**Full Stack Development with MERN**

**API Development and Integration Report**

|  |  |
| --- | --- |
| Date | 20-07-2024 |
| Team ID | SWTID1720019632 |
| Project Name | Project - House Rent App Using Mern |
| Maximum Marks |  |

**Project Title:** House Rent App Using Mern  
**Date:** 20-07-2024  
**Prepared by:** Ganesh.S

**Objective**  
The objective of this report is to document the API development progress and key aspects of the backend services implementation for the House Rent App Using Mern project.

**Technologies Used**

* **Backend Framework:** Node.js with Express.js
* **Database:** MongoDB

**Project Structure**

A screenshot of a computer

Description automatically generated

**Key Directories and Files**

1. **/Controllers**
   * Contains functions to handle requests and responses.
   * propertyController.js performs Getting Property Details Uploaded by Owner from DB. Individual Property Details for Dynamic Pages. Uploading Property Details from Owner To DB
   * userController .js performs registration and login
2. **/models**
   * Includes Mongoose schemas and models for MongoDB collections.
   * propertyModel.js contains the Mongoose schemas and models for Property collections
   * userModel.js contains the Mongoose schemas for models User collections
3. **/routes**
   * Defines the API endpoints and links them to controller functions.
   * propertyRoutes.js API endpoints (/propertyget, /addproperty, /propertyget/:id) in an Express router and links them to corresponding controller functions (PropertyDetails, uploadProperty, IndvidualDetails) from propertyController. Exports the router for use in the main application.
   * userRoutes.js API endpoints (`/register`, `/login`) in an Express router and links them to corresponding controller functions (`Register`, `Login`) from `userController`. Exports the router for use in the main application.

**API Endpoints**  
A summary of the main API endpoints and their purposes:

**User Authentication**

* **POST** **/register -** Registers a new renter
* **GET /login** - Authenticates a renter

**Home page**

* **GET /home** - displays quote and buttons
* **GET /about**– Retrieves all details about us

**Property**

* **GET /details** - Retrieves all property details owner uploads
* **POST /owner** – add property details
* **GET /prp/:id –** get the property details that owner uploaded

**Integration with Frontend**  
The backend communicates with the frontend via RESTful APIs. Key points of integration include:

* **User Authentication:** Tokens are passed between frontend and backend to handle authentication.
* **Data Fetching:** Frontend components make API calls to fetch necessary data for display and interaction.

**Error Handling and Validation**  
Describe the error handling strategy and validation mechanisms:

* **Error Handling:** Centralized error handling using middleware.
* **Validation:** Input validation using libraries like Joi or express-validator.

**Security Considerations**  
Outline the security measures implemented:

* **Authentication:** Secure token-based authentication.
* **Data Encryption:** Encrypt sensitive data at rest and in transit.