Rentify-Presidio Challenge

Rentify is an application where sellers and buyers can meet to facilitate rental transactions.

Technologies Used

- Frontend: React.js, PrimeReact, PrimeFlex
- Backend: Spring Framework, PostgreSQL(Render)
- Hosting:
- Frontend: Vercel(https://rentify-react-ten.vercel.app/)
- Backend: Render (including PostgreSQL)

Note

The application may take up to 1 minute to load due to being hosted on free services.

Getting Started

To get a local copy up and running follow these simple steps.

- Prerequisites
- Node.js
- npm (or yarn)
- Java (JDK 22)
- PostgreSQL

Installation
Clone the repository:

git clone https://github.com/batrick-swaistan/rentify-react.git

Navigate to the project directory:

cd Presideo_Full_Stack

Install frontend dependencies:

cd rentify_reactjs
npm install

Build and start the frontend:

npm run build

npm start

Navigate to the backend directory and install dependencies:
cd .Presideo_Full_Stack/Rentify-Backend
./mvnw install

Update application properties: Update the application.yml file with your PostgreSQL database credentials.

Run the backend application:
./mvnw spring-boot:run

Usage
Once both the frontend and backend are running, you can access the application in your web browser at http://localhost:3000.

API Documentation
This document outlines the endpoints available in the Rentify API, categorized by user roles: Seller, Buyer, and Whitelisted (publicly accessible) URLs.

Authentication

All authenticated endpoints require a Bearer token, which is issued upon successful login. This token must be included in the Authorization header of each request.

Seller Endpoints
These endpoints are designed for sellers to manage their properties and view seller-related details.

SpringBoot Server: https://rentify-api-rentify-api.onrender.com

Add Property

- URL: /rentify/api/seller/addproperty
- Method: POST
- Description: Allows sellers to add a new property to their listing.

Edit Property

- URL: /rentify/api/seller/editproperty/**
- Method: PUT
- Description: Allows sellers to edit an existing property. The property ID should be specified in the URL.

Delete Property

- URL: /rentify/api/seller/deleteProperty/**
- Method: DELETE
- Description: Allows sellers to delete a property. The property ID should be specified in the URL.

Get All Sellers

- URL: /rentify/api/seller/getallsellers
- Method: GET
- Description: Retrieves a list of all sellers.

Get Properties by Seller

- URL: /rentify/api/seller/propertybyseller/**
- Method: GET
- Description: Retrieves all properties listed by a specific seller. The seller ID should be specified in the URL.

Seller Details
- URL: /rentify/api/seller/sellerdetails
- Method: GET
- Description: Retrieves details about the logged-in seller.

Get Property Details
-
- URL: /rentify/api/seller/property/**
- Method: GET
- Description: Retrieves details of a specific property. The property ID should be specified in the URL.

Buyer Endpoints
These endpoints are designed for buyers to interact with properties and view seller details.

Like Property

- URL: /rentify/api/buyer/like/**
- Method: POST
- Description: Allows buyers to like a property. The property ID should be specified in the URL.

View Seller Details

- URL: /rentify/api/buyer/viewSellerDetails/**
- Method: GET
- Description: Allows buyers to view details of a seller. The seller ID should be specified in the URL.

Whitelisted (Public) Endpoints
These endpoints are publicly accessible and do not require authentication.

Authentication

- URL: /rentify/v1/auth/**
- Method: Various (GET, POST, etc.)
- Description: Endpoints related to authentication such as login, registration, and password recovery.
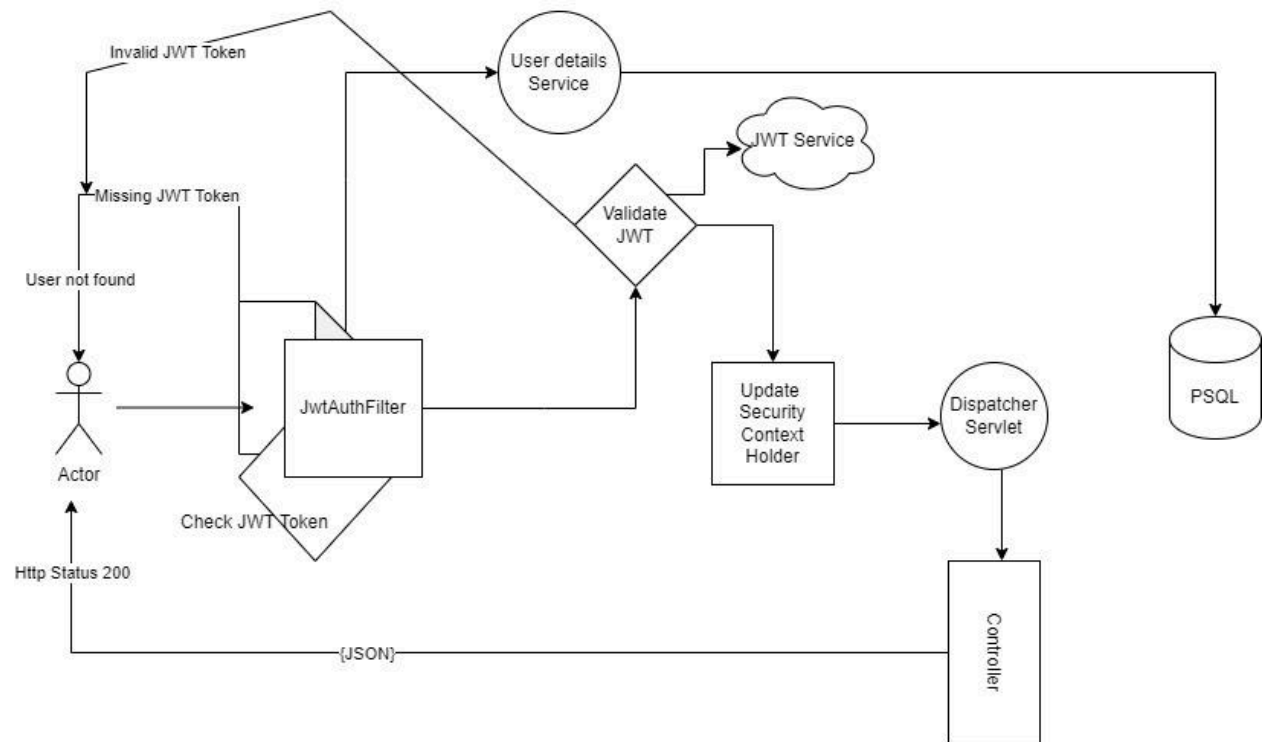
Get All Properties

- URL: /rentify/api/buyer/allProperties
- Method: GET
- Description: Retrieves a list of all properties available on the platform.

Summary
- **Seller Endpoints:** CRUD operations for properties, retrieving seller details, and managing seller-specific information.
- **Buyer Endpoints:** Interacting with properties (like) and viewing seller information.
- Whitelisted Endpoints: Publicly accessible endpoints related to authentication and retrieving all properties.
- This structured approach ensures that each user role has access to the appropriate endpoints, facilitating efficient property management and interaction within the Rentify platform.

Security Architecture

DB ERD :

public

seller_details

- id bigint (IDENTITY)
- email character varying(255)
- firstname character varying(255)
- lastname character varying(255)
- mobile character varying(255)
- password character varying(255)
- role character varying(255)

public

property_details

- id bigint (IDENTITY)
- address character varying(255)
- available date
- bedrooms integer
- image bytea
- likes integer
- name character varying(255)
- rate double precision
- seller_id bigint