**PROJECT REPORT**

Submitted to

**DEPARTMENT OF COMPUTER SCIENCE**



**VIVEKANAND EDUCATION SOCIETY’S**

**COLLEGE OF ARTS, SCIENCE AND COMMERCE,**

**SINDHI SOCIETY, CHEMBUR, MUMBAI 400071.**

**(Autonomous)**

**Hostel Management System**

For Partial Fulfillment for Degree of

**Bachelor of Science (Computer Science)**

**Academic Year (2022-23)**

COODINATOR OF DEPARTMENT COLLEGE GUIDE

**Mr. Kamlakar Bhopatkar Dr. Madhavi Vaidya**

**Mr. Kamlakar Bhopatkar**

SUBMITTED BY

**Priyesh Panchal**

**VIVEKANAND EDUCATION SOCIETY’S**

**COLLEGE OF ARTS, SCIENCE AND COMMERCE**

**Sindhi Society, Chembur, Mumbai 400071. Phones:**

**25227514/25227470.**

**(Autonomous)**

**NAAC Re-Accredited ‘A’ Grade**

# **CERTIFICATE**

This is to certify that **Mr. Priyesh Panchal** of T.Y.B.Sc (Computer Science) affiliated to University of Mumbai has successfully completed a project work entitled

**Hostel Management System**

As partial fulfilment of the requirement for the degree of B.Sc. (Computer Science) for the academic year 2022-2023.

COORDINATOR OF DEPARTMENT COLLEGE GUIDE

**Mr. Kamlakar Bhopatkar Mr. Kamlakar Bhopatkar**

**Dr. Madhavi Vaidya**

Date:

Examiner: College Seal

**Acknowledgement:**

  I have great pleasure in presenting this project entitled “**Hostel Management System**” and I grab this opportunity to convey my immense regards towards all the people who with their invaluable contributions made this project successful.

     It gives me great pleasure in presenting this project report. Its justification will never complete if I don’t express my vote of thanks to our **V.E.S. College** and **Principal Dr. Anita Kanwar**

I sincerely thank and express my profound gratitude to our Project Guide **Mr. Kamlakar Bhopatkar and Dr. Madhavi Vaidya** for timely and prestigious guidance required for the project completion at each phase of the project development.

         I also owe to my friends who have been a constant source of help to solve the problems that cropped up during the development of the project, positive criticism, suggestions, constant support, encouragement and guidance force towards the successful completion of the project.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TOPIC** | **PAGE NO.** |
|  | **PRELIMINARY INVESTIGATION** |  |
| **1** | **Organizational Overview** | **6** |
| **2** | **Description of Current System.** | **7** |
| **3** | **Limitations of the Current System.** | **8** |
| **4** | **Proposed system and its Advantages.** | **9** |
| **5** | **Technologies Used & Basic Scenarios.** | **11** |
|  | **SYSTEM ANALYSIS** |  |
| **5** | **Use Case Diagram.** | **13** |
| **6** | **Activity Diagram.** | **15** |
| **7** | **Entity Relationship Diagram.** | **17** |
|  | **SYSTEM IMPLEMENTATION** |  |
| **8** | **Test Cases.** | **20** |
| **9** | **Screenshots.** | **22** |
| **10** | **Bibliography & References.** | **26** |

**PRELIMINARY INVESTIGATION**

**ORGANIZATIONAL OVERVIEW.**

The initiated project is a dummy project to help reduce a lot of manual inputs to manage hostel data which currently is very difficult to save the records of the students about their rooms and other things. It helps the manual work from which it is very difficult to find the record of the students and the mess bills of the students and information about the ones who had left the hostel years ago.

**DESCRIPTION OF CURRENT SYSTEM.**

The hostel keeps a database of information regarding students who are staying at the hostel. It also incorporates in its database details of fees paid. The tasks mentioned above are performed manually using a register. The paper work also includes new student entry, updates in fee structure, etc.

Existing system has two modules:

(1)     ACCOMMODATION MODULE

(2)     BILLING MODULE

**Accommodation Module:** Accommodation module is first important module of existing system. The main job of this module is to make entry of students applying for accommodation and to provide information regarding vacancies and profile of students already registered. As and when a student applies, he is made aware of the rules and regulation of the stay at the refuge. That done, if the student accepts and promises to abide by the rules, he may be made aware of the vacancies and the students occupying the various rooms, and he may be given the choice to decide on a room provided, the room has the capacity.

**Billing Module:** Billing Module is a second important module of existing Hostel system. The main purpose of this module is to extensively deal in finances, making entry of fees paid up by the students. The quantity of fees paid up is then updated into the fees file. Also an efficient mechanism is set up to ensure that the records set up student wise. The main problem faced is dealing with the databases such as fees register and other types of register in which they have to update, delete. With every transaction, the details are noted down sequentially in the order of time and that too, manually. Thus, use of computer come into existence where there are such tedious and very hard tasks to be performed in the system**.**

**LIMITATIONS OF CURRENT SYSTEM.**

* Time Consuming.
* Editing of data is tedious job.
* No Security of data.
* Maintenance is difficult.
* No obtaining accurate and relevant information.
* Problems of Report generation.

**PROPOSED SYSTEM.**

Hostel Room Allocation System is a web application which aims at computerization of the current procedure of allocating hostel rooms. Currently, the process involves students filling up the forms and submitting them in respective hostel offices which involves lots of paper work, hence less efficient. This system will help the hostel officer to be able to manage the affairs of the hostel. This system will provide full information about a student in the hostel. It will show rooms available or not and number of people in a particular room. There will also be an administrator module which will accessed by the administrator and has the ability to delete, add and edit employee records. This system will be developed based on Software Development Life Cycle (SDLC) with PHP and XAMPP server. There will be predefined criteria for the reservation to the hostels. He/she checks the attested application forms of the students obtained from the internet and verify it with the student database. If the student are found eligible then they are allotted to the hostel room.

Modules involved in the project:

* **Administrator module:**

The Administrator can:

* Allot different students to the different hostels.
* Vacate the students from the hostel.
* Edit the details of the students & modify the student records.
* **User module:**
* Can submit the application form.
* Can view the notice board.
* Can submit the vacating form.
* **Application module:**

This section provides a form to the students which can be filled by them, and a copy of the filled page can be taken in the printed form. This is later submitted to the hostel authorities and can be verified by them before allotting them to the respective hostel rooms.

**ADVANTAGES OF PROPOSED SYSTEM.**

* Security of data.
* Ensure data accuracy.
* Proper control of higher officials.
* Minimize manual data entry.
* Minimum time needed for the various processing.
* Greater efficiency.
* Better service.
* User friendliness and interactive.
* Minimum time required.

**TECHNOLOGIES USED:**

The technologies used to create this website is as follows:

1. **HTML:** HTML is the standard mark-up language for documents designed to be displayed in a web browser.

2. **CSS:** CSS is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML.

3. **jQuery:** jQuery is a fast, small, and feature-rich JavaScript library.it makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

4. **JavaScript:** JavaScript is a client scripting language which is used for creating web pages. It is a standalone language developed in Netscape. It’s used to make webpages dynamic.

**5. PHP:** PHP (Hypertext PreProcessor) is the most widely used open source and general purpose server side scripting language used mainly in web development to create dynamic websites and applications.

**6. MySQL:** MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. In association with a scripting language such as PHP or Perl (both offered on our hosting accounts) it is possible to create websites which will interact in real-time with a MySQL database to rapidly display categories and searchable information to a website user.

**SYSTEM ANALYSIS**

**USE CASE DIAGRAM.**

A use case diagram depicts the various operations that a system performs. It contains use cases, actors, and their relationships. Use cases are the sequence of actions that form a single unit of work for an actor. An actor represents a user who is external to the system and interacts with the use case.

**ELEMENTS OF USE CASE DIAGRAM:**

**Actors**

An actor portrays any entity (or entities) that perform certain roles in a given system. The different roles the actor represents are the actual business roles of users in a given system. An actor in a use case diagram interacts with a use case. For example, for modelling a banking application, a customer entity represents an actor in the application. Similarly, the person who provides service at the counter is also an actor. But it is up to you to consider what actors make an impact on the functionality that you want to model. If an entity does not affect a certain piece of functionality that you are modelling, it makes no sense to represent it as an actor. An actor is shown as a stick figure in a use case diagram depicted "outside" the system boundary.

**Use Cases**

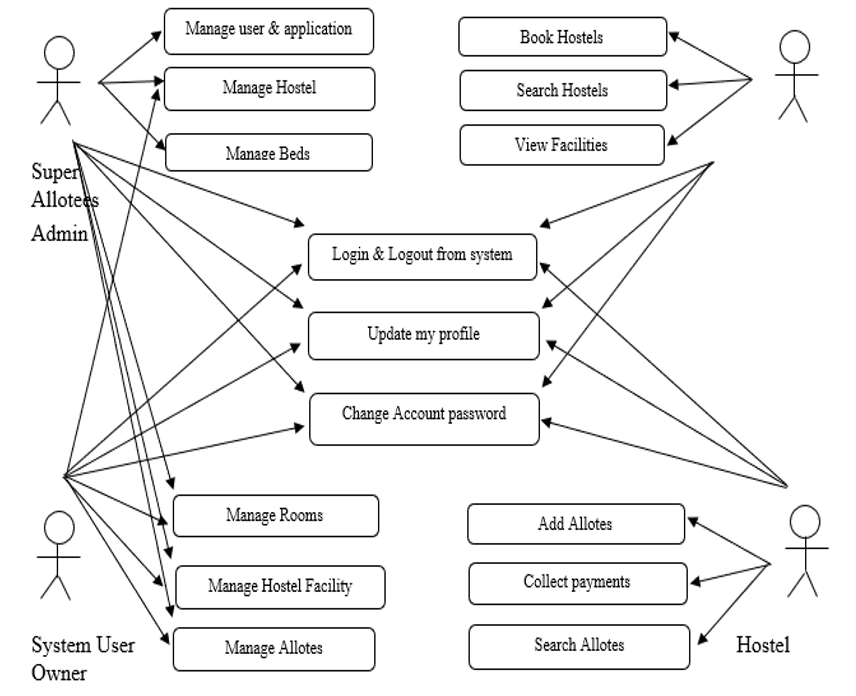
A use case in a use case diagram is a visual representation of distinct business functionality in a system. The key term here is "distinct business functionality." To choose a business process as a likely candidate for modelling as a use case, you need to ensure that the business process is discrete in nature. As the first step in identifying use cases, you should list the discrete business functions in your problem statement. Each of these business functions can be classified as a potential use case. Remember that identifying use cases is a discovery rather than a creation. As business functionality becomes clearer, the underlying use cases become more easily evident. A use case is shown as an ellipse in a use case diagram.

**System Boundary**

A system boundary defines the scope of what a system will be. A system cannot have infinite functionality. So, it follows that use cases also need to have definitive limits defined. A system boundary of a use case diagram defines the limits of the system. The system boundary is shown as a rectangle spanning all the use cases in the system.

**Relationships: The following relationships can be established among use cases**

* **Extends:** A use case may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label “**«extend»**".
* **Includes:** A use case may include another. Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case. The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behavior from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "**«include»**".

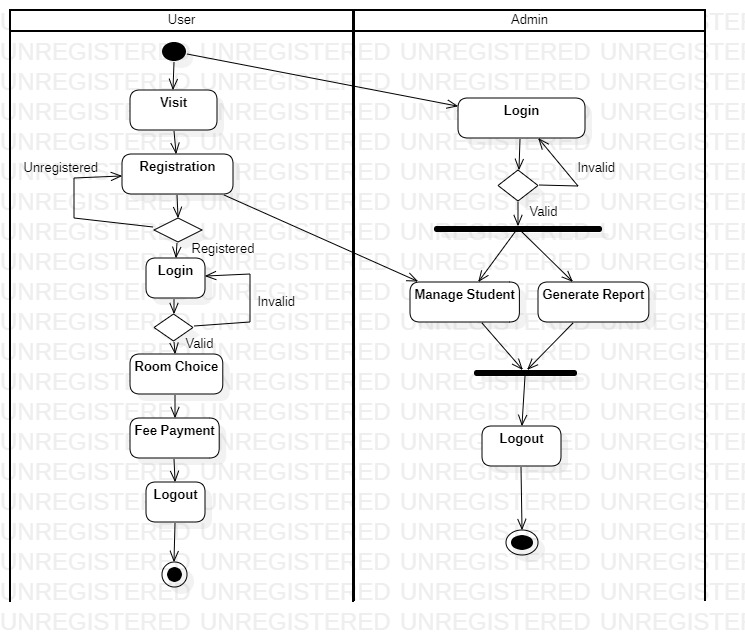


**ACTIVITY DIAGRAM.**

An activity diagram visually presents a series of actions or flow of control in a system similar to a [flowchart](https://www.smartdraw.com/flowchart/) or a [data](https://www.smartdraw.com/data-flow-diagram/) [flow](https://www.smartdraw.com/data-flow-diagram/) [diagram](https://www.smartdraw.com/data-flow-diagram/). Activity diagrams are often used in business process modeling. They can also describe the steps in a [use](https://www.smartdraw.com/use-case-diagram/) [case](https://www.smartdraw.com/use-case-diagram/) [diagram](https://www.smartdraw.com/use-case-diagram/). Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

In between there are ways to depict activities, flows, decisions, guards, merge and time events and more.

* **Initial State or Start Point: -** A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram.
* **Activity or Action State: -** An action state represents the non-interruptible action of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.
* **Action Flow: -** Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.
* **Decisions and Branching: -** A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."
* **Synchronization: -** A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.
* **Swim lanes: -** Swim lanes group related activities into one column.
* **Final State or End Point: -** An arrow pointing to a filled circle nested inside another circle represents the final action state.



**Entity Relationship Diagram**

Data models are tools used in analysis to describe the data requirement and assumptions in the system from a top-down perspective. They also set the stage for design of databases later on in the SDLC.

**There are three basic elements in ER model:**

**Entities** are the “things” about which we seek information.

**Attributes** are the data we collect about the entities.

**Relationships** provide the structure needed to draw information from multiple entities.

**Entity:** It represents a collection of objects o things in the real world whose individual members or instances have the following characteristics:

Each can be identified uniquely in some fashion.

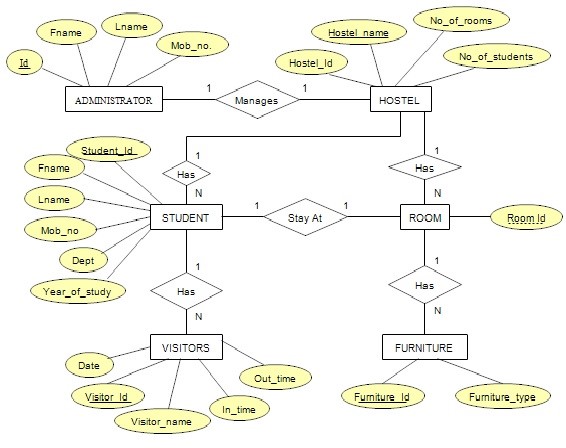
Each plays a necessary role in the system we are building.

Each can be described by one or more data elements.

**Attributes:** They express the properties of entities. Attributes having unique values are called candidate keys (Primary key).

**Relationships:** They describe the association between entities. They are characterized by cardinality as follows:

* **One-to-One relationship** means an instance of the first entity is associated with only one instance of second entity.
* **One-to-Many relationship** means that one instance of the first entity is related to many instances of second entity, while an instance of second entity is associated with only instance of the first entity



**SYSTEM IMPLEMENTATION**

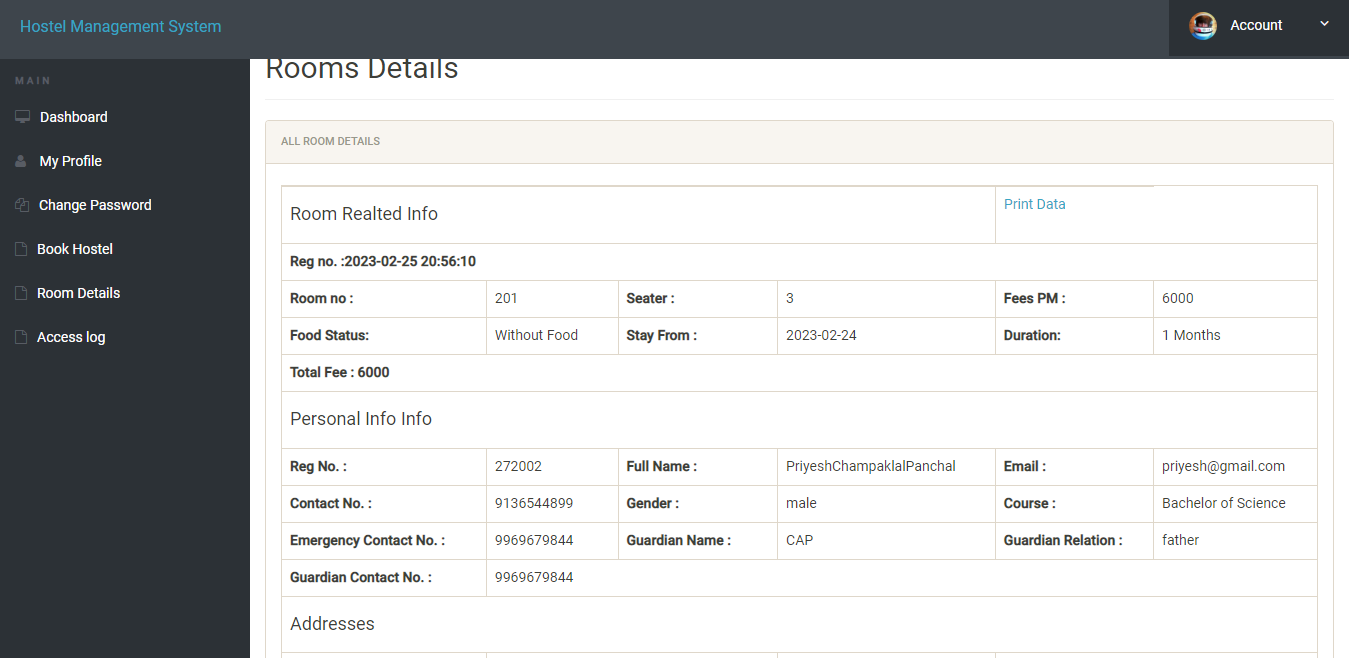
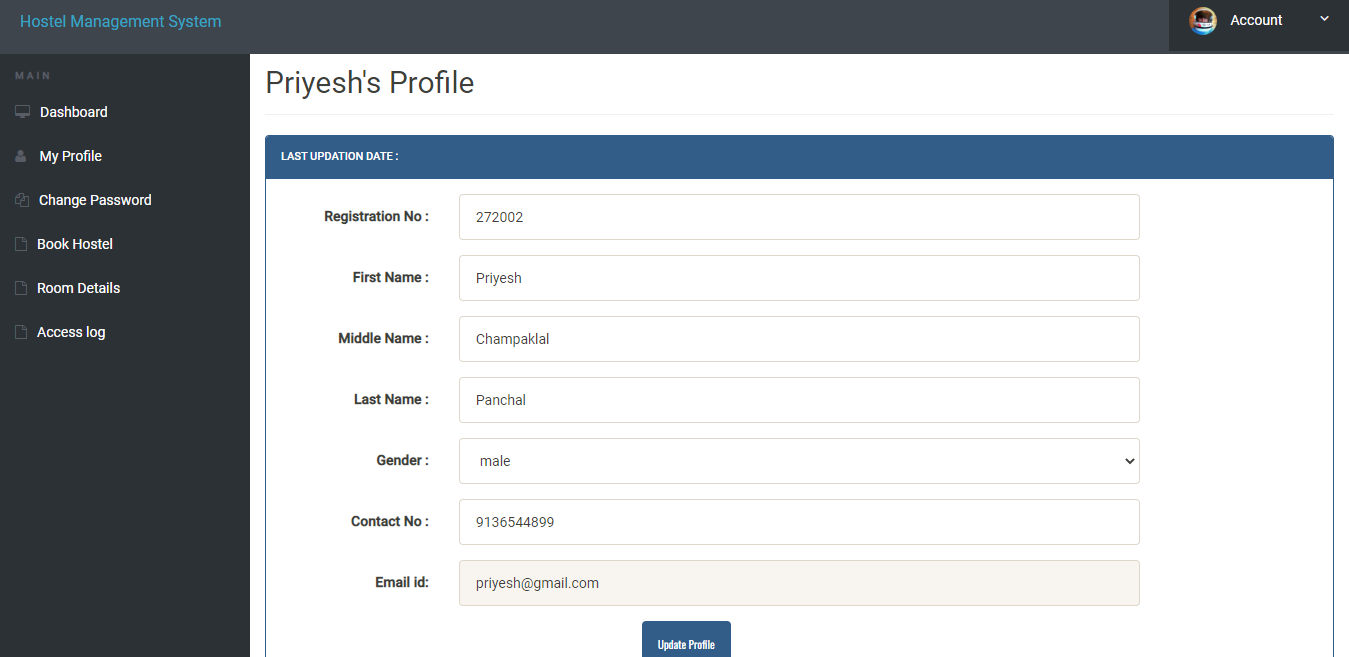
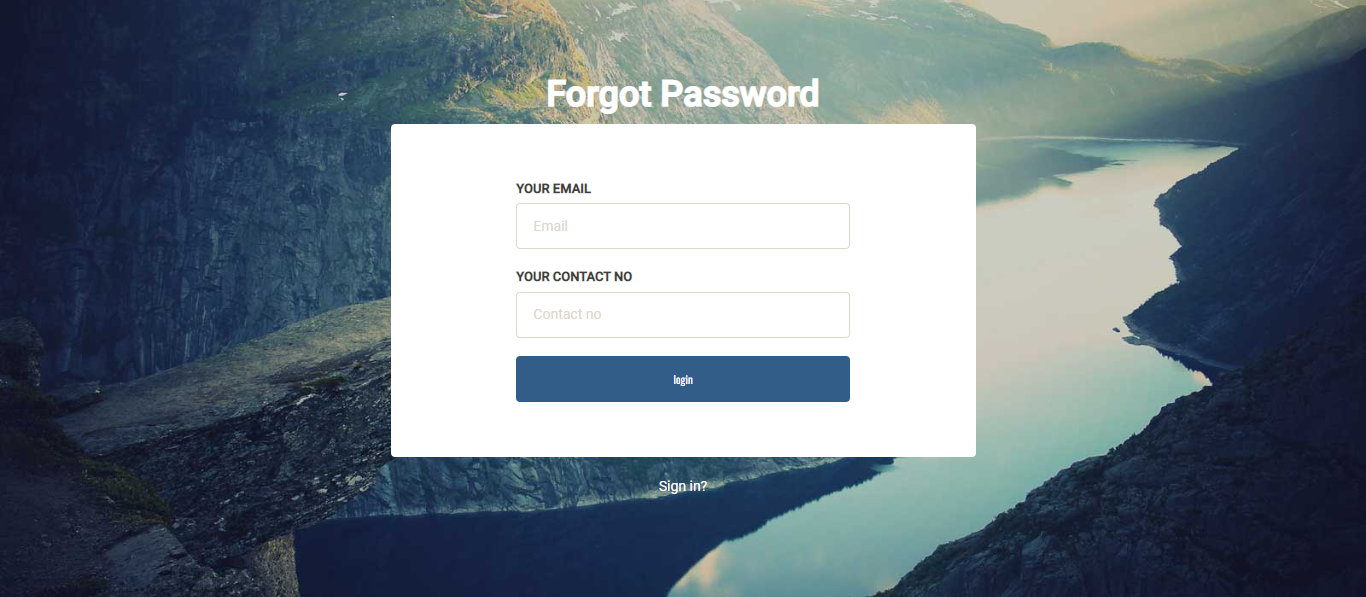
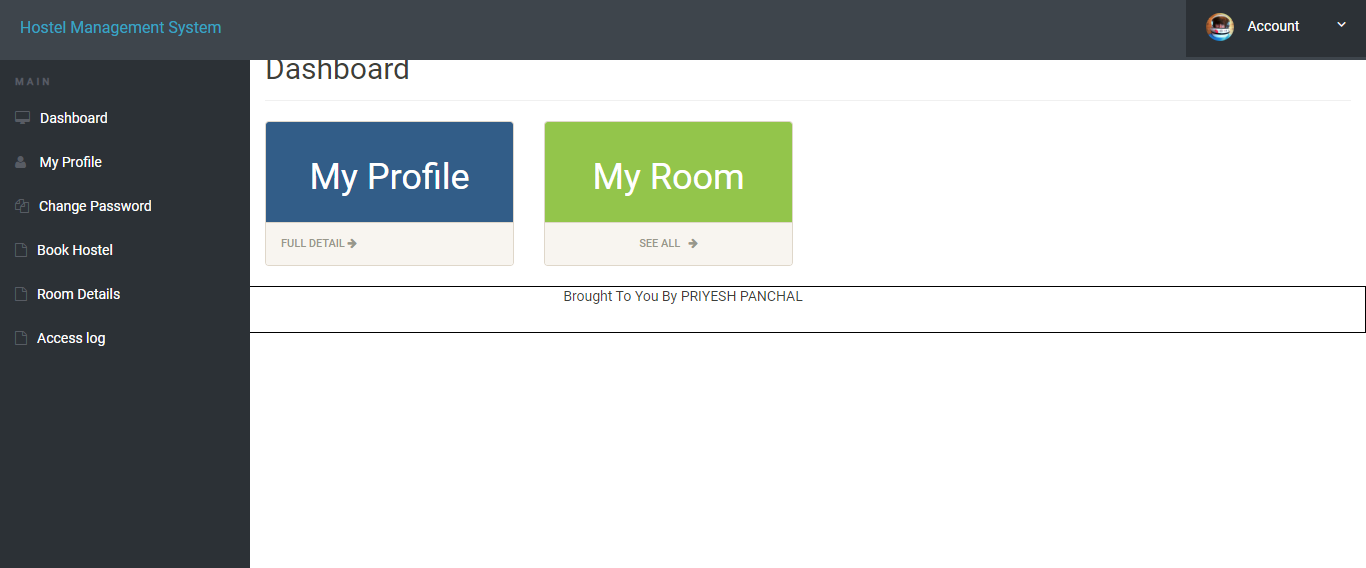
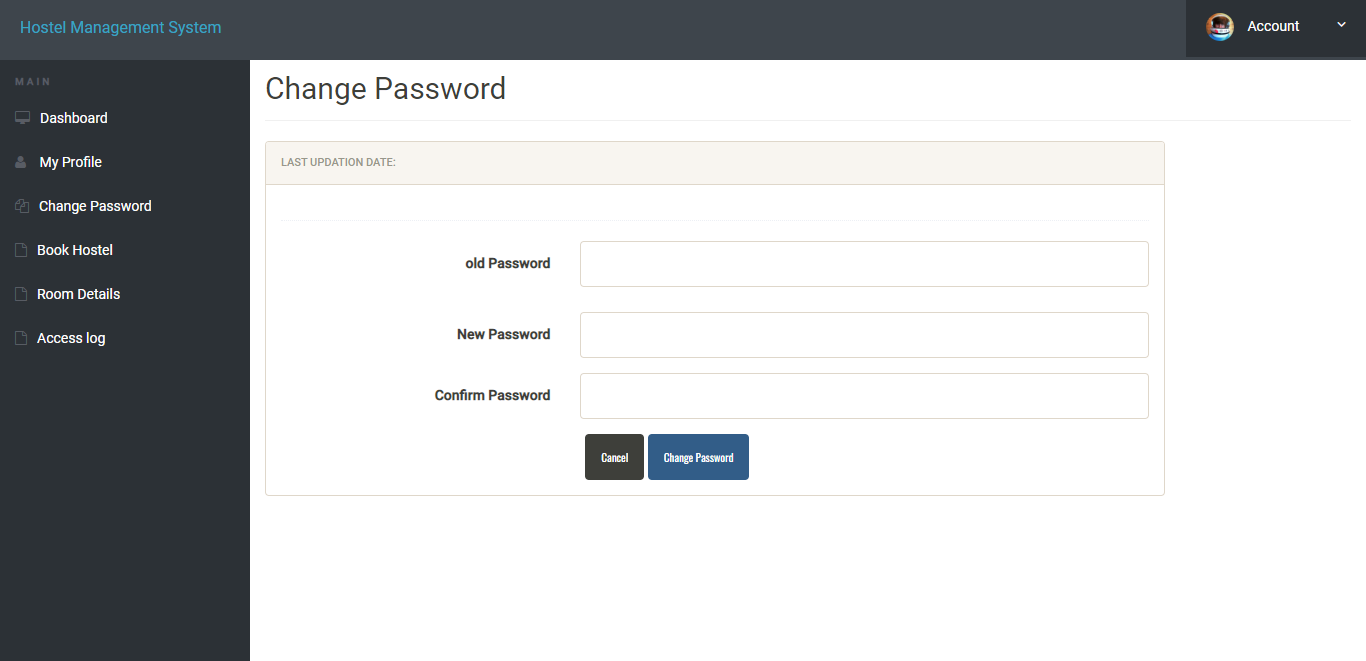
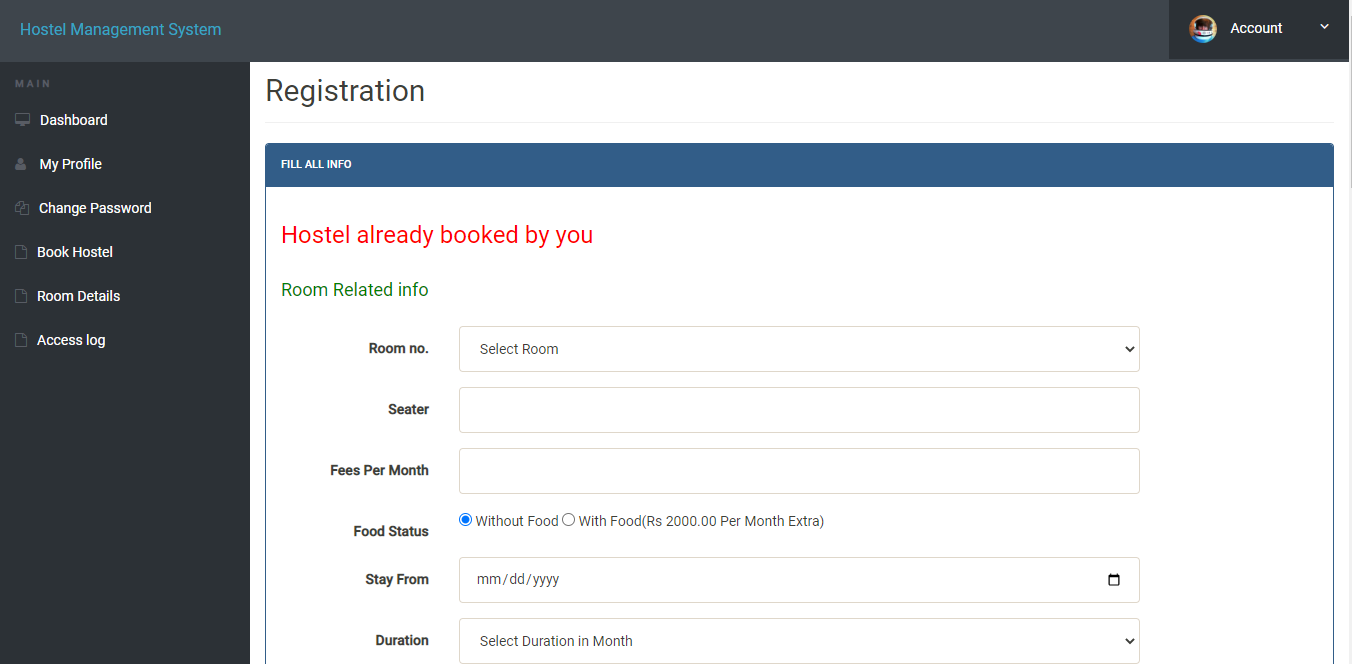
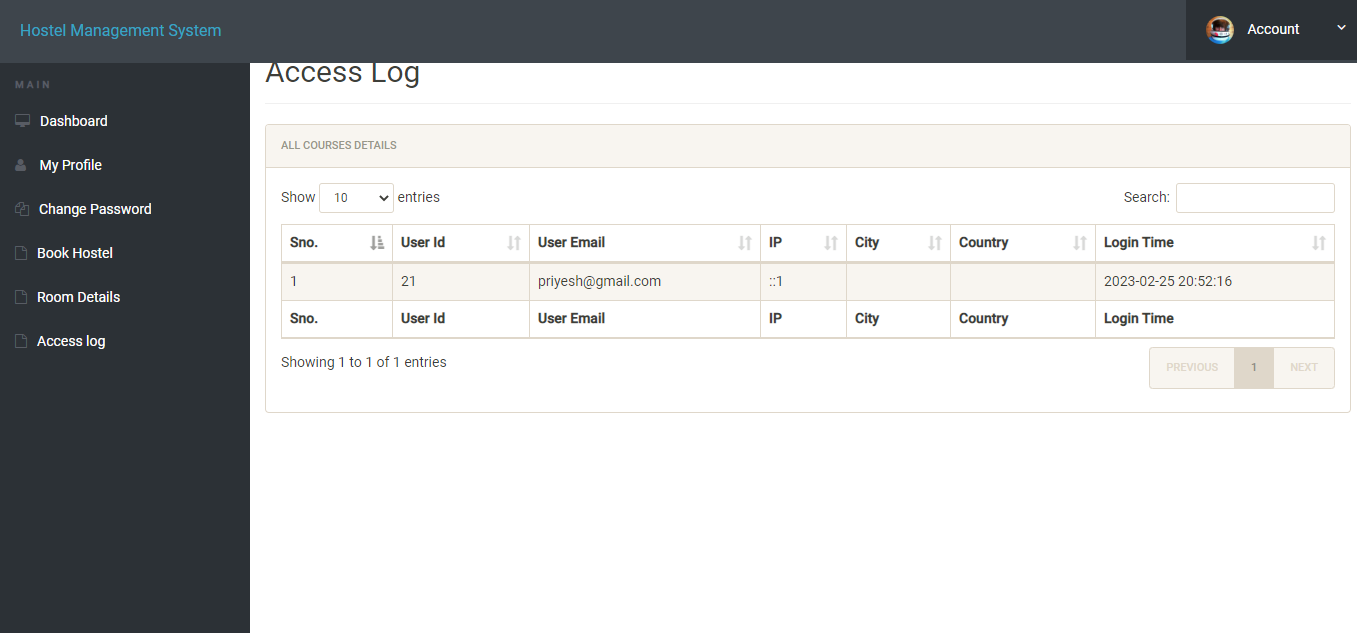
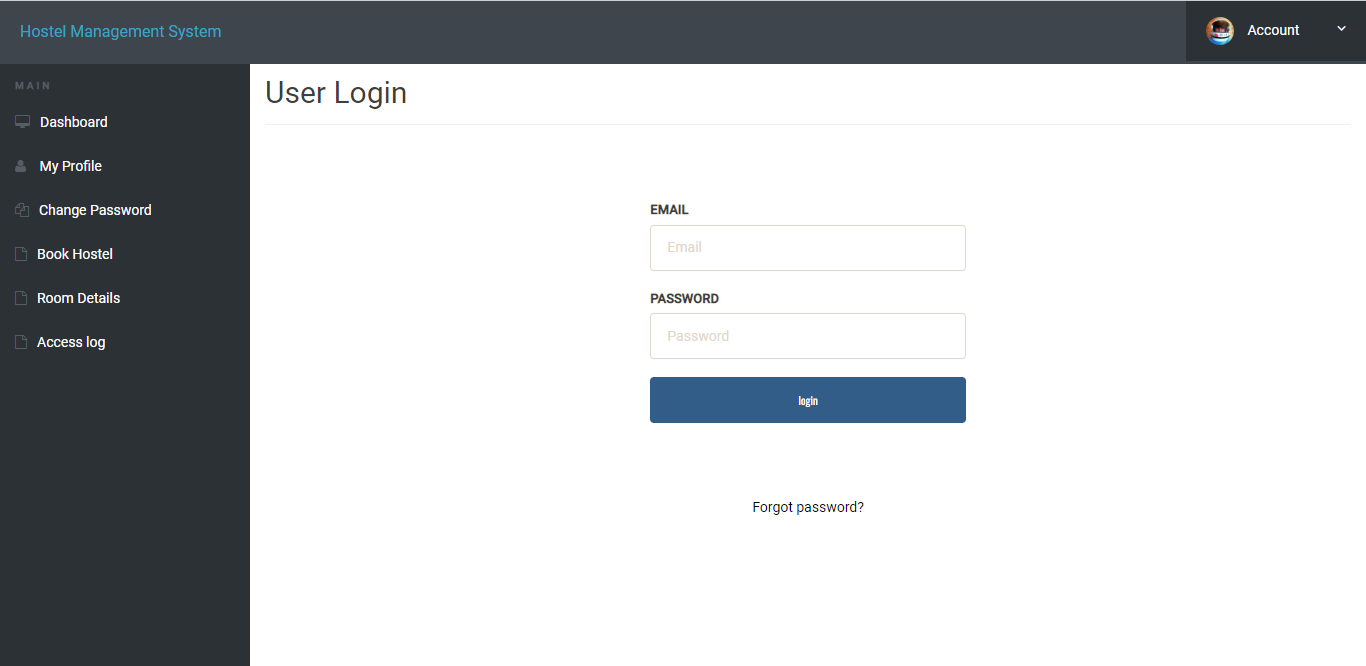
**TEST CASES.**

AP – Admin Module

UP – User Module

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Cases ID | Module | Form | Test Conditions | Steps | Input Test Data | Expected Output | Actual Output | Status |
| AP | Admin | Admin | Validate User  Login | 1)Enter  User  username 2)Enter User password | 1)username 2)password | It must show error message  “Incorrect username or password” | It showed error message  “Incorrect username or password” | Pass |
| UP1 | User | User Login | Validate User Login | 1)Enter  User  username 2)Enter User password | 1)username 2)password | It must show error message  “Incorrect Credentials” | It showed error message  “Incorrect Credentials” | Pass |
| UP2 | User | User Login | Validate User Login | 1)Enter  User  username 2)Enter User password | 1)username 2)password | It must show error message  “Login Successful” | It showed error message  “Login Successful” | Pass |
| UP3 | User | User  Registration | Validate  User  Registration | Enter invalid details | 1)First Name 2)Last Name 3)Username  4)password | Form should not be submitted, required fields would be shown wherever required. | Form was not submitted, required fields showed wherever required. | Pass |
| UP4 | User | User  Registration | Validate  User  Registration | Enter invalid details | password | It must show an message “please match the requested format” | It showed message “please match the requested format” | Pass |
| UP5 | User | User Registration | Validate  User  Registration | Enter invalid details | 1)First Name 2)Last Name 3)Username  4)password | It must show an info message  “Account signed up successfully” | It showed info message  “Account signed up successfully” | Pass |
| UP6 | User | Contact US | Validate  Contact details | Enter invalid details | 1)Name  2) Email id  3)Subject  4)Message | It should not be submitted, required fields would be shown wherever required. | It was not submitted, required fields showed wherever required. | Pass |
| UP7 | User | Contact US | Validate  Contact details | Enter invalid details | Email Id | It must show an message “please match the requested format” | It showed message “please match the requested format” | Pass |

**SCREENSHOTS.**



**BIBLIOGRAPHY & REFERENCES.**

* https://www.w3schools.com/
* https://www.php.net/manual/en/langref.php
* https://www.geeksforgeeks.org/
* https://www.tutorialspoint.com/index.html
* https://stackoverflow.com/