**Requirements**

1. User Authentication

-Allows users to securely register, log in, and access protected resources.

API Endpoints:

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/register/ | Register a new user |
| POST | /api/login/ | Authenticate user login |
| GET | /api/profile/ | Fetch logged-in user’s profile (auth required) |

Input Specifications:

**POST /api/register/**

{

"name": "John Doe",

"email": "john@example.com",

"password": "SecurePassword123"

}

**POST /api/login/**

{

"email": "john@example.com",

"password": "SecurePassword123"

}

Output Specifications:

Success (Register/Login):

{

"message": "User registered successfully",

"token": "JWT\_TOKEN\_HERE"

}

Validation Rules:

* email: Must be unique and valid format
* password: Minimum 8 characters, must contain letters and numbers
* name: Required, max 50 characters

Performance Criteria:

* Authentication operations should complete in < 300ms
* Tokens should expire in a secure, configurable time

1. Property Management

Allow hosts to create, update, view, and delete property listings.

API Endpoints:

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/properties/ | Create a new property |
| GET | /api/properties/ | List all available properties |
| GET | /api/properties/:id/ | View a single property |
| PUT | /api/properties/:id/ | Update property (host only) |
| DELETE | /api/properties/:id/ | Delete property (host only) |

Input Specifications:

POST /api/properties/

{

"title": "Cozy Beach House",

"location": "Mombasa, Kenya",

"price\_per\_night": 120.00,

"description": "Ocean-view, 2-bedroom house",

"host\_id": 1

}

Output Specifications:

{

"id": 101,

"title": "Cozy Beach House",

"location": "Mombasa, Kenya",

"price\_per\_night": 120.00,

"host": {

"id": 1,

"name": "Jane Doe"

}

}

Validation Rules:

* title: Required, max 100 characters
* location: Required
* price\_per\_night: Must be positive float
* host\_id: Must refer to an existing user with role = "host"

Performance Criteria:

* Fetch properties in < 500ms with pagination
* Allow filtering by location and price

1. Booking System

Allow guests to book properties for specific dates and manage bookings.

API Endpoints:

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/bookings/ | Create a new booking |
| GET | /api/bookings/ | List user’s bookings |
| GET | /api/bookings/:id/ | View booking detail |

Input Specifications:

POST /api/bookings/

{

"user\_id": 3,

"property\_id": 101,

"start\_date": "2025-07-10",

"end\_date": "2025-07-15"

}

Output Specifications:

{

"id": 5001,

"user\_id": 3,

"property\_id": 101,

"start\_date": "2025-07-10",

"end\_date": "2025-07-15",

"status": "confirmed"

}

**Validation Rules:**

* start\_date and end\_date must be in future
* end\_date must be after start\_date
* Prevent overlapping bookings for the same property
* Ensure user is authenticated

**Performance Criteria:**

* Booking confirmation should respond within < 400ms
* System should handle concurrency