# Rentify

## Overview

This document provides detailed information about the Real Estate Application, which allows users to register and manage real estate properties. Users can register as either a Seller or a Buyer. Sellers can list properties, view their listed properties, and get notifications when a buyer shows interest. Buyers can browse available properties, get details of the property owners, and like properties they are interested in. Notifications are sent via email using Nodemailer. The frontend of the application is built with React.

## Table of Contents

1. [Features](#features)

2. [Prerequisites](#prerequisites)

3. [Installation](#installation)

4. [Usage](#usage)

- [Registration and Authentication](#registration-and-authentication)

- [Seller Functionalities](#seller-functionalities)

- [Buyer Functionalities](#buyer-functionalities)

5. [Email Notification Setup](#email-notification-setup)

6. [API Endpoints](#api-endpoints)

7. [Frontend Implementation](#frontend-implementation)

8. [Contributing](#contributing)

9. [License](#license)

10. [Contact](#contact)

## Features

### Seller

- Register and log in as a Seller.

- Add new properties.

- View all properties listed by the seller.

- Receive email notifications when a buyer shows interest in a property.

### Buyer

- Register and log in as a Buyer.

- Browse all available properties.

- Get owner details via email.

- Like properties and show interest.

## Prerequisites

- Node.js installed

- MongoDB installed and running

- Nodemailer configured with your email service provider (e.g., Gmail, SMTP server)

## Installation

1. Navigate to the project directory:

cd real-estate-app

2. Install dependencies:

npm install

3. Create a `.env` file in the root directory and add your environment variables:

PORT=3000

MONGODB\_URI=your\_mongodb\_uri

JWT\_SECRET=your\_jwt\_secret

EMAIL\_SERVICE=your\_email\_service

EMAIL\_USER=your\_email@example.com

EMAIL\_PASS=your\_email\_password

4. Start the server:

npm start

5. Navigate to the `client` directory to set up the React frontend:

cd client

6. Install frontend dependencies:

npm install

7. Start the React development server:

npm start

## Usage

### Registration and Authentication

1. \*\*Register as a Seller or Buyer\*\*:

- Navigate to `/register` and select your role (Seller or Buyer).

- Fill in the registration form and submit.

2. \*\*Login\*\*:

- Navigate to `/login` and enter your credentials.

### Seller Functionalities

1. \*\*Add Property\*\*:

- Navigate to `/properties/add`.

- Fill in the property details and submit.

2. \*\*View Listed Properties\*\*:

- Navigate to `/properties/my-listings` to see all properties you have listed.

3. \*\*Email Notifications\*\*:

- When a buyer shows interest in one of your properties, you will receive an email with their details.

### Buyer Functionalities

1. \*\*Browse Properties\*\*:

- Navigate to `/properties` to see all available properties.

2. \*\*Get Owner Details\*\*:

- Click on a property to view more details.

- Request owner details, and you will receive an email with the owner's contact information.

3. \*\*Like a Property\*\*:

- Click the "Like" button on a property to show your interest.

## Email Notification Setup

To set up email notifications, configure NodeMailer in the application:

1. Open `nodemailer.config.js` and configure it with your email service provider:

const nodemailer = require('nodemailer');

const transporter = nodemailer.createTransport({

service: process.env.EMAIL\_SERVICE,

auth: {

user: process.env.EMAIL\_USER,

pass: process.env.EMAIL\_PASS,

},

});

module.exports = transporter;

2. Ensure your `.env` file has the correct email configuration:

EMAIL\_SERVICE=gmail

EMAIL\_USER=your\_email@gmail.com

EMAIL\_PASS=your\_email\_password

## API Endpoints

### Authentication

- `POST /register` - Register a new user (Seller or Buyer)

- `POST /login` - Log in a user and get a JWT token

### Seller

- `POST /properties/add` - Add a new property (authenticated as Seller)

- `GET /properties/my-listings` - Get all properties listed by the seller (authenticated as Seller)

### Buyer

- `GET /properties` - Get all available properties (authenticated as Buyer)

- `POST /properties/:id/like` - Like a property (authenticated as Buyer)

- `POST /properties/:id/contact-owner` - Request owner details via email (authenticated as Buyer)

## Frontend Implementation

The frontend of the Real Estate Application is built using React. Here are the key steps and components:

1. \*\*Project Structure\*\*:

- `src/components` - Contains React components.

- `src/pages` - Contains different pages like Home, Login, Register, etc.

- `src/services` - Contains services for API calls.

- `src/App.js` - Main application component.

2. \*\*Key Components\*\*:

- `Register.js` - Registration form for Sellers and Buyers.

- `Login.js` - Login form.

- `PropertyList.js` - Displays a list of properties.

- `PropertyDetail.js` - Displays details of a single property.

- `AddProperty.js` - Form to add a new property.

3. \*\*API Integration\*\*:

- Use `axios` for making API calls to the backend.

- Handle authentication using JWT tokens stored in localStorage.

4. \*\*Styling\*\*:

- Use CSS modules or styled-components for styling React components.

## Contributing

1. Fork the repository.

2. Create a new branch (`git checkout -b feature-branch`).

3. Make your changes.

4. Commit your changes (`git commit -am 'Add new feature'`).

5. Push to the branch (`git push origin feature-branch`).

6. Create a new Pull Request.