

# **HOUSE RENT APP USING MERN**

**TEAM LEADER**

**SANJAI.E**

**TEAM MEMBER**

HEMANTH.B

MADHAN.R

HARISH.M

# ABSTRACT

- The need for creative solutions that streamline the home-finding and rental process has been fueled by the real estate industry's growing desire for digital platforms.
- This project presents the development of a House Rent Application using the MERN stack, that comprises of MongoDB, Express.js, React.js, and Node.js.
- The application offers a modern and effective platform for listing and renting homes, facilitating smooth communication between landlords and tenants.

# INTRODUCTION

- The House Rent App is an easy-to-use tool made to make renting out and managing properties easier.
- This software provides both tenants and landlords with a simple and effective way to list their properties or find a new place to live.
- The app's sophisticated search filters, safe payment integration, and real-time communication tools facilitate the process of matching renters with rental homes that suit their individual requirements.

# EXISTING SYSTEM

- However, there are several limitations associated with such a system that can impact its overall performance and user experience, if the app gains a large user base; the MERN stack may struggle with managing high traffic and large databases without significant optimization.
- The use of real-time features like chat or notifications may also lead to performance issues if not carefully managed.
- Tenants can search for apartments based on location, price, and preferences on traditional rental websites, which typically allow property owners to display available rents. These systems are less effective and more difficult to maintain since they frequently rely on different technologies for front-end and back-end functions.

# PROPOSED SYSTEM

The proposed system will be built using the MERN stack

## **User-friendly interface:**

The app offers a responsive design, easy navigation, and personalized dashboards.

## **Real-time scheduling:**

The platform offers real-time availability, instant booking, live notifications, and messaging for efficient rental management.

## **Secure User Authentication and Authorization :**

The platform uses JWT for secure user access and role-based authorization for landlords and tenants.

### **Payment Integration :**

The platform integrates Stripe or PayPal for secure payments and allows users to track their transaction history.

### **Reviews and Ratings :**

Tenants can leave reviews for properties and rate landlords, fostering transparency and informed decisions.

### **Admin Panel and Management :**

The admin panel enables system management, user monitoring, and dispute resolution for landlords and tenants.

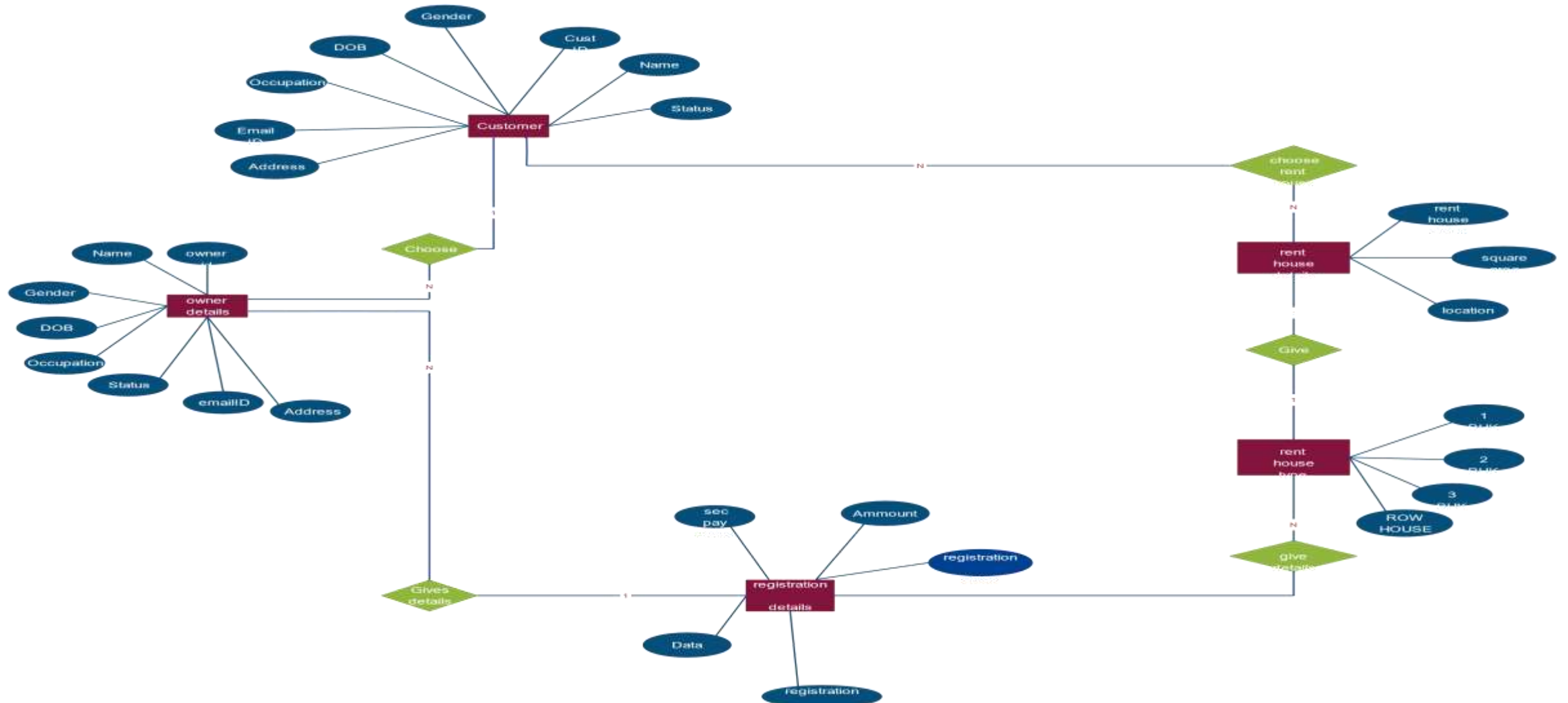
### **Scalability and Performance:**

The platform uses MongoDB for scalable data storage and a Node.js backend for fast, efficient API services.

# **FUTURE IMPLEMENTATIONS**

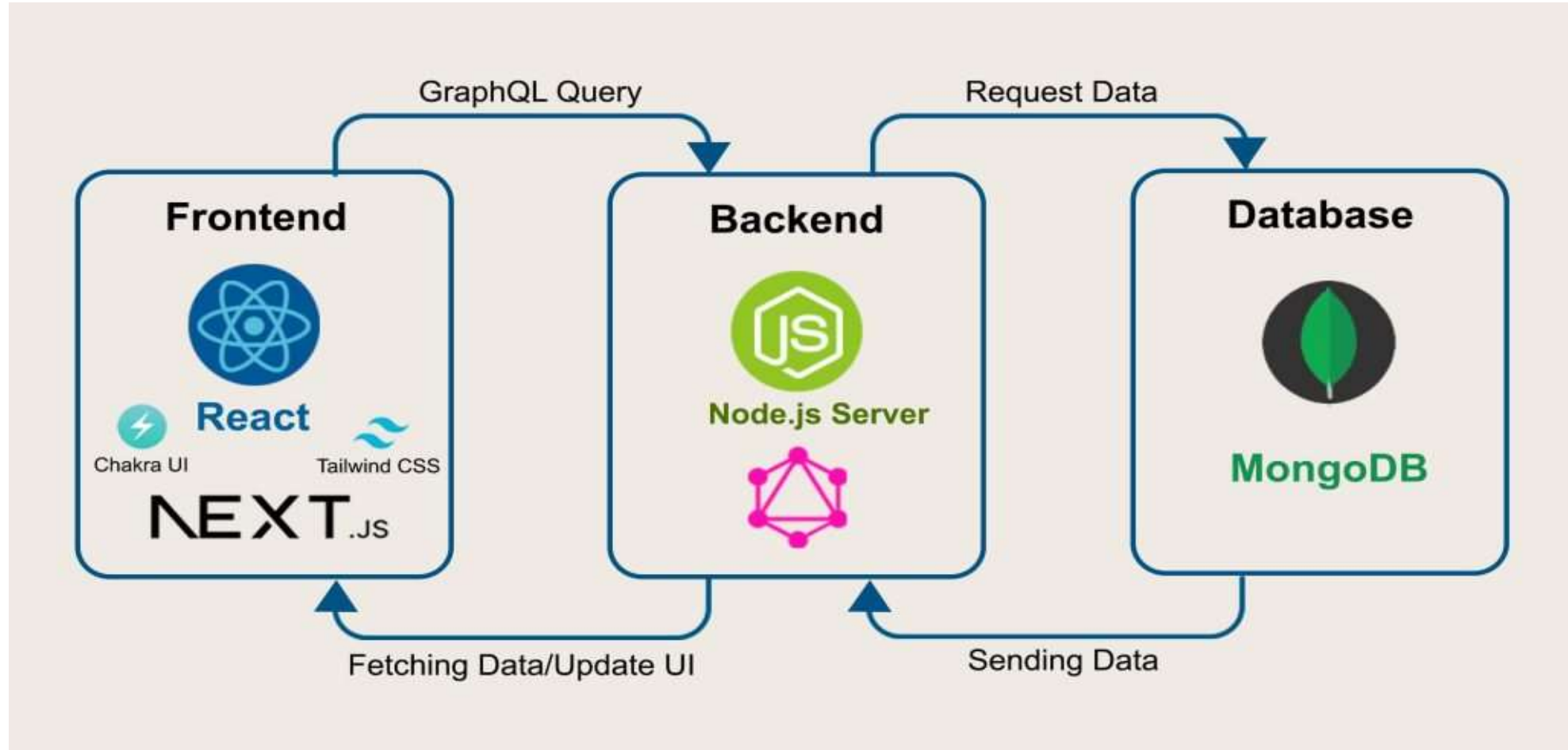
- **Advanced Search and Filtering**
- **User Profile Customization**
- **Mobile Application**
- **Multilingual Support**
- **Automated Booking and Payment Systems**

# ER DIAGRAM

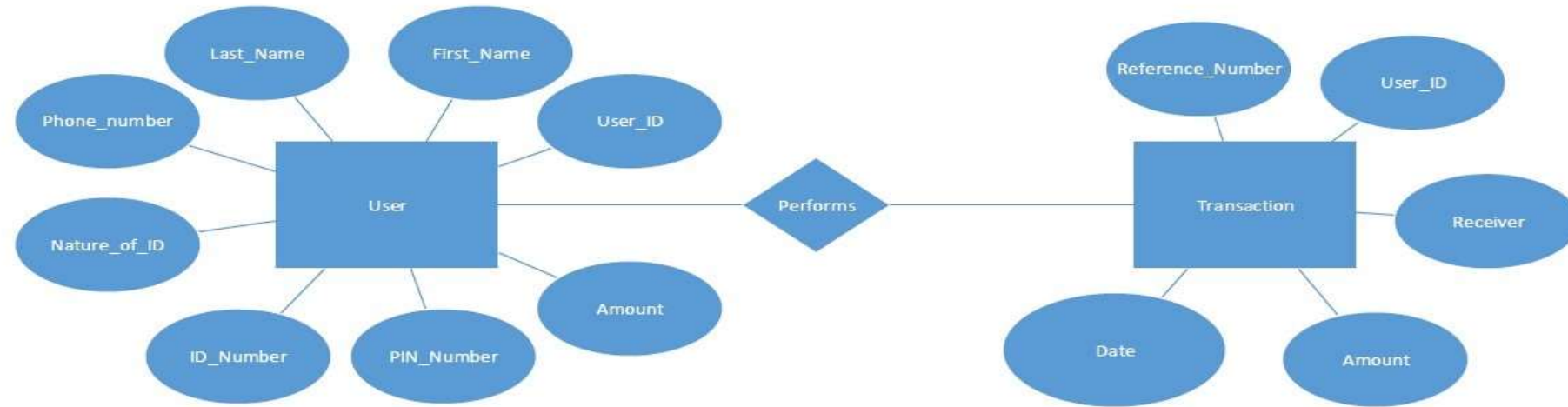




# ARCHITECTURE DIAGRAM



# FLOW DIAGRAM



# FRONT-END DEVELOPMENT

- The frontend is built using **React.js** to create a responsive and interactive user interface.
- **Technologies Used:**  
React , HTML, CSS, Bootstrap.
- React components handle the Booking scheduling process and user interactions dynamically. It includes components like the House Rent search page, Managing booking form, and user login/signup features.
- **JWT (JSON Web Tokens)** is implemented to provide secure login and authentication.
- Features like real-time status updates and notifications are implemented using **React's** state management.

# BACK-END DEVELOPMENT

The backend of the **House Rent App using Mern** is built using **Node.js** and **Server.js**, providing the core functionality for managing bookings , user data, and communication between the frontend and database.

## Database Design:

- **Users:** Stores landlords/tenants' details
- **Properties:** Stores property details (owner, price, location)
- **Bookings:** Manages tenant bookings
- **Payments:** Handles payment transactions

# BACK-END DEVELOPMENT

## API ENDPOINTS:

- **Users:** Register, login, profile management
- **Properties:** CRUD (Create, Read, Update, Delete) for properties
- **Bookings:** Manage bookings (create, update, cancel)
- **Payments:** Process payments
- **4 Authentication JWT** For user authentication Role-based Access Separate permissions for landlords and tenants

# BACK-END DEVELOPMENT

## Database Management (MongoDB):

- **MongoDB** is used as the NoSQL database for the house rent application, storing crucial data such as user profiles, house listings, rental agreements, and user reviews.
- Data is organized into collections (e.g., **Users**, **Houses**, **Appointments**) and is managed using Mongoose, which is a MongoDB Object Data Modeling (ODM) library.
- Mongoose provides schemas to define the structure of the data, enabling data validation and ensuring smooth interactions with MongoDB.

# INTEGRATION

- The frontend (React.js) interacts with the backend through **RESTful APIs**. React makes **GET** and **POST** requests to the server, which processes data and returns appropriate responses.
- The backend communicates with **MongoDB** using **Mongoose** for managing user profiles, house listings, bookings and reviews. API requests fetch and update this data in real-time.
- The backend uses **JWT tokens** for secure user authentication. When users log in or sign up, their credentials are validated, and a token is returned to allow access to protected routes.

# MODEL


## REGISTRATION PAGE

React App

localhost:3000/login

Google Axis Bank Indian Bank HDFC Bank NetBa... YouTube Samacheer Kalvi 1...

RentEase Home Login Register

  
Sign In

Email Address  
sanjaie12@gmail.com

Password  
....

SIGN UP

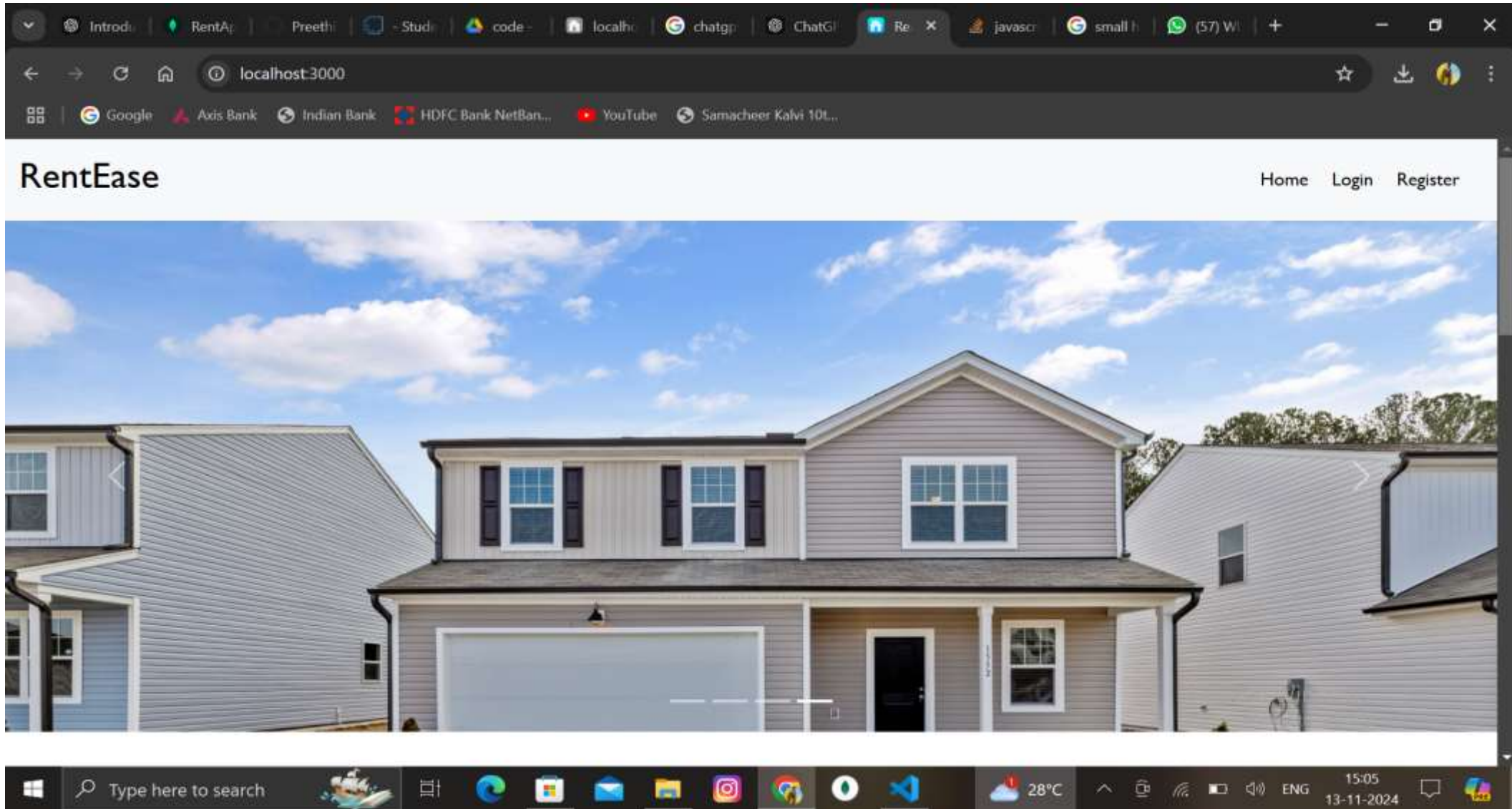
forgot password? [Click here](#) Have an account? [Sign Up](#)

© 2024 Copyright: RentEase

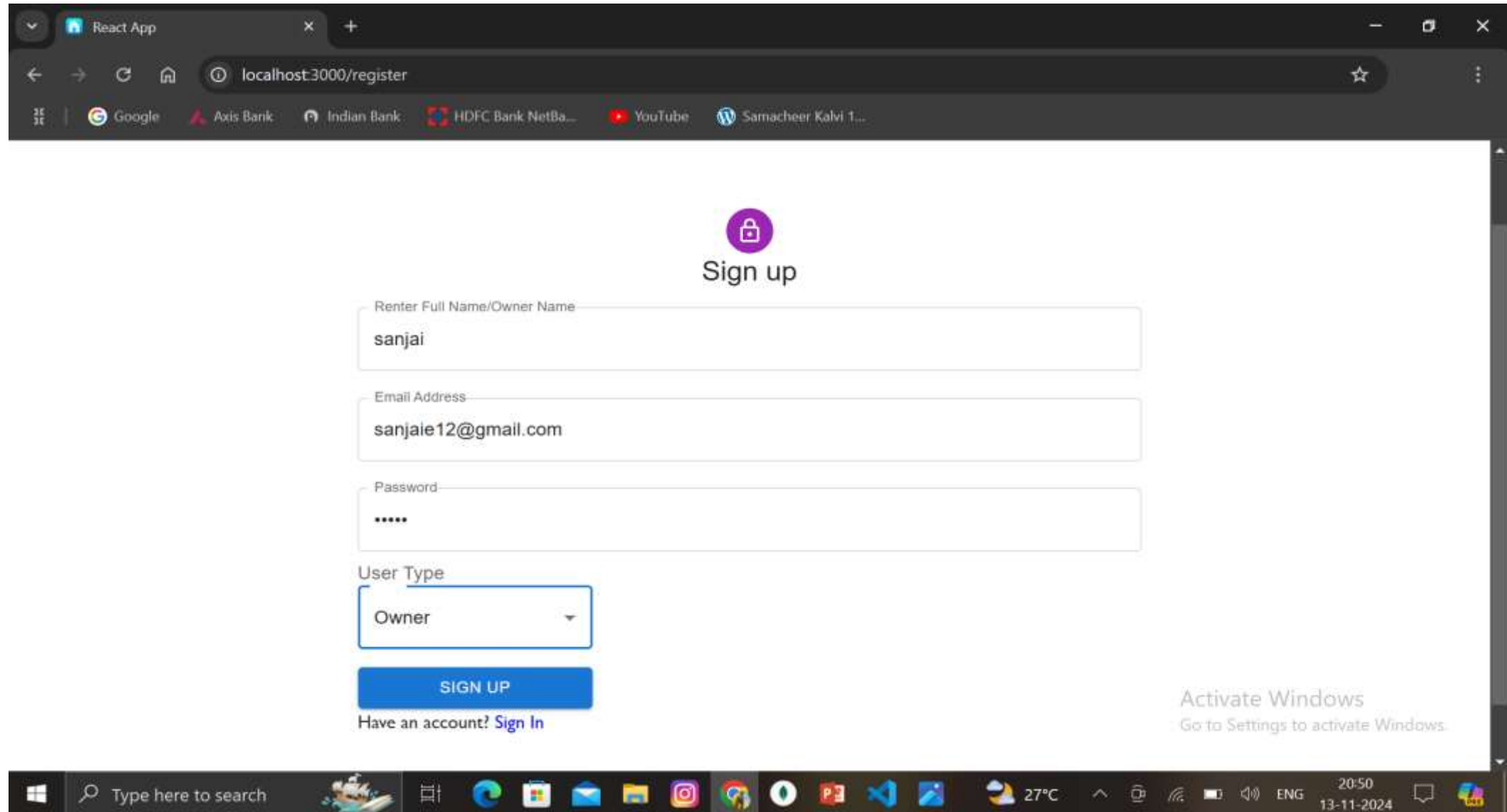
Activate Windows  
Go to Settings to activate Windows.



# HOME PAGE



# LOGIN PAGE



React App

localhost:3000/register

Sign up

Renter Full Name/Owner Name

sanjai

Email Address

sanjaie12@gmail.com

Password

.....

User Type

Owner

SIGN UP

Have an account? [Sign In](#)

Activate Windows  
Go to Settings to activate Windows.

Type here to search

27°C

20:50  
13-11-2024

# HOUSE DETAILS



**Location:**

pillayar kovil street

**Property Type:**

commercial

**Ad Type:**

rent

**Owner Contact:**

7904193510

**Availability:**

Available

**Property Amount:**

Rs.2500000

[Get More Info of the Property](#)

Get Info



**Location:**

kolapakkam vandalur chennai

**Property Type:**

land/plot

**Ad Type:**

sale

**Owner Contact:**

9123559541

**Availability:**

Available

**Property Amount:**

Rs.25

[Get More Info of the Property](#)

Get Info



**Location:**

murgan kovil 1st street

**Property Type:**

residential

**Ad Type:**

rent

**Owner Contact:**

9884117829

**Availability:**

Unavailable

**Property Amount:**

Rs.15000

Not Available

Activate  
Go to Setti



**Owner Contact:** 7904193510

**Availability:** Available

**Property Amount:** Rs.2500000

**Location:** pillayar kovil street

**Property Type:** commercial

**Ad Type:** rent

**Additional Info:** Its road side

## Your Details to confirm booking

Full Name

sanjai

Phone Number

6369367834

Book Property

# CONCLUSION

This project, “**HOUSE RENT APPLICATION USING MERN,**” offers a robust solution to modernize the house rental process. It is built using scalable web technologies that enhance the user experience for both tenants and landlords while streamlining property management workflows. The application exemplifies how the MERN stack can be used to develop responsive and efficient web applications in the real estate sector. With features such as user registration, house listings, booking management, and secure authentication, it addresses the needs of users in a digital marketplace, making the house rental experience more accessible and efficient

**THANK YOU**