**API Development and Integration Report**

|  |  |
| --- | --- |
| Date | 21-07-24 |
| Team ID | SWTID1720019378 |
| Project Name | Project - HouseHunt |
| Maximum Marks |  |

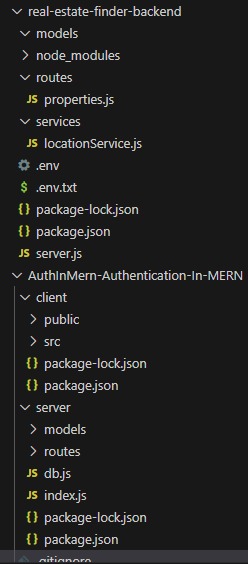
**Project Title:** HouseHunt  
**Date:** 21-07-24  
**Prepared by:** Apoorva Saini

**Objective**  
 The objective of this report is to document the API development progress and key aspects of the backend services implementation for the HouseHunt project.

**Technologies Used**

* **Backend Framework:** Node.js with Express.js
* **Database:** MongoDB
* **Authentication:** JWT (JSON Web Token)

**Project Structure**



**Key Directories and Files**

1. **/controllers**
   * Contains functions to handle requests and responses.
2. **/models**
   * Includes Mongoose schemas and models for MongoDB collections.
3. **/routes**
   * Defines the API endpoints and links them to controller functions.
4. **/middlewares**
   * Custom middleware functions for request processing.
5. **/config**
   * Configuration files for database connections, environment variables, etc.

**6. /services**

* Contains external service integration logic, such as the Street Map API.

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

**User Authentication**

* **POST /api/user/register** - Registers a new user.
* **POST /api/user/login** - Authenticates a user and returns a token.

**User Management**

* **GET /api/user/-** Retrieves user information by ID.
* **PUT /api/user/**- Updates user information by ID.

**Workout Plans**

* **GET /api/workoutplans** - Retrieves all workout plans.
* **POST /api/workoutplans** - Creates a new workout plan.

**Equipment**

* **GET /api/equipment** - Retrieves all equipment.
* **POST /api/equipment** - Adds new equipment.

**Monthly Plans**

* **GET /api/monthlyplans** - Retrieves all monthly plans.
* **POST /api/monthlyplans** - Creates a new monthly plan

**Street Map Integration**

* **GET /api/streetmap** - Retrieves map data for specified locations using the Street Map API.
* **POST /api/streetmap** - Fetches and stores new map data from the Street Map API.

.

**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 transion