****

**Myanmar Institute of Theology**

**Liberal Arts Programs**

**Computer Department**

Systems Analysis and Design

***Final Project:***

under the topic of

**Hostel Management System**

*Prepared by*

| Allen Htoo | ID: MIT-LAP0859 |
| --- | --- |
| Aung Myat Kaung | ID: MIT-LAP0867 |
| Christopher Lone Toe | ID: MIT-LAP0886 |
|  |  |

*Presented to*

Saya Min Latt

Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Studies

September, 2023

# **Contents**

* Introduction
* Project Planning
  + Objective
  + Scope
  + Constraint
  + Estimate Development Cost(COCOMO)
  + Feasibility Analysis
    - Economic Feasibility
      * Benefits (Tangible and Intangible)
      * Costs (Recurring and One Time)
      * ROI (Return on Investment)
      * Break-even Analysis
    - Operational Feasibility
    - Technical Feasibility
    - Project Schedule
* Project Analysis and Design
  + Requirement Analysis
    - Functional Requirement
    - Non-Functional Requirement
  + New Project Design
    - Data Flow Diagram
      * Context Diagram
      * Level 0
      * Level 1
    - ERD
    - Data Definition
    - Business Logic Modelling
      * Structure English
* Input/output Design *(Screenshots of System)*
* Conclusion

**Introduction**

Hostel management systems make it easy for everyone involved in hostel operations. As an admin, you will have access to control and manage various aspects of the hostel including admin accounts, student management, room management, course management, and user access logs, allowing them to manage the hostel operations smoothly. On the user side, our system provides easy management of their own personal accounts, the ability to book a hostel room, access to room details, and a personal access log. Our goal is to simplify the hostel management process for both administrators and users, making it more efficient and user-friendly in one convenient place.

**Problem Statement**

Managing hostels is hard because there is no good system in place. We need a new computerized system that can help with managing students, rooms, courses, and logs. This new system should be easy to use, keep information safe, and help keep track of bookings, room details, and log access. With this new system, managing hostels will be easier and everyone involved (hostel staff and users) will have a better experience. It will take the operation of the Hostel by providing faster access to data and allowing management data, modification, and deletion of data in a systematic and reliable manner.

**Project Planning**

**Objectives**

The objective of the Hostel Management System is to provide an efficient and streamlined process for managing the day-to-day operations of a hostel. The system aims to automate the various tasks involved in managing the hostel such as managing room allocation, maintaining student records, managing staff and user information, and billing payment methods.

The Hostel Management System offers a well-organised, effective, and user-friendly platform that improves the overall experience for both students and staff and automates the hostel's management. Additionally, this solution can help reduce the workload of the hostel staff, allowing them to focus on delivering high-quality services to the students.

In summary, the objective of the Hostel Management System is to improve the management of a hostel by automating the various tasks involved, enhancing the experience of both the students and staff, and reducing the workload of the hostel staff.

**Project Scope**

Hostel Management System is designed for Hostels like small private schools and universities. That system will be used for maintaining the record of the student's information and room system. This project aims at automating the hostel management to work smoothly for all activities. Updating and modifying students' data will be easily achievable and all the calculations of the billing process will be more accurate.

**Constraints**

Time: The development of a basic Hostel Management System may take time from 4-10 months depending on the project's requirements and the complexity of the system.

**Estimate Development Cost (COCOMO)**

Budget: If we hire a team of 3-5 personnel depending on their skills and experience level, the hiring costs for a year will be around $25,000. Other things like software development tools, third-party software licenses, hardware requirements, and hiring personnel will cost over $29,000.

Hardware: A basic Hostel Management System may require a server, storage devices, and networking equipment. The total costs for hardware requirements can be from $3000 - $6000.

**Feasibility Analysis**

**1. Economic Feasibility**

**1.1 Benefits (Tangible and Intangible)**

**1.1.1 Tangible Benefit**

● Staff

● Paper Cost

● Accurate Billing errors

**1.1.2 Intangible Benefit**

● Smooth transaction

○ It is easier for students to transfer money without having to queue up in lines and effective for hostel staff to collect money without any intricate work, despite cyber security.

● Improve time management

○ A hostel management system can assist in simplifying the time-consuming task. Data can be accessed at any time and from any location, and it is completely error-free.

● Data Security

○ A Manual hostel management system would leave a variety of registration papers lying on the table, and could be accessed by anyone without any security. Digital hostel management system will provide cloud-based security and this system will help the administrators to set up different roles for users so that specific data is accessed by only authorized users.

● Improve general management

○ The hostel software allows the staff to dynamically allocate rooms and resources. Moreover, it will also be effective in time and effort management such as providing access at any time and from any location, and it is completely error-free. There is also improvement in managing and updating all the hoster records securely on the cloud.

● Automate Communication

○ It is excessively stressful for administrators to communicate admission-related issues and respond to students’ queries. By using the hostel management system, administrators can respond to queries and support students with instant notifications via the website and mobile devices.

**1.2 Costs (Recurring and One Time)**

**One Time Cost**

Development cost : $18,000 (Developer (3) - $18,000)

Hardware cost: $4,000

Training cost: $2,000

Total cost: $24,000

**Recurring Cost (Yearly)**

Hardware Maintenance cost: $2000

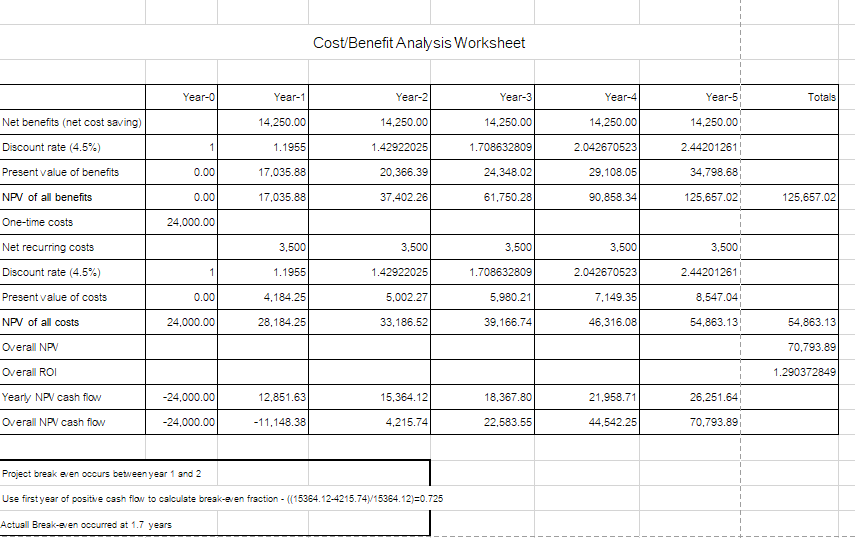
System Maintenance cost: $1000

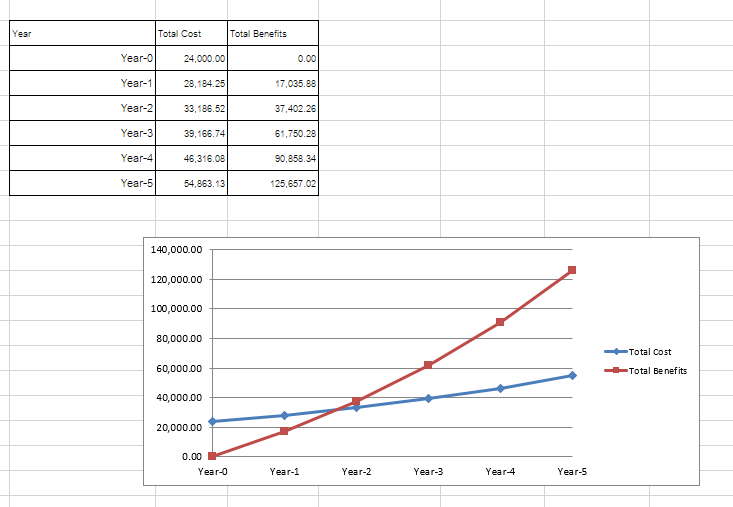
Backup cost: $500

Total cost: $3,500

**Tangible Benefits**

| No. | Benefits | Current System | New System | Tangle Benefits |
| --- | --- | --- | --- | --- |
| 1 | Staff  - Current system = 10  - New system = 5  - Salary per staff = 1500 | $12,000 | $6,000 |  |
| 2 | Paper Cost | $5,000 | $500 |  |
| 3 | Accurate Billing errors  ● Current System  -Guest per year = 250  -Average Booking per guest = $100  -Billing errors per year = 20%  ● New System  -Guest per year = 250  -Average Booking per guest = $100  -Billing errors per year = 5% | $5,000 | $1,250 |  |
|  | Total | $22,000 | $7,750 | = $14,250 |

**1.3 ROI (Return on Investment) & Break-even Analysis** 



**Operational Feasibility**

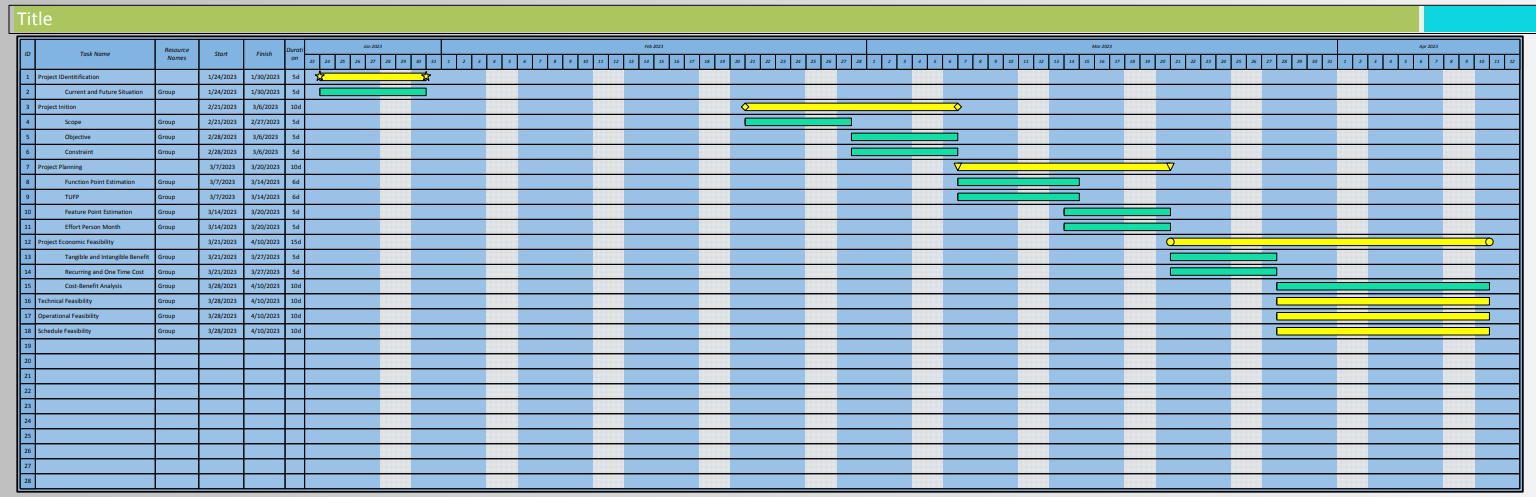
As we are living in the modern world, which is significantly portrayed as the “Era of Science”, it is really necessary to upgrade the way we behave, think, and the standards of our living. Nowadays, most manual jobs are being replaced by digital processing systems as one of the results of advanced computer programs and their powerful efficiency. Therefore, upgrading our hostel management process from a manual to a preprogrammed computerized system will not only make the process easier to handle but also ensure a more secure service.

Our hostel management system is developed in the simplest way and it doesn’t contain any intricate interface or process. That is why it would be easy to use even for people with little computer knowledge. Normally, when any new candidate system is introduced to an initial workflow, it requires special effort to educate, sell and train the staff on new ways to conduct it properly. As this project is a user-friendly version there is not much training required for people to use. Our product will not only make the task of administrator easy but also reduce the time consumption. Therefore, we would like to propose that this product is operationally feasible.

**Technical Feasibility**

A hostel management system is software that is designed to manage the daily operations of a hostel, including user reservation, room booking, check-in and check-out process, billing and payment processing, and reports. The feasibility of this system depends on several factors, such as the availability of hardware and software resources, the system’s scalability and reliability, security and data privacy requirements, and compatibility with the current system. This system has been developed using the PHP programming language, databases, and frameworks. In Addition, this system has been designed to integrate with other relevant systems, such as payment gateways, and to provide a seamless user experience.

**Project Schedule**



**Project Analysis and Design**

**1.1 Requirement Analysis**

**1.1.1 Functional Requirement**

* **All the manual processes of the system should be computerized.**
* **User**

1. Users should be allowed to register and log in to the system by themselves.
2. The system should allow users to update their account information and personal information.
3. The system should allow users to register bookings for hostels online.
4. The system should allow users to view their rooms’ details after booking.
5. The system should allow users to check their log-in access.

* **Admin**

1. The Admin should be able to register for users and manage users’ information.
2. The Admin should be able to upgrade the admin account email and password.
3. The Admin should be able to view and manage bookings by students, track their progress, and user access logs.
4. The Admin should be able to add new courses and manage courses.
5. The Admin should be able to add new rooms and manage rooms.

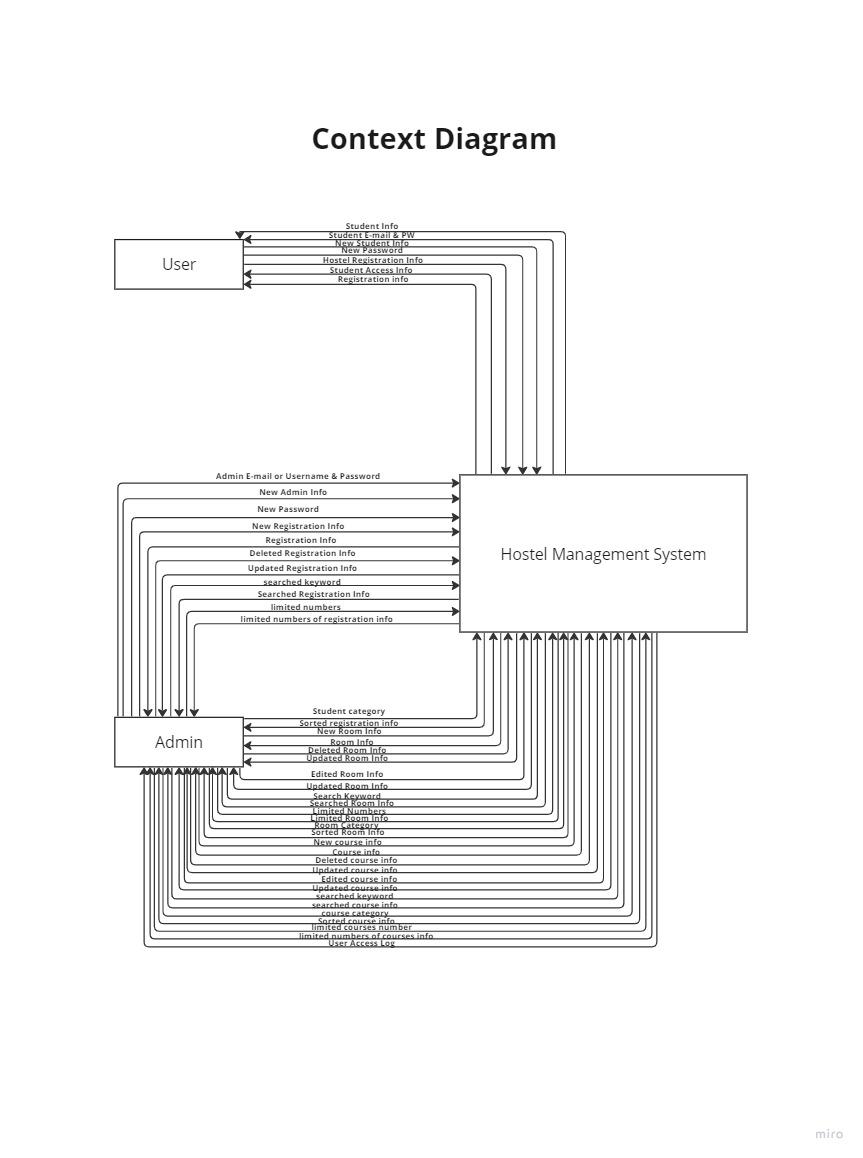
**1.1.2 Non-Functional Requirement**

1. **Performance:** The system should be able to handle a large amount of data and process requests efficiently to ensure quick response times. It should be capable of handling concurrent users without significant performance degradation.
2. **Scalability:** The system should be scalable to accommodate an increasing number of hostels, rooms, and users as the organization grows.
3. **Reliability:** The system should be reliable and available for use at all times.
4. **Security:** The system should ensure the confidentiality, integrity, and availability of data.
5. **Usability:** The system should have an intuitive user interface that is easy to navigate and use. It should provide clear and concise error messages, helpful prompts, and appropriate feedback to guide users through the system.
6. **Disaster Recovery:** The system should have backup and disaster recovery mechanisms to protect against data loss or system failures. It should include regular backups and procedures for data restoration in the event of an unforeseen incident.

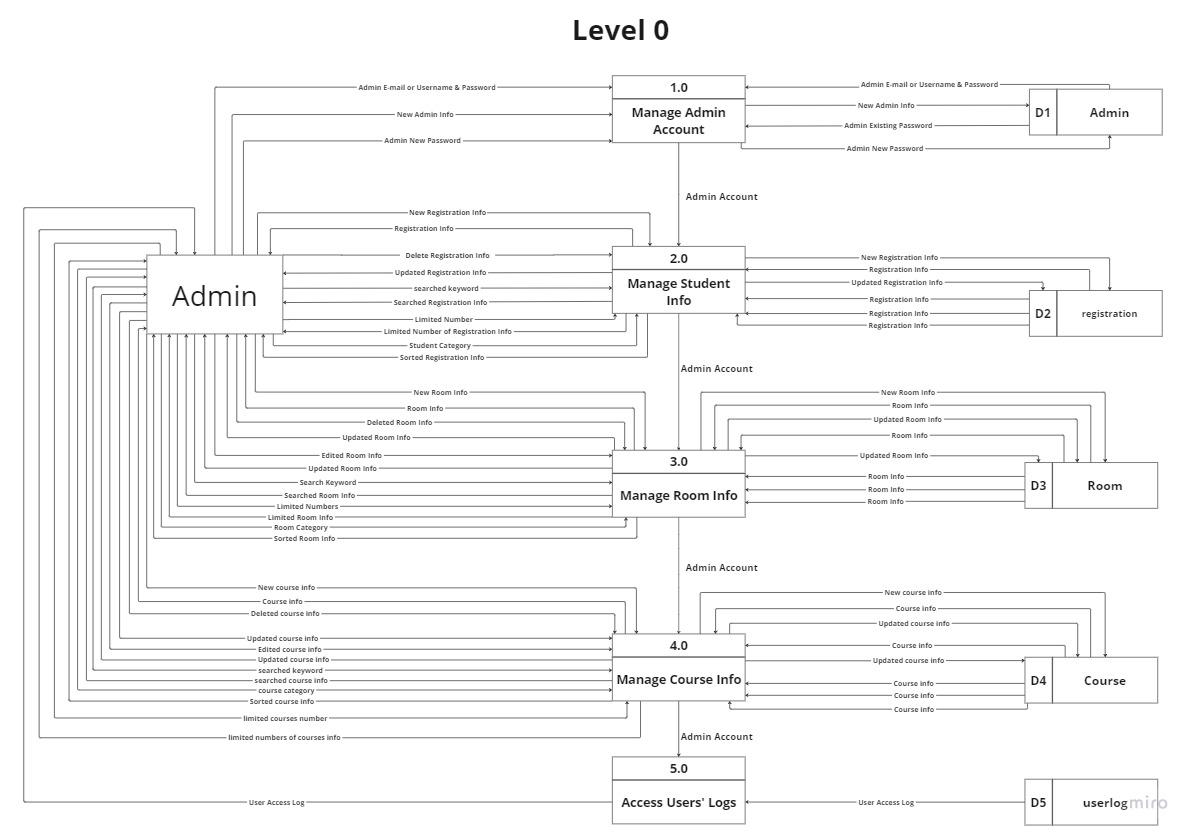
**New Project Design**

**1.1 Data Flow Diagram**

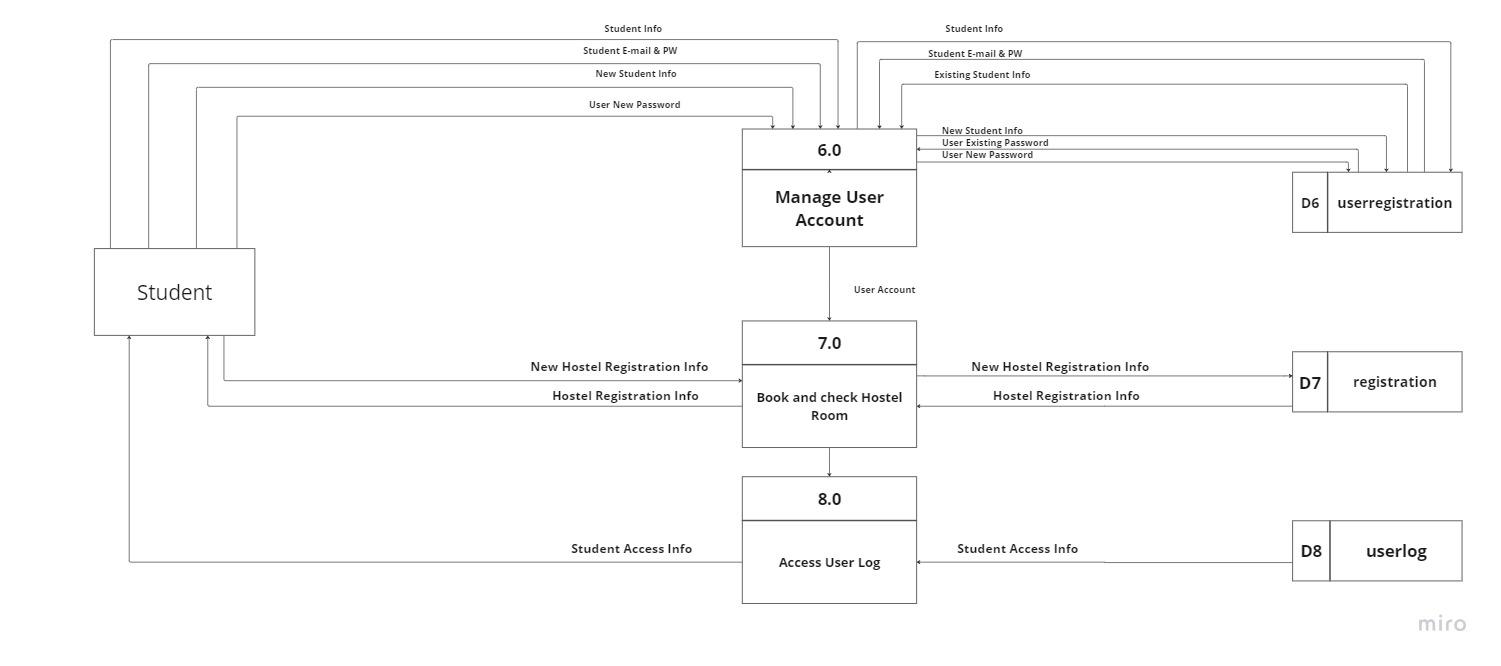
* **Context Diagram**



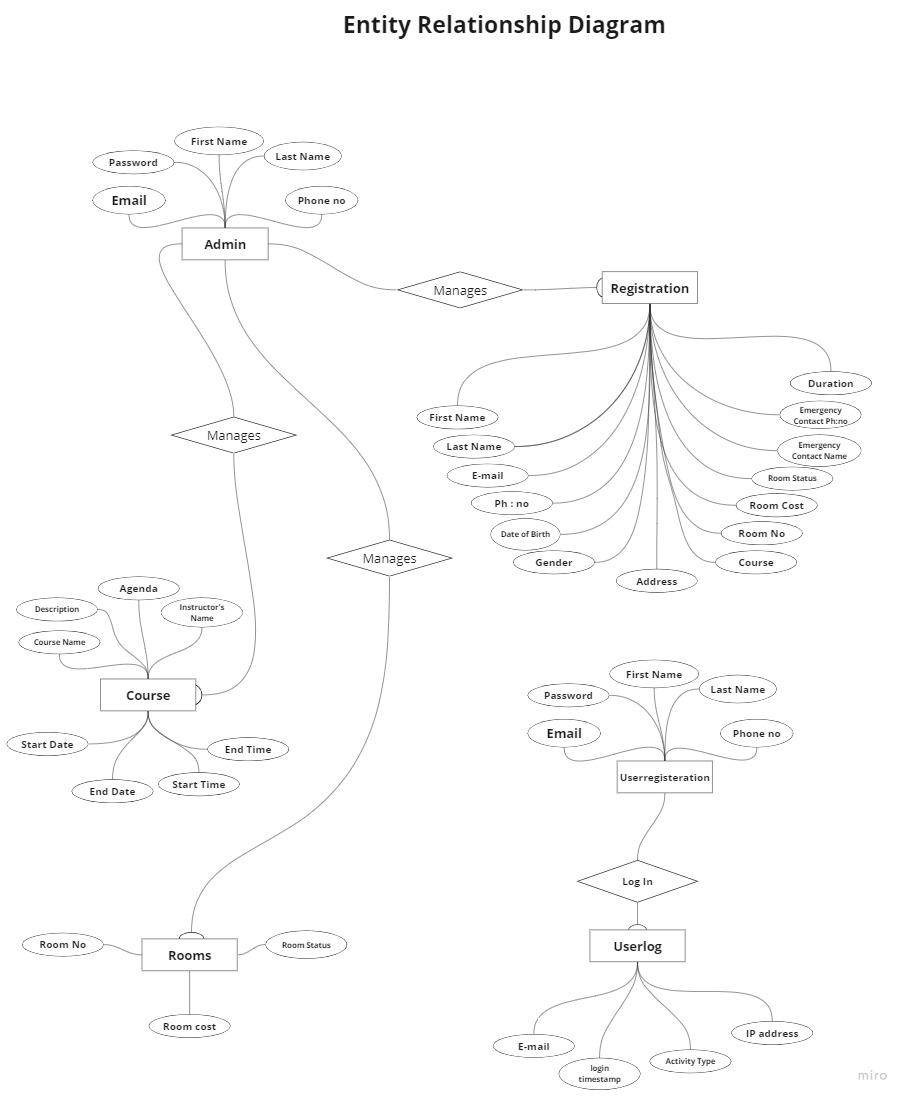
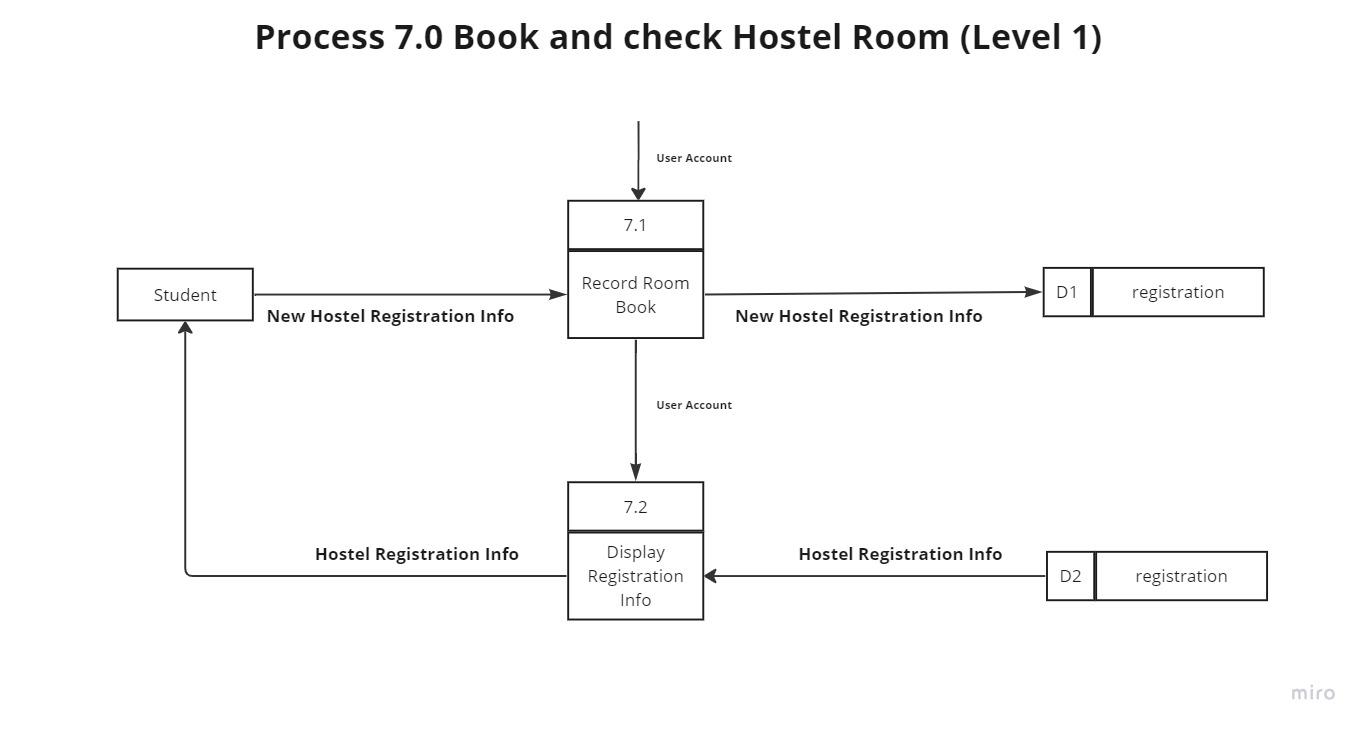
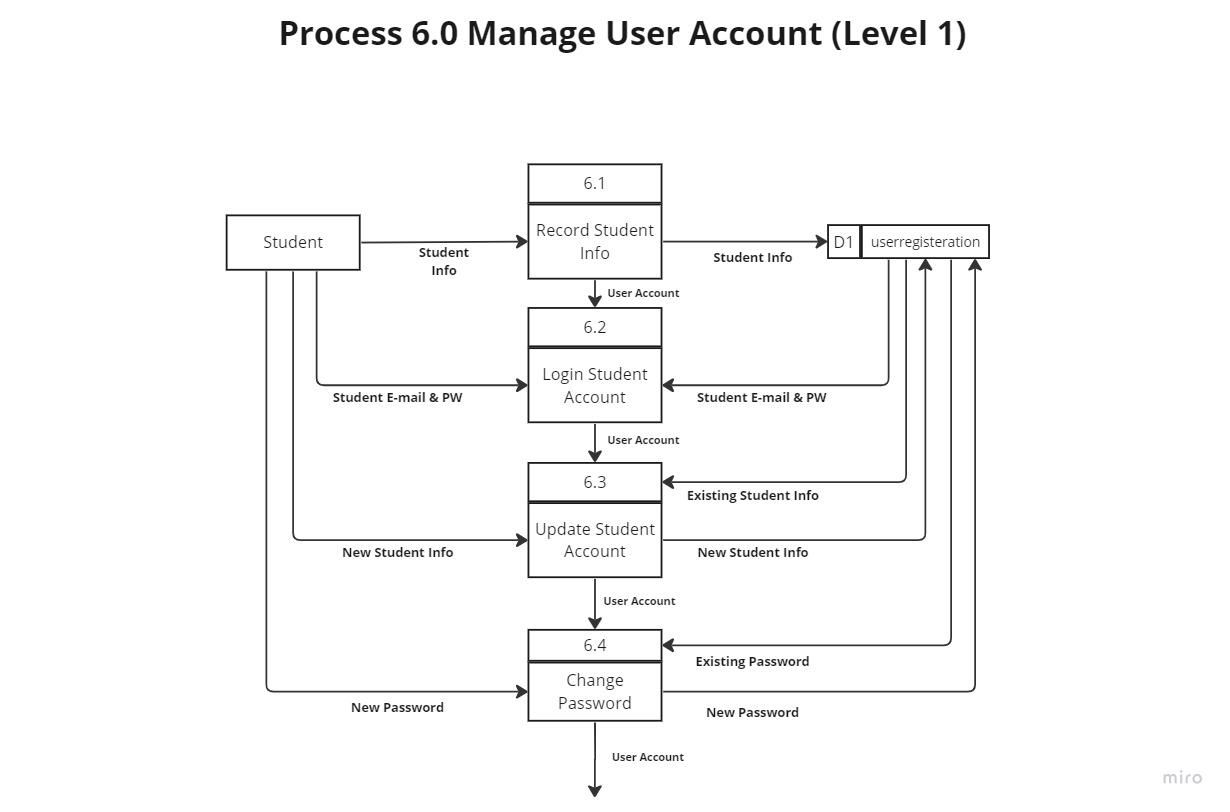
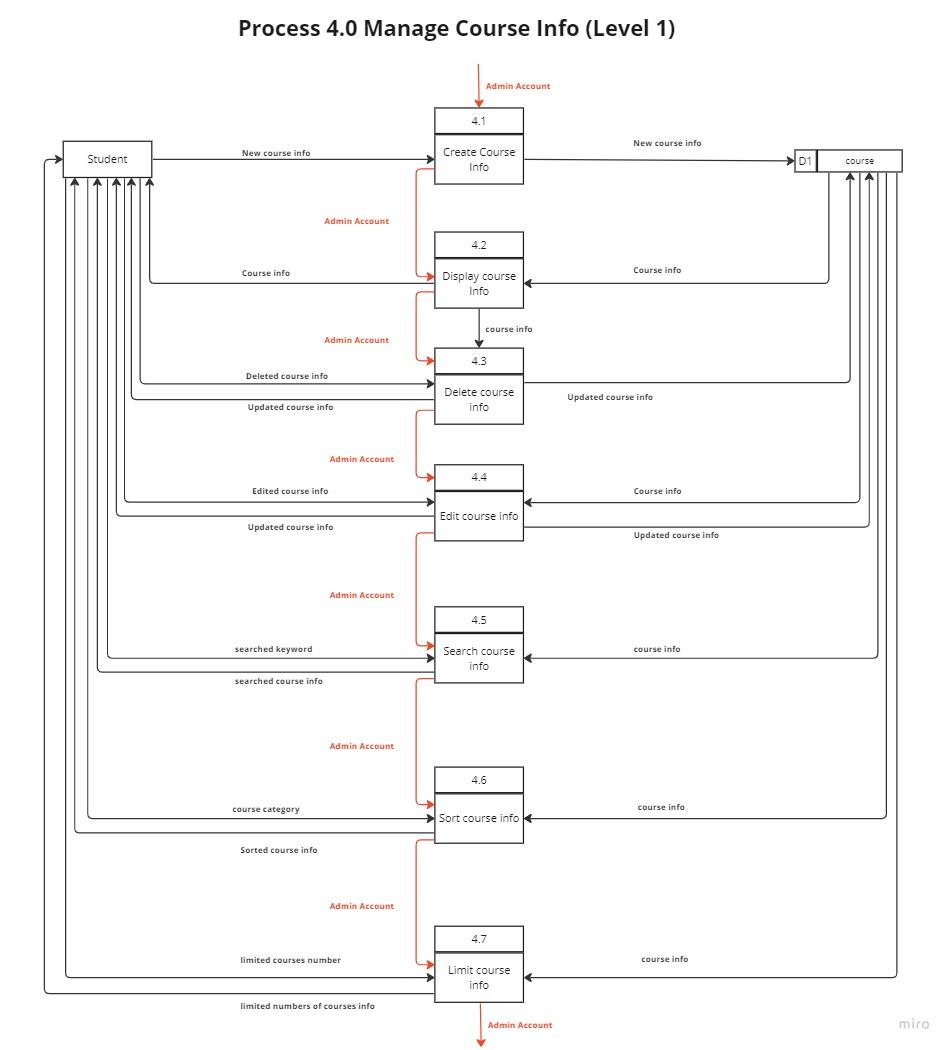
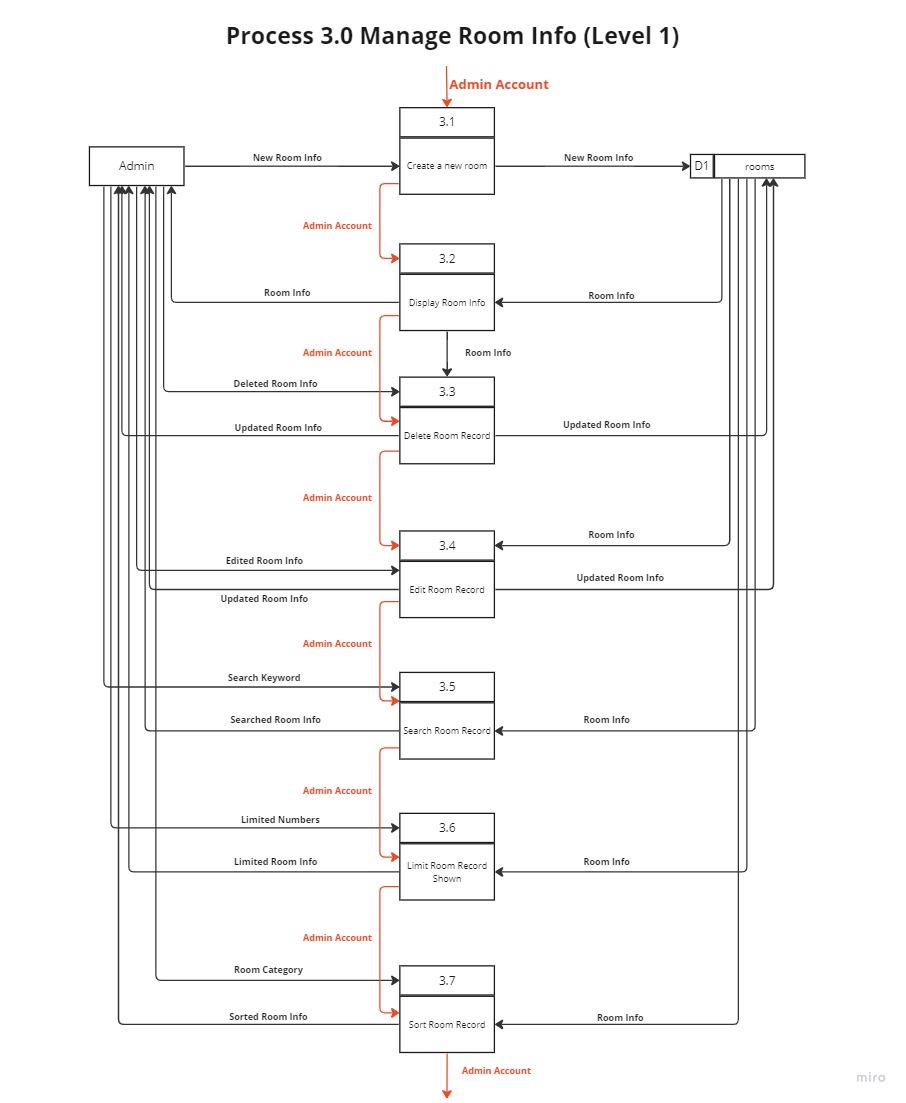
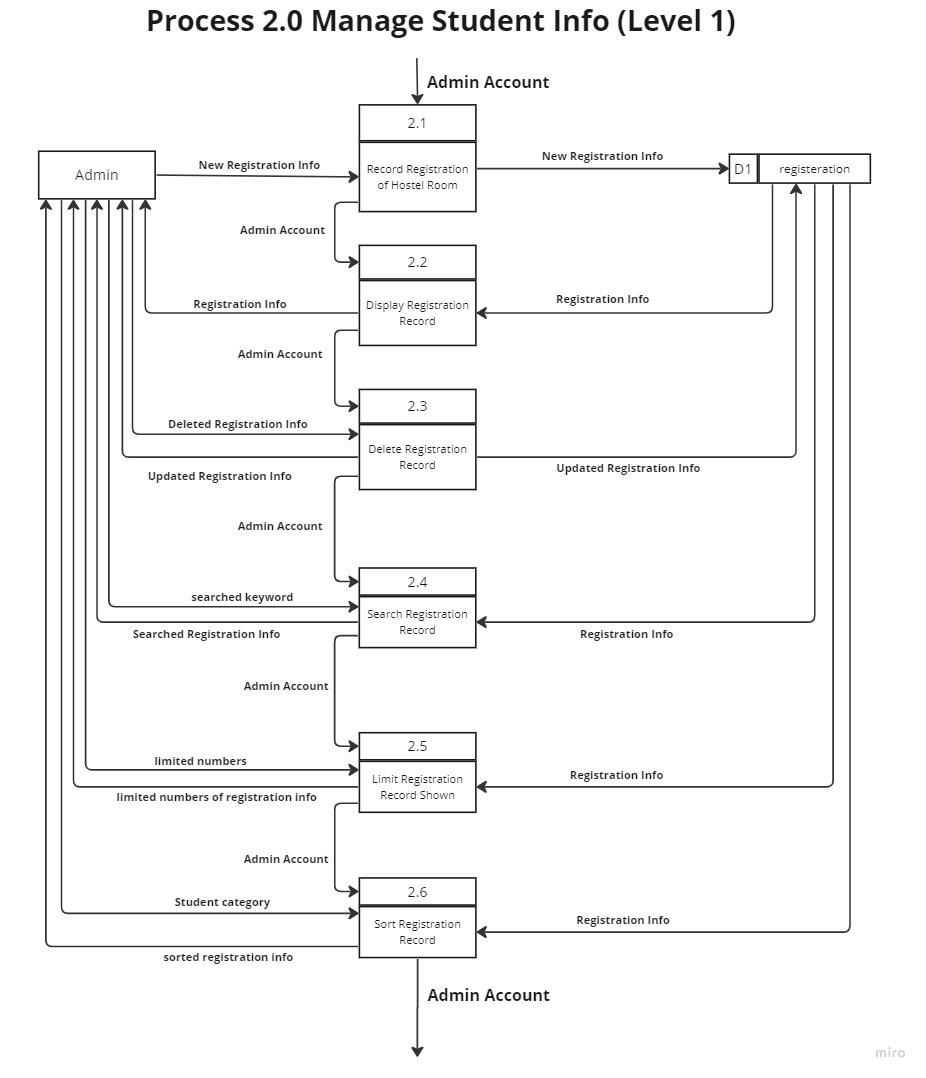
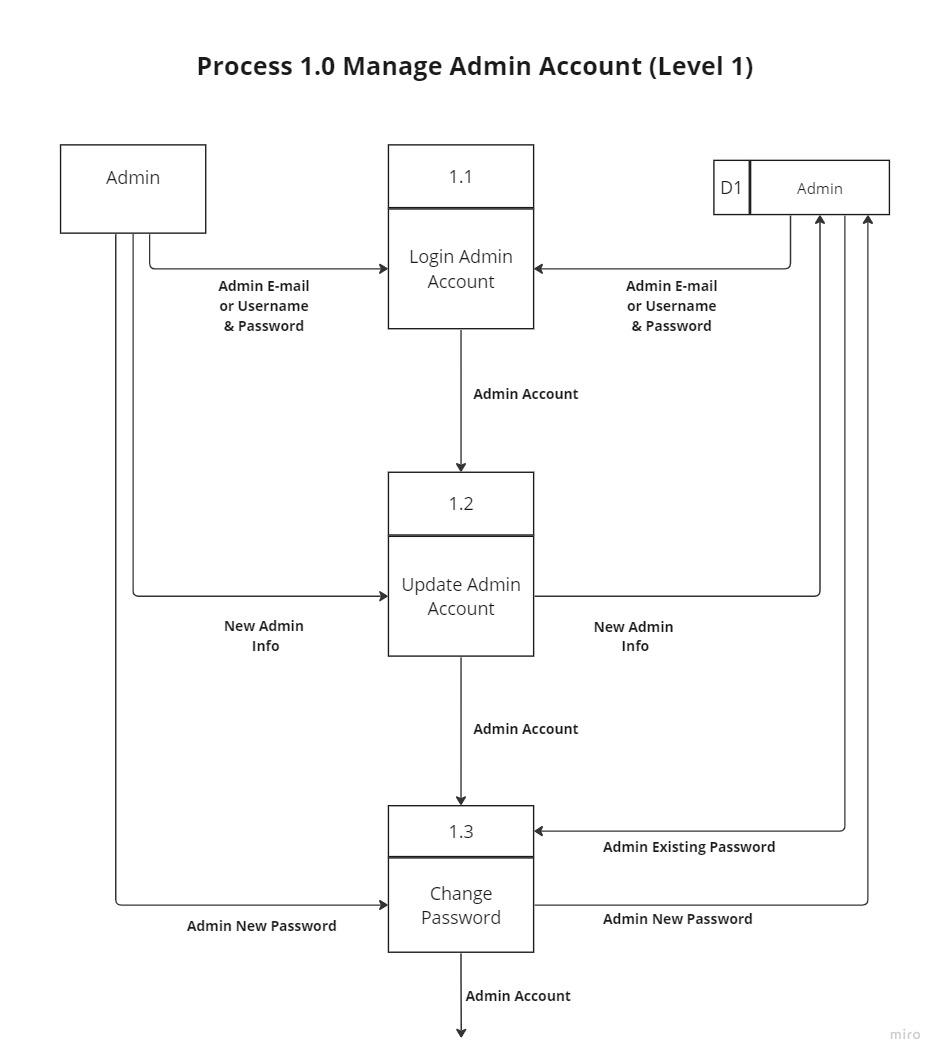
* **Admin Level 0**



* **User Level 0**

****

* **Level 1**

****

**Data Definition**

| Entity Name | Data | Data Definition |
| --- | --- | --- |
| Admin | Admin E-mail or Username & Password | - Admin Email  - Admin Password |
| New Admin Info | - Admin First Name  - Admin Last Name  - Admin Email  - Admin Password  - Admin Phone No. |
| Admin New Password | - Admin Password |
| registration | New Registration Info,  Registration Info,  Delete Registration Info,  Searched Registration Info,  Updated Registration Info,  Hostel Registration Info, | - Student First Name  - Student Last Name  - Student Email  - Student Phone No.  - Student Address  - Student Date of Birth  - Student Gender  - Student Course  - Student Duration  - Student Room No.  - Student Room Status  - Student Room Cost  - Student Food Status  - Student Check In Date  - Student Emergency Contact Name  - Student Emergency Contact  - Student Phone No. |
| Limited Number of Registration Info,  Sorted Registration Info | - Student First Name  - Student Last Name  - Student Email  - Student Course  - Student Room No.  - Student Room Cost  - Student Check In Date |
| room | New Room Info,  Room Info,  Deleted Room Info,  Updated Room Info,  Edited Room Info,  Updated Room Info,  Limited Number of Registration Info,  Sorted Registration Info | - Room No.  - Limited No. of People  - No. of People Staying  - Room Status  - Room Cost |
| Course | New Course Info,  Course Info,  Deleted course info,  Updated Course Info,  Edited course info, | - Course Name  - Description  - Agenda  - Limited No. of People  - No. of People Attending  - Instructor’s Name  - Start Time  - End Time  - Start Date  - End Date |
| sorted course info,  limited numbers of courses info | - Course Name  - No. of People Attending  - Instructor’s Name  - Start Date  - End Date |
| userlog | User Access Log | - Email  - Login Timestamp  - Activity Type  - IP Address |
|  | Student Access Log | - Student Email  - Student Login Timestamp  - Student Activity Type  - Student IP Address |
| userregistration | Student Info,  New Student Info, | - Student Email  - Student Password  - Student First Name  - Student Last Name  - Student Phone No. |
|  | Student E-mail & PW | - Student Email  - Student Password |
|  | User New Password | - Student Password |

**Business Logic Modelling**

* **Structure English**

**Process 1.1 Login Admin Account**

DO

GET Admin Email or Username and Password from User.

FIND Admin Email or Username and Password in Admin Data File.

BEGIN IF

IF Admin Email or UserName and Password are in Admin Data File.

THEN

GRANT Login Access.

ELSE SHOW Error Message

END IF

UNTIL END of File

**Process 1.2 Update Admin Account**

DO

READ Admin Information in Admin Data File.

GET New Admin Information from User.

UPDATE New Admin Information in Admin Data File.

UNTIL END of File

**Process 1.3 Change Password**

DO

READ Admin Existing Password in Admin Data File.

GET Admin New Password from User.

UPDATE Admin New Password in Admin Data File.

UNTIL END of File

**Process 2.1 Record Registration of Hostel Room**

DO

GET New Registration Info from User.

RECORD New Registration Info in Registration Data File.

UNTIL END of File

**Process 2.2 Display Registration Record**

DO

READ Registration Info from Registration Data File

SHOW Registration Info.

UNTIL END of File

**Process 2.3 Delete Registration Record**

DO

GET Deleted Registration Information to delete from User.

DELETE Deleted Registration Information in Registration Data File.

UPDATE Updated Registration Information from Registration Data File.

SHOW Updated Registration Information from Registration Data File.

UNTIL END of File

**Process 2.4 Search Registration Record**

DO

READ Registration Information from Registration Data File.

GET Searched Keyword From User.

FIND matching Registration Information

SHOW matched Registration Information

UNTIL END of File

**Process 2.5 Limit Registration Record Shown**

DO

READ Registration Information from Registration Data File.

GET Limited Numbers From User.

LIMIT numbers of Registration Info shown.

SHOW Registration Information

UNTIL END of File

**Process 2.6 Sort Registration Record**

DO

READ Registration Information from Registration Data File.

GET Student Category From User.

SORT Registration Information by Student Category.  
 SHOW Registration Information

UNTIL END of File

**Process 3.1 Create A New Room**

Do

GET New Room Info From User

STORE New Room Info in Rooms Data File

UNTIL END of File

**Process 3.2 Display Room Info**

Do

READ Room Info From Rooms Data File

SHOW Room Info.

UNTIL END of File

**Process 3.3 Delete Room Record**

Do

GET Deleted Room Information to delete from User.

DELETE Deleted Room Information in Rooms Data File.

UPDATE Updated Room Information from Rooms Data File.

SHOW Updated Room Information from Rooms Data File.

UNTIL END of File

**Process 3.4 Edit Room Record**

Do

READ Room Information From Rooms Data File.

EDIT Edited Room Information in Rooms Data File

UPDATE Updated Room Information from Rooms Data File.

SHOW Updated Room Information from Rooms Data File.

UNTIL END of File

**Process 3.5 Search Room Record**

Do

READ Room Information From Rooms Data File.

GET Searched Keyword From User.

FIND matching Room Information

SHOW matched Room Information

UNTIL END of File

**Process 3.6 Limit Room Record Shown**

Do

READ Room Information From Rooms Data File.

GET Limited Numbers From User.

LIMIT numbers of Room Info shown.

SHOW Room Information

UNTIL END of File

**Process 3.7 Sort Room Record**

Do

READ Room Information From Rooms Data File.

GET Room Category From User.

SORT Room Information by Room Category.  
 SHOW Room Information

UNTIL END of File

**Process 4.1 Create Course Info**

Do

GET New Course Info From User

STORE New Course Info in Course Data File

UNTIL END of File

**Process 4.2 Display Course Info**

Do

READ Course Info From Course Data File

SHOW Course Info.

UNTIL END of File

**Process 4.3 Delete Course Info**

Do

GET Deleted Course Information to delete from User.

DELETE Deleted Course Information in Course Data File.

UPDATE Updated Course Information from Course Data File.

SHOW Updated Course Information from Course Data File.

UNTIL END of File

**Process 4.4 Edit Course Info**

Do

READ Course Information From Course Data File.

EDIT Edited Course Information in Course Data File

UPDATE Updated Course Information from Course Data File.

SHOW Updated Course Information from Course Data File.

UNTIL END of File

**Process 4.5 Search Course Info**

Do

READ Course Information From Course Data File.

GET Searched Keyword From User.

FIND matching Course Information

SHOW matched Course Information

UNTIL END of File

**Process 4.6 Sort Course Info**

Do

READ Course Information From Course Data File.

GET Course Category From User.

SORT Course Information by Course Category.  
 SHOW Course Information

UNTIL END of File

**Process 4.7 Limit Course Info**

Do

READ Course Information From Course Data File.

GET Limited Numbers From User.

LIMIT numbers of Course Info shown.

SHOW Course Information

UNTIL END of File

**Process 6.1 Record Student Info**

Do

GET Student Info from User.

RECORD Student Info in User Registration Data File.

UNTIL END of File

**Process 6.2 Login Student Account**

Do

GET Student Email and Password from User.

FIND Student Email and Password in User Registration Data File.

BEGIN IF

IF Student Email and Password are in User Registration Data File.

THEN

GRANT Login Access.

ELSE SHOW Error Message

END IF

UNTIL END of File

**Process 6.3 Update Student Account**

Do

READ Student Information in User Registration Data File.

GET New Student Information from User.

UPDATE New Student Information in User Registration Data File.

UNTIL END of File

**Process 6.4 Change Password**

Do

READ Student Existing Password in User Registration Data File.

GET Student New Password from User.

UPDATE Student New Password in User Registration Data File.

UNTIL END of File

**Process 7.1 Record Room Book**

DO

GET Hostel Registration Information from User.

RECORD Hostel Registration Information in Registration Data File.

UNTIL END of File

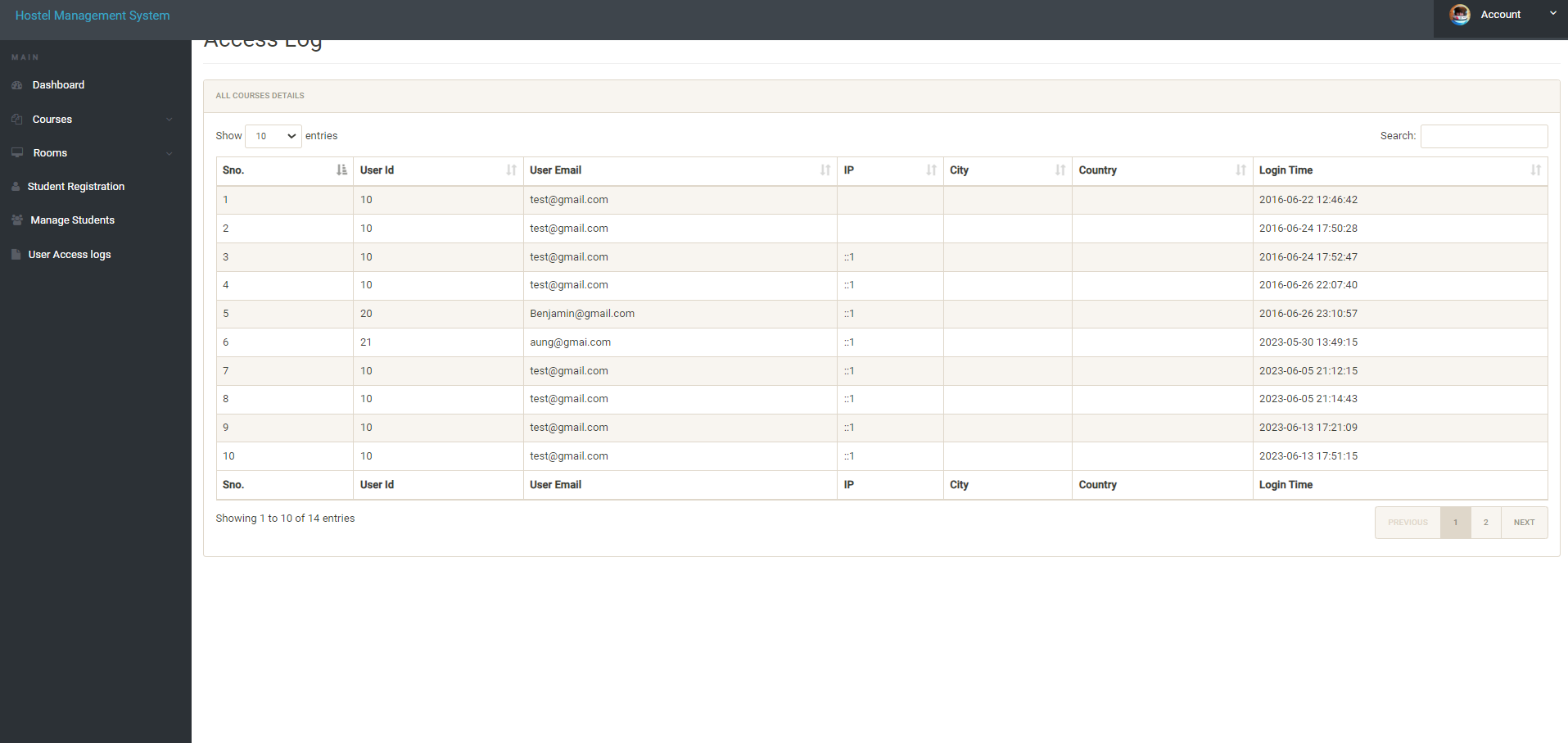
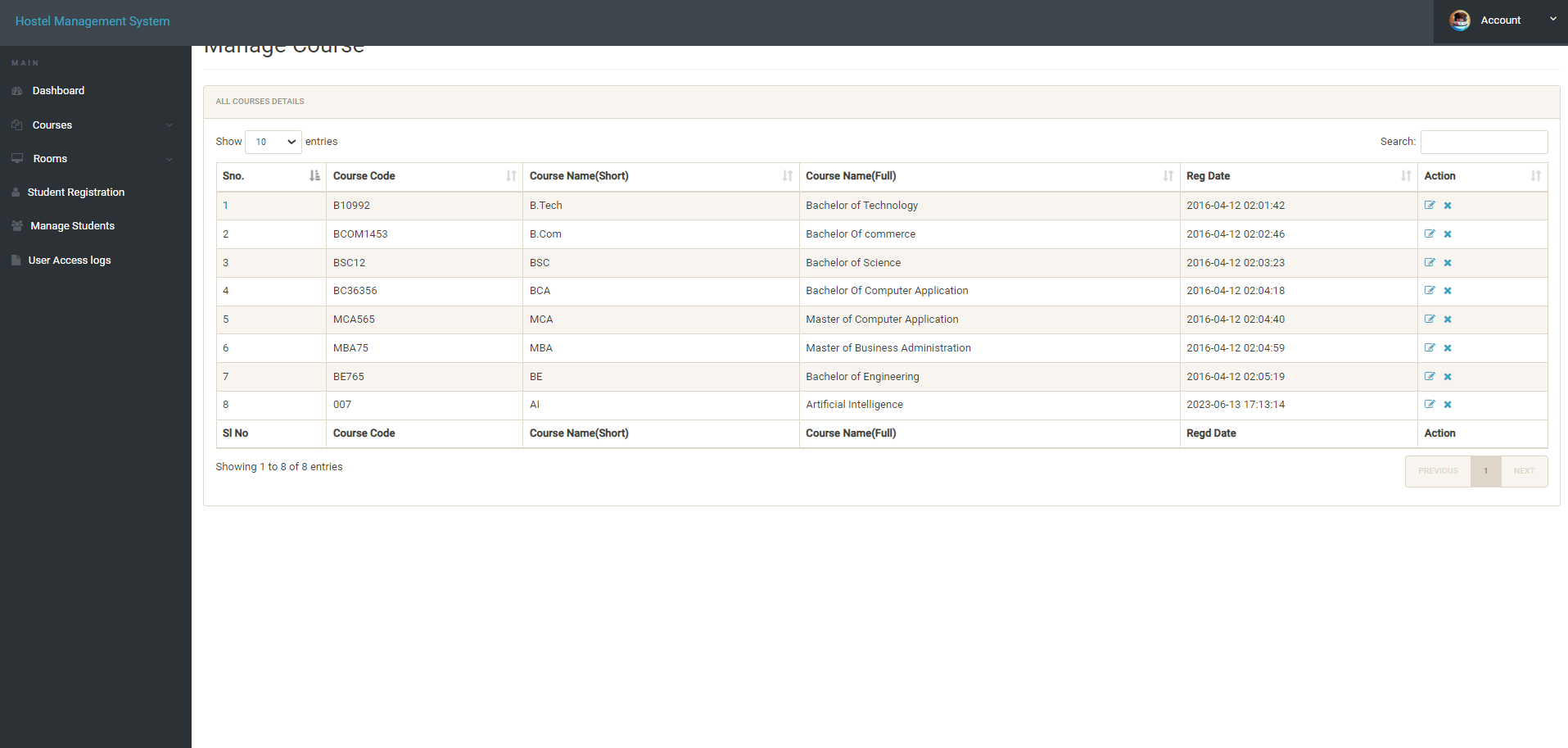
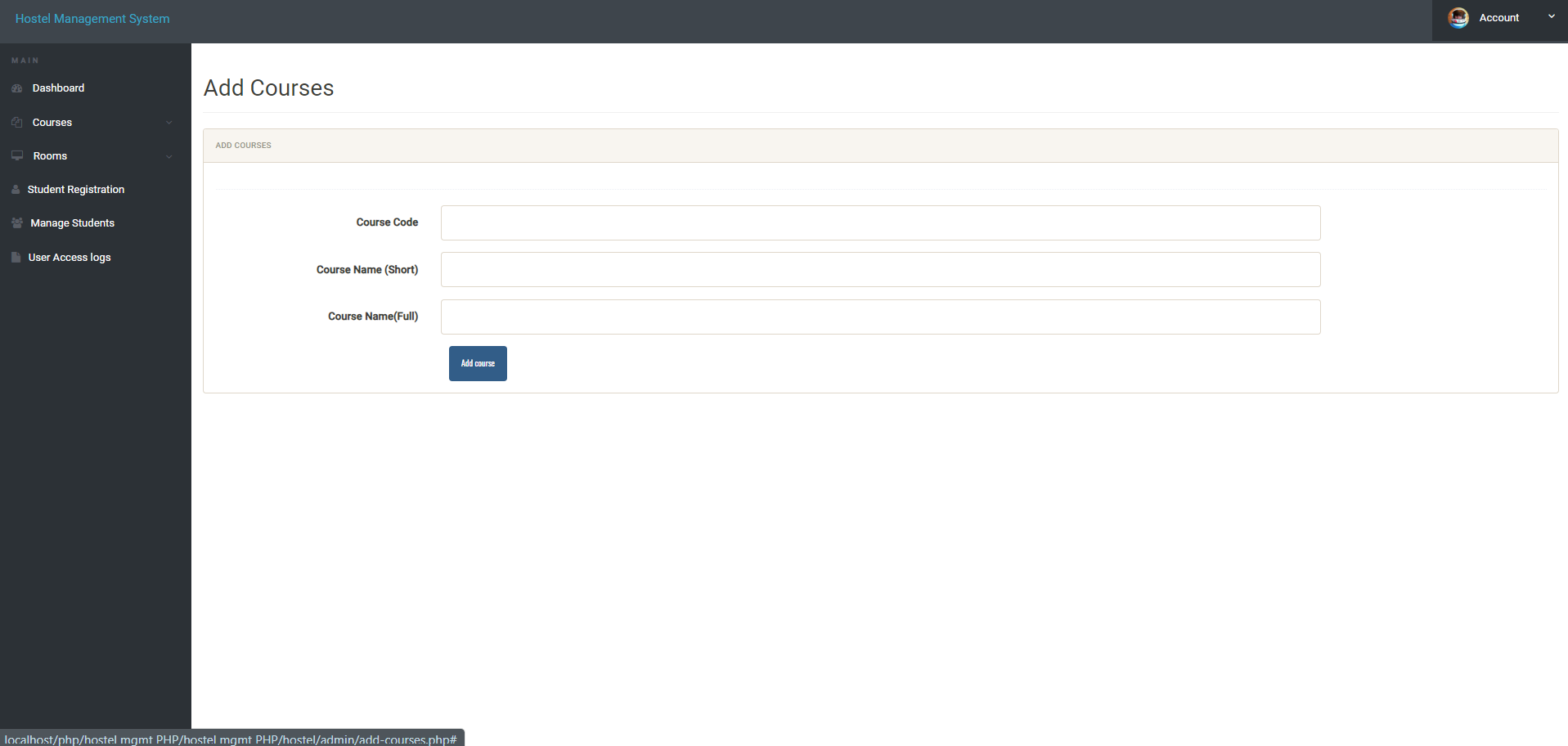
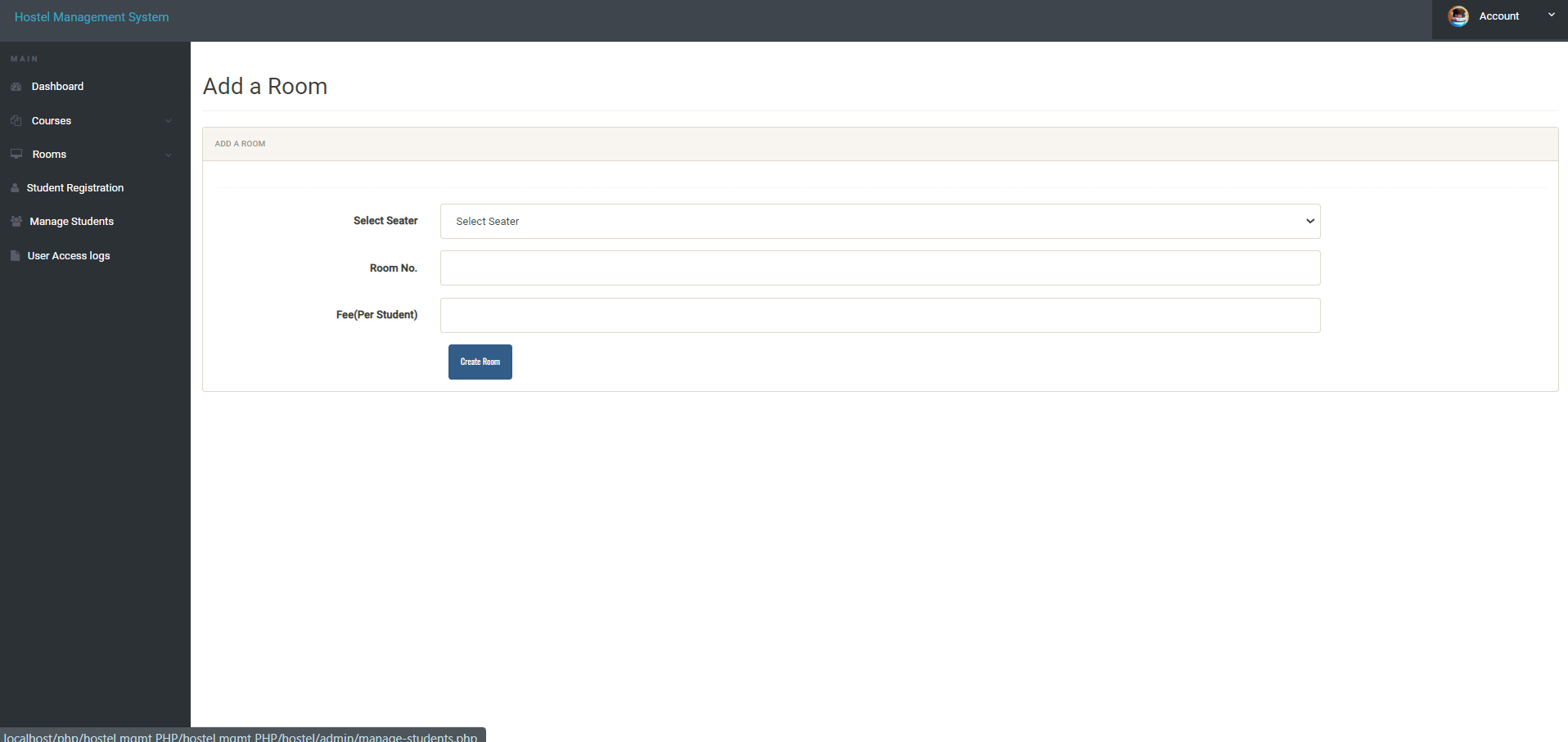
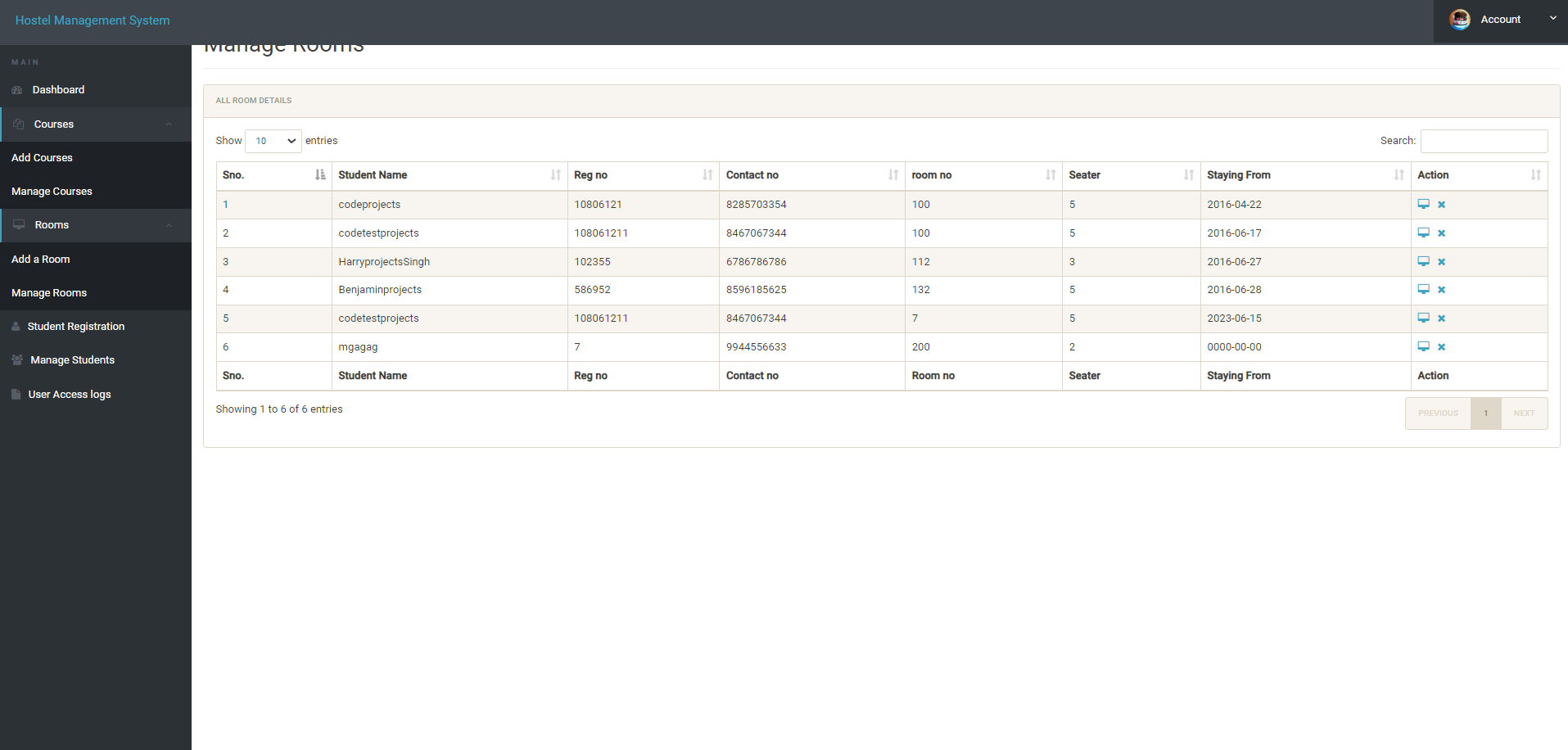
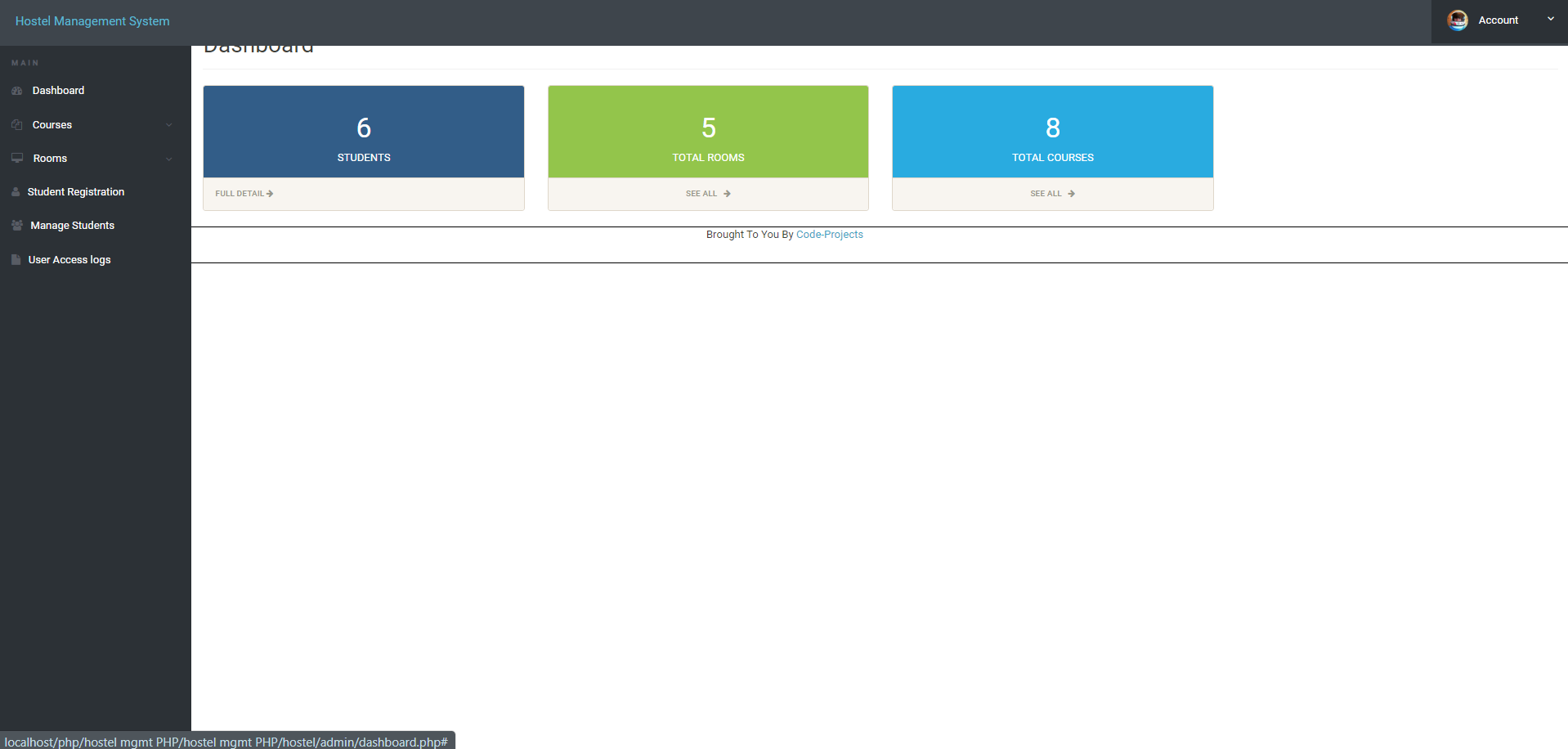
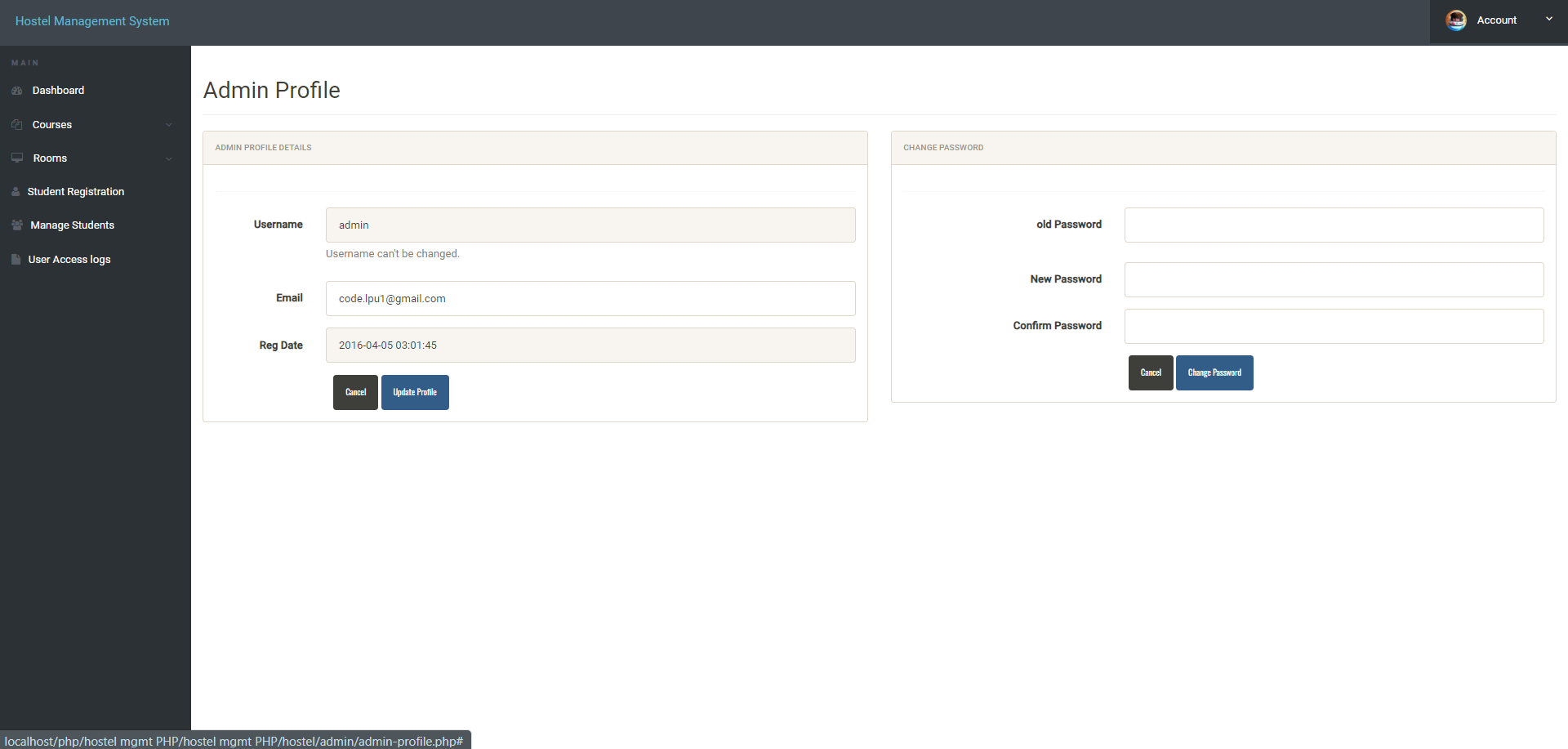
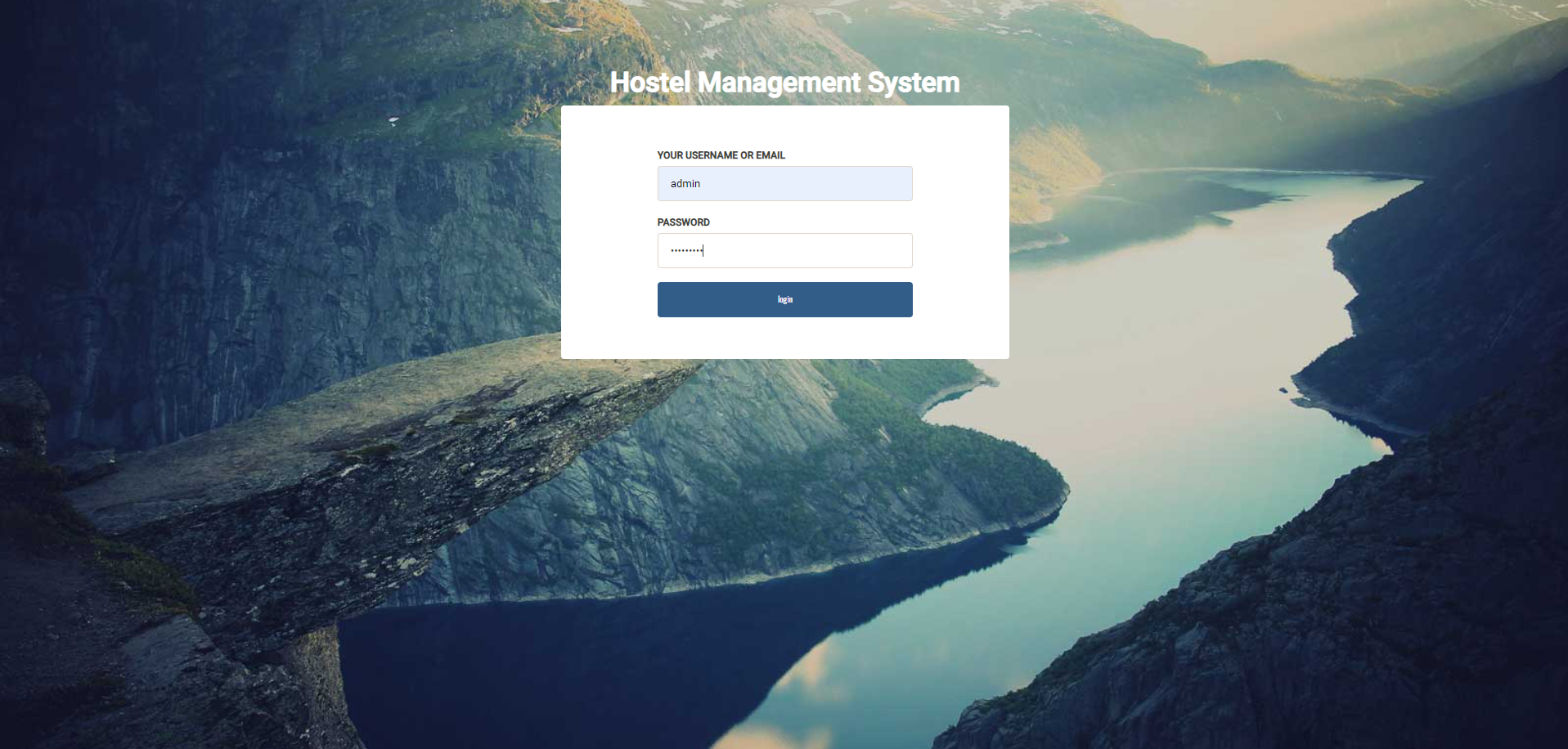
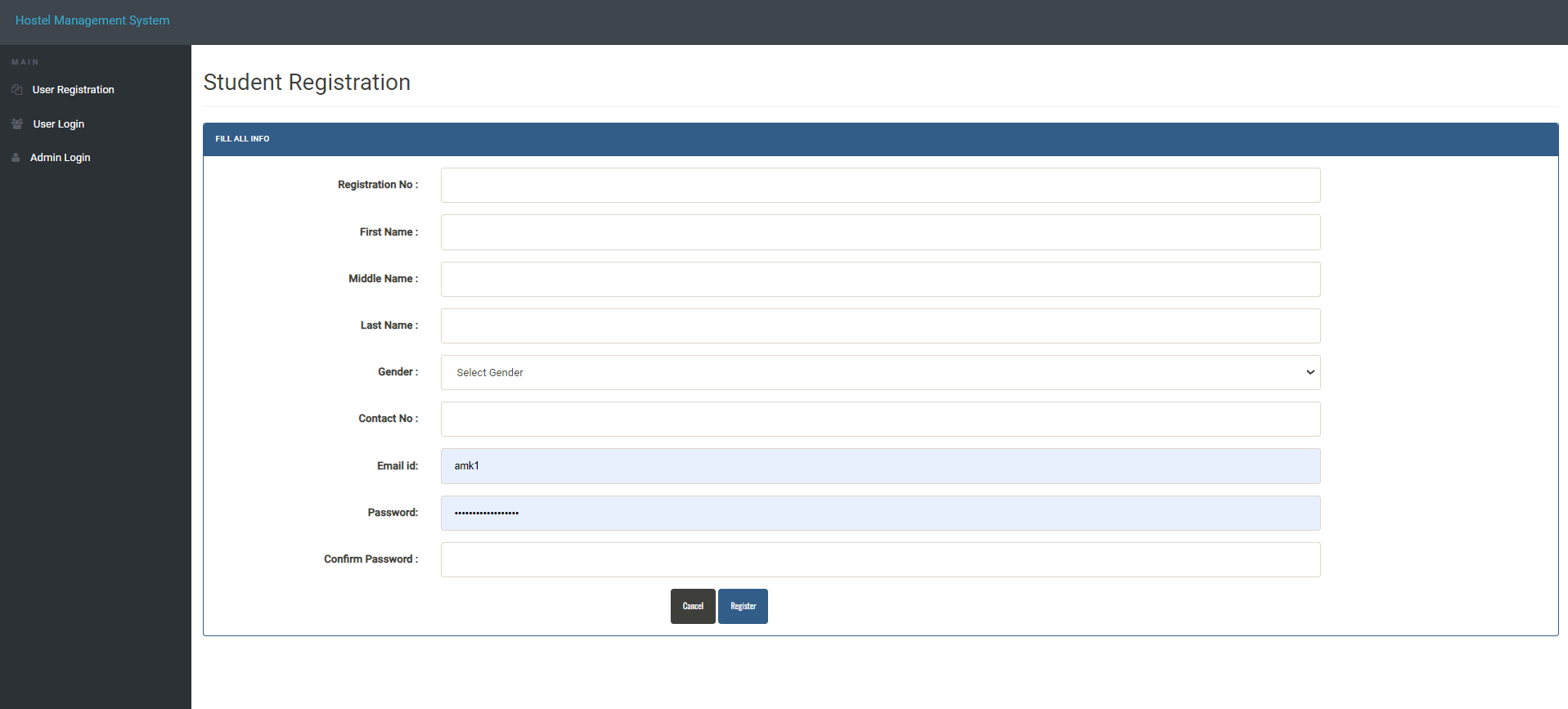
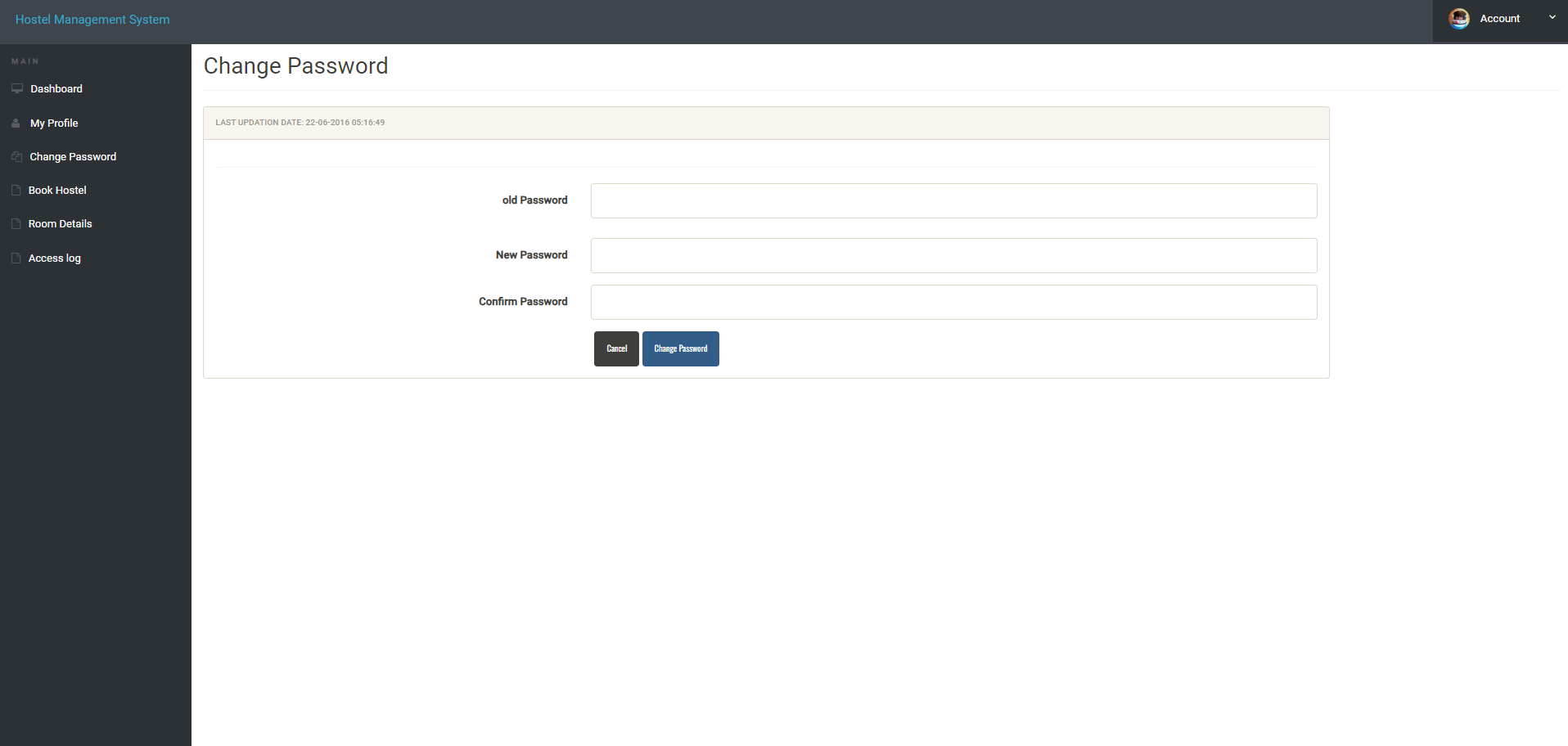
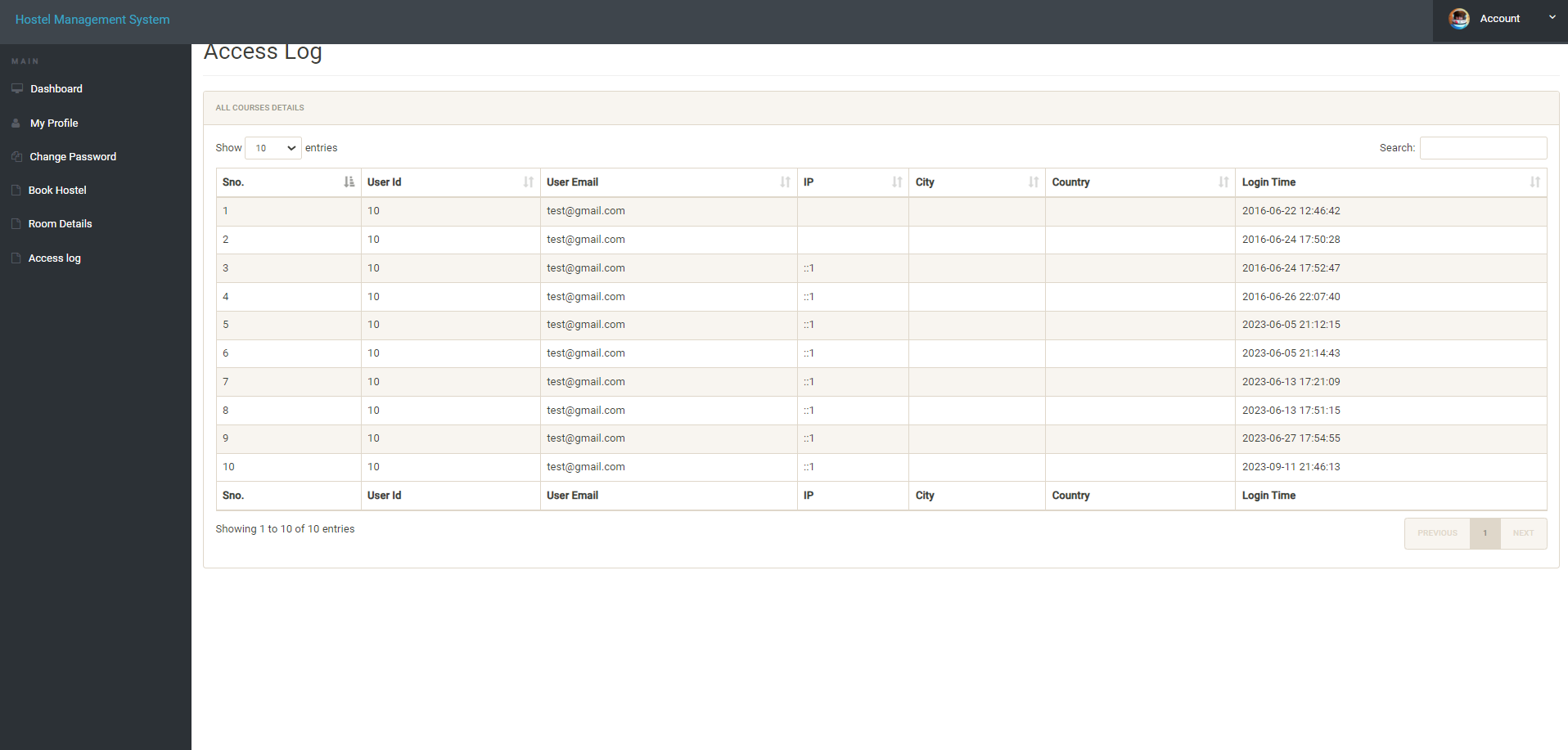
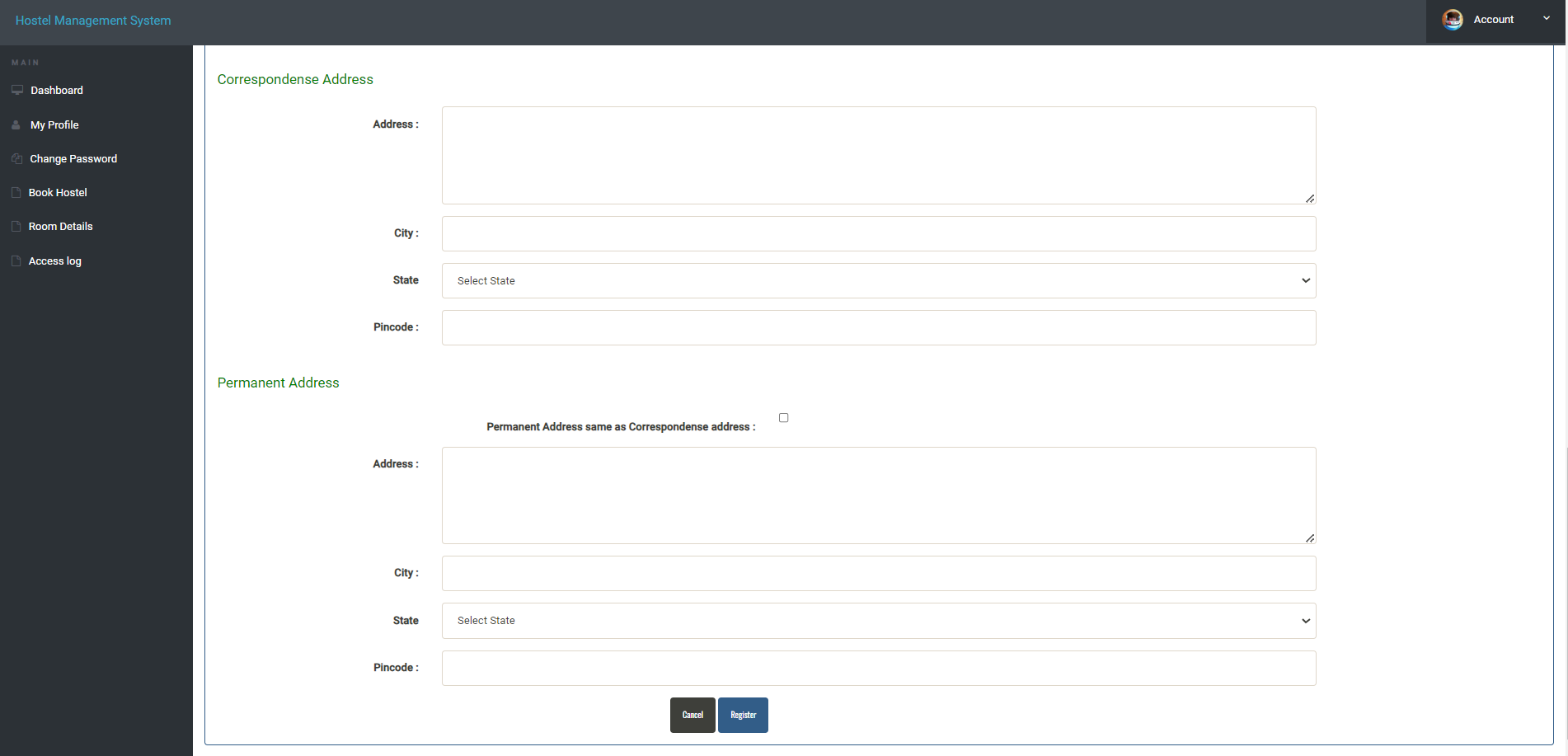
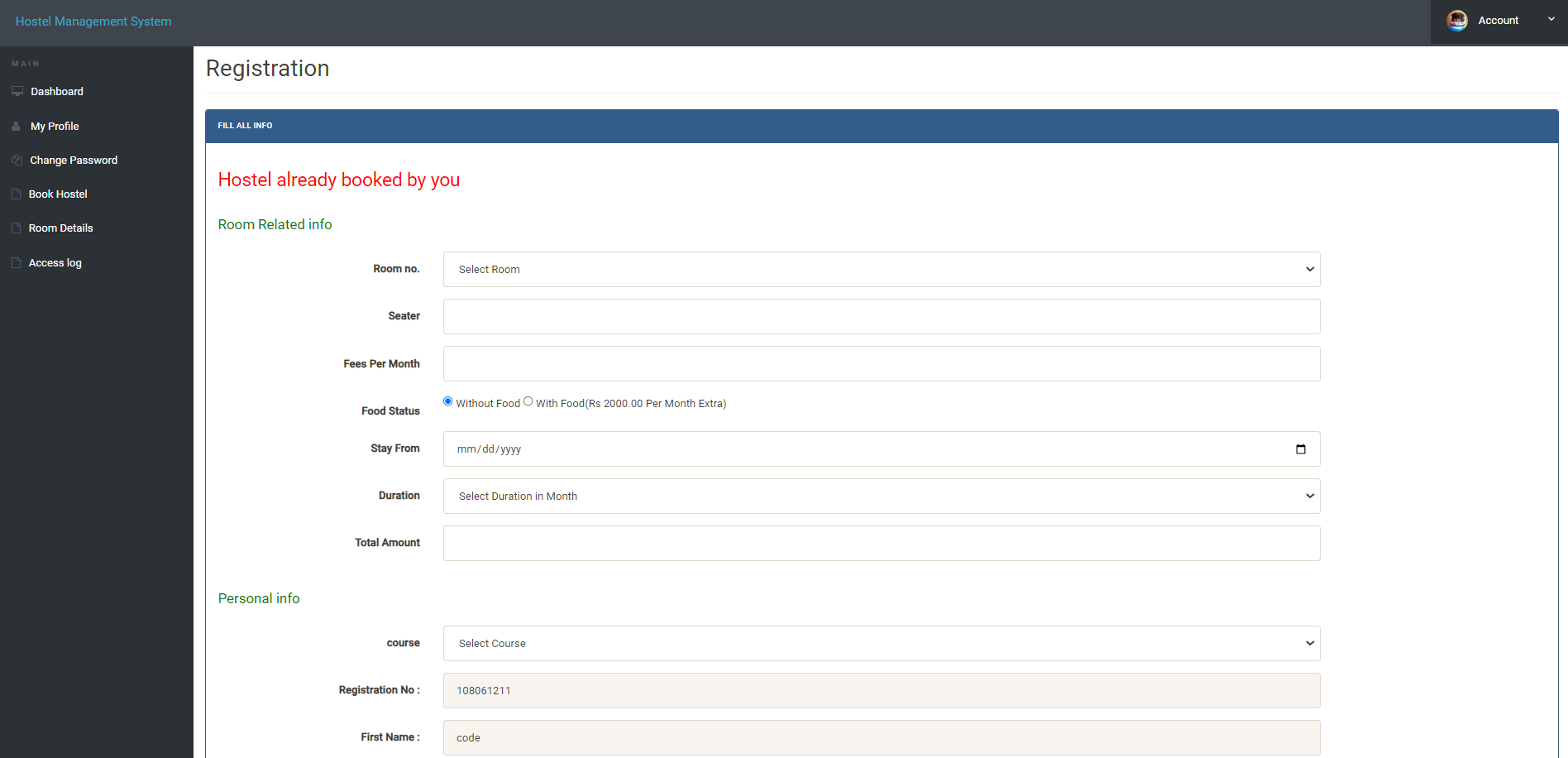
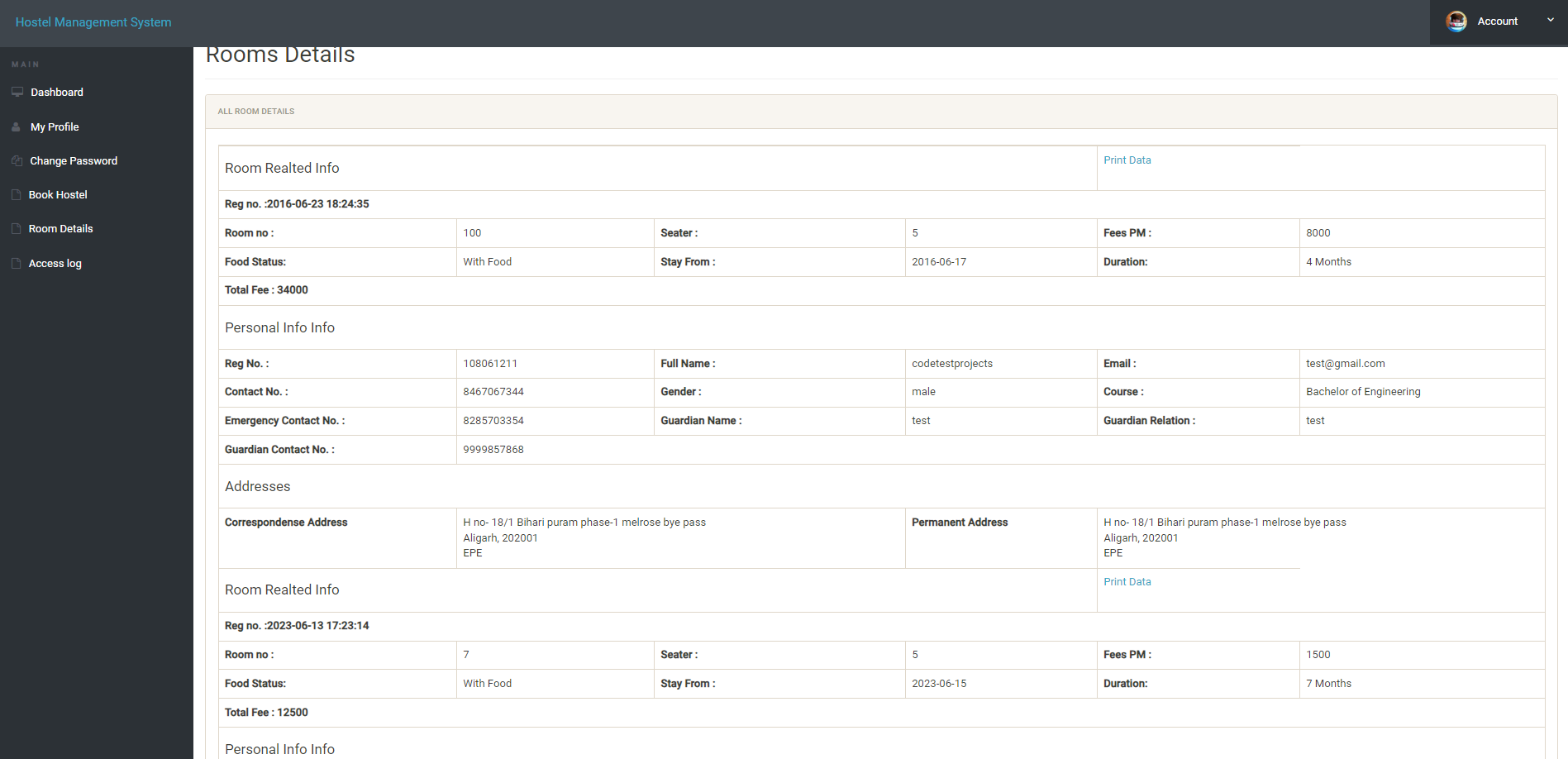
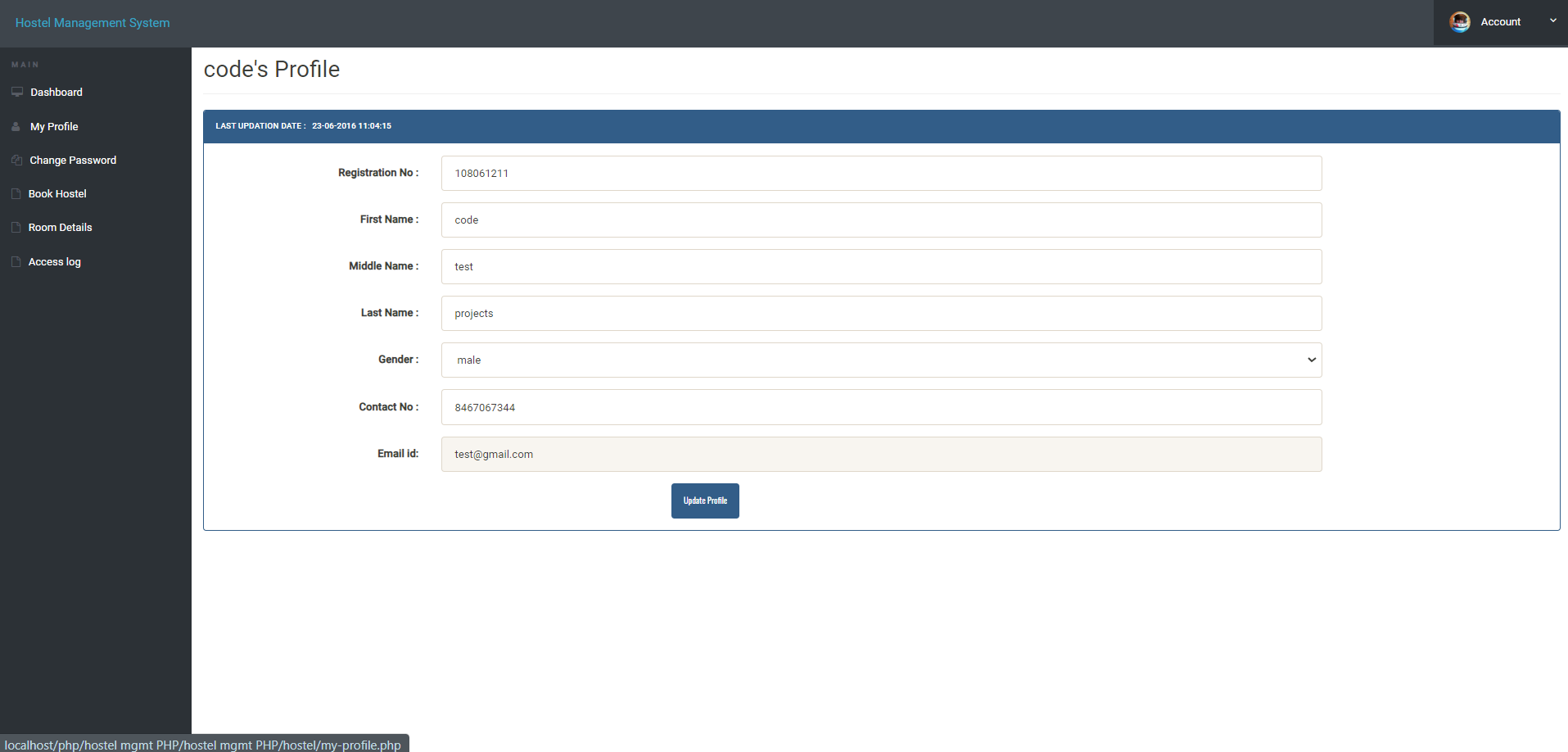
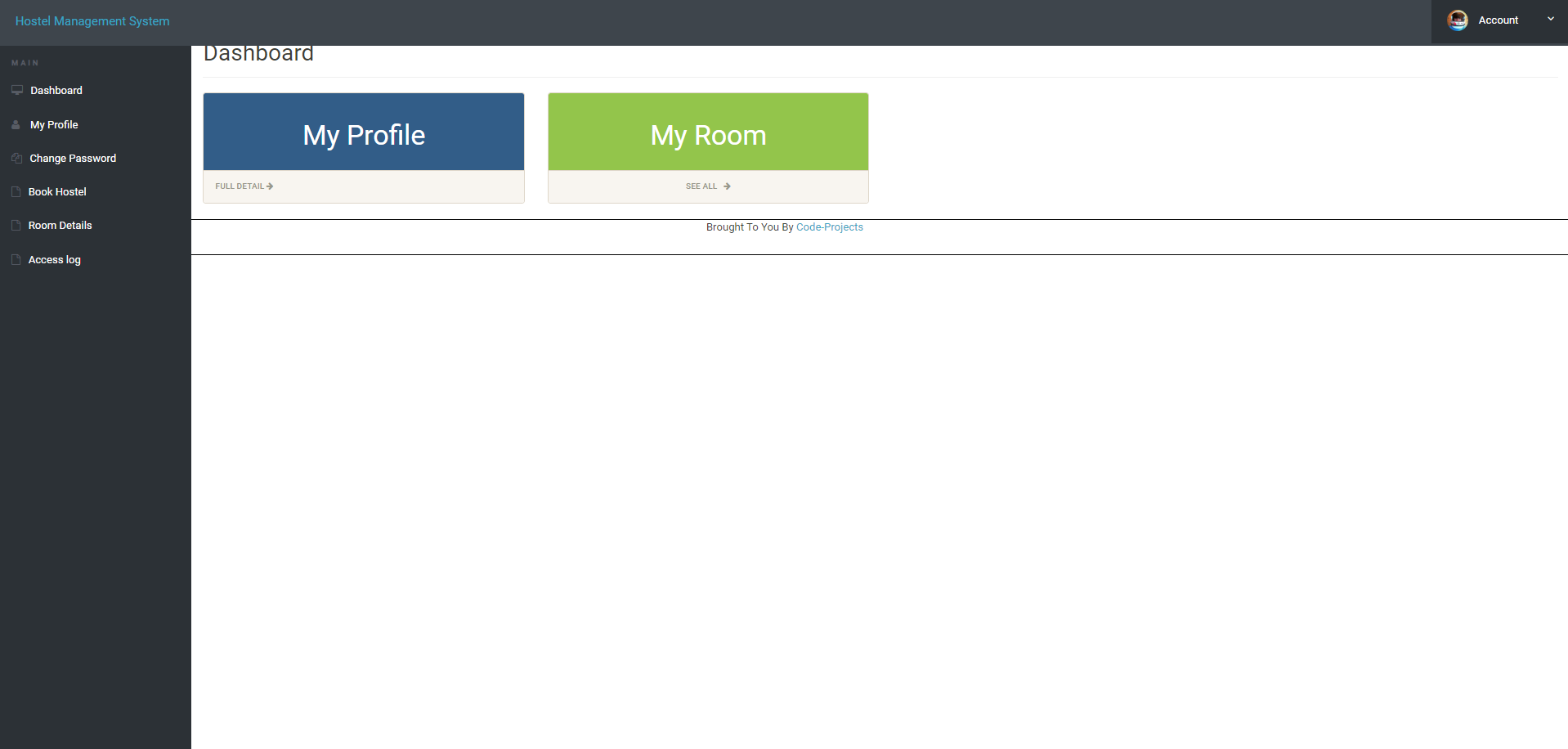
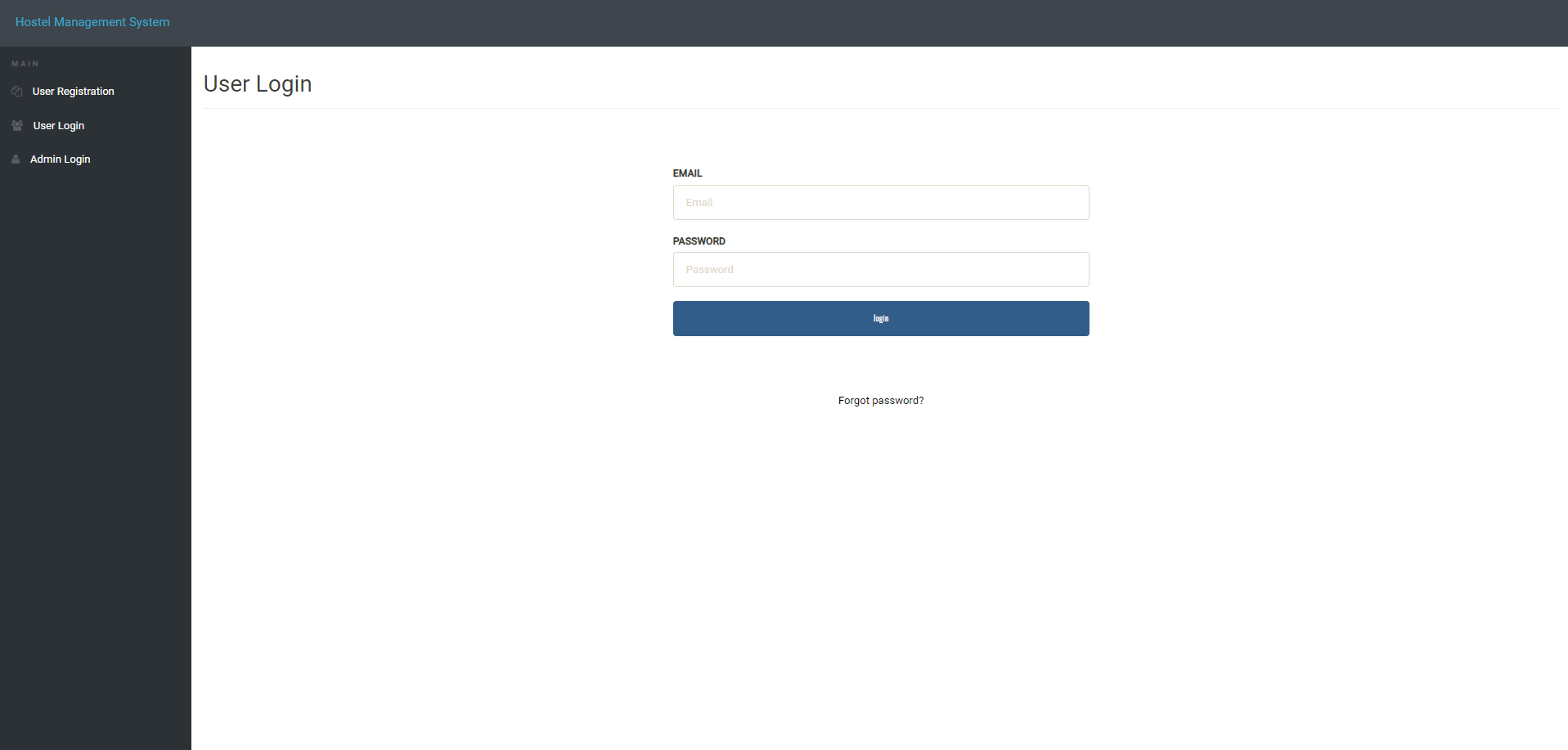
**Process 7.2 Display Registration Info**

DO

READ Hostel Registration Information from Registration Data File.

SHOW Hostel Registration Information.

UNTIL END of File

**Input/output Design *(Screenshots of System)***

**Conclusion**

In conclusion, the development of a Hostel Management System represents an incredible step forward in streamlining and enhancing the efficiency of Hostel operations. With its potential to improve service quality, reduce errors, and save time and resources, the Hostel Management System stands as a valuable tool for hostel owners and administrators looking to modernize their operations and provide a seamless and convenient experience for their guests.