VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-560014

DBMS LABORATORY

Mini Project Report

On

“HOSTEL MANAGEMENT SYSTEM”

Submitted in partial fulfillment of the requirement of V Semester DBMS Laboratory

Submitted by

S. UDAY KIRAN	(1DT21IS124)
SAHANA M	(1DT21IS125)
SAI BALAJI M	(1DT21IS127)
YESHWANTH SINGH S	(1DT21IS174)

Under the guidance of

Mrs. Supriya R.K
Asst. Professor
Dept. of ISE
DSATM, Bangalore.

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)
(All B.E Courses Accredited by NBA, New Delhi)
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082
2023-24

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)
(All B.E Courses Accredited by NBA, New Delhi)
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the mini-project work entitled **"HOSTEL MANAGEMENT SYSTEM"** is carried out by **S. UDAY KIRAN (1DT21IS124), SAHANA M (1DT21IS125), SAI BALAJI M (1DT21IS127), YESHWANTH SINGH S (1DT21IS174) in** partial fulfillment for the requirement of V semester DBMS Laboratory in **Information Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2023-2024. It is certified that all the corrections/ suggestions indicated for the given internal assessment have been incorporated in the report. This report has been approved as it satisfies the academic requirements with respect to the mini-project work.

Signature of the Guide

Mrs. Supriya R.K
Asst. Professor, Dept. of ISE
DSATM, Bangalore.

Signature of the HOD

Dr. Nandini Prasad K S
Dean (Foreign Affairs) and HOD
Dept. of ISE
DSATM, Bangalore.

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

ACKNOWLEDGEMENT

We express our profound gratitude to **Dr. Ravishankar M, Principal, DSATM, Bangalore,** for providing the necessary facilities and an ambient environment to work.

We are grateful to **Dr. Nandini Prasad K S, Dean (Foreign Affairs) & HOD, Department of Information Science and Engineering, DSATM, Bangalore,** for her valuable suggestions and advice throughout my/our work period.

We would like to express my deepest gratitude and sincere thanks to our guide **Mrs. Supriya R.K, Assistant Professor, Department of Information Science and Engineering, DSATM, Bangalore,** for her keen interest and encouragement in the project whose guidance made the project into reality. We would like to thank all the staff members of **Department of Information Science and Engineering** for their support and encouragement during the course of this project.

Definitely most, We would like to thank our parents, all my family members and friends, without whose help and encouragement this project would have been impossible.

S. UDAY KIRAN(1DT21IS124)
SAHANA M(1DT21IS125)
SAI BALAJI M (1DT21IS127)
YESHWANTH SINGH S (1DT21IS174)

ABSTRACT

The report presents a detailed analysis of The Hostel Management System, developed using PHP and MySQL, which offers a comprehensive solution tailored for hostel businesses, particularly catering to student accommodations. It features an intuitive administrator dashboard providing centralized control over hostel operations, including student record management and room allocation. Students benefit from a user-friendly dashboard enabling easy access to essential functionalities such as booking accommodations and updating profiles. Both administrators and users can manage profiles effortlessly, fostering a sense of ownership and accountability. The system simplifies student records management and optimizes room allocation, enhancing operational efficiency. Detailed room insights aid administrators in informed decision-making, while course management facilitates seamless integration of educational programs within hostel premises. Secure password management and streamlined booking processes ensure convenience and security for users, with an access log feature enhancing system security and accountability. Additionally, responsive design ensures optimal user experience across various devices, ensuring accessibility and usability regardless of the platform.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The project is designed so as to be used by Admin to add students in the hostel. It is an online system through which Admin can add students to hostel, view available hostels, add the new hostel, view students, view number of students, search student, see the room allotments and other features.

The main objective of the Project on Hostel Management System is to manage the details of Hostel, Students, Student visitors, Student Registration. It manages all the information about Hostel, Student Registration, Warden, Visitors. The project is totally built at administrative end and thus only the administrator is guaranteed the access.

The project aims at Business process automation, i.e., we have tried to computerize various processes of Hostel Management System.

➢ In computer system the person has to fill the form.

➢ To utilize resources in an efficient manner by increasing their productivity through automation.

➢ The system generates types of information that can be used for various purposes.

➢ It satisfies the user requirement.

➢ Be easy to understand by the user and operator.

➢ Be easy to operate.

➢ Have a good user interface.

➢ Be expandable.

CHAPTER 2

REQUIREMENT ANALYSIS

The requirement analysis specifies the requirements needed to develop a project. In this phase, we collect the requirements needed for designing the project. The requirements collected are then analyzed and carried to the next phase.

2.1 FRONTEND TECH STACK

- HTML
- CSS
- JavaScript

2.2 BACKEND

For backend we have used PHP, MySql as database and Apache using Xampp for the connectionto the server. The project is then run in the localhost on the browser. The required software are:

- PHP
- MYSQL
- XAMPP

2.3 FEATURES

- Online System for Admin: The project provides an online platform accessible to administrators, enabling them to perform various tasks related to hostel management.
- Student Management: Admin can add students to the hostel, view student details, search for specific students, and manage student registrations.
- Hostel Management: Admin can view available hostels, add new hostels to the system, and manage room allotments.
- Business Process Automation: The project automates various processes involved in hostel

management, reducing manual efforts and increasing productivity.

- Efficient Resource Utilization: By automating tasks, the system aims to utilize resources efficiently, optimizing staff productivity and hostel facilities.

- Information Generation: The system generates different types of information that can be utilized for various purposes, such as reporting, analysis, and decision-making.

- User Satisfaction: The project aims to satisfy user requirements by providing a user-friendly interface, easy operation, and intuitive features.

- Expandability: It is designed to be expandable, allowing for future enhancements and additional features.

CHAPTER 3

DESIGN

3.1 E-R DIAGRAM

An Entity – Relationship model (ER model) describes inter-related things of interest ina specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

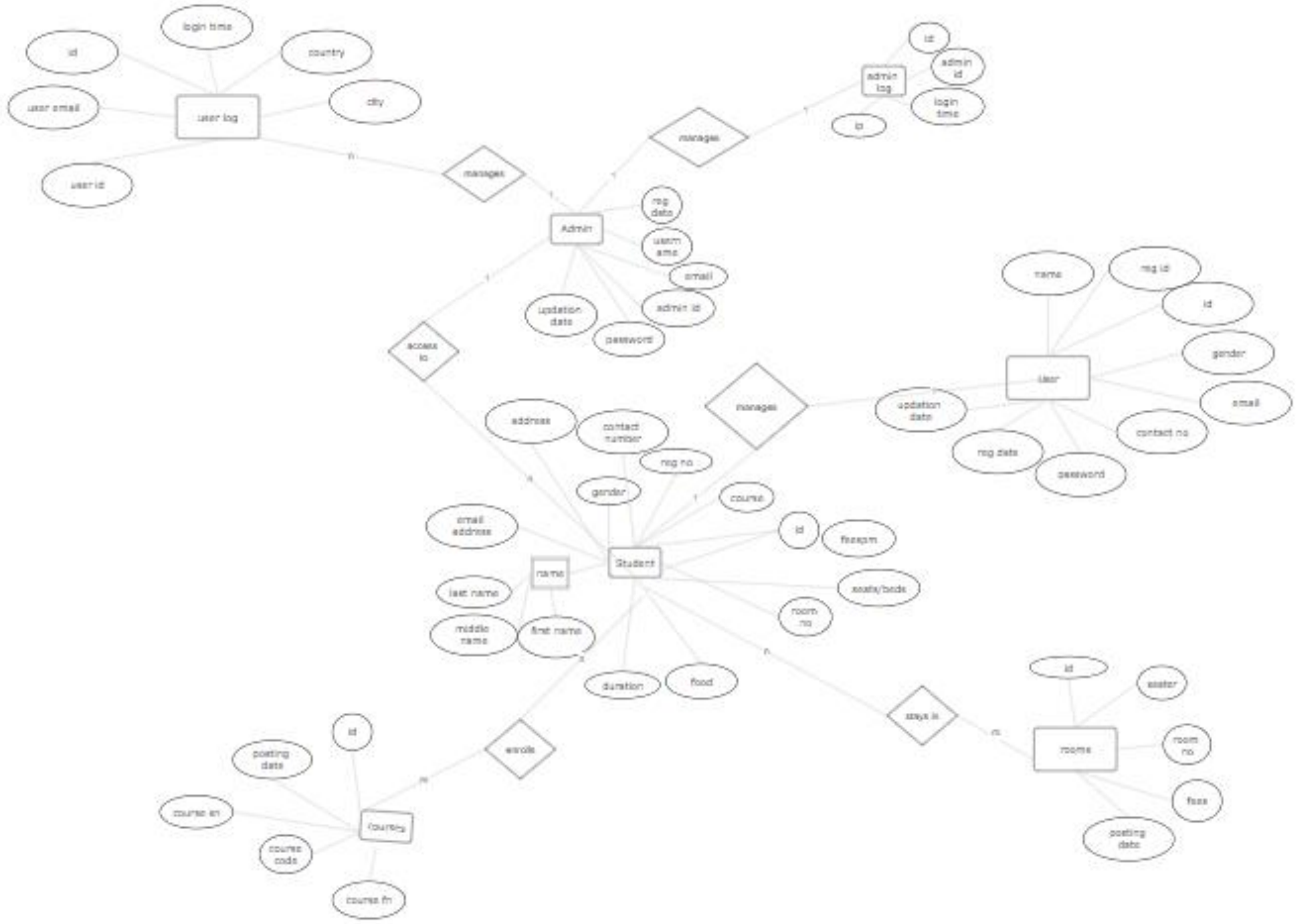


FIGURE 3.1.1: E-R DIAGRAM OF THE HOSTEL MANAGEMENT SYSTEM

3.2 DATABASE SCHEMA DIAGRAM

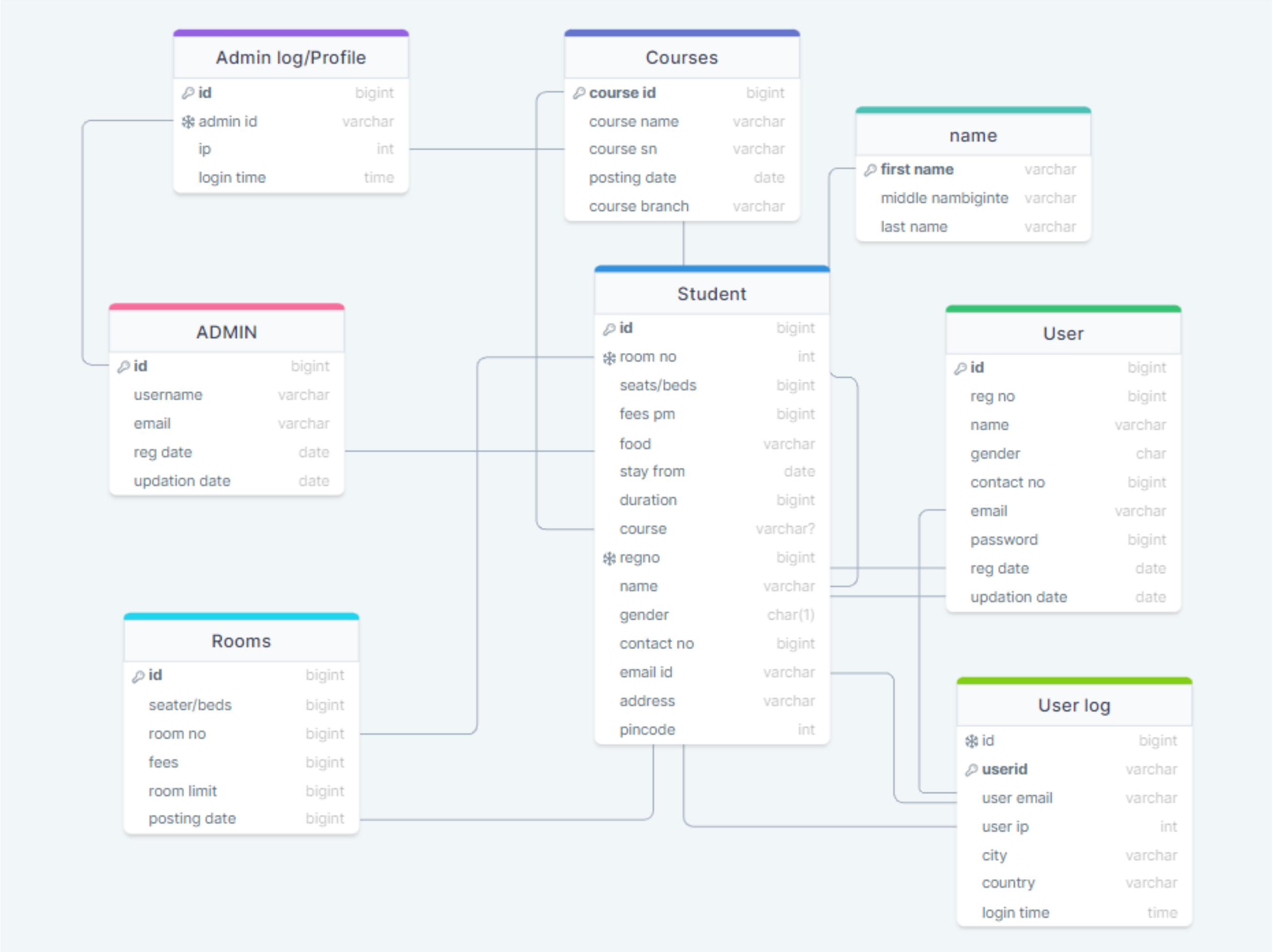


FIGURE 3.2.1 DATABASE SCHEMA DIAGRAM

The schema diagram visually represents the structure of the database tables and their relationships. Each table is depicted with its attributes, primary keys, and foreign keys, providing a comprehensive overview of the database design.

The schema diagram serves as a visual representation of the database structure, illustrating the relationships between different entities and their attributes. It provides a clear and concise overview of how data is organized within the system, facilitating better understanding and communication among developers, stakeholders, and users. By visually depicting the database schema, the diagram helps ensure consistency and accuracy in database design and implement ultimately contributing to the efficiency and effectiveness of the inventory management system. A database schema diagram is essential for comprehending the database's structure and relationships, aiding communication among stakeholders and facilitating planning and design. It supports normalization to ensure data integrity, enables performance optimization through informed decision-making, and serves as documentation for maintenance, ensuring consistency and efficiency. Additionally, it enables scalability for future growth, laying the groundwork for seamless database development, maintenance, and management processes, ultimately enhancing the effectiveness of database-driven applications.

3.3 REALTIONAL SCHEMA BETWEEN THE ENTITIES

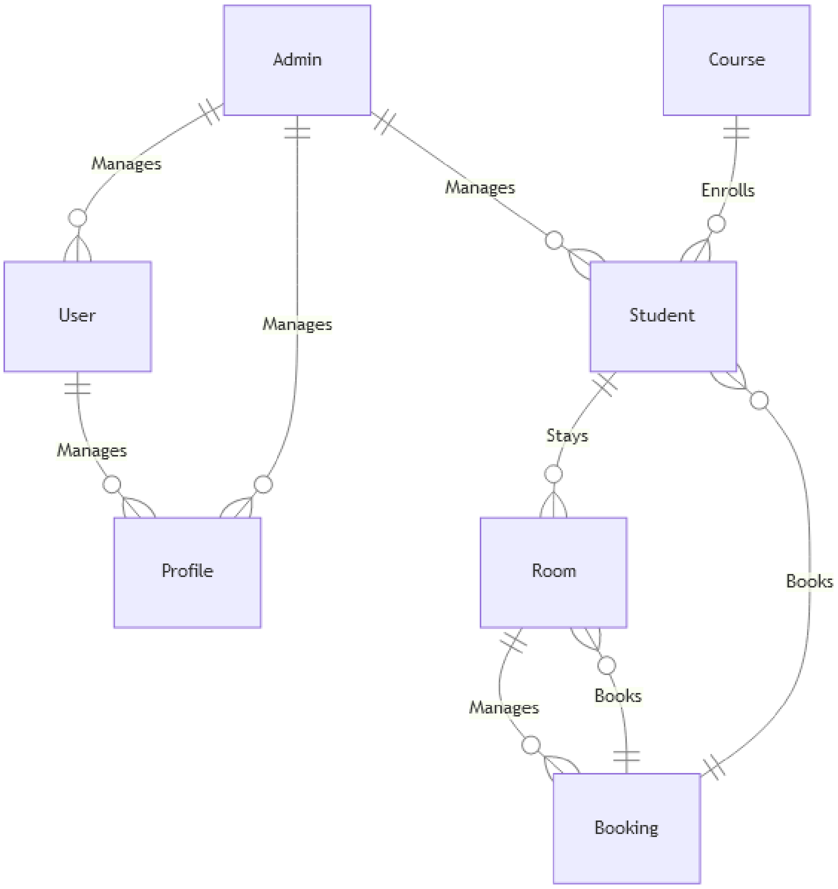


FIGURE 3.3.1 RELATIONAL SCHEMA DIAGRAM

CHAPTER 4

IMPLEMENTATION

In this chapter, the actual implementation of the inventory management system is detailed, covering both front-end and back-end development, as well as the database management systemused.

4.1 Front-End Development

This section focuses on the technologies and methodologies employed in the front-end development of the inventory management system.

4.1.1 HTML Structure

- Description of the HTML structure used to create the user interface.
- Explanation of the layout, including header, footer, navigation bar, and content sections.
- Use of semantic HTML elements for better accessibility and SEO.

4.1.2 CSS Styling

- Overview of the CSS stylesheets utilized to enhance the visual presentation of the application.
- Description of the styling principles followed, such as CSS resets, modularization, and responsive design.
- Use of CSS frameworks like Bootstrap for rapid development and consistent styling.

4.1.3 Bootstrap Integration

- Discussion on integrating Bootstrap framework for responsive and mobile-first design.
- Explanation of how Bootstrap components and utilities are used to streamline UI development.
- Customization of Bootstrap styles to match the design requirements of the inventory management system.

4.1.4 JavaScript Functionality

• Discussion on the JavaScript functions implemented for client-side interactivity and dynamic behavior.

• Explanation of event handling, form validation, DOM manipulation, and asynchronous data retrieval using AJAX.

• Integration of JavaScript libraries or frameworks for enhanced functionality, such as jQuery or Vue.js.

4.2 Back-End Development

This section delves into the server-side scripting and logic implemented to handle data processing, authentication, and server interactions.

4.2.1 PHP Scripts

• Description of the PHP scripts responsible for server-side processing and generation of dynamic content.

• Overview of PHP frameworks or libraries used, such as Laravel or CodeIgniter, if applicable.

• Explanation of PHP coding standards, security practices, and performance optimizations.

4.2.2 XAMPP Environment

• Discussion on setting up and configuring XAMPP for local development and testing purposes.

• Overview of the Apache web server, MySQL database, and PHP runtime environment included in XAMPP.

• Explanation of how XAMPP simplifies the development process by providing a complete development stack.

4.3 Database Management

This section focuses on the design and management of the MySQL database used to store and retrieve inventory data.

4.3.1 Database Schema

• Introduction of the database schema designed to organize and structure the inventory data.

• Explanation of the tables, columns, primary keys, foreign keys, and relationships defined in the schema.

• Use of entity-relationship diagrams (ERDs) to visualize the database structure.

4.3.2 SQL Queries

• Overview of the SQL queries used to perform CRUD (Create, Read, Update, Delete) operations on the database.

• Description of complex queries for data retrieval, filtering, sorting, and aggregation.

• Discussion on optimizing SQL queries for performance and efficiency

CHAPTER 5

MYSQL

The provided SQL code represents the database schema for an Hostel management system encompassing tables to store information related to administrators, hostel rooms, courses, registrations, user logins, and states.

Administrative Functionality:

The admin table stores essential information about administrators, including their unique IDs, usernames, emails, passwords, registration dates, and update timestamps. Additionally, the adminlog table records login activities of administrators, tracking details such as their IDs, IP addresses, and login timestamps.

Course Management:

The courses table serves as a repository for information regarding different courses offered by the hostel. It contains fields such as course IDs, course codes, short names, full names, and posting dates.

Hostel Room Allocation:

Room allocation details are managed through the rooms table, which stores information about individual rooms, including their IDs, seating capacity, room numbers, fees per month, and posting dates.

Registration Information:

The registration table holds crucial data about hostel registrations, encompassing a wide range of details such as registration IDs, room numbers, seating capacity, fees, food status, stay duration, course information, personal details of registrants including names, genders, contact information, guardian details, correspondence and payment addresses, and posting dates.

User Interaction Tracking:

User interactions with the system are monitored and logged in the userlog table, which records user IDs, email addresses, IP addresses, login timestamps, cities, and countries. This table provides valuable insights into user activity within the hostel management system.

User Registration Details:

The userregistration table stores information about users who register with the system. It includes fields such as user IDs, registration numbers, names, genders, contact information, emails, passwords, registration dates, and update timestamps.

Data Integrity and Performance Optimization:

Each table is equipped with primary keys and indexes to ensure data integrity and optimize query performance. The AUTO_INCREMENT attribute is utilized to automatically generate unique identifiers for primary key fields.

RELATIONS-

The provided database schema consists of several entities (tables) interconnected through various relationships. Let's outline the relationships between these entities:

1. **One-to-Many Relationship:**
    - The **admin** table has a one-to-many relationship with the **adminlog** table. One administrator can have multiple login activities recorded in the **adminlog** table.
2. **One-to-Many Relationship:**
    - The **courses** table has a one-to-many relationship with the **registration** table. One course can have multiple registrations associated with it, as students enroll in courses offered by the hostel.
3. **One-to-One Relationship:**
    - The **registration** table has a one-to-one relationship with the **rooms** table. Each registration is typically associated with one specific room where the student will reside during their stay in the hostel.

4. **One-to-One Relationship:**
   - The **registration** table also has a one-to-one relationship with the **userregistration** table. Each registration corresponds to a single user registration, capturing personal details of the individual registering for hostel accommodation.
5. **One-to-Many Relationship:**
   - The **states** table has a one-to-many relationship with the **registration** table. Many registrations may have the same state information, indicating the state from which the registrants come.
6. **One-to-Many Relationship:**
   - The **userregistration** table has a one-to-many relationship with the **userlog** table. One user registration can have multiple login activities recorded in the **userlog** table.

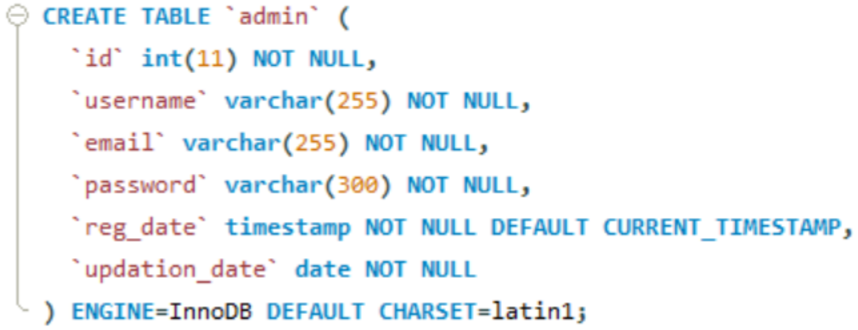
ATTRIBUTES AND CONSTRAINTS-

The provided database schema consists of several entities (tables) interconnected through various attributes and having specific constraints. Let's discuss the primary keys, foreign keys, super keys, and integrity constraints present in the tables of the hostel database schema:

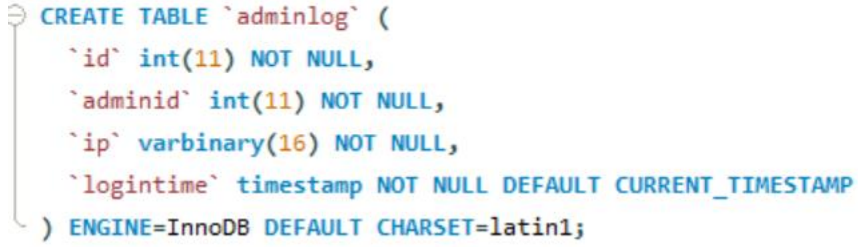
1. **Primary Keys (PK):**
   - Each table typically has a primary key defined, which uniquely identifies each record within the table. Primary keys are essential for data integrity and indexing purposes. In the provided schema:
     - **admin** table: **id** is the primary key.
     - **courses** table: **id** is the primary key.
     - **registration** table: **id** is the primary key.
     - **rooms** table: **id** is the primary key.
     - **states** table: **id** is the primary key.
     - **userlog** table: **id** is the primary key.
     - **userregistration** table: **id** is the primary key.

2. **Foreign Keys (FK):**
    - Foreign keys establish relationships between tables by referencing the primary key of another table. They enforce referential integrity, ensuring that values in one table's foreign key column correspond to values in another table's primary key column. In the provided schema:
        - **registration** table: **course** column references **id** column in the **courses** table.
        - **registration** table: **roomno** column references **id** column in the **rooms** table.
3. **Superkeys:**
    - A superkey is a set of attributes that uniquely identifies each tuple within a relation. In other words, it's a superset of a candidate key. In the provided schema:
        - Each table's primary key can be considered a superkey since it uniquely identifies each record.
4. **Integrity Constraints:**
    - Integrity constraints enforce rules on data to maintain data accuracy and consistency. Common types of integrity constraints include NOT NULL, UNIQUE, CHECK, and DEFAULT constraints.
        - In the provided schema, certain columns are defined with NOT NULL constraints to ensure they always contain values.
        - Unique constraints may be implied by the use of primary keys, ensuring uniqueness within the specified column(s).
        - Default constraints are set for certain columns, specifying default values if no value is provided during data insertion.
        - For example, the **registration** table has various columns such as **firstName**, **gender**, **contactno**, **emailid**, etc., which are essential for a registration record and thus have NOT NULL constraints.

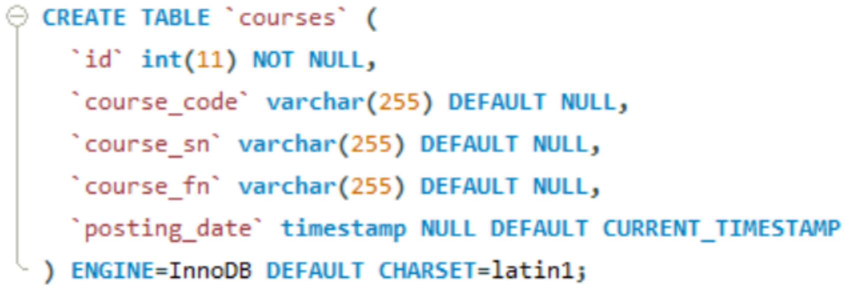
5.1 CODE



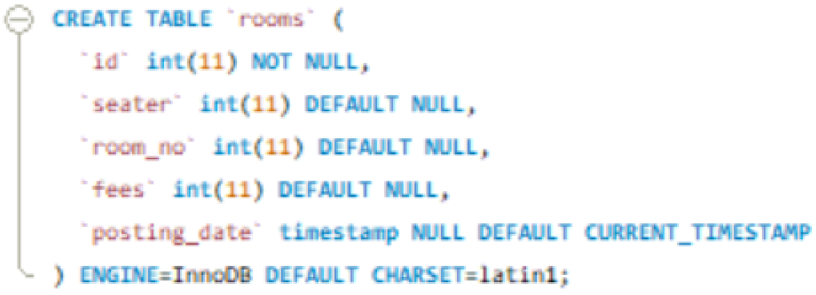
```
CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(300) NOT NULL,
  `reg_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updation_date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



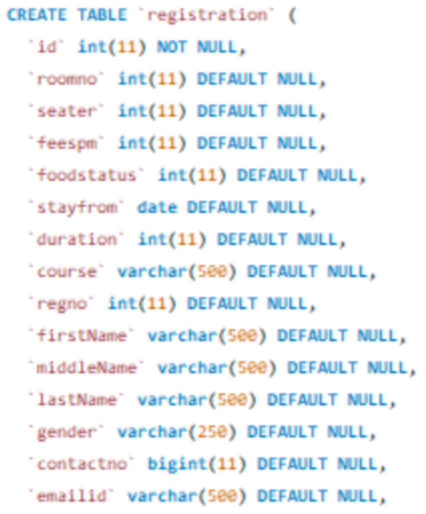
```
CREATE TABLE `adminlog` (
  `id` int(11) NOT NULL,
  `adminid` int(11) NOT NULL,
  `ip` varbinary(16) NOT NULL,
  `logintime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



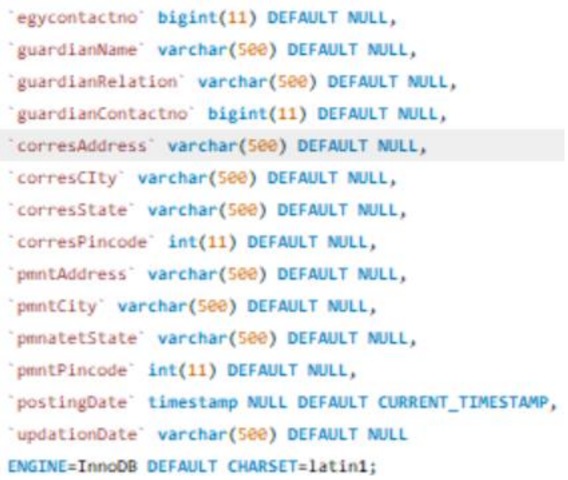
```
CREATE TABLE `courses` (
  `id` int(11) NOT NULL,
  `course_code` varchar(255) DEFAULT NULL,
  `course_sn` varchar(255) DEFAULT NULL,
  `course_fn` varchar(255) DEFAULT NULL,
  `posting_date` timestamp NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



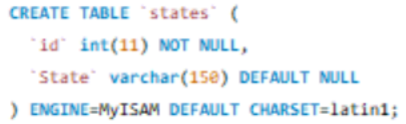
```
CREATE TABLE `rooms` (
  `id` int(11) NOT NULL,
  `seater` int(11) DEFAULT NULL,
  `room_no` int(11) DEFAULT NULL,
  `fees` int(11) DEFAULT NULL,
  `posting_date` timestamp NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



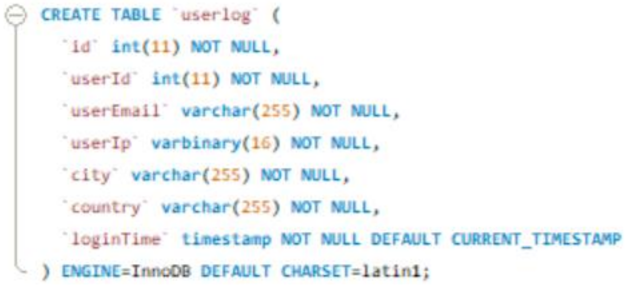
```
CREATE TABLE `registration` (
  `id` int(11) NOT NULL,
  `roomno` int(11) DEFAULT NULL,
  `seater` int(11) DEFAULT NULL,
  `feespm` int(11) DEFAULT NULL,
  `foodstatus` int(11) DEFAULT NULL,
  `stayfrom` date DEFAULT NULL,
  `duration` int(11) DEFAULT NULL,
  `course` varchar(500) DEFAULT NULL,
  `regno` int(11) DEFAULT NULL,
  `firstName` varchar(500) DEFAULT NULL,
  `middleName` varchar(500) DEFAULT NULL,
  `lastName` varchar(500) DEFAULT NULL,
  `gender` varchar(250) DEFAULT NULL,
  `contactno` bigint(11) DEFAULT NULL,
  `emailid` varchar(500) DEFAULT NULL,
```



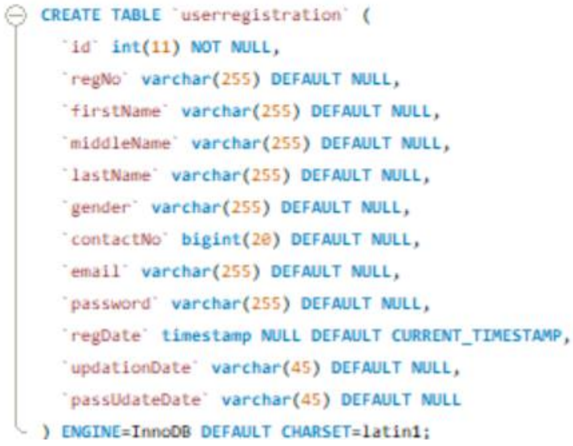
```
`egycontactno` bigint(11) DEFAULT NULL,
`guardianName` varchar(500) DEFAULT NULL,
`guardianRelation` varchar(500) DEFAULT NULL,
`guardianContactno` bigint(11) DEFAULT NULL,
`corresAddress` varchar(500) DEFAULT NULL,
`corresCIty` varchar(500) DEFAULT NULL,
`corresState` varchar(500) DEFAULT NULL,
`corresPincode` int(11) DEFAULT NULL,
`pmntAddress` varchar(500) DEFAULT NULL,
`pmntCity` varchar(500) DEFAULT NULL,
`pmnatetState` varchar(500) DEFAULT NULL,
`pmntPincode` int(11) DEFAULT NULL,
`postingDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
`updationDate` varchar(500) DEFAULT NULL
ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



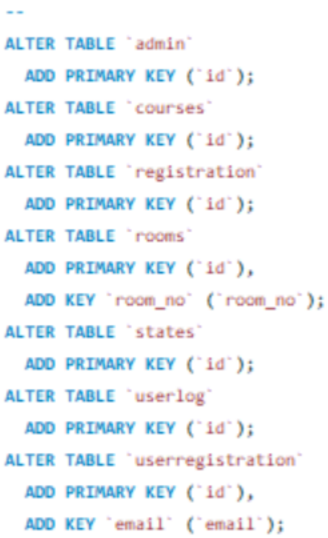
```
CREATE TABLE `states` (
  `id` int(11) NOT NULL,
  `State` varchar(150) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```



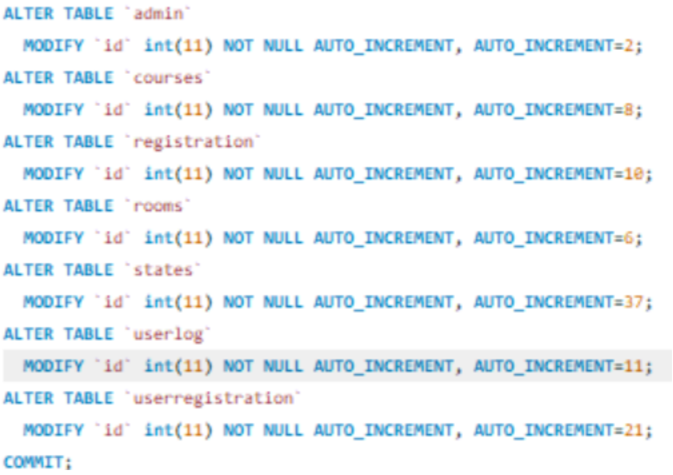
```
CREATE TABLE `userlog` (
  `id` int(11) NOT NULL,
  `userId` int(11) NOT NULL,
  `userEmail` varchar(255) NOT NULL,
  `userIp` varbinary(16) NOT NULL,
  `city` varchar(255) NOT NULL,
  `country` varchar(255) NOT NULL,
  `loginTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
CREATE TABLE `userregistration` (
  `id` int(11) NOT NULL,
  `regNo` varchar(255) DEFAULT NULL,
  `firstName` varchar(255) DEFAULT NULL,
  `middleName` varchar(255) DEFAULT NULL,
  `lastName` varchar(255) DEFAULT NULL,
  `gender` varchar(255) DEFAULT NULL,
  `contactNo` bigint(20) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `regDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `updationDate` varchar(45) DEFAULT NULL,
  `passUdateDate` varchar(45) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `courses`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `registration`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `rooms`
  ADD PRIMARY KEY (`id`),
  ADD KEY `room_no` (`room_no`);
ALTER TABLE `states`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `userlog`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `userregistration`
  ADD PRIMARY KEY (`id`),
  ADD KEY `email` (`email`);
```

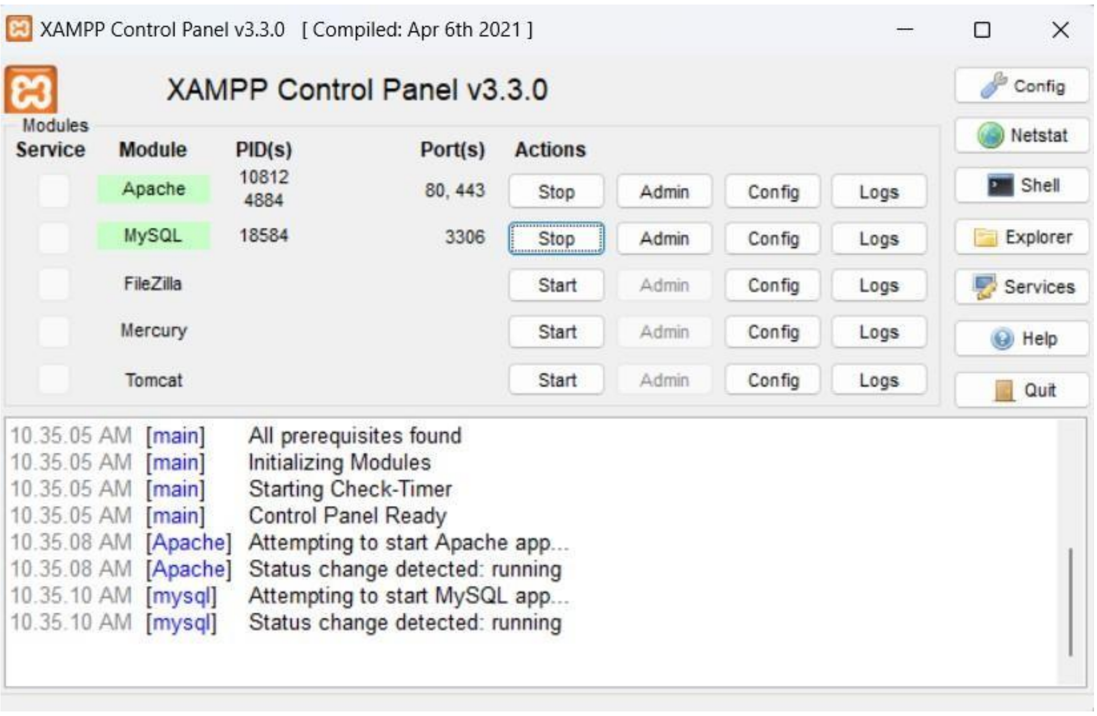


```
ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
ALTER TABLE `courses`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
ALTER TABLE `registration`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;
ALTER TABLE `rooms`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
ALTER TABLE `states`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=37;
ALTER TABLE `userlog`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
ALTER TABLE `userregistration`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;
COMMIT;
```

CHAPTER 6

SNAPSHOTS

For presenting the functionality of the hostel management system, a comprehensive approach will be employed to showcase various sections such as the dashboard, registration pages, room allocation, administrative tools, and other relevant features. Multiple screenshots will be captured for each section, highlighting different aspects of the system's functionality and user interfaces. These screenshots will be organized into distinct categories or galleries, facilitating easy navigation and exploration of the system's capabilities. Each screenshot will be accompanied by annotations or captions, providing brief descriptions and emphasizing key features or functionalities depicted in the images. This presentation style ensures clarity and accessibility, allowing stakeholders to visually comprehend the system's capabilities across different modules and interfaces effectively.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	10812 4884	80, 443	Stop	Admin	Config	Logs
	MySQL	18584	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
10.35.05 AM  [main]     All prerequisites found
10.35.05 AM  [main]     Initializing Modules
10.35.05 AM  [main]     Starting Check-Timer
10.35.05 AM  [main]     Control Panel Ready
10.35.08 AM  [Apache]   Attempting to start Apache app...
10.35.08 AM  [Apache]   Status change detected: running
10.35.10 AM  [mysql]    Attempting to start MySQL app...
10.35.10 AM  [mysql]    Status change detected: running
```

FIGURE 6.1 XAMPP DASHBOARD

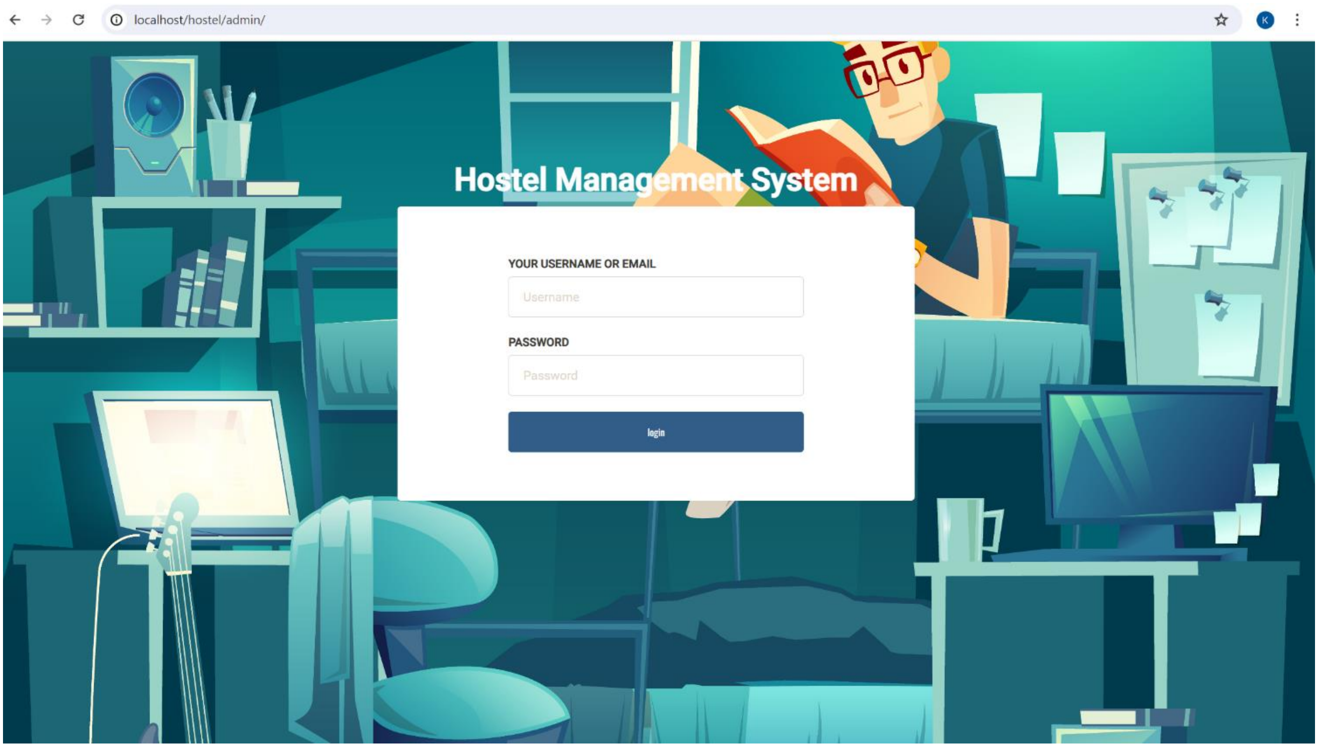
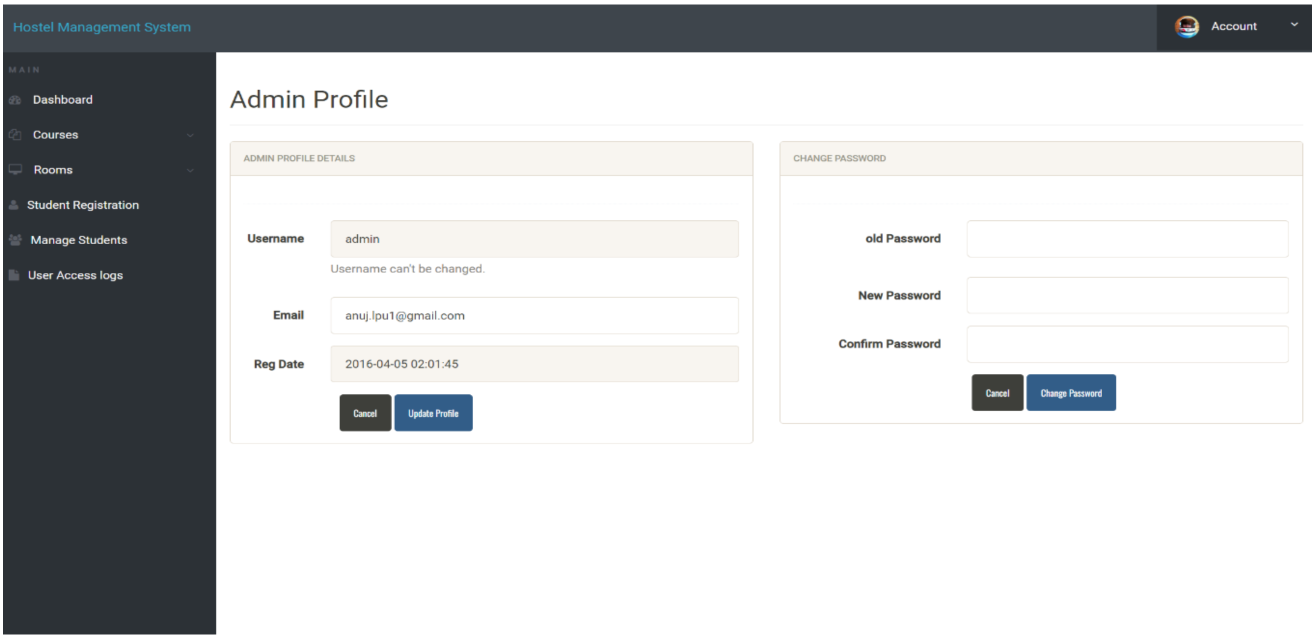


FIGURE 6.2. DASHBOARD OF THE HOSTEL MANAGEMENT SYSTEM (ADMIN LOGIN)



Admin Profile

ADMIN PROFILE DETAILS

Username: admin

Username can't be changed.

Email: anuj.lpu1@gmail.com

Reg Date: 2016-04-05 02:01:45

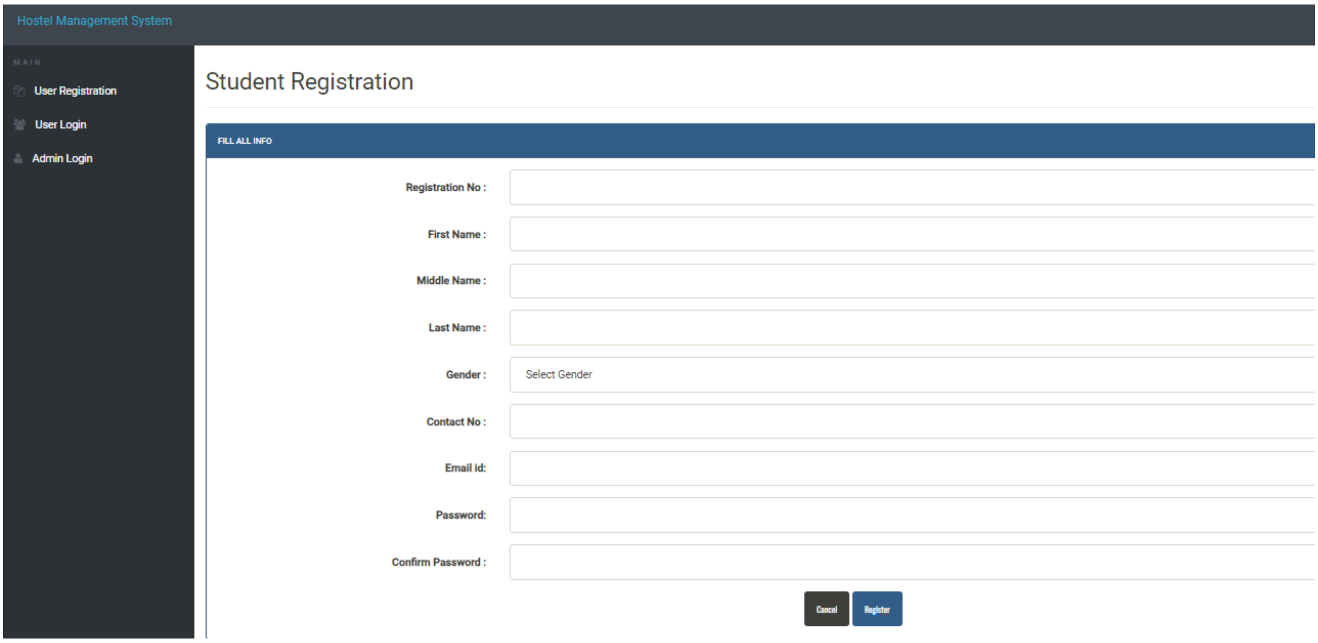
CHANGE PASSWORD

old Password

New Password

Confirm Password

FIGURE 6.3 DASHBOARD OF ADMIN PROFILE



Student Registration

FILL ALL INFO

Registration No :

First Name :

Middle Name :

Last Name :

Gender : Select Gender

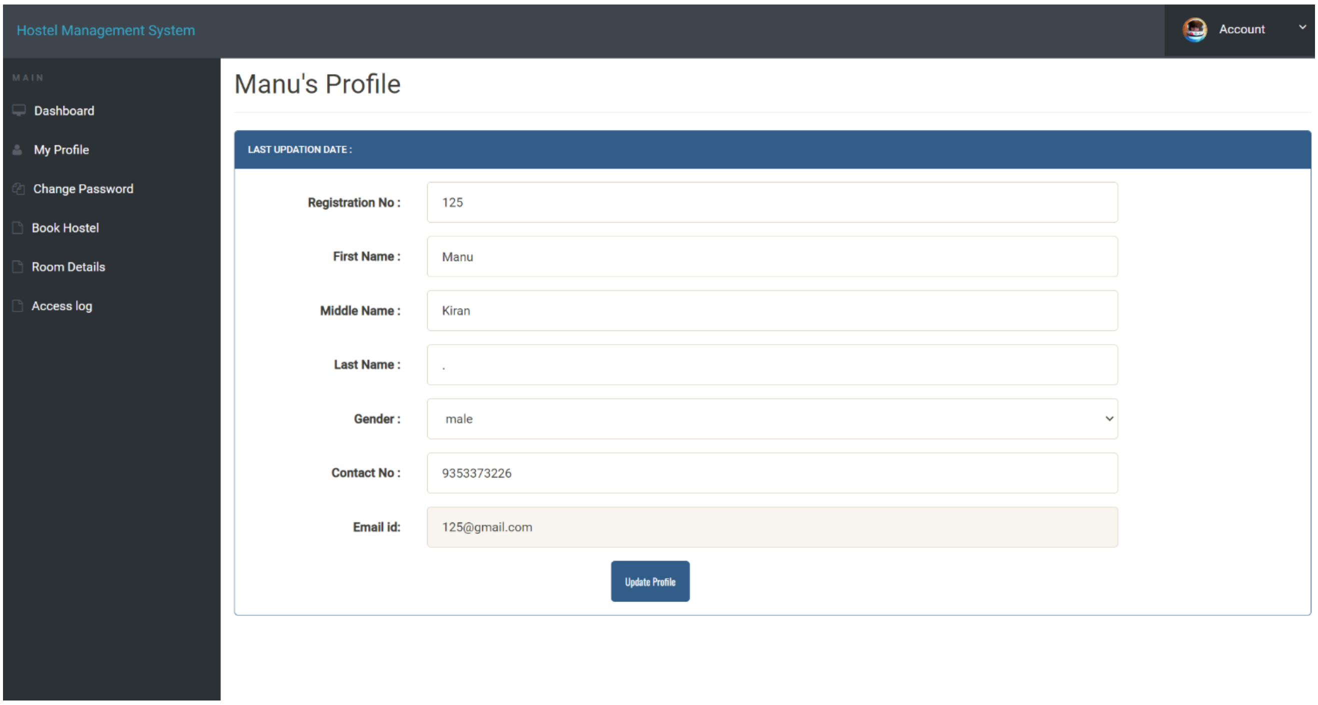
Contact No :

Email id:

Password:

Confirm Password :

FIGURE 6.4 STUDENT REGISTRATION



Manu's Profile

LAST UPDATION DATE :

Registration No : 125

First Name : Manu

Middle Name : Kiran

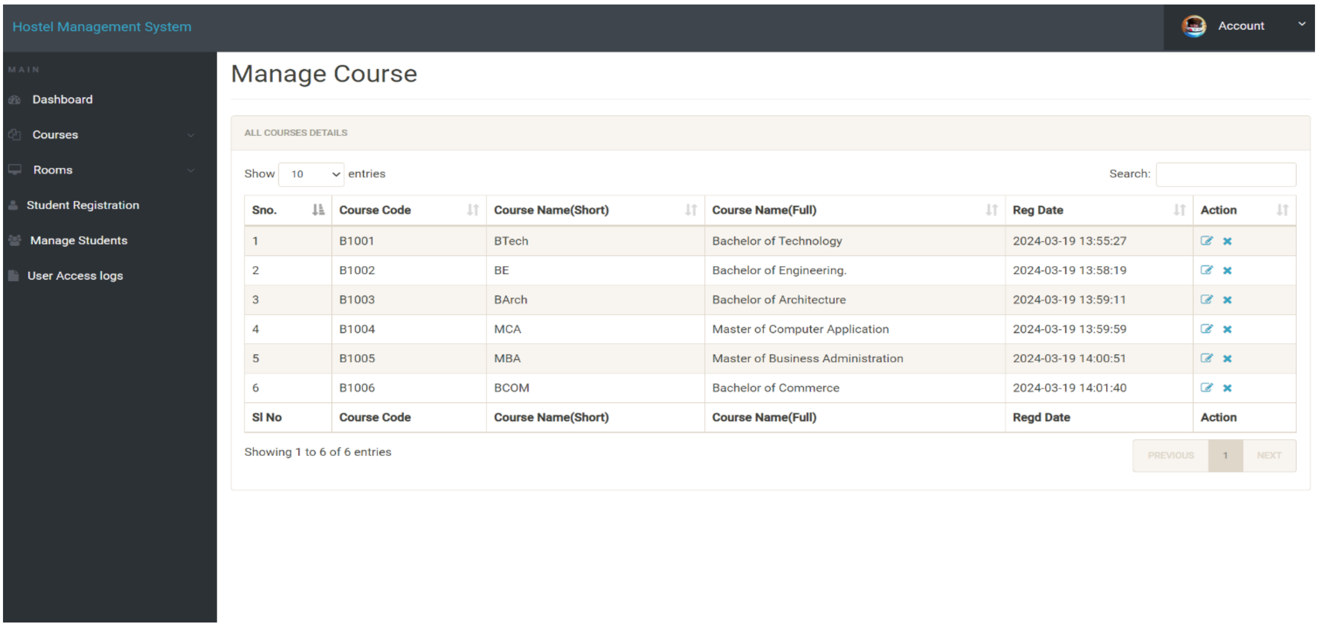
Last Name : .

Gender : male

Contact No : 9353373226

Email id: 125@gmail.com

FIGURE 6.5 STUDENT PROFILE



Manage Course

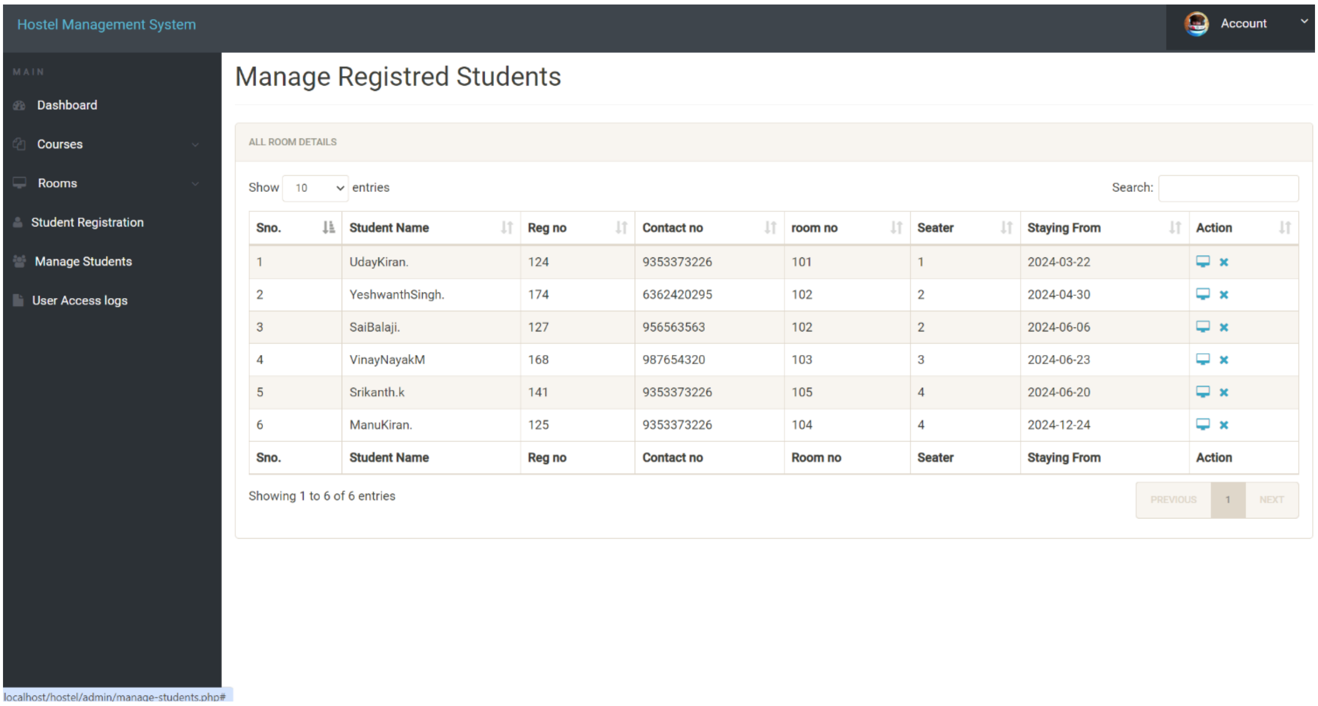
ALL COURSES DETAILS

Show 10 entries Search:

Sno.	Course Code	Course Name(Short)	Course Name(Full)	Reg Date	Action
1	B1001	BTech	Bachelor of Technology	2024-03-19 13:55:27	✎ ✖
2	B1002	BE	Bachelor of Engineering.	2024-03-19 13:58:19	✎ ✖
3	B1003	BArch	Bachelor of Architecture	2024-03-19 13:59:11	✎ ✖
4	B1004	MCA	Master of Computer Application	2024-03-19 13:59:59	✎ ✖
5	B1005	MBA	Master of Business Administration	2024-03-19 14:00:51	✎ ✖
6	B1006	BCOM	Bachelor of Commerce	2024-03-19 14:01:40	✎ ✖
Sl No	Course Code	Course Name(Short)	Course Name(Full)	Regd Date	Action

Showing 1 to 6 of 6 entries

FIGURE 6.6 MANAGING THE COURSES (BY ADMIN)



Manage Registred Students

ALL ROOM DETAILS

Show 10 entries Search:

Sno.	Student Name	Reg no	Contact no	room no	Seater	Staying From	Action
1	UdayKiran.	124	9353373226	101	1	2024-03-22	🖥 ✖
2	YeshwanthSingh.	174	6362420295	102	2	2024-04-30	🖥 ✖
3	SaiBalaji.	127	956563563	102	2	2024-06-06	🖥 ✖
4	VinayNayakM	168	987654320	103	3	2024-06-23	🖥 ✖
5	Srikanth.k	141	9353373226	105	4	2024-06-20	🖥 ✖
6	ManuKiran.	125	9353373226	104	4	2024-12-24	🖥 ✖
Sno.	Student Name	Reg no	Contact no	Room no	Seater	Staying From	Action

Showing 1 to 6 of 6 entries

FIGURE 6.7 MANAGING THE REGISTERED STUDENTS

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENTS

The project is only a humble venture to satisfy the needs to manage the paper work. Several user-friendly coding has also adopted. The computerization of the system speeds up the process. This package shall prove to be a powerful package in satisfying some requirements of the hostel. The objective of software planning is to provide a frame work that enables the admin to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

7.1. Future Enhancements-

For the future development of the hostel management system, several enhancements could significantly improve its functionality and user experience. Firstly, integrating online payment gateways would streamline fee collection processes, offering students a secure platform to conveniently pay hostel fees. This integration would not only reduce administrative workload but also enhance convenience for students, facilitating timely payments and improving financial management for the hostel. Additionally, developing a mobile application for the system would extend its accessibility, allowing students to access registration details, room allocation information, and make inquiries directly from their smartphones. This mobile application would cater to the growing trend of mobile usage and provide students with greater flexibility in managing their hostel-related tasks on-the-go.

Furthermore, enhancing reporting and analytics capabilities within the system would enable administrators to derive valuable insights from hostel occupancy rates, fee collections, and room utilization. By leveraging advanced reporting tools, administrators can make data-driven decisions to optimize hostel operations, improve resource allocation, and enhance overall efficiency. Moreover, implementing automated notification systems would enhance communication between the hostel administration and students, providing timely updates on room allocations & fee payment deadlines.

REFERENCES

[1] MySQL. https://www.mysql.com/.

[2] PHP: Hypertext Preprocessor. https://www.php.net/.

[3] JavaScript. https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[4] Bootstrap. https://getbootstrap.com/.

[5] Laravel. https://laravel.com/.

[6] HTML. https://www.w3schools.com/html/.

[7] CSS. https://www.w3schools.com/css/.

[8] Git. https://git-scm.com/.

[9] Apache Friends: XAMPP. https://www.apachefriends.org/.

[10] Open Source Initiative: MySQL. https://opensource.org/licenses/MySQL.

[11] Apache Friends. XAMPP License. https://www.apachefriends.org/index.html.

[12] Elmasri .& Navathe, S. B. Fundamentals of Database System(7th ed.).Pearson.