MANIPAL UNIVERSITY
JAIPUR

Minor Project

Project - Synopsis

MCA - III Sem

Subject Code: CA7131

Submitted By

JATIN ISRANI

23FS20MCA00081

DIVYANSHU SINGH CHAUHAN

23FS20MCA00044

Faculty Coordinator

Dr. Linesh Raja, Associate Professor

Dr. Govind Murari Upadhyay, Assistant Professor

DEPARTMENT OF COMPUTER APPLICATIONS

A. Title of the Project:

Hostel Visitor Interface

This project is titled **Hostel Visitor Interface**, but the website is dedicated to NS Hostel. The NS Hostel website is created to attract students by offering clear details about room options, amenities, and how to book. It has an easy-to-use design that works well on all devices, plus a dark mode for comfortable browsing. The site encourages bookings with clear buttons and a visually appealing layout that shows why staying at NS Hostel is a great choice. Overall, it aims to increase online visibility and get more students interested in staying at the hostel.

B. Team Details:

- **Team Leader**: Divyanshu Singh Chouhan
- **Team Members**: Jatin Israni
- **Affiliation**: Manipal University Jaipur
    - Frontend, Database : Divyanshu Singh Chouhan
    - Backend , Database : Jatin Israni

C. Introduction:

The Hostel Visitor Interface project is all about creating an easy-to-use website for NS Hostel. This site aims to attract students by giving them all the information they need about rooms, amenities, and how to book a stay. It features a clean design that looks great on any device—be it a computer, tablet, or smartphone. Plus, there's a dark mode for comfortable browsing at night. The website showcases the benefits of staying at NS Hostel and encourages students to make bookings. The main goal is to boost the hostel's online visibility and get more students interested in staying there. For this project, we used a few key technologies: **MongoDB, Express.js, React.js, and Node.js**. (MERN stack) By combining these tools, we aim to create a solid and efficient website that offers a great experience for users and is easy to manage.

D. Objectives:

The objective of our website is to provide an informative and visually appealing platform for NS Hostel, aimed at attracting students who are seeking comfortable and affordable accommodation near Manipal University, Jaipur. The site highlights key features of the hostel, including room options, amenities, and booking details, to give potential residents a clear understanding of what NS Hostel offers.

Additionally, the website ensures:

- A user-friendly experience with smooth navigation and responsive design across different devices.
- Clear calls-to-action, such as booking links, to drive engagement and conversions.
- Dark mode toggle to enhance accessibility and user comfort.
- The ultimate goal is to create a compelling online presence that helps drive hostel bookings and communicates the benefits of staying at NS Hostel.

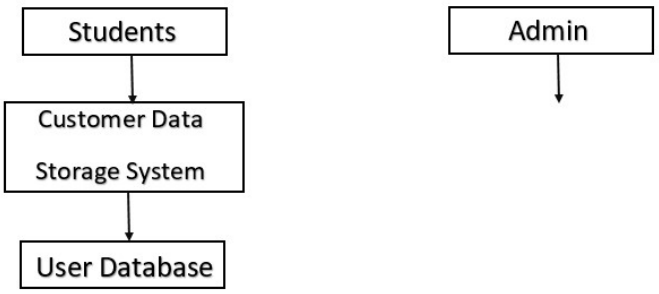
E. Data Flow Diagrams (DFD):

Zero Level DFD (Context Level) for Customer Data Storage: This diagram provides a high-level overview of how customer data is handled in the system.

Components:

- **External Entities:**
    - **Students:** Users providing their information.
    - **Admin:** The person managing customer data.

Diagram Representation:

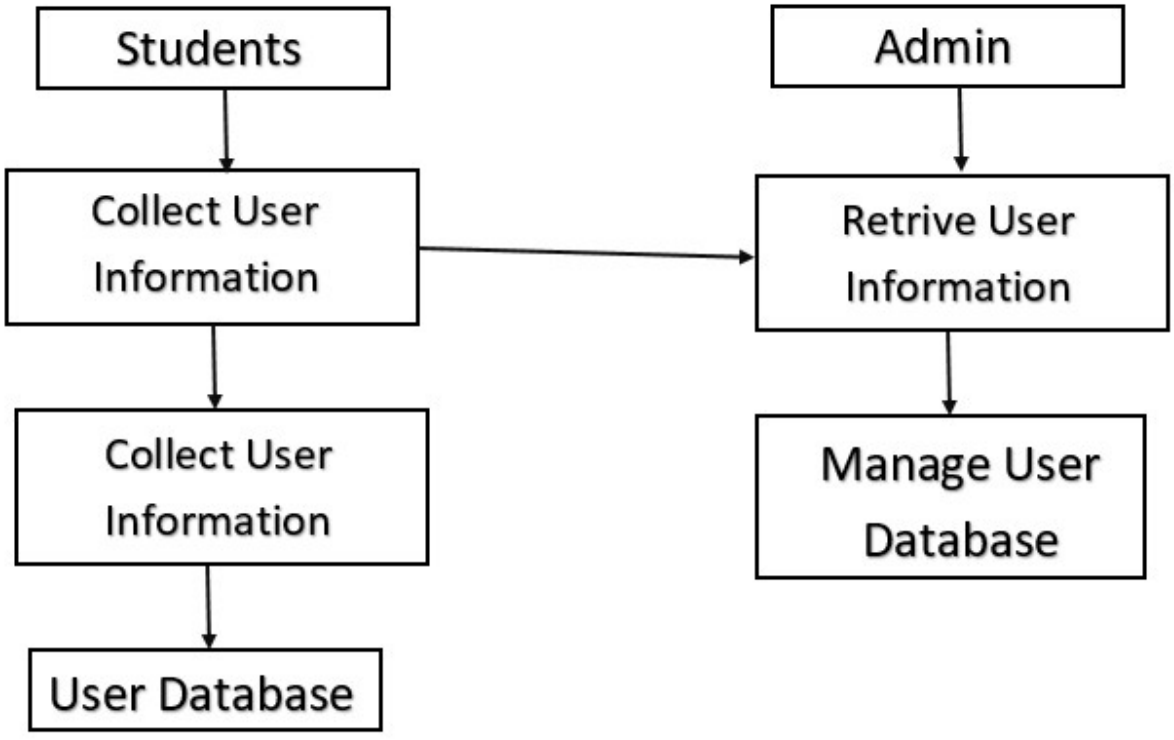


First Level DFD for Customer Data Storage: This diagram breaks down the main customer data storage process into sub-processes.

Components:

- **Processes:**
    - **Collect User Information:** Gathering details from students.
    - **Store User Information:** Saving the collected data in the database.
    - **Retrieve User Information:** Admin retrieves and manages user data.

Diagram Representation:



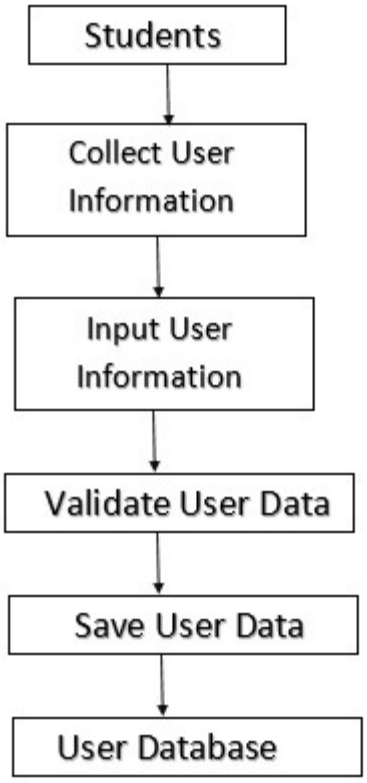
Second Level DFD for Collecting and Storing User Information

This level provides more detail on how user information is collected and stored.

Components:

- **Processes:**
    - **Input User Data:** Students enter their information.
    - **Validate User Data:** Check the accuracy of the provided data.
    - **Save User Data:** Store the validated information in the database.

Diagram Representation:



F. ER Diagram:

ER Diagram Components

Entities:

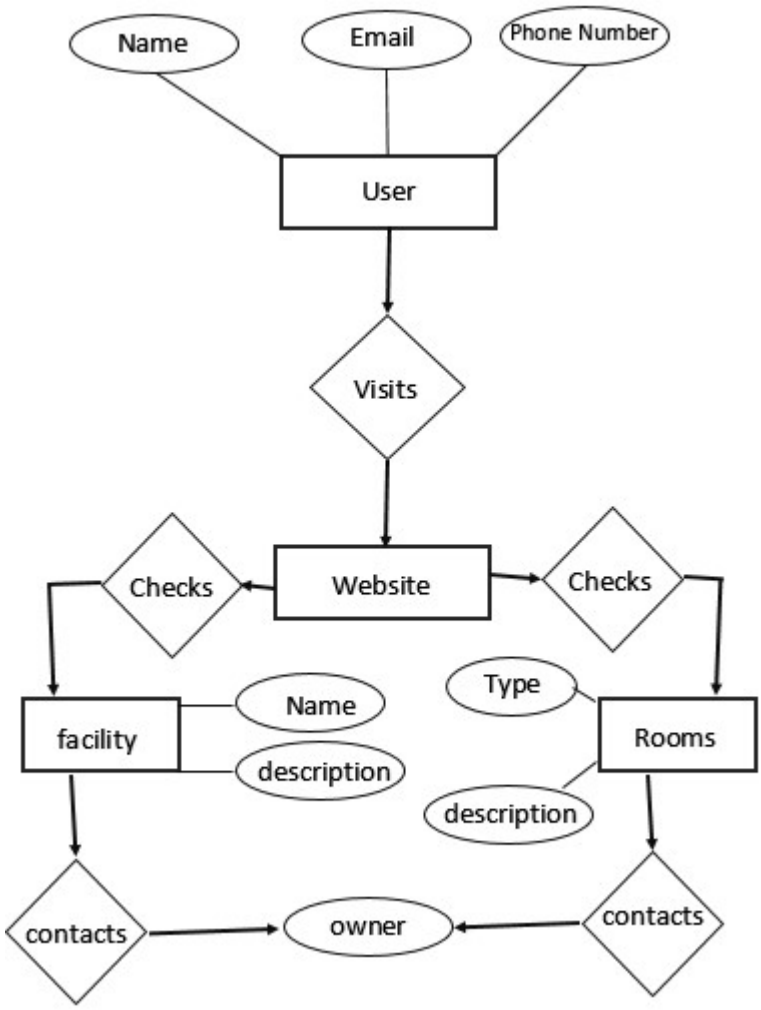
1. **User**
    - **Attributes:**
        - **UserID** (Primary Key)
        - **Name**
        - **Email**
        - **Phone Number**
2. **Room**
    - **Attributes:**
        - **RoomID** (Primary Key)
        - **RoomType**
        - **Price**
        - **Description**
3. **Facility**
    - **Attributes:**
        - **FacilityID** (Primary Key)
        - **FacilityName**
        - **Description**

Relationships:

- **User - Website Interaction:** Users can provide their details through the website (e.g., registration or inquiries), but there are no booking processes included in this model.

- **Room - Facility:** A room can have multiple facilities associated with it, indicating a one-to-many relationship (one room can offer many facilities).

ER Diagram Representation:-



G. Project Timeline:

The **project timeline** outlines the key phases of development for the **Hostel Visitor Interface** project, detailing the start and end dates of each task:

Task	Start Date	End Date	Duration
Requirements Gathering	[12-9-2024]	[18-9-2024]	7 Days
System Design (DFD, ERD)	[19-9-2024]	[23-9-2024]	5 Days
Frontend Development	[24-9-2024]	[8-10-2024]	15 Days
Backend Development (Node.js)	[9-10-2024]	[25-10-2024]	17 Days
Database Design (MongoDB)	[25-10-2024]	[10-11-2024]	16 Days
Integration and Testing	[11-11-2024]	[25-11-2024]	15 Days
Final Deployment	[26-11-2024]	[31-11-2024]	6 Days

Each phase includes specific deliverables like DFDs, frontend and backend features, and testing to ensure that the system works efficiently before final deployment.

H. Tools / Platform, Hardware and Software Requirement Specifications:

1. Platform:

- **Frontend**: Developed using **CSS, JavaScript ,Recat.js** for creating a responsive and interactive user interface.
- **Backend**: Server-side programming is handled using **Node.js** for processing user requests and managing data.
- **Database**: **MongoDB** is used to store user information, product details, order data, and payment records.
- **Web Server**: **Express.js** is required to host the website locally or on a cloud platform.

2. Hardware Requirements:

- **Processor**: Minimum Intel Core i3 or equivalent for development and testing.
- **RAM**: At least 2 GB of RAM to ensure smooth performance during testing.
- **Storage**: 500 GB HDD or SSD for storing files, databases, and project resources.

3. Software Requirements:

- **Operating System**: The system can be developed and tested on **Windows 10** or **Linux** (Ubuntu).
- **IDE**: **VS Code**, or any text editor for writing and debugging code.
- **Web Browsers**: **Google Chrome**, **Firefox**, or **Edge** for testing the website's compatibility across platforms.

I. References:

Here are some resources and references that can help you get started:

Tutorials and Guides:-

1. MDN Web Docs: Building a Hostel Visitor Interface with Vanilla JavaScript

- A comprehensive guide on building a simple Hostel Visitor Interface using HTML, CSS, and JavaScript.

2. FreeCodeCamp: Build a Hostel Visitor Interface with React

- This tutorial walks you through building a Hostel Visitor Interface using React, covering component creation, state management, and event handling.

3. The Odin Project: JavaScript Hostel Visitor Interface

- A project-based tutorial that guides you through building a Hostel Visitor Interface using pure JavaScript.

GitHub Repositories:-

1. Simple Hostel Visitor Interface using React

- A basic implementation of a Hostel Visitor Interface using React. Explore the code and learn how to manage state and props.

2. MERN Stack Hostel Visitor Interface

- A more complex example using the MERN stack (MongoDB, Express, React, Node.js). This repository demonstrates full-stack development.

Documentation:-

1. React Documentation

- Official documentation for React, including guides and tutorials on building applications.

2. Node.js Documentation

- Official documentation for Node.js, covering everything from installation to building RESTful APIs.