HOSTEL MANAGEMENT SYSTEM

Name : Vasundhara Gadakh Roll no: 331069 Prn no: 22320013	Name : Manasi Hire Roll no: 331070 Prn no: 22320015
Name : Vivek Pahilwan Roll no: 331071 Prn no: 22320030	Name : Aditya Walke Roll no: 331076 Prn no: 22320123

1.Problem Statement:

Managing hostel operations manually can be inefficient and prone to errors, especially with a large number of students. This project aims to develop a Hostel Management System using a DBMS to automate tasks like student registration, room allocation, fee calculation, and record-keeping. The system will help administrators manage hostel activities more efficiently and reduce the chances of mistakes.

The goal is to develop or maintain a database for the data of all the stakeholders, and assets with the following specifications=

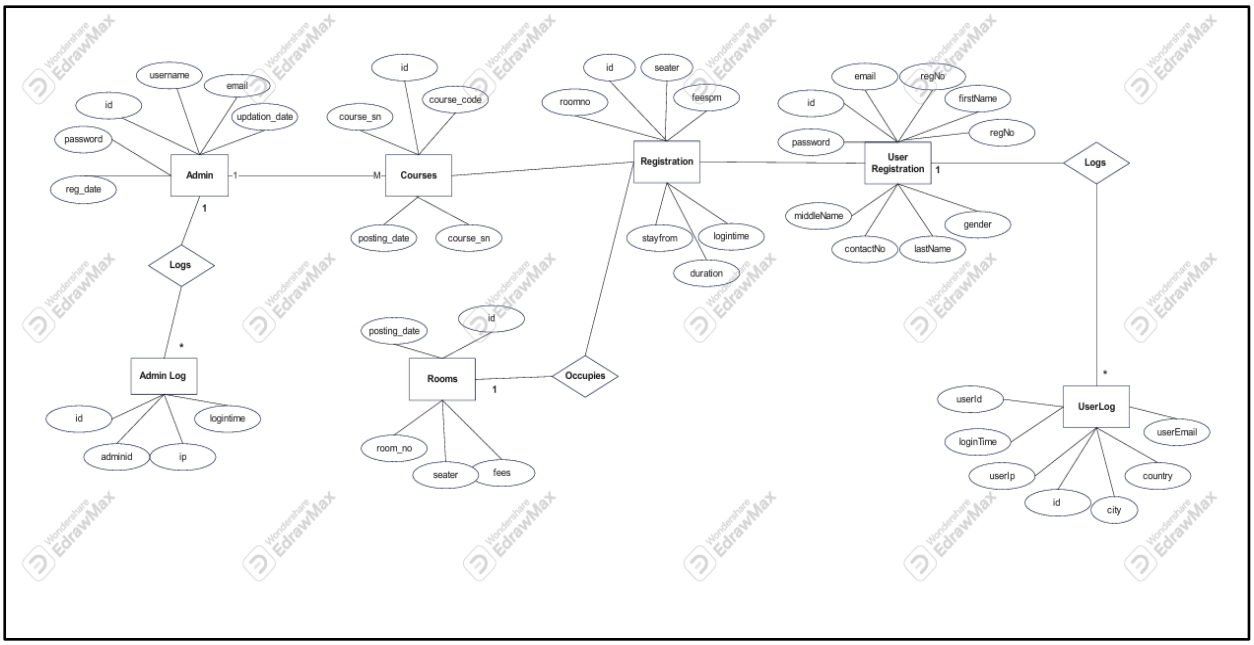
1. To keep proper record and information of a hostel's branch. Hostel Management System plays an important role.

2. Hostel's each branch holds our names, address, number of room and capacity of students.

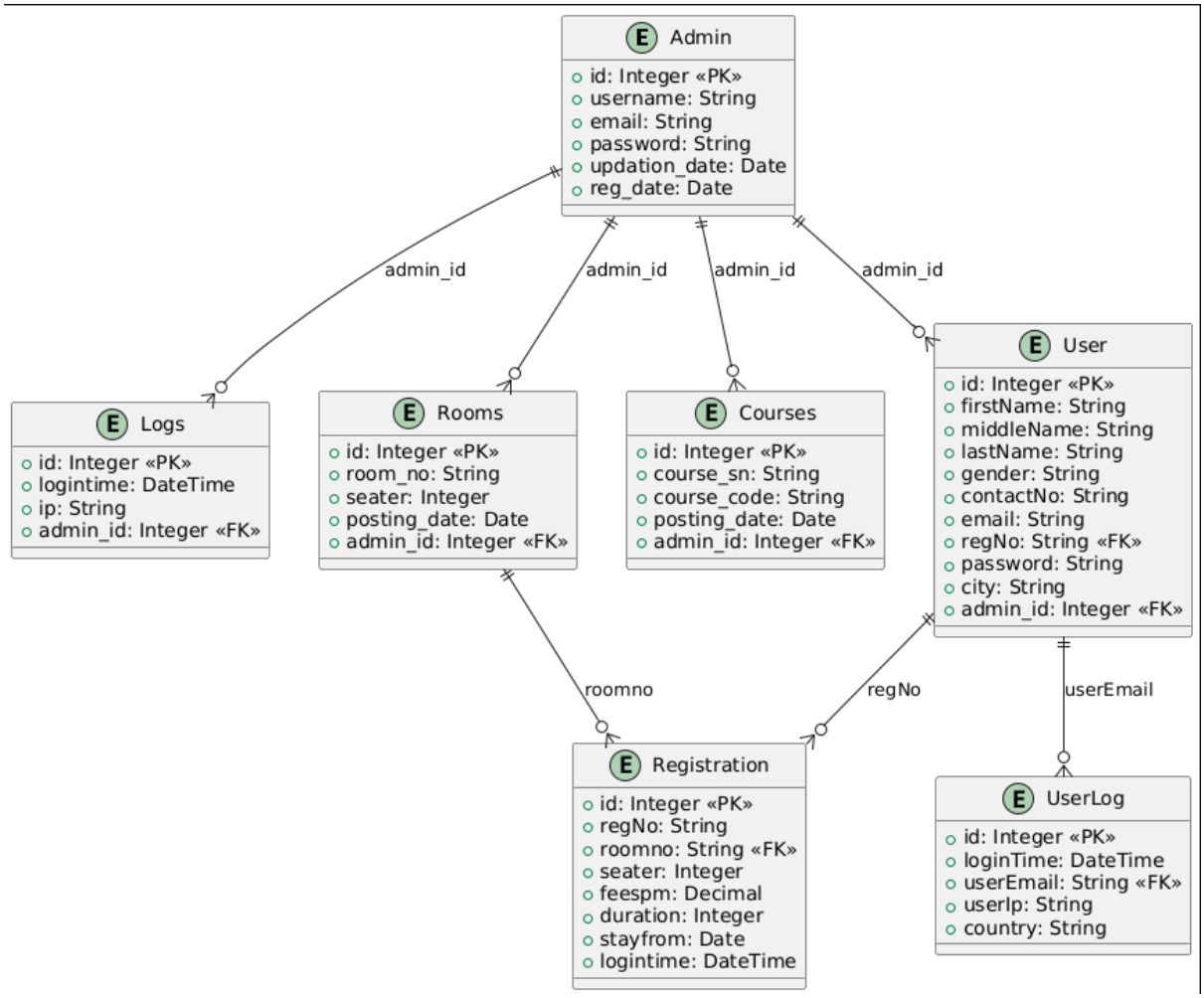
3. Many employees are waiting in hostels having their own name, id, phone number, gender, date of joining, salary, and resigning.

4. In hostel, there are many rooms having their particular room number, capacity of students, and students living in that particular room. For students, their name, phone number, id, address, age, course enrolled in, date of joining and leaving are important.

2. ER-Diagram:


username
email
id
updation_date
password
Admin
reg_date
1
1
Logs
*
Admin Log
id
logintime
adminid
ip
id
course_code
course_sn
M
Courses
posting_date
course_sn
id
seater
roomno
feespm
Registration
stayfrom
logintime
duration
posting_date
id
Rooms
1
Occupies
room_no
seater
fees
email
regNo
id
firstName
password
regNo
User Registration
1
middleName
gender
contactNo
lastName
Logs
*
userId
UserLog
userEmail
loginTime
userIp
country
id
city

3.Schema Diagram:



4.Database Design

4.1 Table Structures and Attributes :

Sr.no	Table_name	Attributes	Data type
1.	admin	id username email password reg_date updation_date	int(11) varchar(255) varchar(255) varchar(300) date
2.	adminlog	id adminid ip logintime -	int(11) int(11) varbinary(16) timestamp
3.	courses	id course_code course_sn course_fn posting_date	int(11) varchar(255) varchar(255) varchar(255) timestamp
4.	registration	id roomno seater feespm foodstatus stayfrom duration course regno firstName middleName lastName gender contactno emailid egycontactno guardianName guardianRelation guardianContactno corresAddress corresCIty corresState corresPincode pmntAddress pmntCity pmnatetState pmntPincode postingDate updationDate - varchar(500)	int(11) int(11) int(11) int(11) int(11) date int(11) varchar(500) varchar(255) varchar(500) varchar(500) varchar(500) varchar(250) bigint(11) varchar(500) bigint(11) varchar(500) varchar(500) bigint(11) varchar(500) varchar(500) varchar(500) int(11) varchar(500) varchar(500) varchar(500) int(11) timestamp (Default: current_timestamp())

5.	rooms	id seater room_no fees posting_date	int(11) int(11) int(11) int(11) timestamp
6.	states	Id state	int(11) (Primary Key) varchar(150)

4.2 Normalization :

Normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. It involves dividing large tables into smaller tables and defining relationships between them. The aim is to eliminate anomalies such as update, delete, and insert anomalies that could occur during database operations.

Example: Students Table

Attributes:

Student_ID (Primary Key)

Student_Name

Student_Age

Room_No

Functional Dependencies:

Student_ID → Student_Name, Student_Age, Room_No

3NF Check:

The table is in 2NF.

There are no transitive dependencies; all non-key attributes depend directly on Student_ID.

5.Implementation Details:

5.1 Frontend Technologies

- **HTML & CSS**: For structuring and styling the user interface (UI).
    - Create forms for student registration, room booking, and fee calculation.
    - Display room availability, registered students, and other dynamic data from the backend.
- **PHP**: For server-side scripting to process requests and interact with the MySQL database.
    - Handle form submissions for student registration and room booking.
    - Fetch data from the MySQL database and display it dynamically (e.g., available rooms, registered students).
    - Calculate fees based on room type and food selection.
- **JavaScript**: Optionally, for dynamic elements like form validation and real-time UI updates without refreshing the page.

5.2 Backend Technology

- **Database Schema**: Tables for students, rooms, bookings, fees, and food options.
- **Students Table**: Stores student details like name, contact, room assigned, etc.
- **Rooms Table**: Details about rooms, seater types, room availability, etc.
- **User Registration Table**: Stores information about which student is assigned which room.
- **MySQL Queries**: To insert, update, and fetch data from the database.
- Insert student details into the students table when registering.
- Calculate fees based on room type and whether the student opts for food, and then update the database accordingly.
- Fetch room availability and student details dynamically to display on the frontend.

5.3 SQL operations :

```sql
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;
```

```sql
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;
```

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

```sql
CREATE TABLE `registration` (
  `id` int(11) NOT NULL,
  `roomno` int(11) NOT NULL,
  `seater` int(11) NOT NULL,
  `feespm` int(11) NOT NULL,
  `foodstatus` int(11) NOT NULL,
  `stayfrom` date NOT NULL,
  `duration` int(11) NOT NULL,
  `course` varchar(500) NOT NULL,
  `regno` varchar(255) NOT NULL,
  `firstName` varchar(500) NOT NULL,
  `middleName` varchar(500) NOT NULL,
  `lastName` varchar(500) NOT NULL,
  `gender` varchar(250) NOT NULL,
  `contactno` bigint(11) NOT NULL,
  `emailid` varchar(500) NOT NULL,
  `egycontactno` bigint(11) NOT NULL,
  `guardianName` varchar(500) NOT NULL,
  `guardianRelation` varchar(500) NOT NULL,
  `guardianContactno` bigint(11) NOT NULL,
  `corresAddress` varchar(500) NOT NULL,
  `corresCIty` varchar(500) NOT NULL,
  `corresState` varchar(500) NOT NULL,
  `corresPincode` int(11) NOT NULL,
  `pmntAddress` varchar(500) NOT NULL,
  `pmntCity` varchar(500) NOT NULL,
  `pmnatetState` varchar(500) NOT NULL,
  `pmntPincode` int(11) NOT NULL,
  `postingDate` timestamp NOT NULL DEFAULT current_timestamp(),
  `updationDate` varchar(500) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

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

--
```

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

--
```

```sql
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;

--
```

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

--
```

DML Commands :

```
INSERT INTO `admin` (`id`, `username`, `email`, `password`, `reg_date`, `updation_date`) VALUES
(1, 'admin', 'admin@mail.com', 'D00F5D5217896FB7FD601412CB890830', '2020-09-08 20:31:45', '2022-09-24');
```

```
INSERT INTO `courses` (`id`, `course_code`, `course_sn`, `course_fn`, `posting_date`) VALUES
(1, 'BTH123', 'B.Tech', 'Bachelor  Of Technology', '2020-09-23 00:45:13'),
(2, 'BCOM18', 'B.Com', 'Bachelor Of Commerce ', '2020-09-23 00:45:13'),
(3, 'BSC296', 'BSC', 'Bachelor  of Science', '2020-09-23 00:45:13'),
(4, 'BCOA55', 'BCA', 'Bachelor Of Computer Application', '2020-09-23 00:45:13'),
(5, 'MCA001', 'MCA', 'Master of Computer Application', '2020-09-23 00:47:13'),
(6, 'MBA777', 'MBA', 'Master In Business Administration', '2020-09-23 00:54:13'),
(7, 'BE069', 'BE', 'Bachelor of Engineering', '2020-09-23 00:59:13'),
(8, 'BIT353', 'BIT', 'Bachelors In Information Technology', '2021-03-07 06:59:05'),
(9, 'MIT005', 'MIT', 'Master of Information Technology', '2022-04-03 13:03:19');
```

```
INSERT INTO `rooms` (`id`, `seater`, `room_no`, `fees`, `posting_date`) VALUES
(1, 5, 100, 5000, '2020-09-20 04:24:06'),
(2, 4, 201, 5000, '2020-09-20 04:24:06'),
(3, 2, 200, 5000, '2020-09-20 04:33:06'),
(4, 3, 112, 5000, '2020-09-20 04:33:30'),
(5, 5, 132, 5000, '2020-09-20 04:28:52'),
(6, 4, 11, 5000, '2021-03-07 05:01:02'),
(7, 2, 269, 5000, '2022-04-03 14:39:22'),
(8, 1, 310, 5000, '2022-04-03 14:41:36'),
(9, 1, 330, 5000, '2022-04-03 14:41:53'),
(10, 1, 1000, 5000, '2022-09-24 11:50:52'),
(11, 3, 1001, 5000, '2022-09-24 11:51:29');
```

```
INSERT INTO `states` (`id`, `State`) VALUES
(1, 'Alabama'),
(2, 'Alaska'),
(3, 'Arizona'),
(4, 'Arkansas'),
(5, 'California'),
(6, 'Colorado'),
(7, 'Connecticut'),
(8, 'Delaware'),
(9, 'Florida'),
(10, 'Georgia'),
(11, 'Hawaii'),
(12, 'Idaho'),
(13, 'Illinois'),
(14, 'Iowa'),
(15, 'Kansas'),
(16, 'Kentucky'),
(17, 'Louisiana'),
(18, 'Maine'),
(19, 'Marryland'),
(20, 'Massachusetts'),
(21, 'Michigan'),
(22, 'Minnesota'),
(23, 'Mississippi'),
(24, 'Missouri'),
(25, 'Nevada'),
(26, 'New Jersey'),
(27, 'New York'),
(28, 'North Carolina'),
(29, 'North Dakota'),
(30, 'Ohio'),
(31, 'Oklahoma'),
(32, 'South Carolina'),
(33, 'South Dakota'),
(34, 'Texas'),
(35, 'Virginia'),
(36, 'Washington');
```

```sql
INSERT INTO `userregistration` (`id`, `regNo`, `firstName`, `middleName`, `lastName`, `gender`, `contactNo`, `email`, `password`, `regDate`, `updationDate`, `passUdateDate`) VALUES
(19, 'CA002', 'Bruce', 'E.', 'Murphy', 'Male', 1346565650, 'bruce@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2020-11-05 04:46:33', '', ''),
(20, 'CA003', 'Richard', 'J.', 'Summers', 'Male', 1325658800, 'richards@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2020-11-05 04:54:33', '', ''),
(21, 'CA004', 'Ross', 'S.', 'Daniels', 'Male', 6958545850, 'ross@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2021-03-05 04:19:44', '06-03-2021 10:15:29', '06-03-2021 10:35:38'),
(22, 'CA005', 'Colin', 'B', 'Greenwood', 'Male', 7541112050, 'colin@gmail.com', 'e10adc3949ba59abbe56e057f20f883e', '2021-03-06 16:29:57', '', ''),
(24, 'CA006', 'Jennifer', 'J.', 'Frye', 'Female', 7895555544, 'jennifer@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:31:50', '', '03-04-2022 08:21:07'),
(25, 'CA007', 'Bonnie', 'J.', 'Lamar', 'Female', 4580001014, 'bonnie@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:51:00', '', ''),
(26, 'CA008', 'Adam', 'A.', 'Rios', 'Male', 4785690010, 'adam@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:52:28', '', ''),
(27, 'CA009', 'Nancy', 'W.', 'Vasquez', 'Female', 3547777770, 'nancy@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:53:19', '', ''),
(28, 'CA010', 'Jerry', 'A.', 'Burdine', 'Male', 8520001450, 'jerry@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:53:58', '', ''),
(29, 'CA011', 'James', 'K.', 'Fischer', 'Male', 4785470014, 'jamesf@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:54:44', '', ''),
(30, 'CA012', 'Darlene', 'D.', 'Kenyon', 'Female', 3547896580, 'darlene@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:57:04', '', ''),
(31, 'CA013', 'Joseph', 'H.', 'Peterson', 'Male', 4587450010, 'joseph@mail.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-04-03 14:57:51', '', ''),
(32, 'CA014', 'Liam', 'K.', 'Moore', 'Male', 7854441014, 'liamoore@mail.com', '5f4dcc3b5aa765d61d8327deb882cf99', '2022-04-03 15:00:04', '', ''),
(33, 'CD/2545/21', 'Erick', 'Omundi', 'Omari', 'Male', 785963245, 'resress4078@outlook.com', 'e10adc3949ba59abbe56e057f20f883e', '2022-09-24 11:55:42', '', '');
```

6. FUTURE SCOPE

- **Enhanced Features**: Additional features such as online payment gateways and automated invoicing can be integrated into the system for a more comprehensive financial management experience.
- **Mobile Application**: Developing a mobile-friendly version of the system would allow users, especially students, to manage room bookings, fee payments, and other tasks more conveniently on their smartphones.
- **Advanced Analytics**: Integrating data analytics tools to generate reports and insights into student preferences, occupancy rates, and financial health would assist in more informed decision-making.
- **Cloud Integration**: Migrating the system to cloud-based platforms can provide scalability, allowing for real-time data access and backup, while also enabling multiple hostel branches to use the same system without geographical limitations
- **AI and Automation**: Future iterations could incorporate AI for automating room allocation based on preferences and optimizing hostel resource usage, such as energy or space management.

7.CONCLUSION

The Hostel Management System successfully automates essential hostel operations, providing a robust solution for managing student registrations, room allocations, fee calculations, and record-keeping. By leveraging DBMS technology, the system reduces manual errors, improves operational efficiency, and enhances data management. This project has achieved its goal of creating a streamlined platform for administrators to handle hostel-related activities with greater ease and reliability. The combination of a well-structured database design and an intuitive user interface ensures that all stakeholders can access accurate and real-time data, leading to a more organized and effective hostel management process.