

## 1. Project Overview:

Develop and implement a module for real estate management system.

### Full Stack Application Django+React

Technologies that I used:

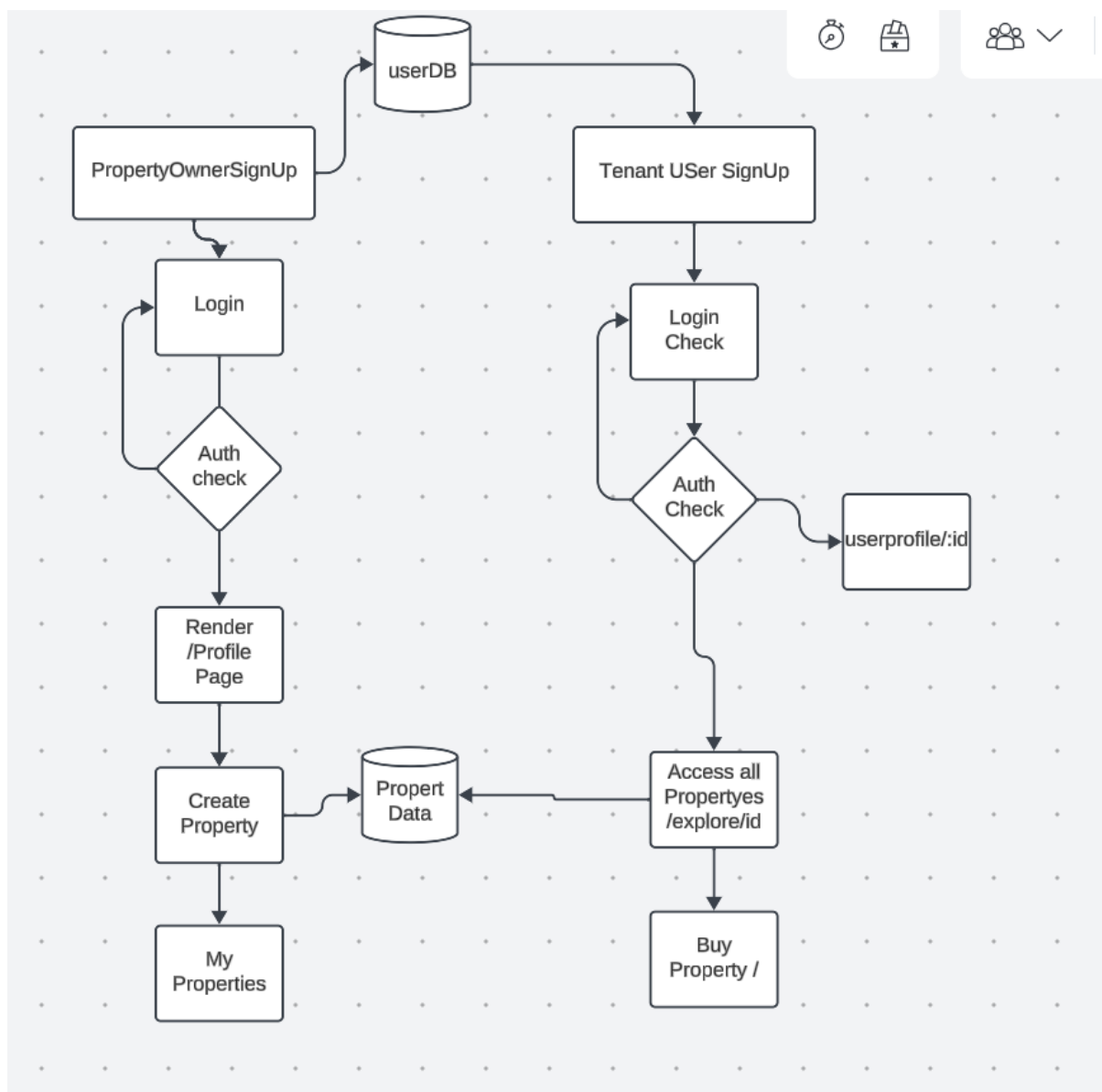
FrontEnd : Reactjs,HTML5,CSS3,JSES6,

BackEnd : Django, Django Rest Framework, Django CORS,

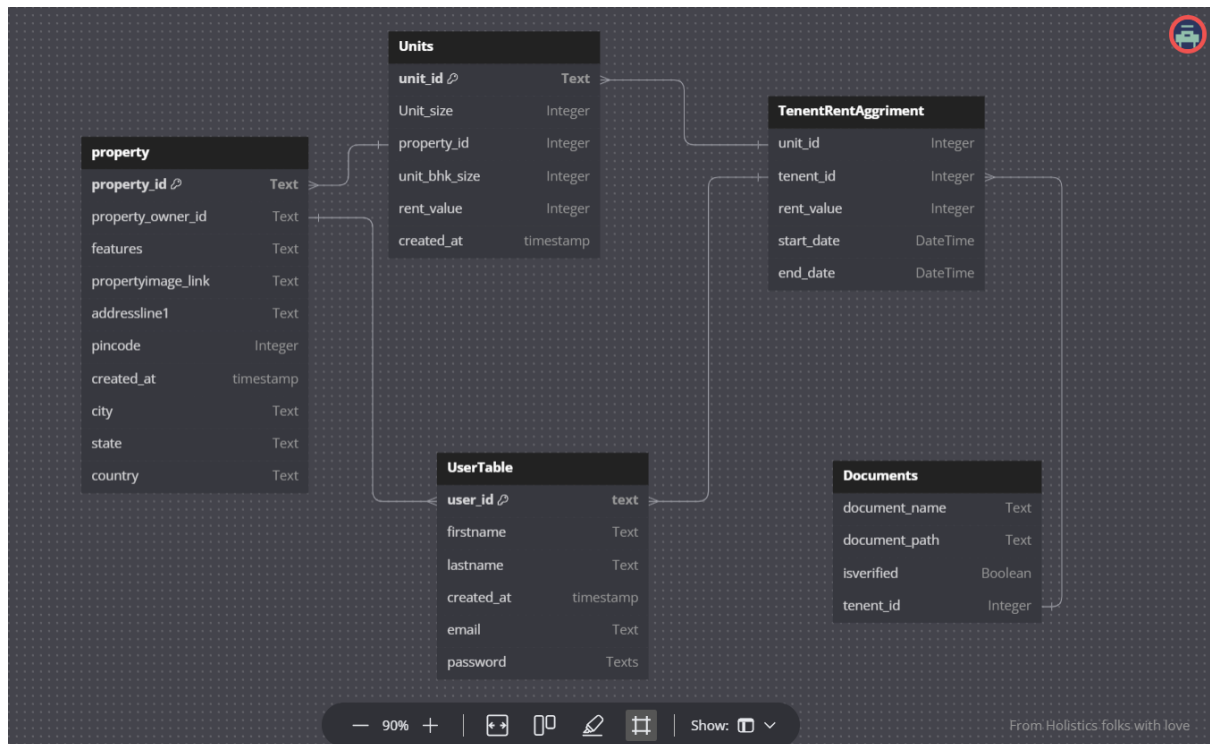
DataBase : SQLITE3,

Plugins and dependencies: uuid v4, react shimmer Loading, react Loading, React Router Dom, Django Rest Framework.

### My Application Flow:



## DataBase ER Diagram:



### Step 1:

Login as a User

Same Signin/Signup for Property Owner and Tenant User

After the SignIn Validation User Redirect To the Profile Page,

If the Credentials Not Valid The User Will Stay On The Same Page

The screenshot shows the Moonhive web application interface. The header includes the Moonhive logo and navigation links: Home, Profile, About, contact, and a Sign Out button. The main content area features a light blue rounded rectangle containing the login and registration form.

**Form Fields:**

- First Name:
- Last Name:
- Email:
- Password:

**Buttons:**

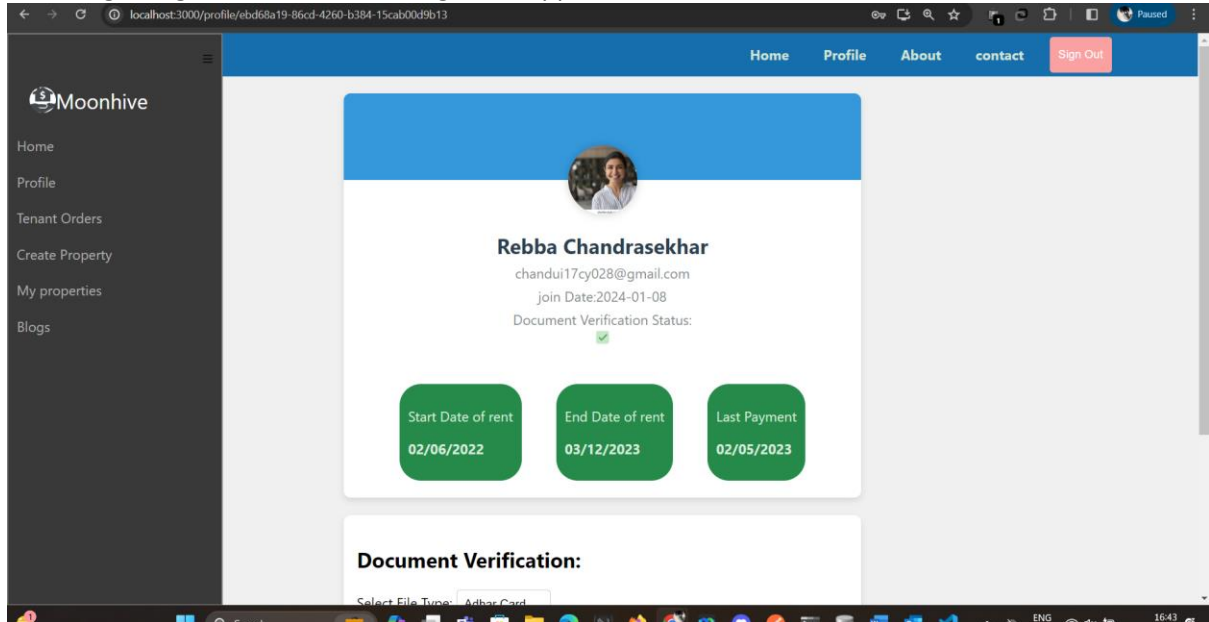
- Sign In (dark blue button)
- Register (dark blue button)

**Text:**

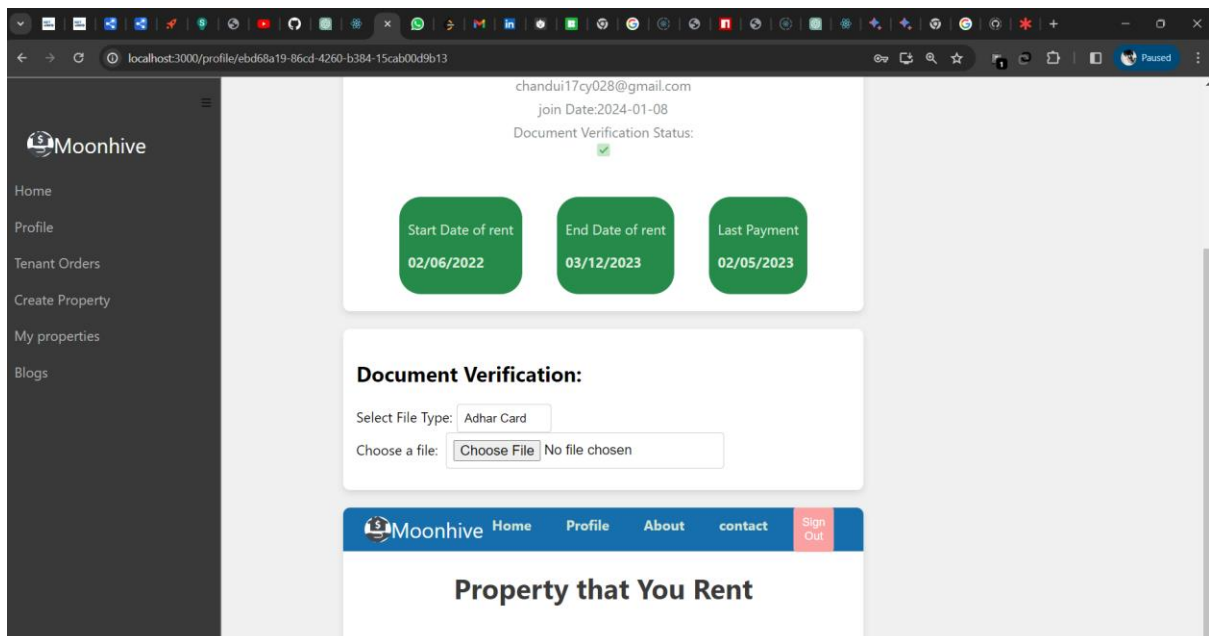
- \*Dont have account please register\*

## Step 2: /profile

After Login Page beautiful Profile Page Will Appear He

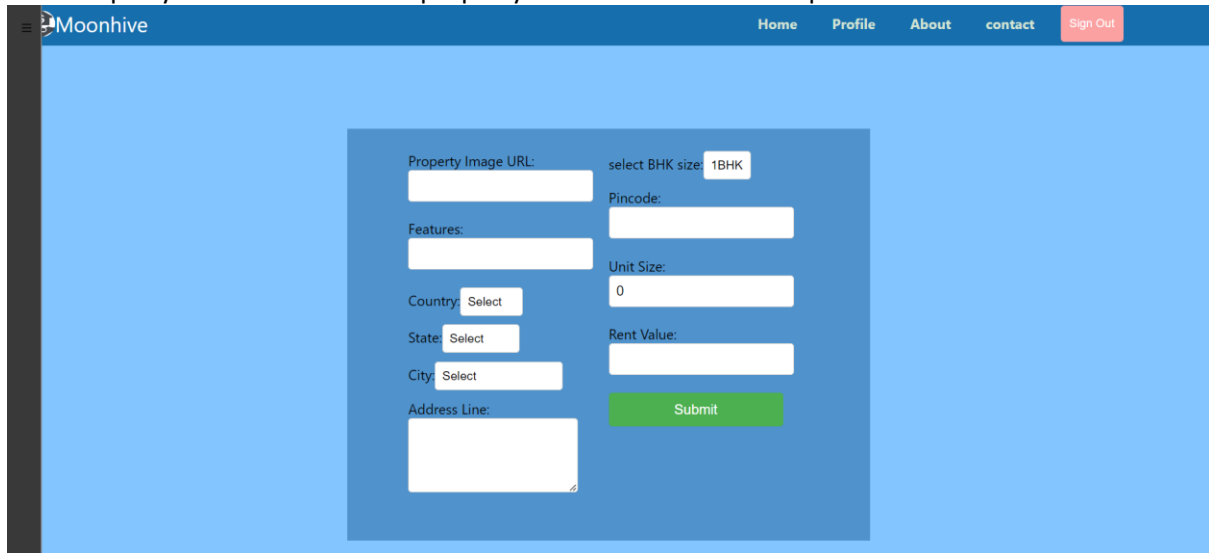


Start Date End Date of his Rent Apartment  
Below he can See his Rented Property Details.  
Now Tenant Can Rent Properties in



### Step3:

Now Property owner Can Post the property details that he want to post for Rent.

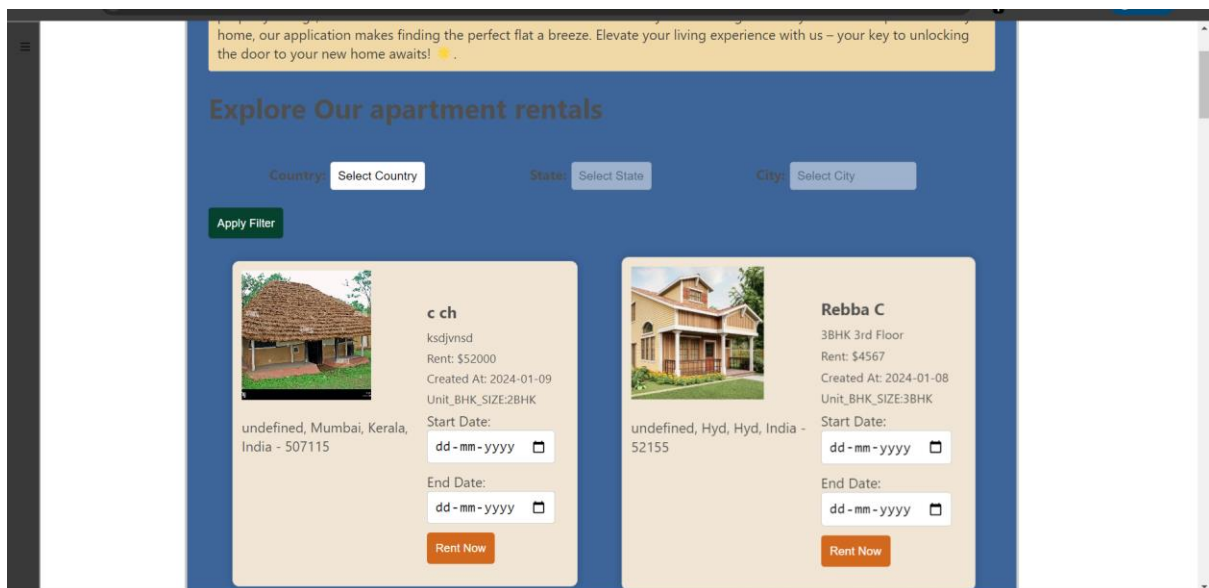


The screenshot shows the Moonhive website's property posting form. The form is set against a light blue background. It includes fields for 'Property Image URL', 'select BHK size' (with '1BHK' selected), 'Pincode', 'Features', 'Unit Size' (with '0' selected), 'Country', 'State', 'City' (all with 'Select' dropdowns), 'Address Line', and 'Rent Value'. A green 'Submit' button is at the bottom right of the form area. The top navigation bar includes 'Home', 'Profile', 'About', 'contact', and a 'Sign Out' button.

### Step4:

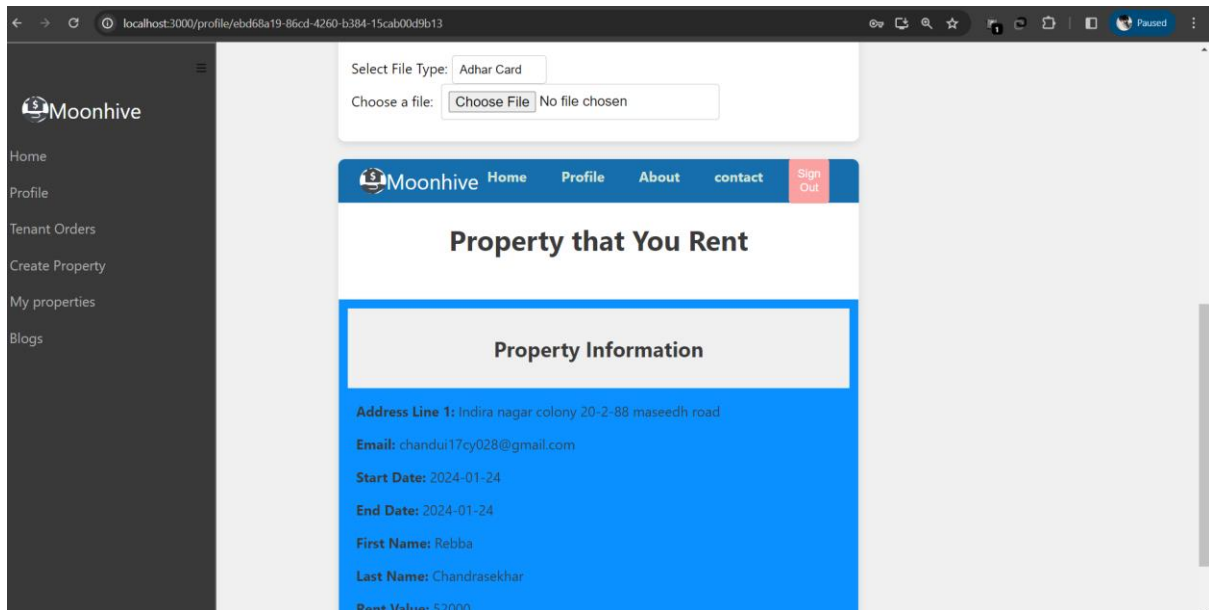
Now Tenant Can See and Rent the Apartment that he want Filter By country, State, City.

Tenant Should Select Start Date And End Date To Rent Unit.

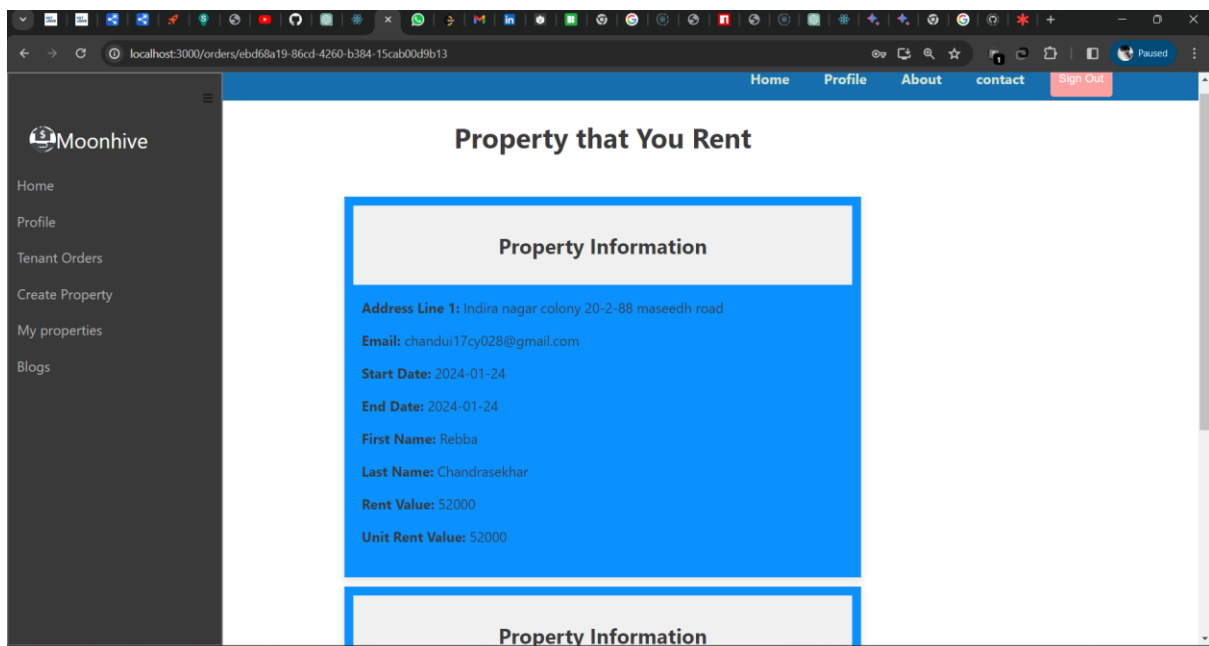


The screenshot displays the 'Explore Our apartment rentals' section of the Moonhive website. At the top, there's a yellow banner with a message. Below it, filters for 'Country', 'State', and 'City' are available, each with a 'Select' dropdown. An 'Apply Filter' button is positioned below the filters. Two apartment listings are shown: 1. 'c ch' with a rent of \$52000, created on 2024-01-09, located in Mumbai, Kerala, India. 2. 'Rebba C' with a rent of \$4567, created on 2024-01-08, located in Hyderabad, India. Each listing includes a photo, a title, rent amount, creation date, unit size, location, and date pickers for 'Start Date' and 'End Date'. A 'Rent Now' button is at the bottom of each listing card.

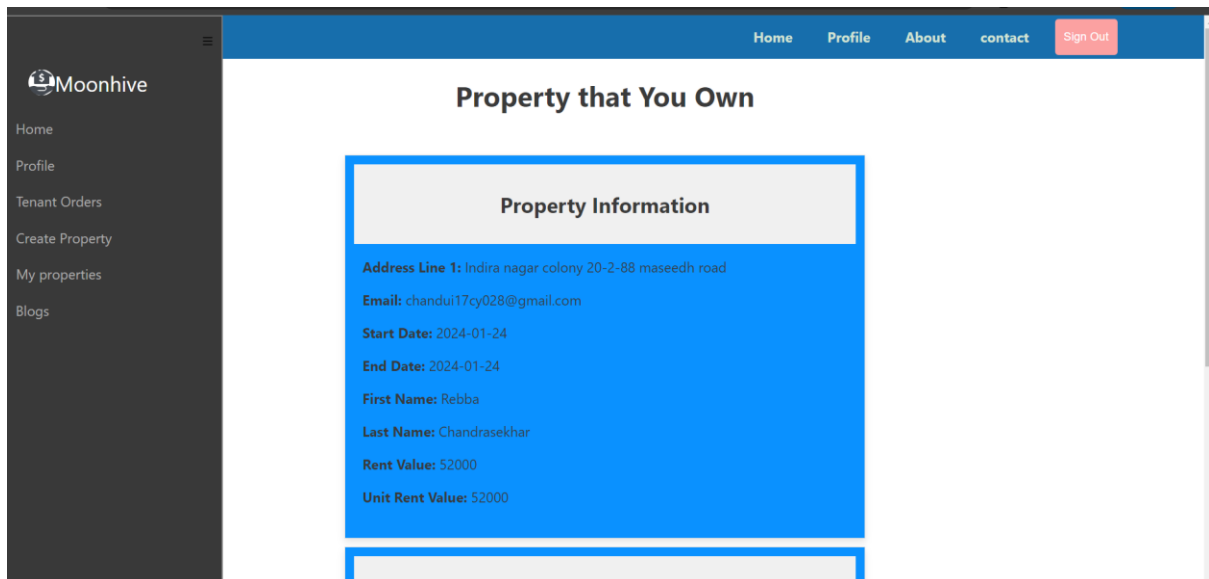
Now Tenant Can See What Product HE Bought in Profile



Tenant can see all his property Rent in orders



Property that he Create



### Key Features:

1. Property owner can create Multiple Property and Tenant can Rent Multiple Property Units
2. Tenant can also Create Properties

### How To Run The Application:

Step1:

git clone <https://github.com/Chandrasekharrebba99/FullStackRenApp.git>

cd to the APP

command1: npm install

command2: npm start

the above will start the client react App

how to run Back end:

cd to server inside the client app

command: py manage.py runserver 0.0.0.0:8000

Run with your IP Address which is acts as a host

Please check The Domain address and replace the client API URL Request with your IP Address