**Campus Mediate - Project Documentation**

**1. Introduction**

**Project Name:** Campus Mediate  
**Description:** A web application that connects **hostel managers** and **students**, allowing them to view available accommodations, make advance bookings, check room availability, and process online payments.  
**Purpose & Objectives:**

* Provide **students** with an easy-to-use platform for finding and booking hostels.
* Enable **hostel managers** to manage bookings and payments efficiently.
* Generate reports on bookings and transactions.
* Provide a geo-map for **locating hostels near campus.** **Target Audience:** Students, hostel managers, and university staff.

**2. Project Overview**

**System Architecture**

Campus Mediate follows a **client-server architecture**, where:

* **Frontend:** Developed with ***HTML****,* ***CSS****,* and ***JavaScript*** for an ***intuitive UI.***
* **Backend:** Built using ***Node.js*** and ***Express.js*** for handling ***API requests.***
* **Database:** ***MongoDB*** for real-time data operations and ***MySQL*** for structured financial transactions.

**Key Features:**

* Hostel Listings & Availability Check
* Online Booking & Payments
* User Authentication & Role Managemen­­t
* Reports Generation
* Hostel Geo-Mapping

**3. Installation Guide**

**Prerequisites**

* Node.js and npm installed
* MongoDB and MySQL set up.
* VS Code editor.
* Git for version control

**Setup Steps**

1. Clone the repository:
2. git clone https://github.com/your-repo/campus-mediate.git
3. cd campus-mediate
4. Install dependencies:
5. npm install
6. Set up environment variables in a .env file.
7. Start the development server:
8. npm start

**4. Usage Guide**

**Running the Application**

* Access the frontend via http://localhost:3000
* API server runs on http://localhost:5000

**User Roles & Permissions**

* **Student:** Can browse hostels, make bookings, and payments.
* **Hostel Manager:** Can list hostels, manage bookings, and view reports.
* **Admin:** Manages all users, hostels, and transactions.

**5. API Documentation**

**Endpoints Overview**

* POST /api/auth/register - User registration
* POST /api/auth/login - User login
* GET /api/hostels - Fetch all hostels
* POST /api/bookings - Make a booking
* GET /api/payments - Retrieve payment history

**6. Database Schema**

**MongoDB (Real-time Operations)**

* **User Collection**: Stores student and manager details.
* **Hostel Collection**: Stores hostel listings and availability.
* **Bookings Collection**: Stores student booking details.

**MySQL (Financial Transactions)**

* **Payments Table**: Stores payment transactions with timestamps.
* **Reports Table**: Maintains structured booking and payment records.

**7. Code Structure**

campus-mediate/

├── backend/

│ ├── controllers/

│ ├── models/

│ ├── routes/

│ ├── config/

│ ├── server.js

│

├── assets/

│ ├── css/├── hostel-details.css

│ ├── hostel-upload.css

│ ├── mediate2.css

│ ├── js/├── hostel-details

│ ├── hostel-upload.js

│ ├── mediate2.js

│

├── ├── images/├── image-1

│ ├── image-2

│ ├── image-3

│

├── mediate.html

├── hostel-upload.html

├── hostel-details.html

├── checkinform.html

│

├── .env

├── package.json

├── README.md

**8. Deployment Guide**

**Frontend Deployment (Netlify)**

1. **Push your project to GitHub:**
2. git add .
3. git commit -m "Initial commit"
4. git push origin main
5. **Deploy on Netlify:**
   * Sign up at [Netlify](https://www.netlify.com/).
   * Click **"New Site from Git"** and select your GitHub repository.
   * Configure **build settings:**
     + **Publish Directory:** The root directory containing index.html
   * Click **Deploy** and get a live URL.

**Backend Deployment (AWS)**

1. **Set up an AWS EC2 instance** and install Node.js.
2. **Push backend code to AWS and configure environment variables.**
3. **Use PM2 to keep the server running:**
4. pm2 start server.js
5. **Set up a reverse proxy with Nginx for better performance.**

**9. Security Considerations**

* **JWT Authentication** for user sessions.
* **Role-based access control** for different users.
* **Encryption** for sensitive data.

**10. Troubleshooting & FAQs**

* **Issue:** Database not connecting?
  + **Solution:** Ensure MongoDB/MySQL services are running and .env variables are correctly set.
* **Issue:** Payment processing fails?
  + **Solution:** Check API key configuration and transaction logs.

**11. Future Improvements**

* Mobile application version.
* AI-based recommendations for students.
* Advanced analytics for hostel managers.

**12. Contributors & Acknowledgments**

* **Developers:** Your Name & Team
* **Tools Used:** HTML, CSS, JavaScript, Node.js, Express, MongoDB, MySQL, Stripe API

**13. License & Legal Information**

* Licensed under MIT License.
* Terms of use should be agreed upon before accessing the platform.

This documentation serves as a guide for developers, administrators, and users of **Campus Mediate**. 🚀