REAL TIME CHAT WEBSITE

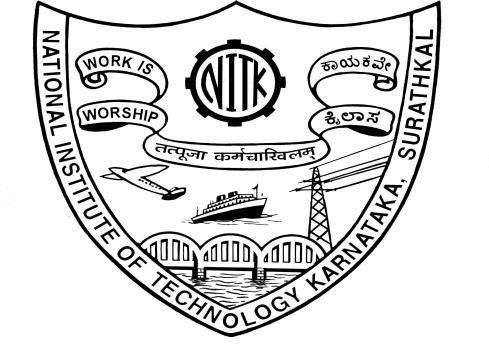
### Web Technologies and Applications (IT254-Minor) Report Submitted in partial fulfillment of the requirements for the degree of B.Tech (Minor) In INFORMATION TECHNOLOGY

### By

### Abhi Krishna A (211EE203)

### Vishal (211EE227)

### Sumukh S K (211EE153)



DEPARTMENT OF INFORMATION TECHNOLOGY NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL, MANGALORE -575025

NOVEMBER, 2023

**DECLARATION**

I hereby *declare* that the *Web Technologies and Applications (IT254-Minor) Project* titled **Real Estate Management Website**, which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfillment of the requirements for the award of the Degree of B.Tech (Minor) in the department of Information Technology, is a ***bonafide report of the work carried out by me.*** The material contained in this project has not been submitted to any University or Institution for the award of any degree.

Abhi Krishna A 211EE203

Vishal 211EE227

Sumukh S K 211EE153

Place : NITK, Surathkal

Date : 20th November 2023

**CERTIFICATE**

This is to certify that the Web Technologies and Applications (IT254-Minor) project titled **“Real Estate Management Website”** has been presented by Abhi Krishna A (211EE203), Vishal (211EE227), Sumukh S K (211EE153), students of V semester B.Tech. (Minor), Department of Information Technology, National Institute of Technology Karnataka, Surathkal on 20th November, during the odd semester of the academic year 2022 - 2023, in partial fulfillment of the requirements for the award of the degree of B.Tech(Minor) in Information Technology.

Guide: Dr. Janani Signature of the Guide with Date

Place : NITK, Surathkal

Date : 20th November 2023

**CONTENTS**

[1 INTRODUCTION 6](#_Toc151471066)

[2 OBJECTIVE 7](#_Toc151471067)

[3 METHODOLOGY 8](#_Toc151471068)

[3.1 Node js 8](#_Toc151471069)

[3.2 React Js 8](#_Toc151471070)

[3.3 Mongo DB 8](#_Toc151471071)

[3.4 Express 8](#_Toc151471072)

[3.5 Socket io 9](#_Toc151471073)

[4 FLOWCHART / BLOCK DIAGRAM 10](#_Toc151471074)

[5 IMPLEMENTATION 12](#_Toc151471075)

[5.1 Login & Sign up Page 12](#_Toc151471076)

[5.2 Properties Listing Page 12](#_Toc151471077)

[5.3 Properties Details Page 12](#_Toc151471078)

[5.4 Edit Profile Page 12](#_Toc151471079)

[5.5 Edit Properties Page 12](#_Toc151471080)

[6 RESULTS 13](#_Toc151471081)

[7 REFERENCES 16](#_Toc151471082)

**LIST OF FIGURES**

* 1. ER Diagram of the Database 10
  2. UX Flowchart 11
  3. New User Sign Up Page 13
  4. Login Page 13
  5. Properties Listed Page 14
  6. Favorite Properties Page 14
  7. Property Details Page when Logged in as Owner 15
  8. Property Details Page when Logged in as User 15

# INTRODUCTION

Introducing a modern real-time chat web application designed for seamless instant messaging. This platform offers live communication, enabling users to engage in real-time conversations and group chats. With a user-friendly interface and responsive design, the app facilitates easy message sharing and interaction. Built with cutting-edge technology, it delivers a smooth and interactive experience, redefining online connectivity. Emphasizing speed, reliability, and user engagement, this chat application represents the next level in web-based communication, providing a dynamic and efficient way for users to connect and communicate in real-time.

# OBJECTIVE

The primary objective of this project is to design and implement a cutting-edge Real Time Chat Website that stands out for its user-friendly interface and robust functionality. Our foremost goal is to create a platform that simplifies the property search process, ensuring users' convenience while navigating diverse real estate listings. Some key project functionalities include user login and registration, property listings, and details.

Additionally, we aim to elevate the property listing experience, making it more informative and engaging for buyers and sellers.

# METHODOLOGY

## Node Js

In a real-time chat web application, Node.js serves as the foundational backend technology, managing server-side operations and facilitating instant communication. Leveraging its event-driven architecture, Node.js efficiently handles incoming requests from users, executes server-side logic, and integrates WebSocket technology to enable seamless real-time bidirectional communication. This framework's non-blocking I/O model ensures efficient handling of concurrent connections, vital for a chat app with multiple users engaged in simultaneous conversations. By integrating with Express.js, Node.js streamlines routing and HTTP request handling, enhancing backend efficiency. Overall, Node.js powers the application's backend, ensuring low-latency communication, scalability, and efficient server-side operations crucial for real-time messaging.

## React Js

In a real-time chat web application, React.js functions as the client-side framework, responsible for building an interactive and dynamic user interface. Leveraging React's component-based architecture, the application's frontend is divided into reusable and modular components, facilitating a structured UI. React efficiently handles UI rendering and updates, providing a responsive and seamless user experience for instant messaging. Its virtual DOM (Document Object Model) efficiently manages and updates the UI, ensuring optimal performance and responsiveness, even in a dynamic real-time environment like a chat app. Additionally, React's state management capabilities enable the application to handle data changes, UI re-renders, and component updates efficiently, contributing to a smooth and engaging user experience.

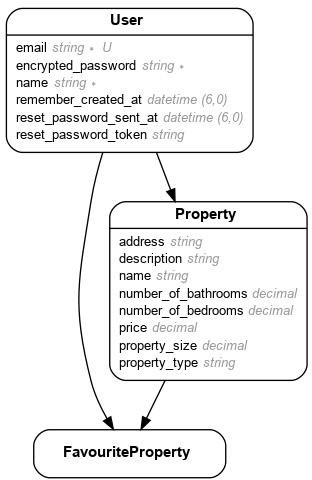
## Mongo DB

MongoDB acts as the robust database solution for a real-time chat web app, storing and managing the vast amounts of dynamic message data. Its NoSQL document-oriented structure allows flexible schema management, facilitating quick and scalable data storage. MongoDB's ability to handle high volumes of real-time data, coupled with its seamless integration with Node.js, ensures efficient storage, retrieval, and management of chat messages, essential for instant communication.

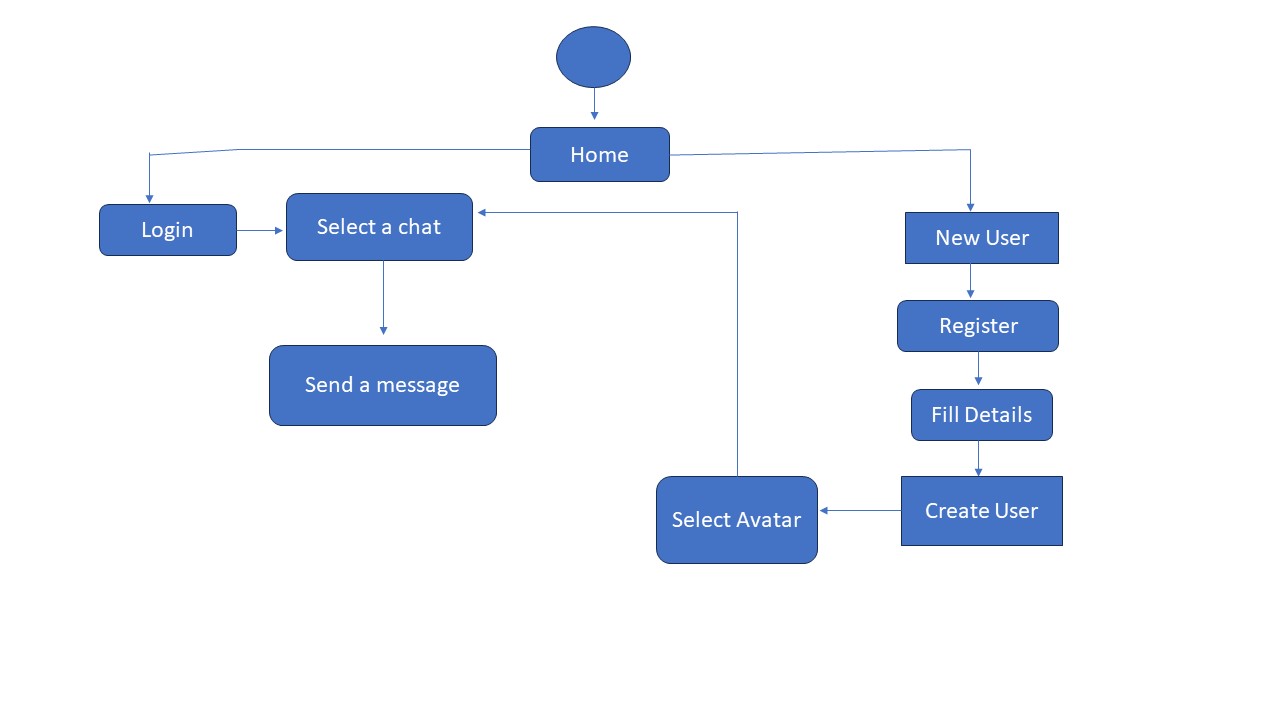
## Socket io

n the real-time chat web application, Socket.IO is pivotal, establishing a persistent connection between the Node.js server and clients' web browsers. This bidirectional communication protocol, built on WebSocket technology, ensures instant message delivery without continuous HTTP requests. Operating on an event-driven model, Socket.IO enables the emission and handling of specific events, such as message sending, ensuring real-time communication. Its functionality extends to broadcasting messages to designated rooms or users, error handling, and automatic reconnection, ensuring a seamless and engaging real-time chat experience.

# FLOWCHART / BLOCK DIAGRAM



*Fig 4.1 - ER Diagram of the Database*



*Fig 4.2 - UX Flowchart*

# IMPLEMENTATION

## Login & Sign up Page

‘Devise’ is a gem in Ruby on Rails that has been used for handling the authentication process on our website. It's a flexible and customizable authentication gem that provides a set of modules and methods to handle user authentication-related functionalities. The login page has been made by modifying the views while keeping the controller unmodified.

## Properties Listing Page

This page shows all the listed properties that are available for sale. One can add new properties with the help of a button on this page. It features a sidebar with user information and a quick access menu. The user can see a few basic details and an image of each property. They can click a button to see more details about each of the properties.

## Properties Details Page

This page gives additional information and details about the selected property. It includes additional pictures, owner details etc. It also has an option for the user to make the given property a favorite poperty. Properties marked as favorite can be viewed in the favorite properties page.

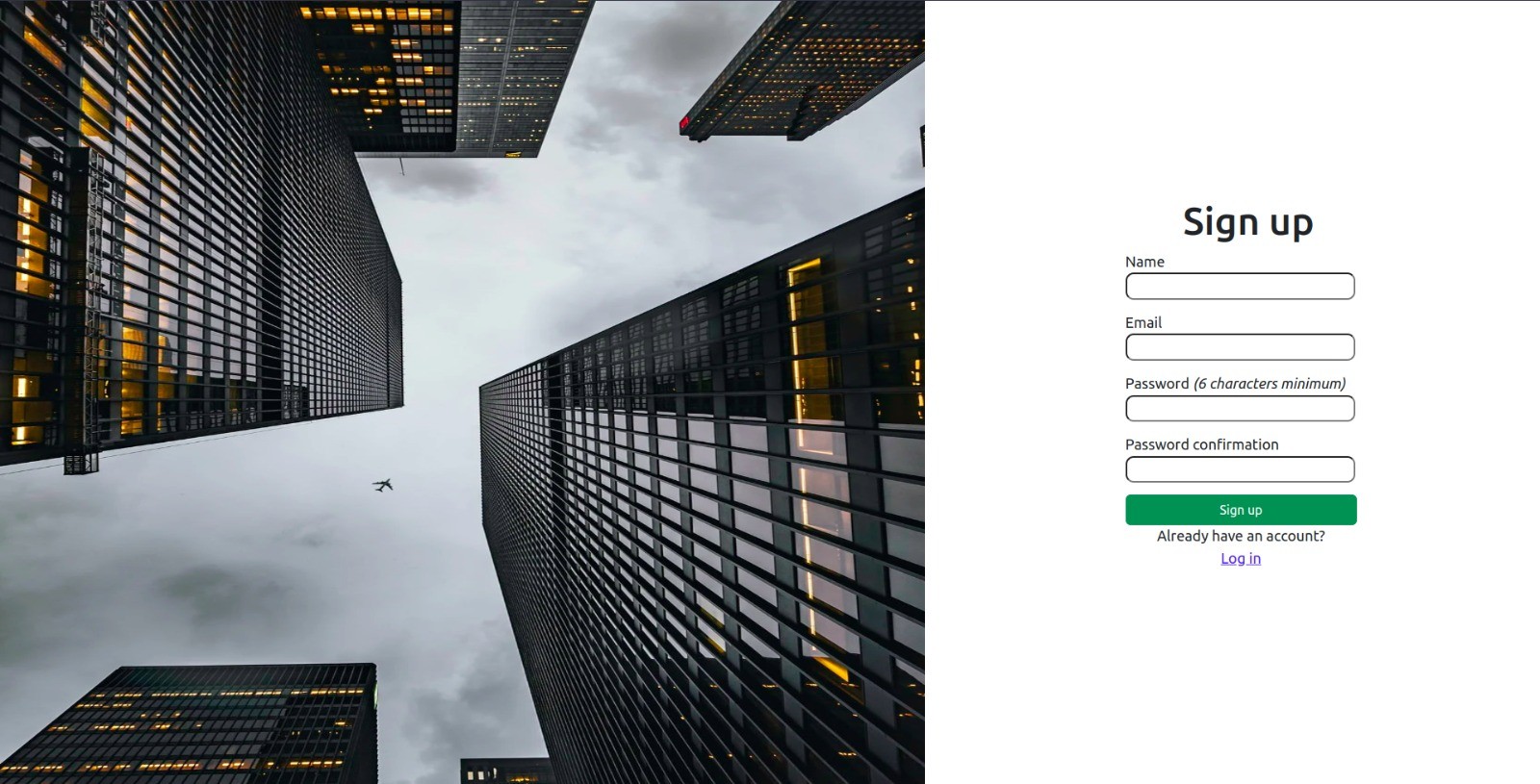
## Edit Profile Page

The user can edit their profile on this page. The user has the power to customize and update their personal information to ensure their profile reflects the most accurate and up-to-date details about them.

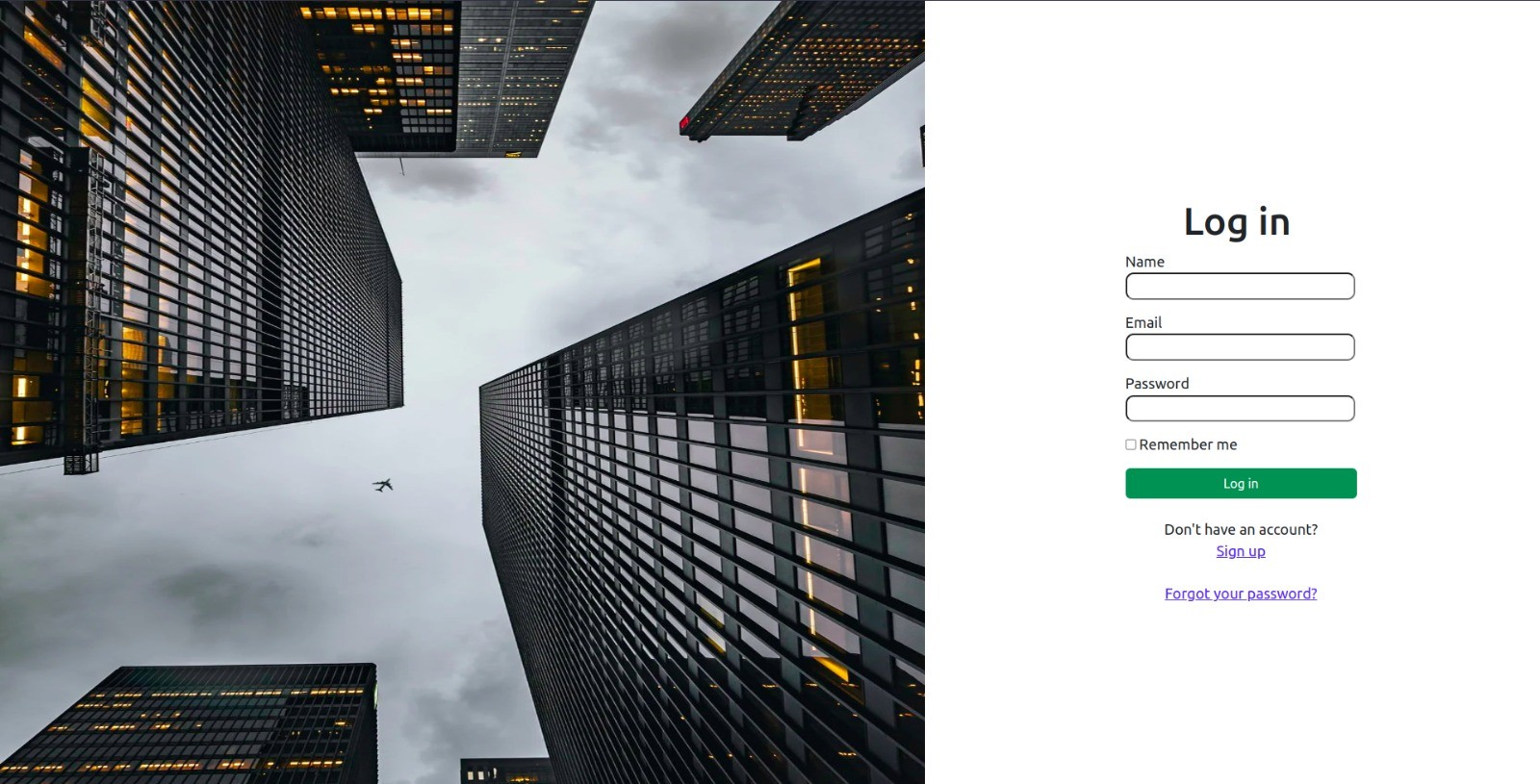
## Edit Properties Page

The user can effortlessly modify the key attributes and settings of his/her properties, ensuring they align perfectly with his/her preferences and needs.

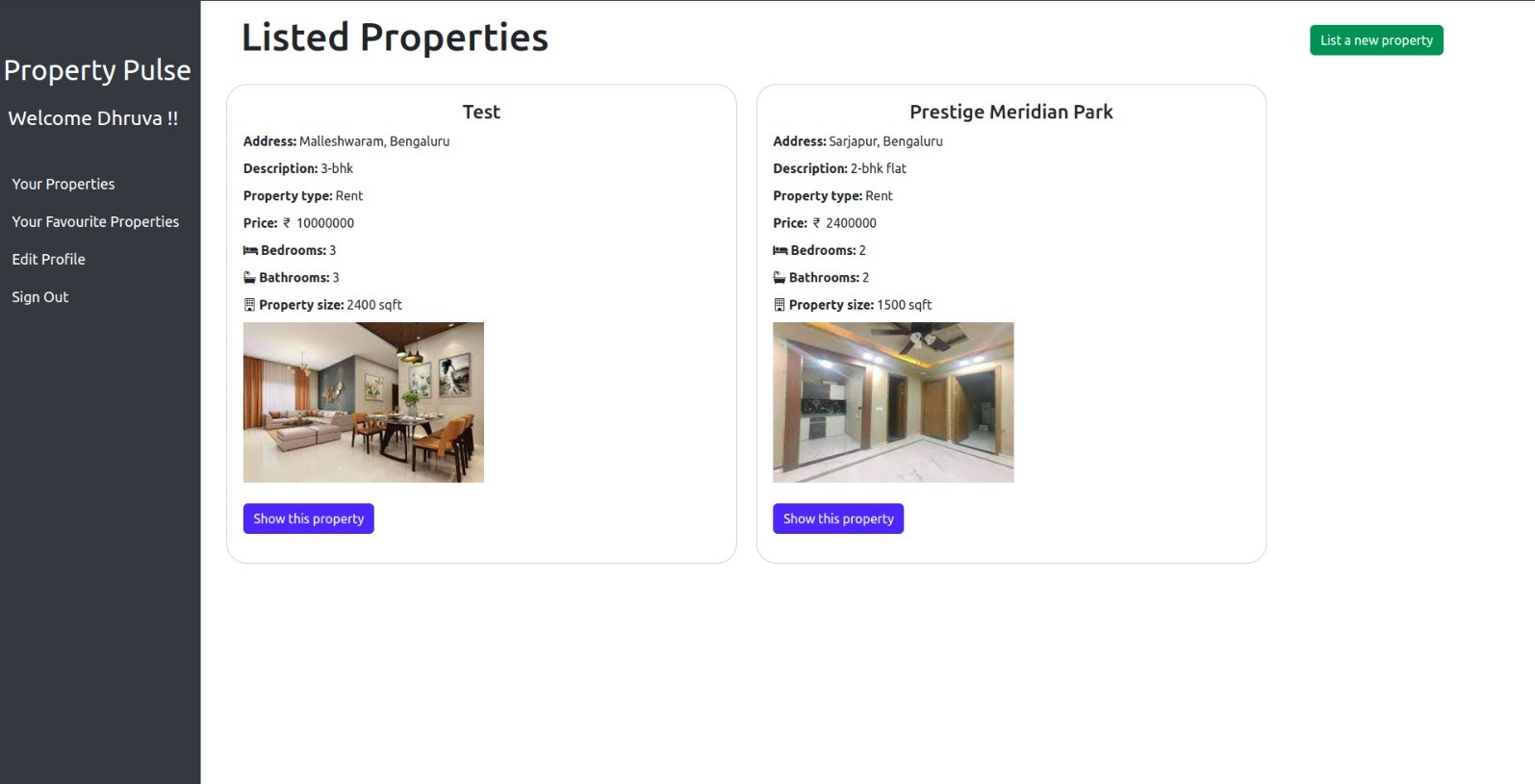
# RESULTS



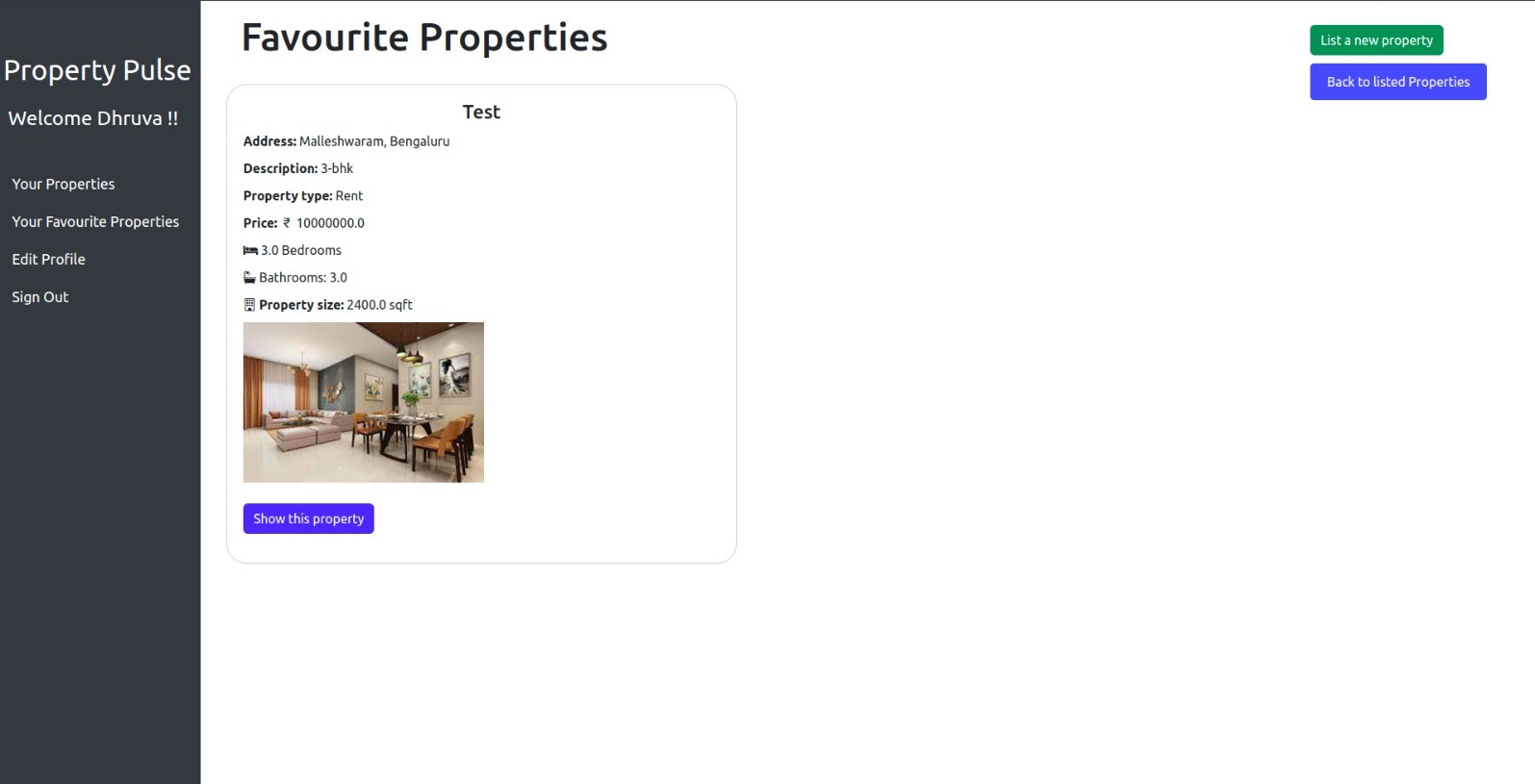
*Fig 6.1 - New User Sign Up Page*



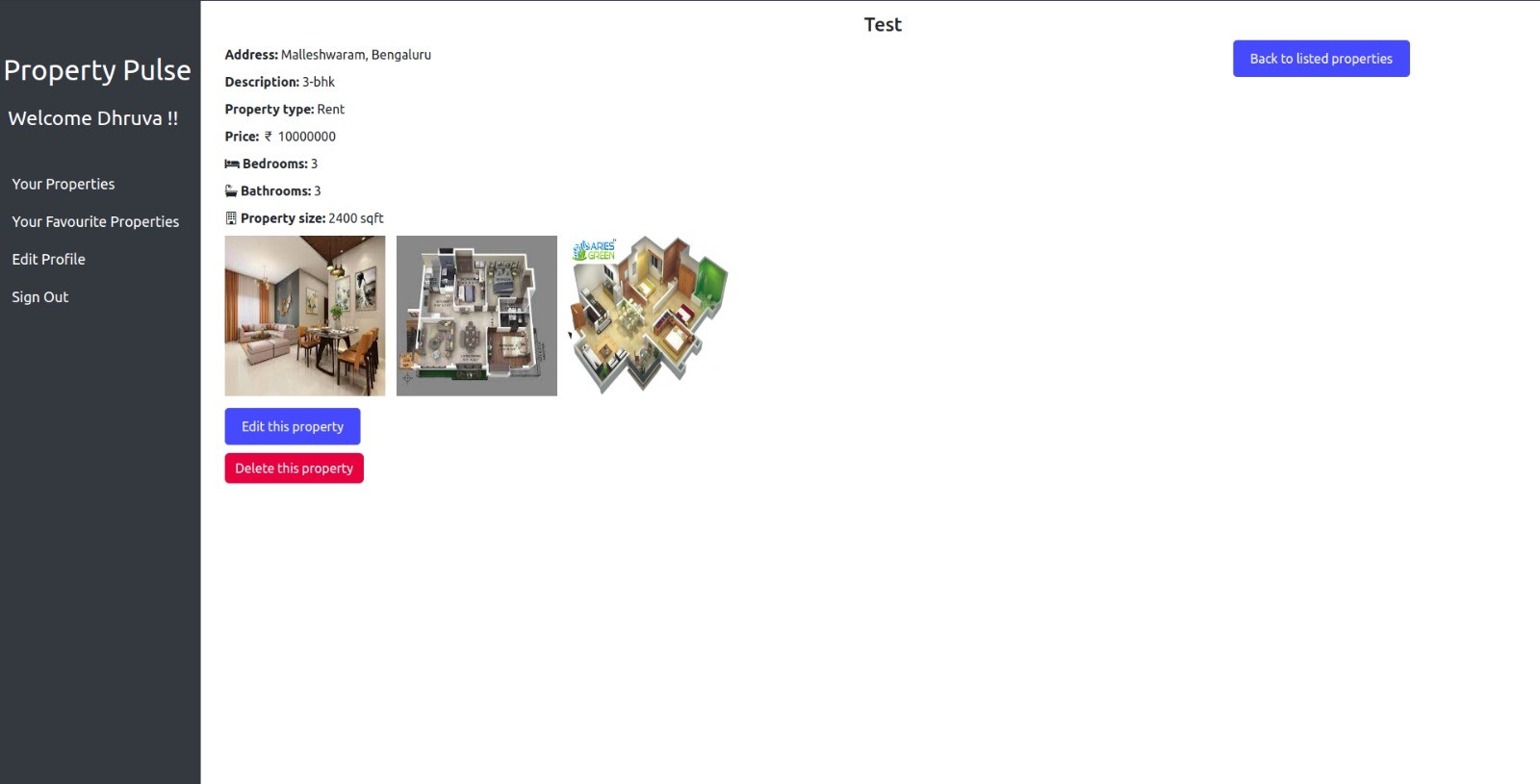
*Fig 6.2 - Login Page*



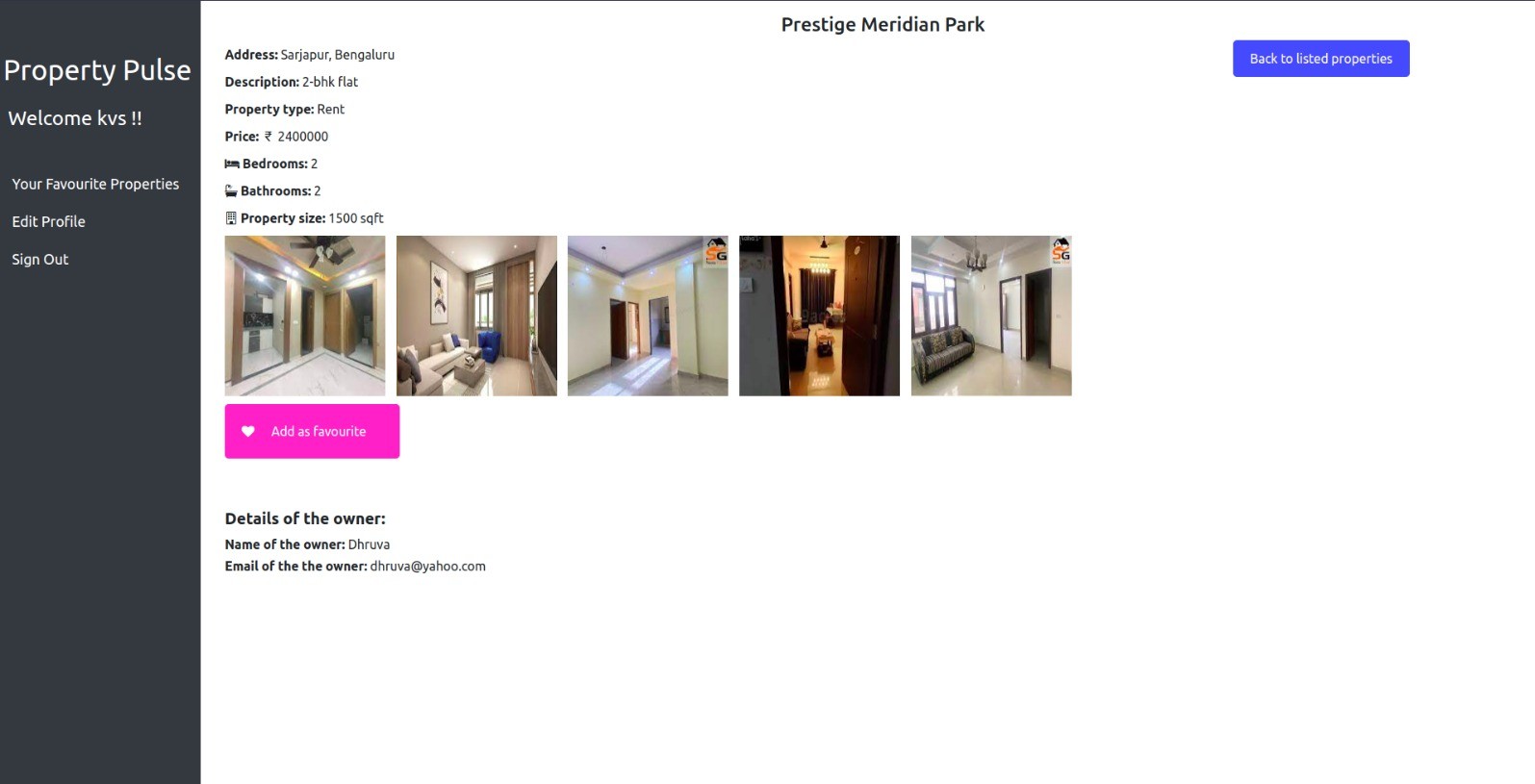
*Fig 6.3 - Listed Properties Page*



*Fig 6.4 - Favorite Properties Page*



*Fig 6.5 - Property Details Page when logged in as Owner*



*Fig 6.6 - Property Details Page when logged in as User*

# REFERENCES

1. <https://apidock.com/rails>
2. <https://guides.rubyonrails.org/>
3. <https://www.geeksforgeeks.org/>
4. <https://stackoverflow.com/>
5. <https://www.sqlite.org/>
6. <https://www.w3schools.com/css/>
7. <https://getbootstrap.com/>