**Apartment Rental System API Documentation**

**Introduction**

Welcome to the Online Apartment Rental System API documentation. This API allows users to interact with the apartment rental system, including user authentication, apartment listings, user requests, and admin operations. The API is built using Tomcat Server 9 and follows the Servlet MVC pattern.

**Base URL**

**Base URL:** https://api.apartmentrentalsystem.com

**Authentication**

**Login**

Endpoint: /auth/login

Method: POST

Description: Authenticate the user and obtain an access token.

Request:

Body:

username: User's username

password: User's password

Response:

access\_token: JWT token for authentication

**Logout**

Endpoint: /auth/logout

Method: POST

Description: Logout the user and invalidate the access token.

Headers:

Authorization: Bearer {access\_token}

**User Registration**

**Register User**

Endpoint: /users/register

Method: POST

Description: Register a new user.

Request:

Body:

username: User's username

password: User's password

Other registration details

Response:

message: Registration successful

**Apartment Listings**

Get Apartments

Endpoint: /apartments

Method: GET

Description: Get a list of all apartments.

Query Parameters:

filter: Filter apartments based on criteria (e.g., Rental cost, 2BHK/3BHK, Floor space, rented out/Vacant)

Headers:

Authorization: Bearer {access\_token}

Response:

List of apartments

**User Requests**

Get User Requests

Endpoint: /requests

Method: GET

Description: Get all requests made by registered users.

Headers:

Authorization: Bearer {access\_token}

Response:

List of user requests

**Submit Request**

Endpoint: /requests/submit

Method: POST

Description: Submit a request to rent an apartment.

Request:

Body:

apartment\_id: ID of the selected apartment

comments: Additional comments from the user

Headers:

Authorization: Bearer {access\_token}

Response:

message: Request submitted successfully

**Admin Operations**

**Add Apartment**

Endpoint: /admin/apartments

Method: POST

Description: Add a new apartment.

Request:

Body:

Apartment details

Headers:

Authorization: Bearer {access\_token}

Response:

message: Apartment added successfully

**Update Apartment**

Endpoint: /admin/apartments/{apartment\_id}

Method: PUT

Description: Update apartment details.

Request:

Body:

Updated apartment details

Headers:

Authorization: Bearer {access\_token}

Response:

message: Apartment details updated successfully

**Remove Apartment**

Endpoint: /admin/apartments/{apartment\_id}

Method: DELETE

Description: Remove an apartment.

Headers:

Authorization: Bearer {access\_token}

Response:

message: Apartment removed successfully

**Approve Request**

Endpoint: /admin/requests/{request\_id}/approve

Method: POST

Description: Approve a user request.

Headers:

Authorization: Bearer {access\_token}

Response:

message: Request approved successfully

**Deny Request**

Endpoint: /admin/requests/{request\_id}/deny

Method: POST

Description: Deny a user request.

Headers:

Authorization: Bearer {access\_token}

Response:

message: Request denied successfully

**Logging Out**

Logout

Endpoint: /auth/logout

Method: POST

Description: Logout from the system.

Headers:

Authorization: Bearer {access\_token}

Response:

message: Logout successful