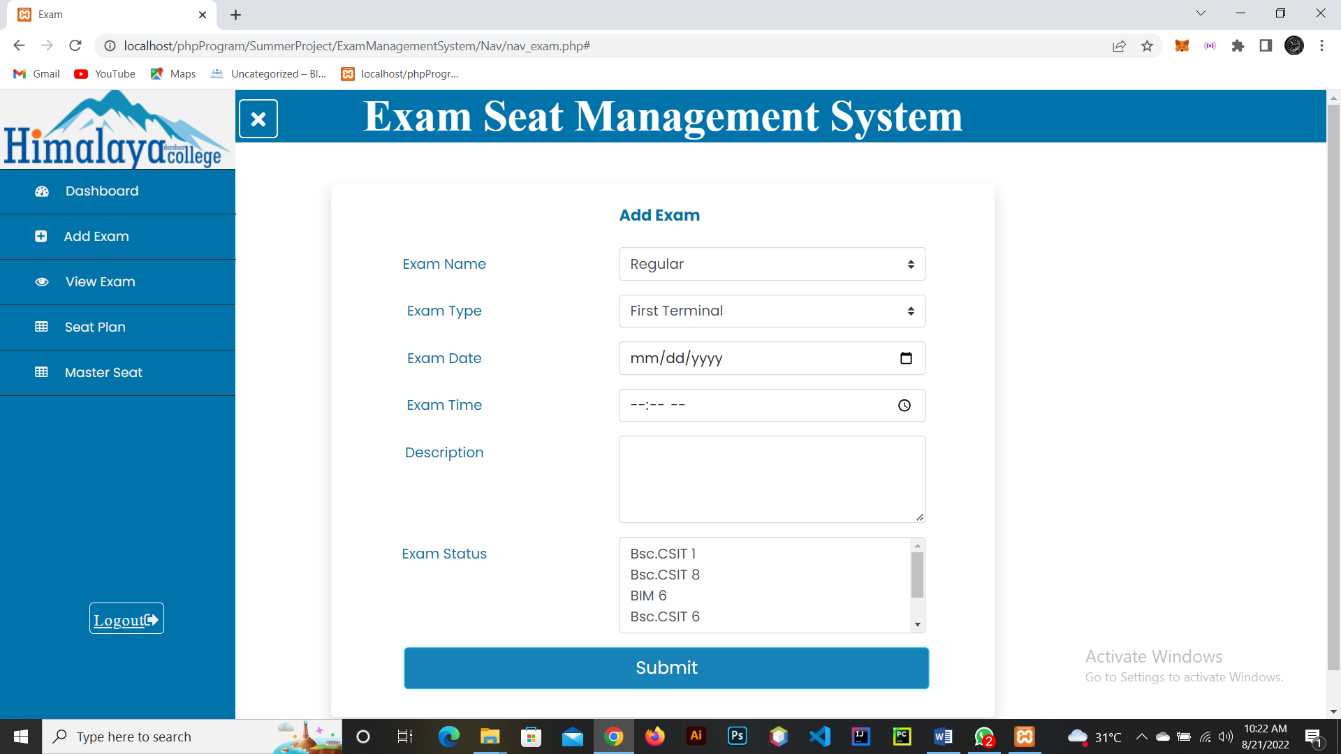
8.1. Add Room



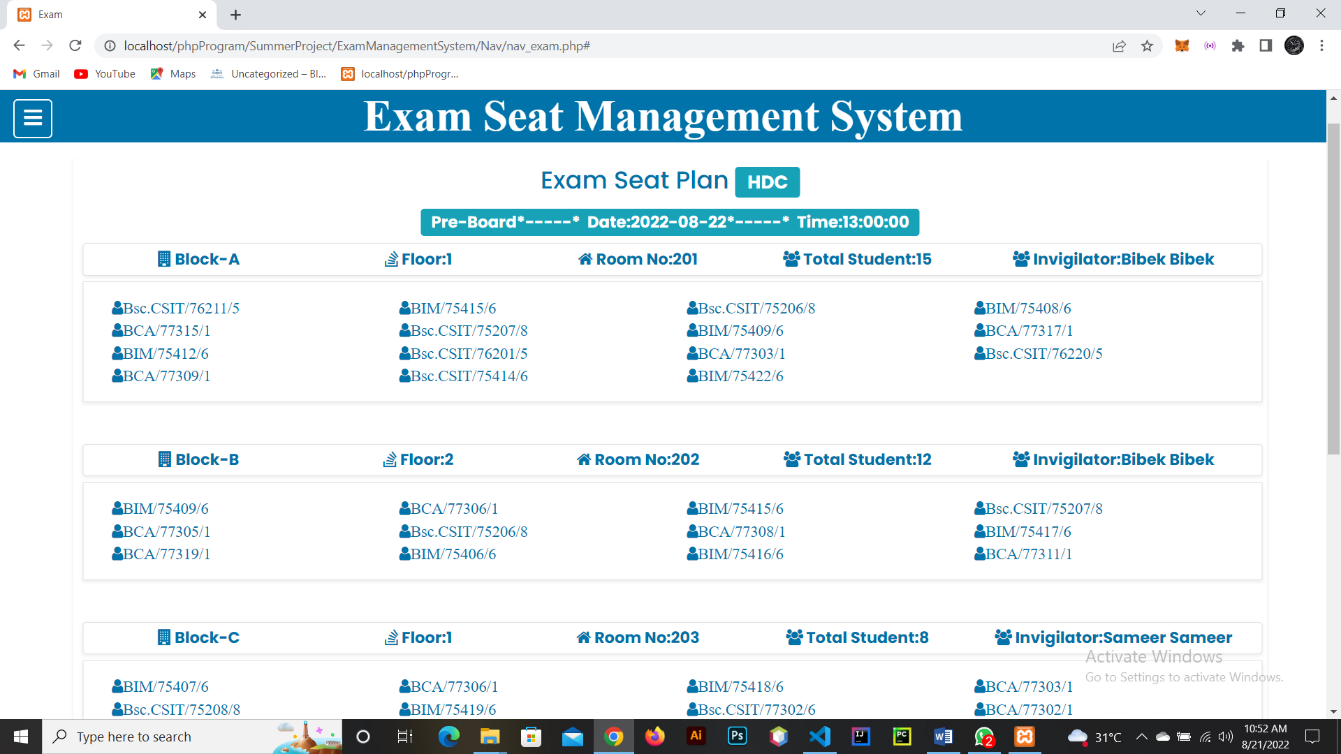
8.2.View Room

9. Exam Management

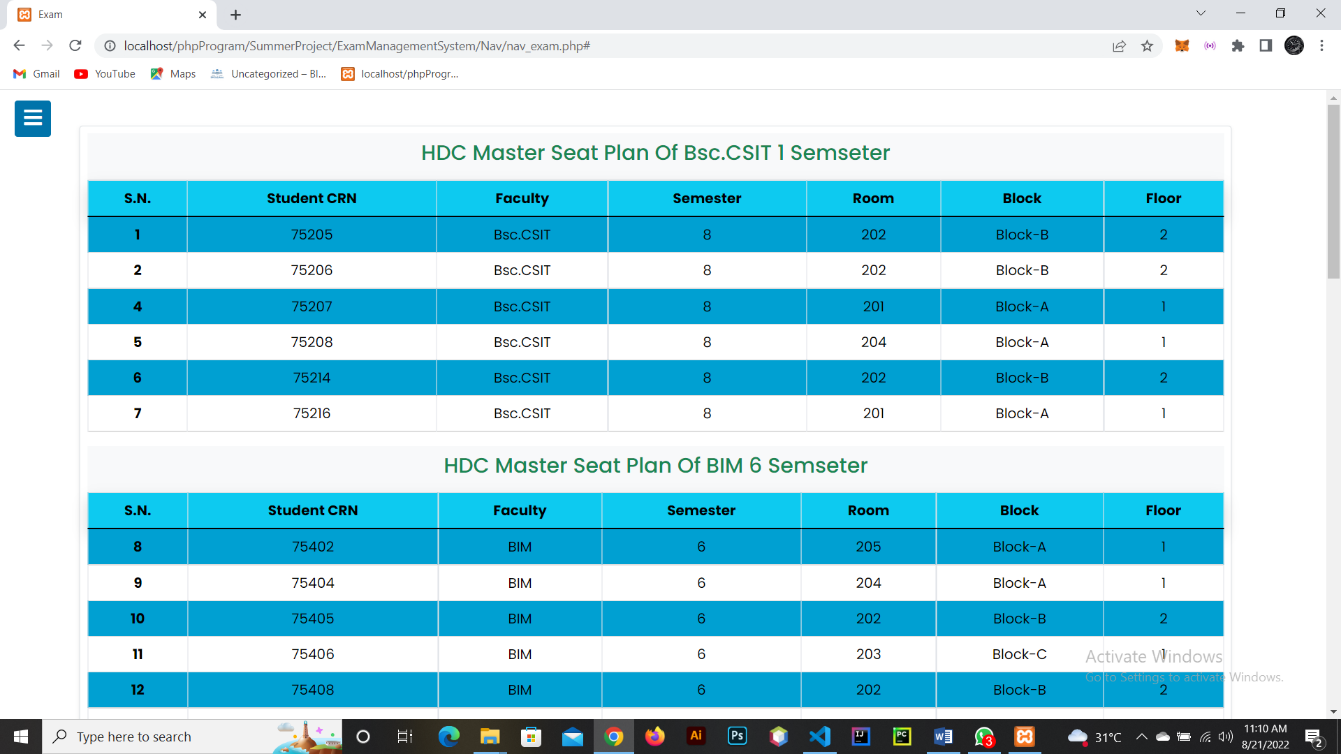
9.1. Add Exam



9.2. Seat Plan



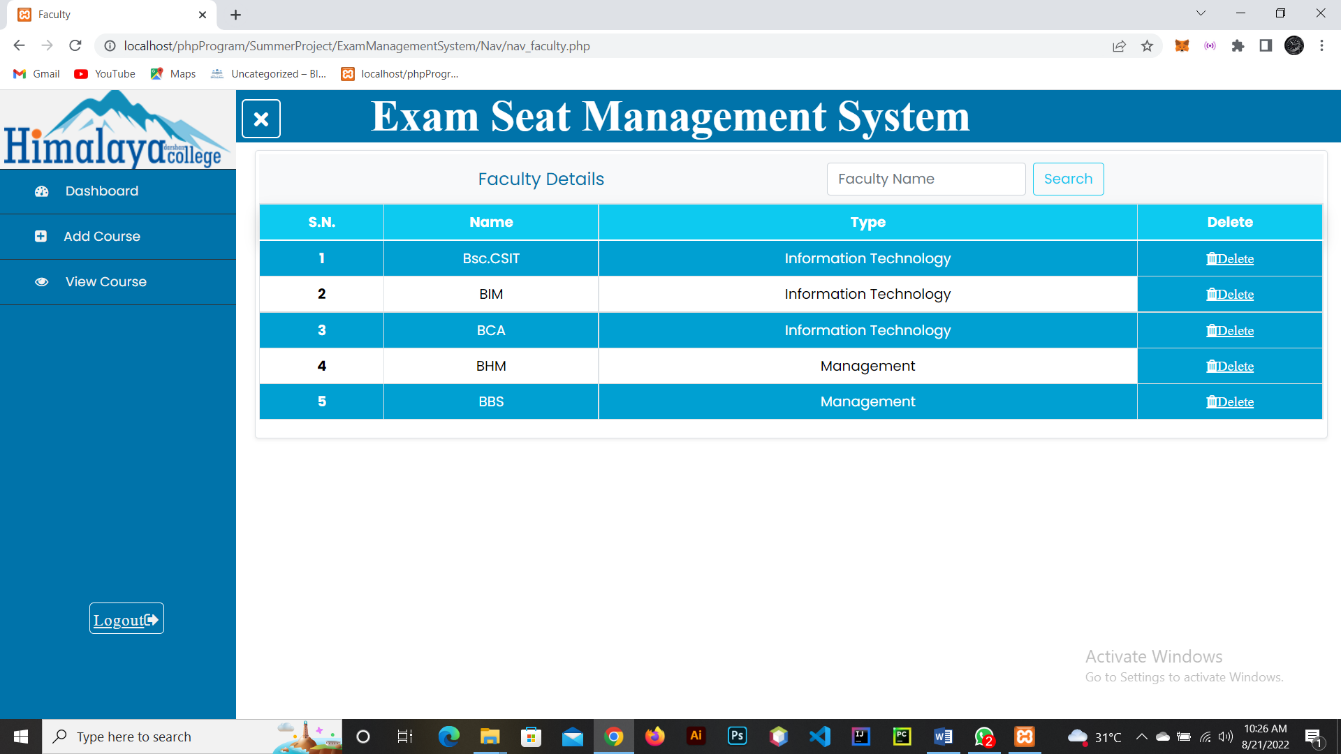
10. Master Seat



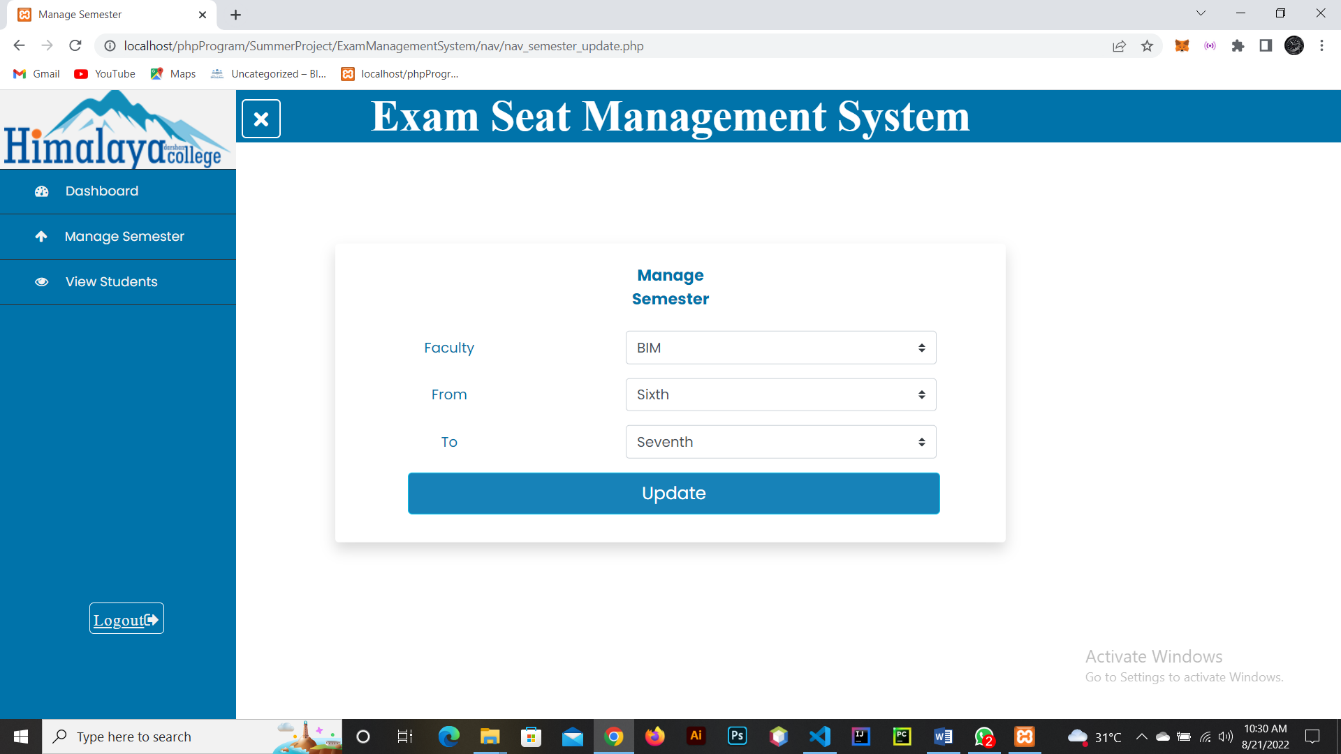


11. Faculty Management

11.1 View Faculty



12. Manage Semester



13. Change Password

