HOME RENTING WEB APPLICATION

A Project Report

Submitted in partial fulfilment of the Requirements for the award of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

VIKAS MUNNA MOURYA 528

Under the Course: Software Engineering,

Guided by Faculty: Elizabeth L George

Assistant Professor

NAGINDAS KHANDWALA COLLEGE(Autonomous)

(Affiliated to University of Mumbai)

MUMBAI, 400 064 MAHARASHTRA 2021-22

(Original Copy of the Approved Proforma of the Project Proposal)

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No.: ……………………   Roll no**: _____**

1. Name of the Student: Vikas Mourya

2. Title of the Project: Home renting web application

3. Name of the Guide: Elizabeth L George

4. Teaching/Industry experience of the Guide:

5. Is this your first submission?   Yes

Signature of the Student: Signature of the Guide:

Date: …………………   Date: …………………….

Signature of the Coordinator:

Date: …………………

NAGINDAS KHANDWALA COLLEGE(Autonomous)
(Affiliated to University of Mumbai)
MUMBAI, 400 064 MAHARASHTRA
DEPARTMENT OF COMPUTER SCIENCE

<u>CERTIFICATE</u>

This is to certify that the project titled, **" Home renting web application "**, is bonafied work of   **VIKAS MUNNA MOURYA** bearing Seat.No: (**NUMBER**) submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in COMPUTER SCIENCE from University of Mumbai. (12, times new roman, justified)

Internal Guide)   **Coordinator**

(Don't write names of lecturers or HOD)

External Examiner

Date: **College Seal**

Abstract

We are stuck with technology when what we really want is just stuff that works. With the current paradigm shift in technological field, there is an urgent need to embrace and appreciate the power of technology. Housing sector remains vigilant to face the challenges of change by employing a new strategy that facilitates easy management of rental houses. Hence there is need to develop a rental house management system that can simplify work for the rental managers so that all their work can be efficient and effective. The Rental Management System is Searching in Based on the Apartment Paying Guest, Office, House in metropolitan cities. The Rental Management System is Based on the Owners and the Customers. The Owner is updated on the Apartment, Office details, House, Paying Guest details. The Customer is details about the Room space, Room rent and the Address Details also. The Rental Management System is best Suitable the owners because time save and the only contact and the eligible person and there is no need to explain the room details on the speak. The Rental Management System is best application in the city place The customer contact and the easily search and the suitable place of Apartment, Office, PG, House and based the Money, Limit Person is based on the suitable house. The Rental Management System is save the time also. The Rental Management System is used to easily identify the suitable place in Save time, cost also. The Rental Management System is best way to search the house, Apartment office, Paying Guest. Hence this system is best applicable for the above reasons making House rental an easy process through an online system

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I'm extremely privileged to have got this all along the completion of our project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I owe my deep gratitude to our project guide **Ms. Elizabeth L George**, who took keen interest in my project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good system.

I also thank **Dr. Mrs. Ancy Jose**, Principal of our college, **Mrs. Sindhu.P.M,** Coordinator of IT/CS of our college.

I am extremely thankful to them for providing such a nice support and guidance, although they had busy schedule managing the corporate affairs.

DECLARATION

I hereby declare that the project entitled, "**Home renting web application"** done at **NAGINDAS KHANDWALA COLLEGE**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (COMPUTER SCIENCE)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

TABLE OF CONTENTS

6.2 User Documentation

CHAPTER 7: CONCLUSIONS

7.1 Conclusion

7.1.1 Significance of the System

7.2 Limitations of the System

7.3 Future Scope of the Project

REFERENCES

www.stackoverflow.com

www.w3school.com

www.google.com

CHAPTER 1: INTRODUCTION

1.1 Background

A House rental is a house that can be used temporarily for a period of time with a fee Renting a house assists people to live in a comfortable housewhen they do not have access to build their own personal homes/houses or. The individual who want to rent a house/room/apartment/home must first contact the House rental company for the desire House/Home/apartment. This can be done online. At this point, this person has to supply some information such as; dates of rental, and type of house. After these details are worked out, the individual renting the house must present a valid Identification Card. Most companies throughout the industry make a profit based of the type of house that are rented

.

The rental houses are categorized into modern Homes, Colonial, apartment, Rentals etc. And customers are free to choose any house of their choice based on their purse and availability of such houses at the time of booking.

Benefits of Online House Rental Services

- This online house rental solution is fully functional and flexible.
- It is very easy to use.
- This online House rental system helps in back office administration by streamlining and standardizing the procedures.
- saves a lot of time, money and labor
- Eco-friendly: The monitoring of the Housing activity and the overall business becomes easy and includes the least of paper work.
- The software acts as an office that is open 24/7.
- It increases the efficiency of the management at offering quality services to the customers.
- It provides custom features development and support with the software.

1.2 Objectives

- To produce a web-based system that allow customer to register and reserve houses online and for the company to effectively manage their House rental business.
- To ease customer's task whenever they need to rent a house.

- To Transform the manual process of renting a house to an online and computerized system
- To validate the house rental system using user feedback and testimonies
- To produce the documentation such as Software Requirement specification, Software Design Description and Software Development References

1.3 Applicability:

Applicability

House Rental is a house/Apartment/home that can be used temporarily for a fee during a specified period. The individual who needs a house must contact a House Rental Owners Through House rental Online System by first checking the available Houses, booking it then The system will contact the house owners to check the House and contract out for a room/home/apartment. This system increases customer retention and simplify House and staff management. The System can be used by Real Estate Companies to increase the house rental market around the world. The system will also help people to rent apartments/houses easily instead of walking and manually renting houses. It is also applicable for landlords who wants to expand their house rental business

CHAPTER 2: GAP ANALYSIS/ DRAWBACK OF EXISTING SYSTEM

Existing System

Currently the most property managers manage property and tenants details on papers. Once customers find a vacant house, they can call or email manager of the houses indicating the size of the house they would like rented to them. The property manager can email them back giving them all the details about the house they are requesting. The details include; Rent per month Deposit paid Terms and conditions to follow acceptance. With the current system recording the details of various activities of user is completely manual and entails a lot of paper work The existing system only provides text-based interface which is not as user friendly as Graphical user interface. Since the system is implemented manually, the response is very slow.

Disadvantages

1. Inconsistency in data entry, room for errors, miss keying information.
2. Large ongoing staff training cost. International Journal of Computer Science and Mobile Computing,
3. System is dependent on good individuals.
4. Reduction in sharing information and customer services & Lack of security. 5. Time consuming and costly to produce reports.

Proposed System

User initially want to sign up and create the account and user logs in the system automatically will show number of rented house in particular places. In this information like owner name ,house rent, address, and mobile number will the user to avoid the house broker, rent payment form, registration form. Each form has several command buttons like new, search, cancel, Back and exit. With the command buttons you can manipulate the database Advantages

1. 1.Eliminate paper-based process
2. Intuitive & user-friendly
3. Customization and flexibility

4. Optimal resource allocation
5. Highly secure

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem definition

Rental house around to rent a house has always been a hassle for people. Especially, on recent times, people have so many priorities based on which they have to rent their house. Some people want their house to be in the commercial space, or some want in a chaos free space. Some people prefer to choose the area of their house relating the religion they belong. Again there are a lot of people who love pets; therefore they want a house which has pet allowance. Basically, in this era of modernism people want to rent their house like online shopping. To rent a house in physical world has become less popular now a days. No one wants to roam around here and there to search for a house. People would prefer a virtual system to rent a house. To decode this situation and to represent a hassle free environment to the people, a dynamic system can be implemented.

That system would give the tenants the best service for renting houses without any kind of hassle. Government can make one unique system where people can rent house based on their priority instead of having so many rental systems. In that system, all the vacant houses of any district of Bangladesh will be listed there. One system will hold every details of every vacant house from any district, any are. To, make the system more liable, there should be a system by which tenants can verify the owner or agent. Also to analysis the place they will rent for house they need to know the location of that. Hence, every information details which have minimum priority to rent a house will hold by the system. There a one special feature for the bachelors so that they can rent houses efficiently as now a day house owner do not want to rent their houses to the bachelors for safety issue.

3.2 Requirements Specification

Function of software:

- Showing products in categories, cart, registration form
- Allow user to add and delete product from cart User functionality:

- Views products and buys them
- Views various offers Admin functionality:

- Keeping records
- Get product deliver on time
- Add new offers

Performance requirements:

- The system should be reliable
- Web pages must be loaded fast

Security requirements:

- The user can only access their own profile
- There is user authentication by entering user id and password

3.3 Planning and Scheduling

System development methodology is a technique that is used to show how the proposed system will be developed. In this case, the methodology used will be a waterfall model.

3.3.1 Waterfall Model It is comprised of the stages that the developer will use when developing the system. It is a sequential model hence, the name waterfall. The developer has to finish with one stage before going to the next one. It comprises of the feasibility study, analysis phase, design phase, coding phase, testing phase, implementation phase and finally the maintenance phase. It is a simple model and easy to use and understand. With waterfall development based methodologies, the analysts and users proceed sequentially from one phase to the next. The deliverables from each phase are voluminous and are presented to the project sponsor for approval as the project moves from phase to phase. Once the phase is approved by the sponsor it ends and the next phase begins.

Diagram of waterfall model
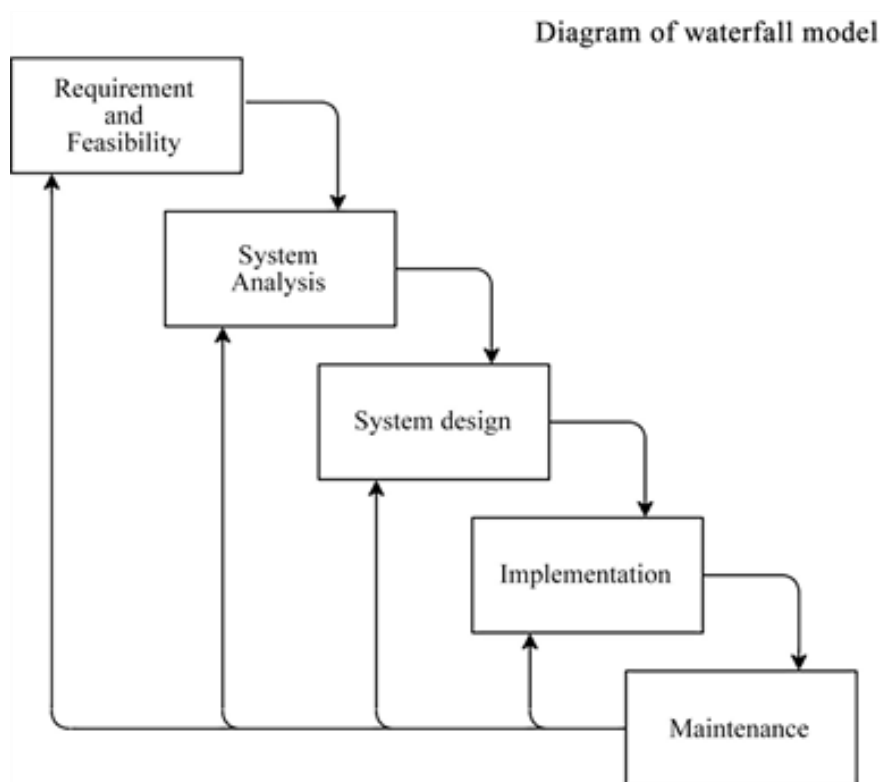
3.3.2 Feasibility study Here, I will carry out a study to gain an understanding of the customers (tenants) current system and problems experienced in this system through interviews, observations, and participations. I will use the obtained data to determine the viability of the system being proposed in terms of technical, economic and social feasibility.

3.3.3 Requirement and analysis At this stage, I will gather information about what the customer needs and define the problems the system is expected to solve. I will also include customers" business context, products functions and its compatibility. I will gather requirement such as software like the programming language to use, database model and hardware needed such as laptop, printers etc

3.3.4 Design At this stage, I will make an overall design of the system architecture and physical design which includes User Interface and Database design. It is at this stage that I will identify any faults before moving onto the next stage. The output of this stage is the design specification which is used in the next stage of implementation. 3.2.1.4 Coding/Implementation At this stage, I will begin coding as per the design specification(s). The output of this step is one or more product components built according to a pre-defined coding standard and debugged, tested and integrated to satisfy the system architecture requirement.

3.3.5 Testing At this stage, I will ensure both individual and integrated whole are methodically verified to ensure they are error free and satisfy customer requirement. I will involve both unit testing of individual code module, system testing of the integrated product and acceptance testing conducted by or on behalf of customer. I will ensure bugs found are corrected before moving to the next stage. I will also prepare, review and publish product documentation at this stage.

3.3.6 Installation It is done once the product has been tested and certified as fit for use. The system is prepared for use at customer site. 3.2.1.7 Maintenance This stage occurs after installation. It involves modifications on the system to improve performance. Such changes are user initiated or as a result of bug being discovered which were initially not known. These modifications are recorded for documentation and system update.

Grant Chart:

	July	August	September	October	November	December	January	February	March
Requirement Gathering	■								
Planning		■							
Analysis			■						
Code				■	■	■			
Design							■		
Testing									■

3.4 Software and Hardware Requirements

- **Hardware Configuration:**

  Processor   :   i3 or above

  RAM   :   Minimum 4 GB RAM

  Hard Disk Drive   :   150 GB HDD / 250 GB SDD

- **Software Configuration:**

  VS code(editor)
  React.js (Library)
  Express.js (server)

MongoDB compass

Node.js (npm)

React-redux

React-thunk

Material-ui (frontend design)

Google api (map, login)

CHAPTER 4: SYSTEM DESIGN

4.1 BASIC MODULES

Modules Details:

The Implemented modules in given below

1.Registration/Login module

2.Booking Module

3 House Posting Module

4.Feedback and contact module

Login modules : Login modules is implemented int he online House Rental Management to only allowed a register person. we have to use this module in security purpose related on the detail.

Registration modules: The online House Rental management system in Registration modules is used to collect the user personal information .It has to collect the address, name, phone number also. The registration module details a restored in the database.

Booking Modules: Online House Rental Management System has an implementation of House Booking where only registered members are allowed to view and book house/office/land/home for rental or lease

Post House Module: House Rental Management System has been implemented with House posting module where the seller/landlord or system administrator can upload house/property details for sale or rent

4.2 Schema Design

MongoDB Compass(NoSQL) document
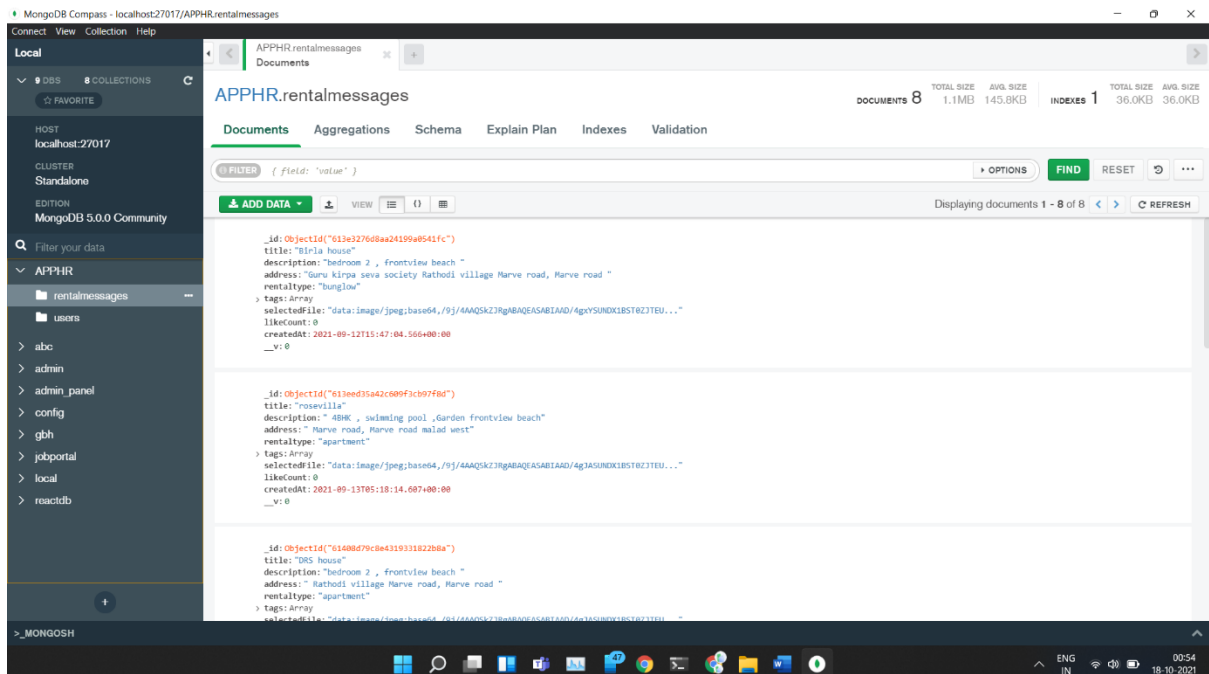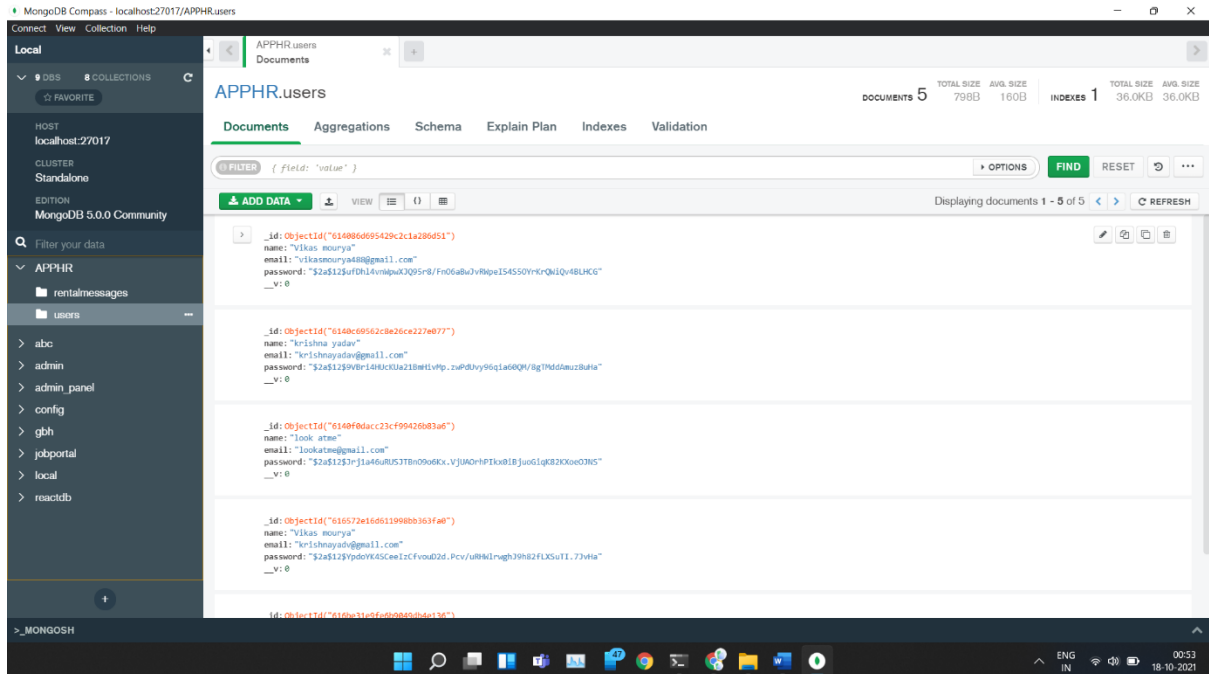
User table

Field Name	Data Type	Field size	Required	Primary Key	Foreign key

User Id	String	10	Yes	Yes	Yes
Name	String	12	No	No	No
Email Id	String	20	No	No	No
Password	String	20	Yes	No	No
Phone_no	Number	10	No	No	No

Property table

Field Name	Data Type	Field Size	Required	Primary Key	Foreign Key
Property Id	String	10	Yes	Yes	Yes
Property name	String	20	Yes	No	No
Property type	String	20	Yes	No	No
Property description	String	20	No	No	No
Property date	Date	-	Yes	No	No
Image	String(base64)	10	Yes	No	No
Price	Number	10	Yes	No	No
Tags	String	10	Yes	No	No
Likes_count	Number	100	No	No	No
Reviews	String	100	No	No	No

Property rent/sales

Field Name	Data Type	Field Size	Required	Primary Key	Foreign key
Pro_rent Id	Number	10	Yes	Yes	Yes
CreateAt date	Date	-	Yes	No	No
PaymentId	Number	20	No	No	Yes
Property_id	Number	10	Yes	Yes	Yes
User Id	String	10	Yes	no	no
Phone_no	Number	-	Yes	no	no

Payment table

Field Name	Data Type	Field Size	Required	Primary Key	Foreign key
Payment Id	Int	10	Yes	Yes	Yes
Payment date	Date	-	Yes	No	No

4.3 Block Diagram
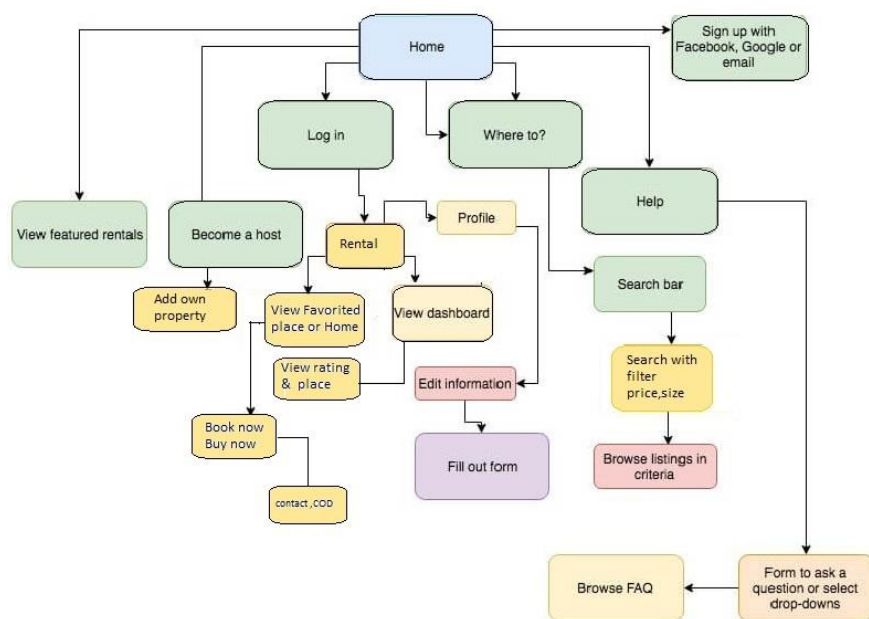
Class Diagram

Home renting WeB Application

Properties Listing

PropertyID (int)
Type_of_property (int)
HostID (int)
Address (string)

get_rent_per_mon ()
getpropertiesPhoto()
add_properties()
 payment()

User

UserID (int)
Name (string)
password (string)
phone no. (int)

Get_user_info()
Notify_user()

User Details

PAN card   (int)
Phone no. (int)
Aadhar no. (int)
Bank details (int)

getBankDetails()

1...*

Add

*

Books

1

Tenant

No_of_booking (int)

books_home()

1

Rented Property

PropertyID (int)
TenantId(int)
Booking_date (int)
Cost_of_booking (int)

Booking_date ()
check_avail()
Add_property()

Has

1

Has

1

Once a property get rented moved to Prented property Class class has also details of previous rentals

Rating

No.of.star (int)

Add_suspect_list()

1

1

Host

Numberofproperties (int)
 Add_property()

1

Forward

1

The Money paid by the Tenants to Host

Pay

1

Payment

PaymentID (int)
PropertiesID (int)
HostID (int)
Amount (int)

Pay_rent()

*

Use Case Daigram

- ◦ **Admin Use Case Diagram**

- ◦ It shows what types of process in the system are done by an admin. Here, admins can log in to the system after registration. Admin can control the database like delete something or update something from the database. Also, they can see the Google map and, they can easily contact others. Admin also reset the password and refresh the page.

- ◦ **Owner Use Case Diagram**

- ◦ It shows what types of processes in the system are done by an owner. Here, owner can register in the system and log into the system. Also, could add or delete the Rental of his/her house.

- ◦ **Tenant Use Case Diagram**

- It shows what types of processes are done by a tenant. Here, tenants can log in to the system after their registration. Tenants can set their own profile and reset the password. Also, contact with others using a contact Details and refresh the page.

Activity Diagram

Flowchart

(FLOWCHART OF TOTAL PROCESS)
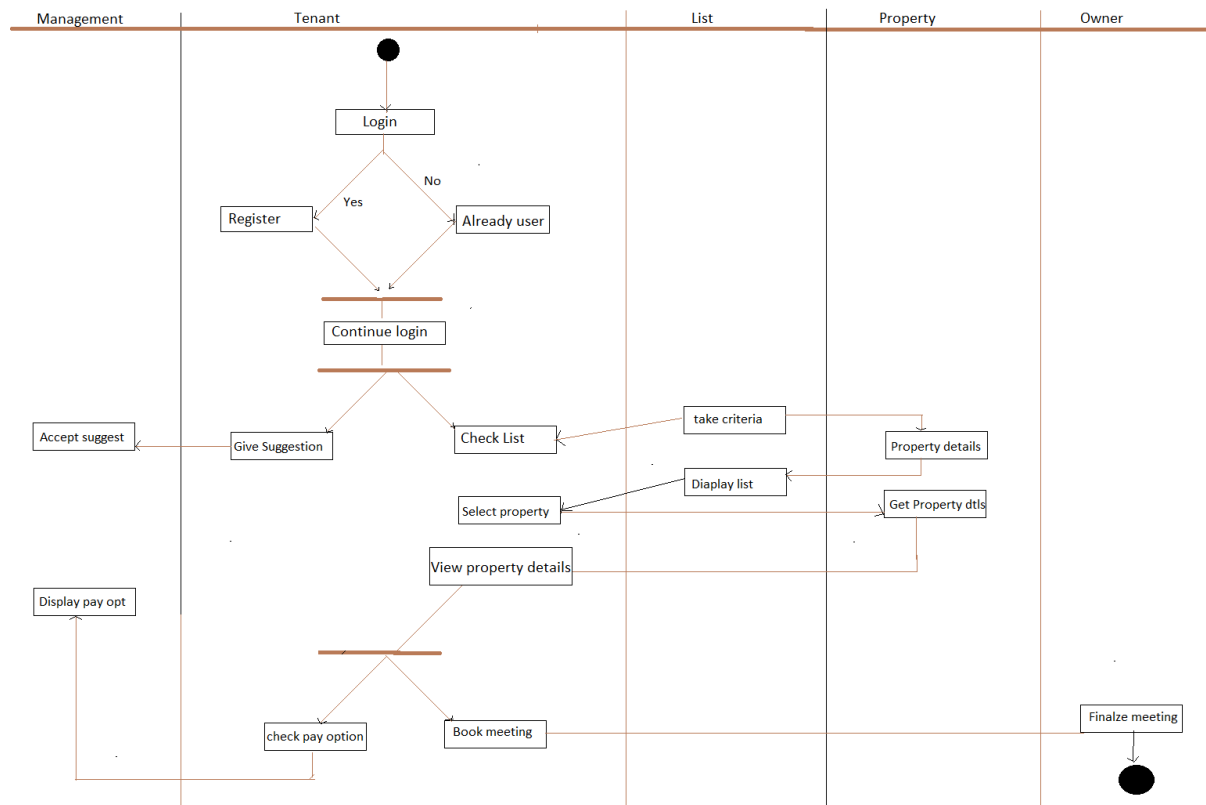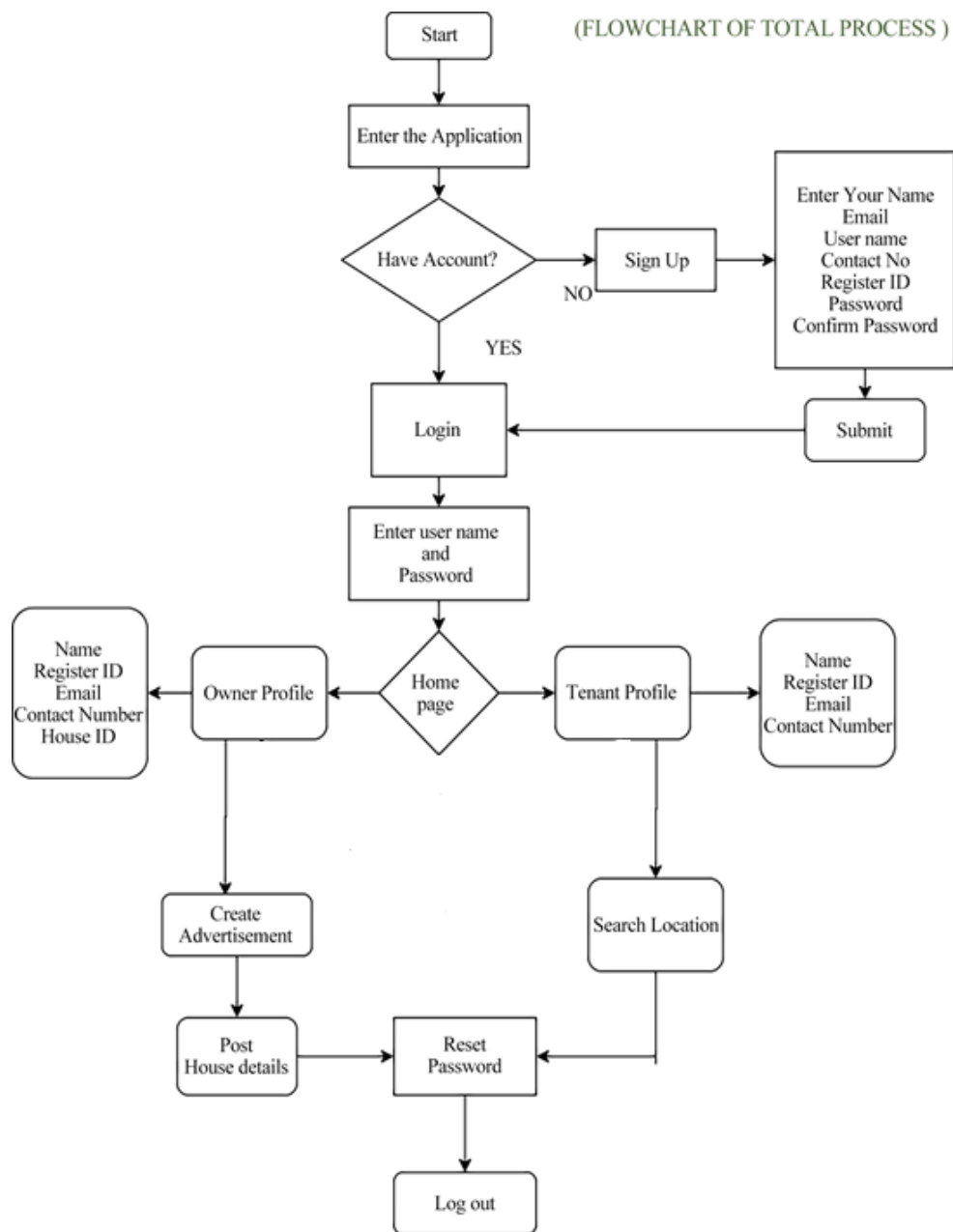
UI interface Design
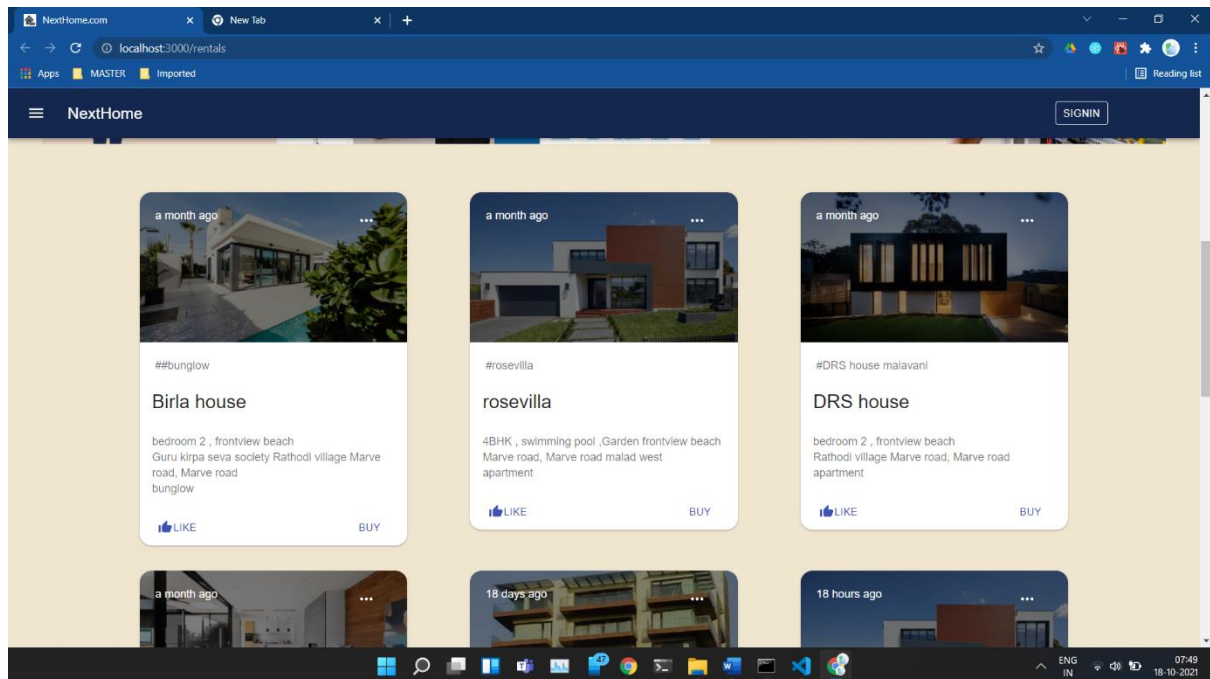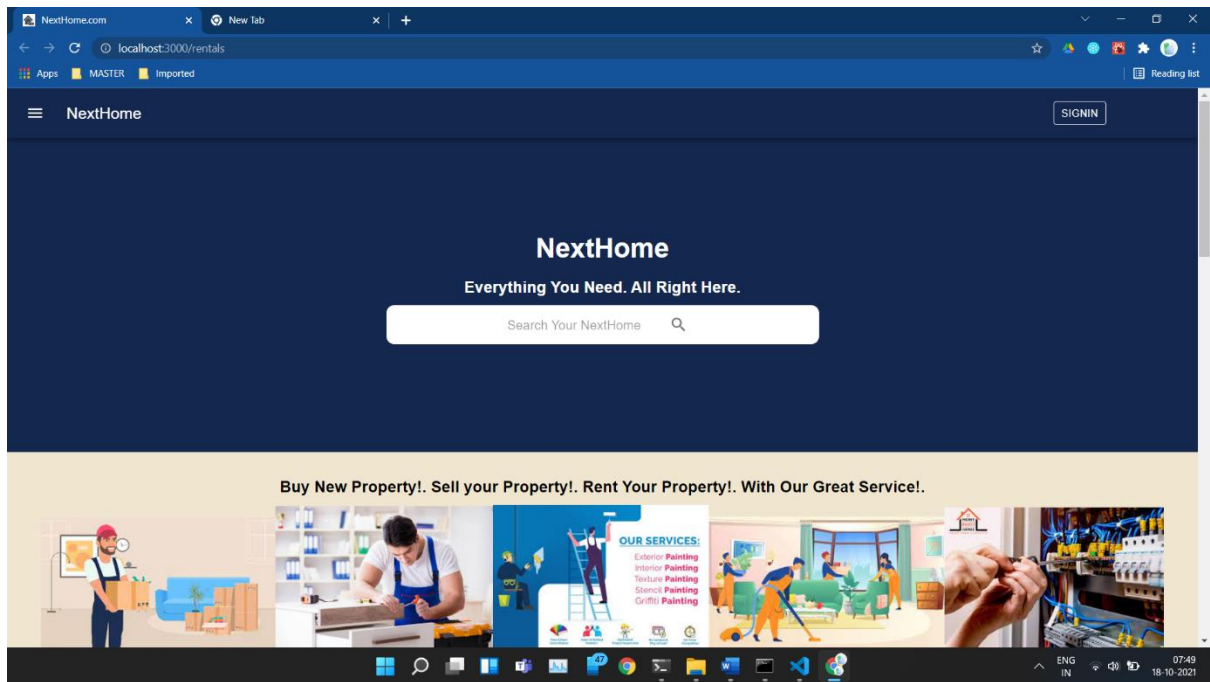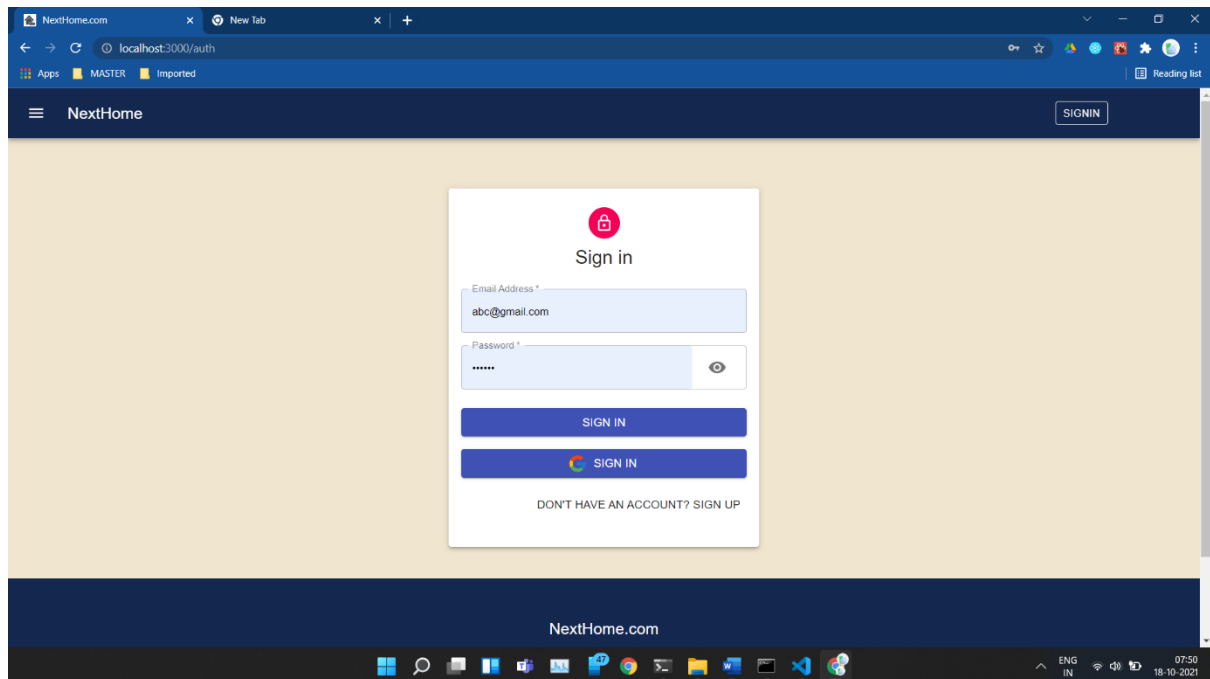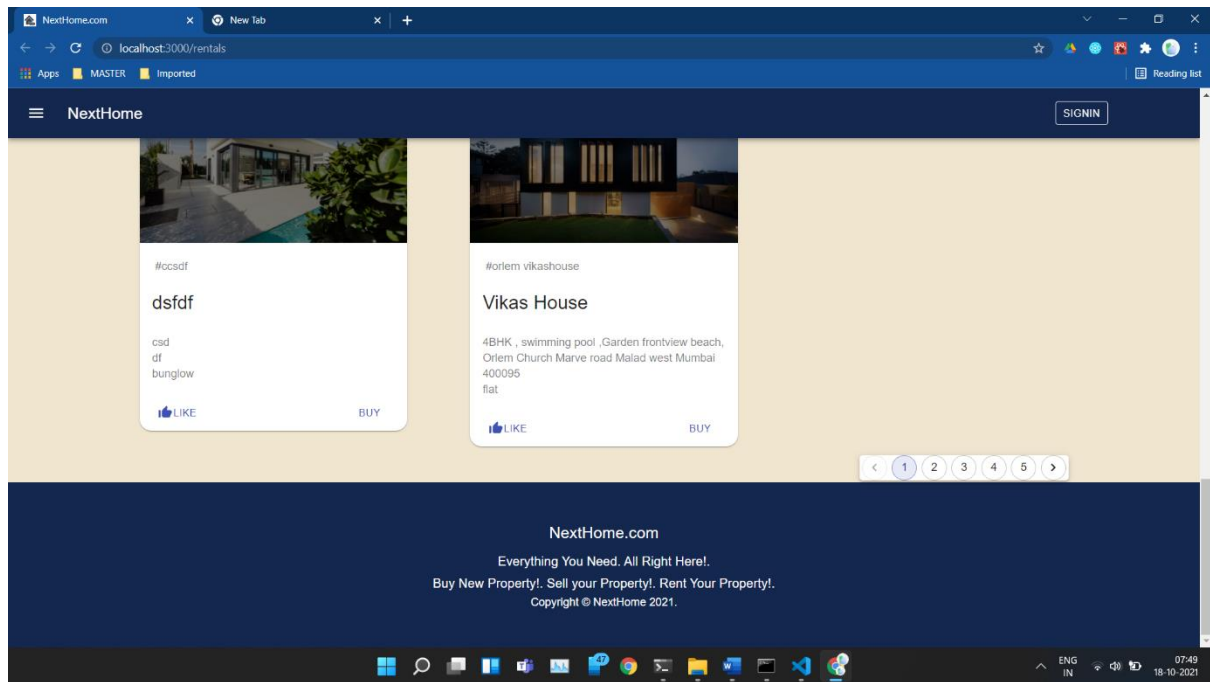
CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Code (Place Core Segments)

Home.js - part_project - Visual Studio Code

```javascript
import React from 'react';
import CssBaseline from '@material-ui/core/CssBaseline';
import Typography from '@material-ui/core/Typography';
import { makeStyles } from '@material-ui/core/styles';
import Container from '@material-ui/core/Container';
import IconButton from '@material-ui/core/IconButton';
import InputBase from '@material-ui/core/InputBase';
import SearchIcon from '@material-ui/icons/Search';
import Paginate from '../pagination';
import {getRentals} from '../../actions/rentals';
import { useState } from 'react';
import { Grid, Paper } from '@material-ui/core';
import { useHistory,useLocation } from 'react-router-dom';
import { useSelector } from 'react-redux';
import Rentalpost from '../Rentals/Rentalpost';
import ChipInput from "material-ui-chip-input";

const useStyles = makeStyles((theme) => ({
  icon: {
    marginRight: theme.spacing(2),
  },
  heroContent: {
    padding: theme.spacing(15, 0, 0),
    height:'400px',
    backgroundColor:'#14274E',
  },
  heroButtons: {
    marginTop: theme.spacing(4),
  },
  indet:{
    backgroundColor:'#fff',
    borderRadius:10,
    padding:1,
    // padding:'100px'
  },
  input:{
```


Navbar.js - part_project - Visual Studio Code

```javascript
import React,{useState,useEffect} from 'react';
import PropTypes from 'prop-types';
import AppBar from '@material-ui/core/AppBar';
import CssBaseline from '@material-ui/core/CssBaseline';
import Divider from '@material-ui/core/Divider';
import Drawer from '@material-ui/core/Drawer';
import Hidden from '@material-ui/core/Hidden';
import IconButton from '@material-ui/core/IconButton';
import List from '@material-ui/core/List';
import ListItem from '@material-ui/core/ListItem';
import ListItemIcon from '@material-ui/core/ListItemIcon';
import ListItemText from '@material-ui/core/ListItemText';
import MenuIcon from '@material-ui/icons/Menu';
import Toolbar from '@material-ui/core/Toolbar';
import Typography from '@material-ui/core/Typography';
import { makeStyles, useTheme } from '@material-ui/core/styles';
import { Switch, Route, Link, BrowserRouter,Redirect} from 'react-router-dom';
import Search from './Search';
import SearchIcon from '@material-ui/icons/Search';
import HomeIcon from '@material-ui/icons/Home';
import Button from '@material-ui/core/Button';
import FormatAlignCenterIcon from '@material-ui/icons/FormatAlignCenter';
import AddCircleOutlineIcon from '@material-ui/icons/AddCircleOutline';
import LockOpenIcon from '@material-ui/icons/LockOpen';
import ContactUs from './ContactUs';
import ContactSupportIcon from '@material-ui/icons/ContactSupport';
import ExitToAppIcon from '@material-ui/icons/ExitToApp';
import InfoIcon from '@material-ui/icons/Info';
import Info from './Info';
import Home from './Home';
import Footer from './Footer';
import RentalF from '../Rentals/RentalF';
import RentalManage from '../Rentals/RentalManage';
import { useDispatch } from 'react-redux';
import {getRentals} from '../../actions/rentals';
import { Avatar } from '@material-ui/core';
import Auth from '../Auth/Auth';
```

```javascript
95          width: '100%',
96      },
97
98  }));
99
100
101 const RentalPosts = (( setCurrentId )) => {
102     const { rentals, isLoading } = useSelector((state) => state.rentals);
103     const classes = useStyles();
104     const dispatch = useDispatch()
105
106     if (!rentals?.length && !isLoading) return 'No posts';
107
108     return (
109         <div>
110
111         <h1> hello vikas</h1>
112             isLoading ? <CircularProgress /> : (
113
114         <Grid container spacing={4} className={classes.card} >
115             {rentals?.map((rental) => (
116                 <Grid key={rental._id}  container item  md={3} >
117                     <Rentalpost rentals={rentals} setCurrentId = {setCurrentId} />
118                     </Grid>
119
120             ))};
121         </Grid>
122         );
123     </div>
124     );
125 };
126
127 export default RentalPosts;
128
```



```javascript
25          const rentals = useSelector((state) => state.rentals);
26
27          const classes = useStyles();
28
29          return (
30              <div className={classes.manage}>
31              <div className={classes.title}>
32                  <h1>Your rental Property</h1>
33                  {/* <h1>Nothing to Show</h1> */}
34  <Container>
35              <Grid container spacing={4} className={classes.card}  >
36                  {rentals.map((rental) => (
37                  <Grid key={rental._id} className={classes.box} item  md={4} >
38                      <Rentalpost rental={rental}  />
39                      </Grid>
40              ))};
41          </Grid>
42          </Container>
43
44
45
46              </div>
47              </div>
48      )
49  }
50
```
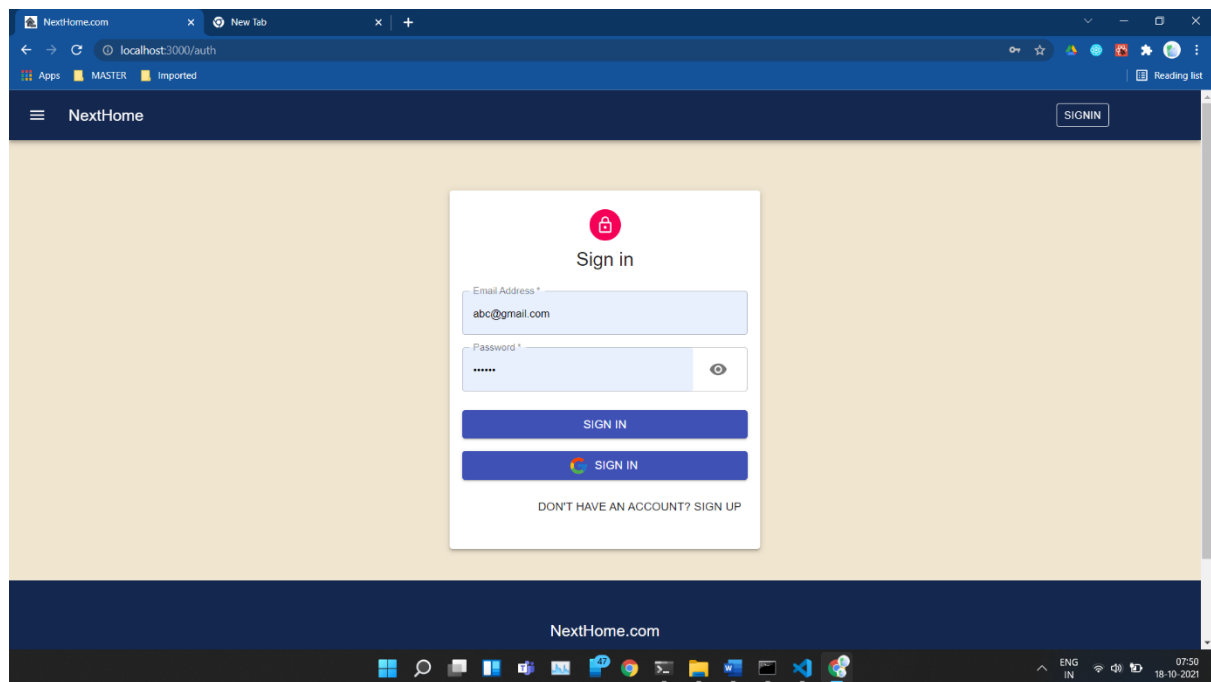
5.2 Testing Approach and Test Cases

1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs

as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.
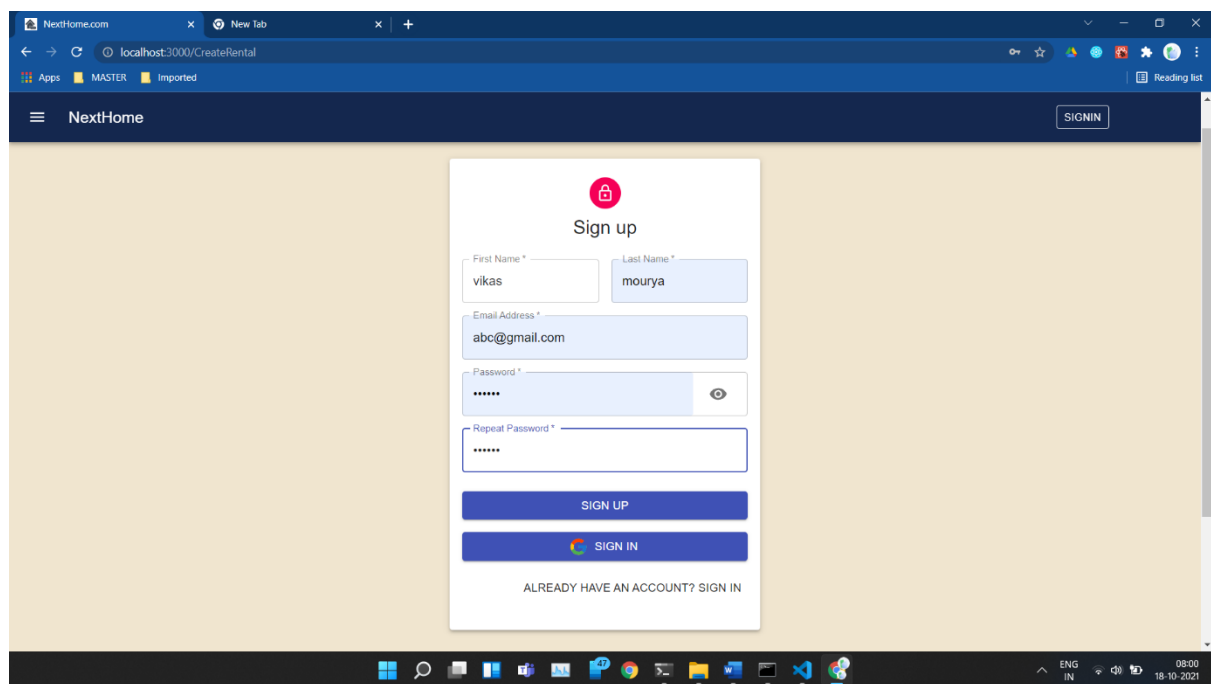
4.3.2 Integrated Testing

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of

multiple software modules, coded by different programmers. Integration Testing focuses on checking data communication amongst these modules.

4.3.3 Beta Testing

Beta Testing is one of the Acceptance Testing types, which adds value to the product as the enduser (intended real user) validates the product for functionality, usability, reliability, and compatibility. Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively

Test Cases

Pre-requisite: Appropriate database with sample record should be ready.

Test Input/data:

User Login:

Test case id	Description	Expected resulted	Actual result	Remark
TC1	Email : Vikasgmail.com Password:Vikas232	Grant login	Unsuccessful	Fail
TC2	Email : Vikas@gmail.com Password:Vikas232	Grant login	Login successful	pass

User Registration:

Test case id	description	Expected result	Actual result	Remark

TC1	FirstName :Vikas LastName: mourya Email:vikas123@gmail.com Contact:1234567890 Password:121234	Successfully registered	Successfully registered	Pass
TC2	FirstName :Vikas LastName: mourya Email:vikas123@gmail.com Contact:1234567890 Password:121234	Successfully registered	Registration unsuccessful Email already exist	Fail

Add Product

Test case id	description	Expected result	Actual result	Remark
TC1	Title:MRS house Desrition: 4BHK, Address: Rathodi Marve malad west mumbai Contact:1234567890 Rentaltype : bunglow Tags: #orlem	Product added Successfully	Product added Successfully	Pass

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

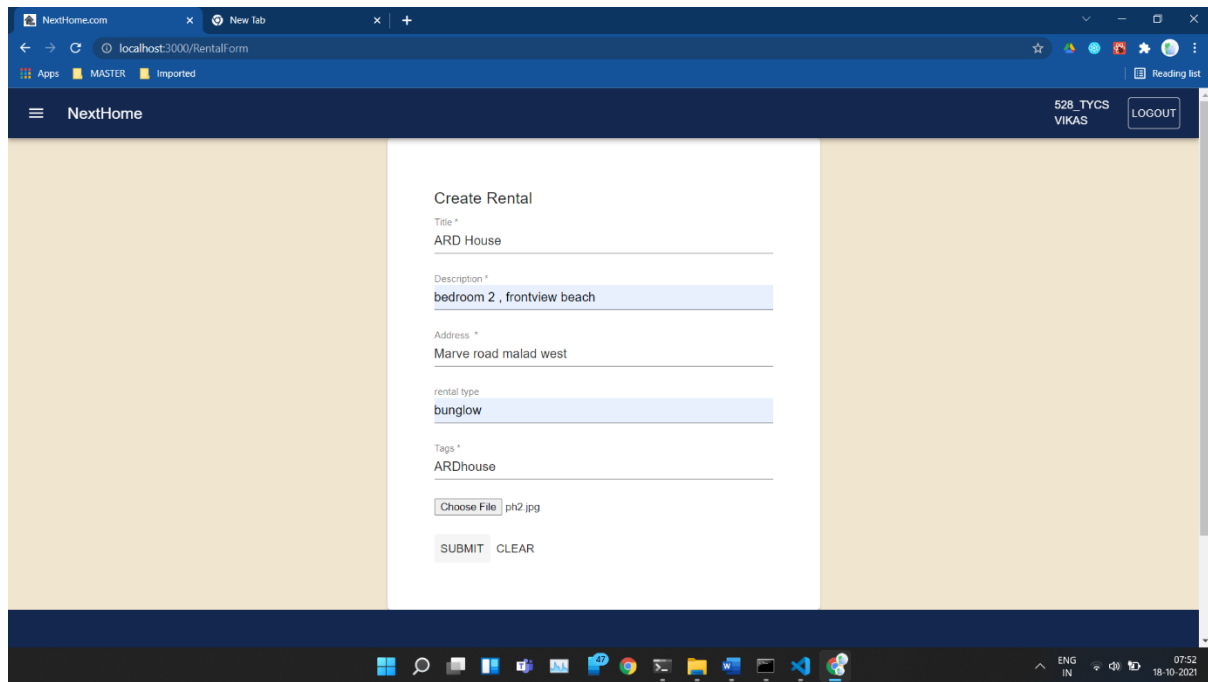6.2 User Documentation:
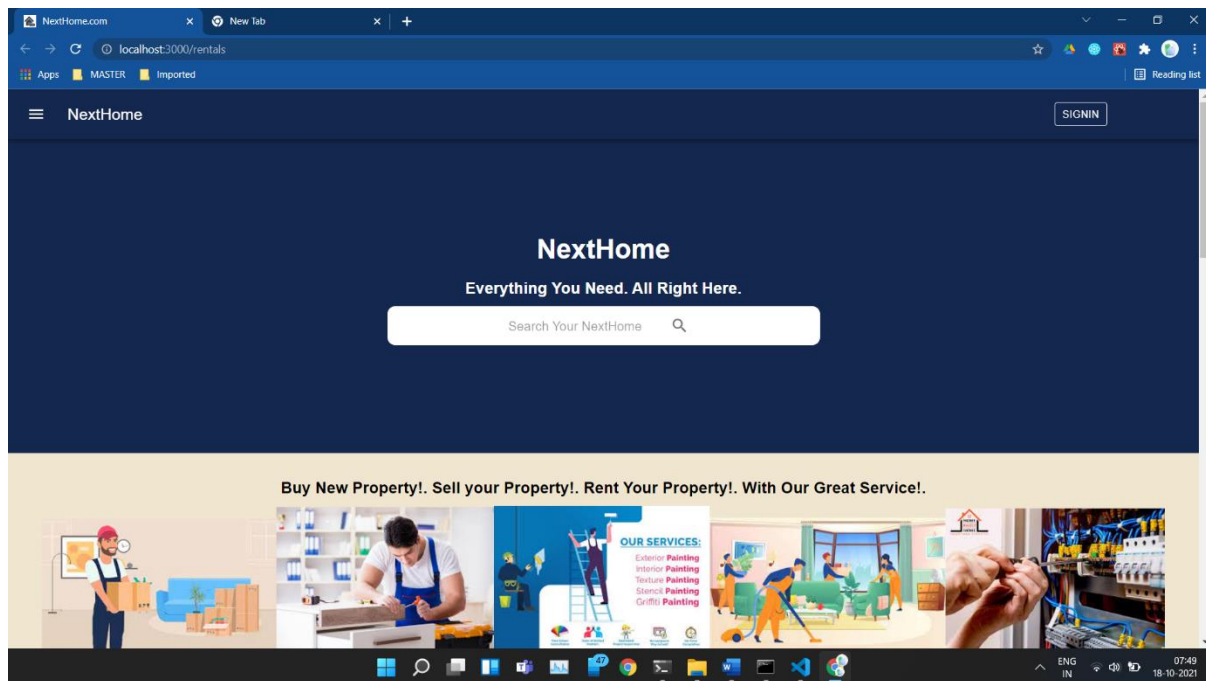
Home page: 1
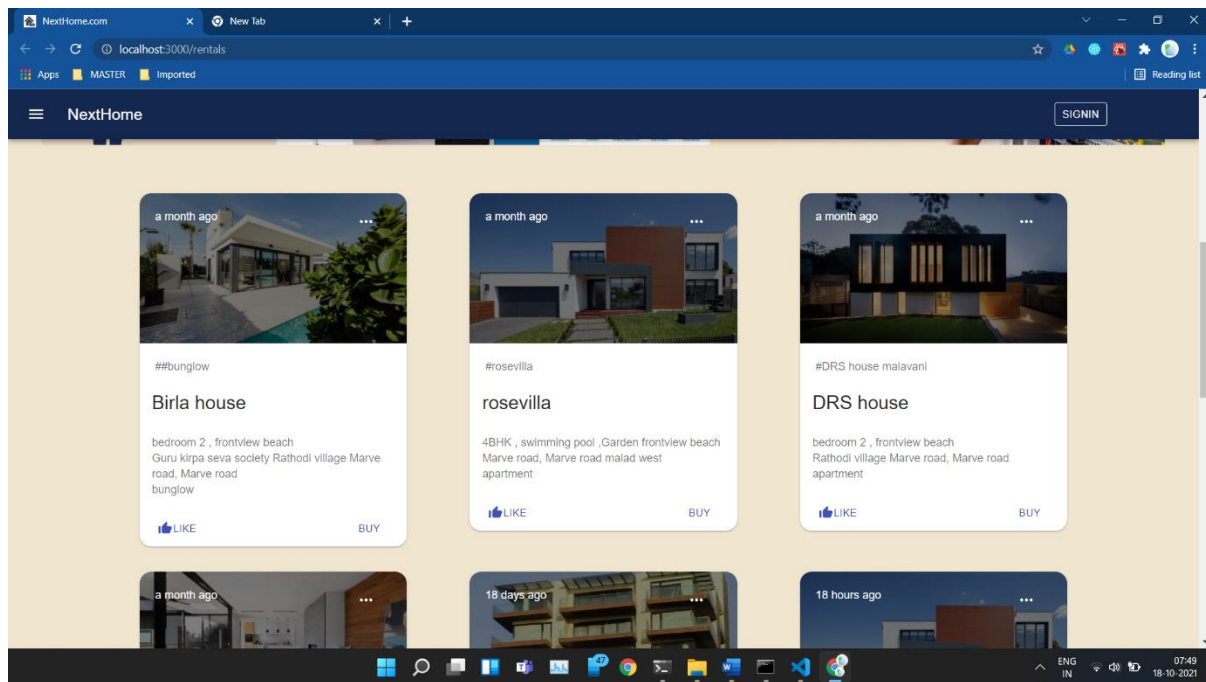
Product/Property page
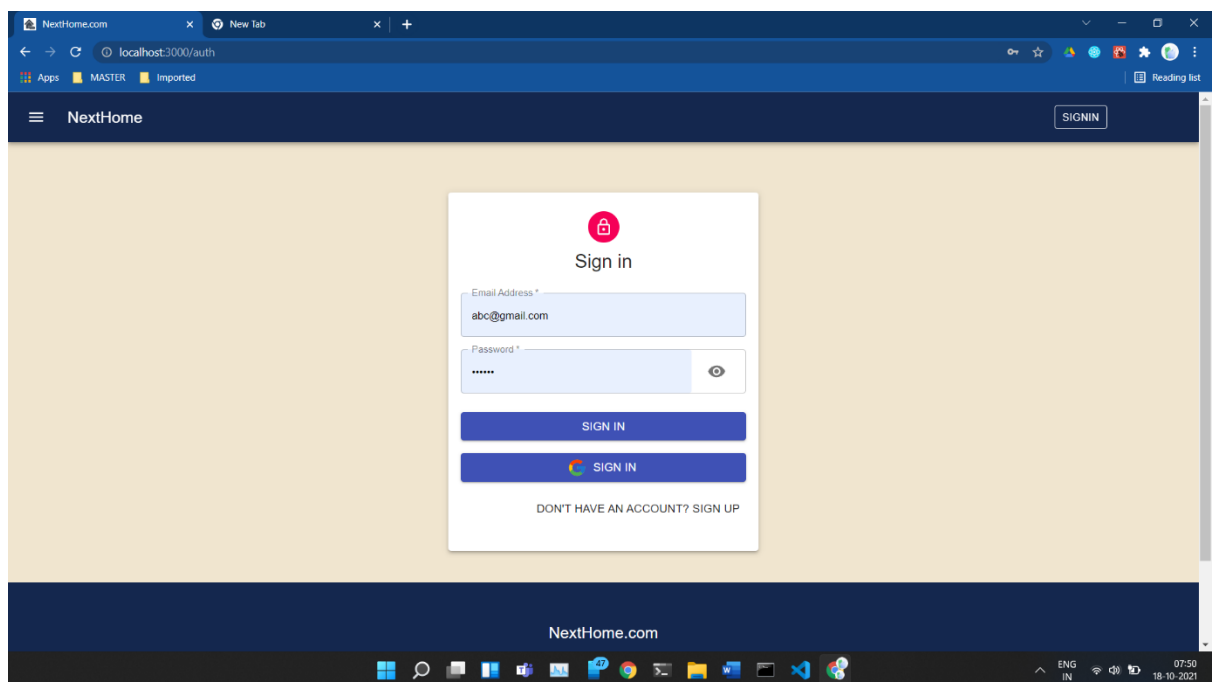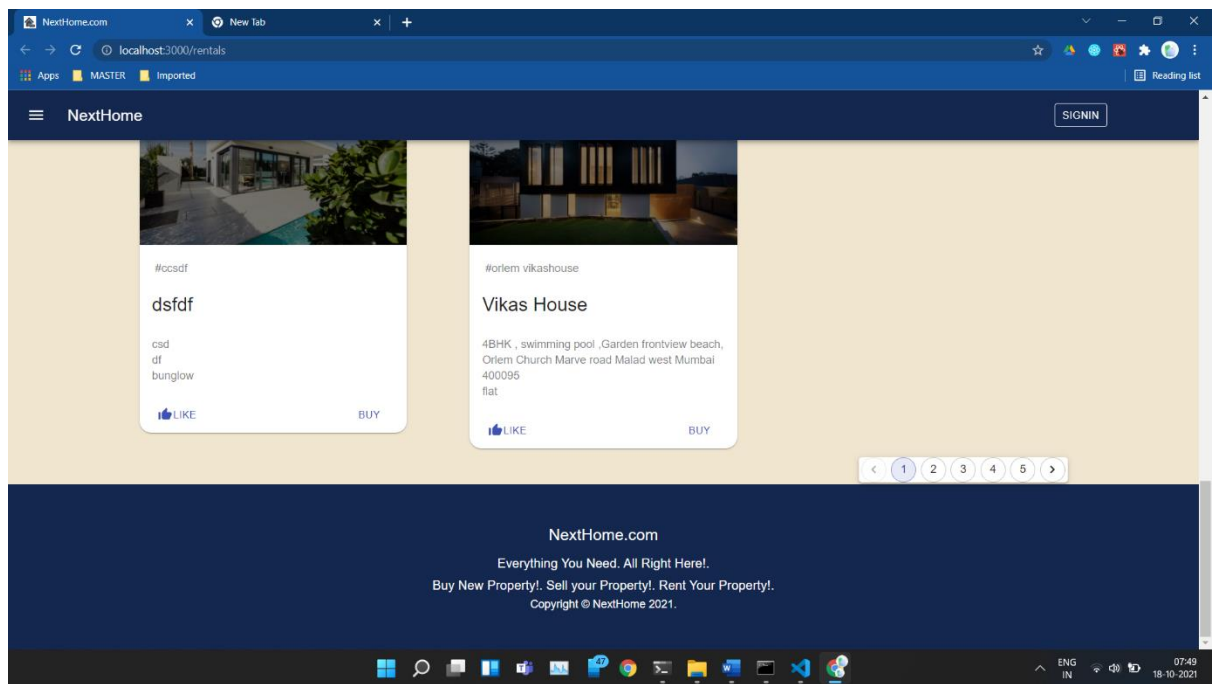
Sign In page
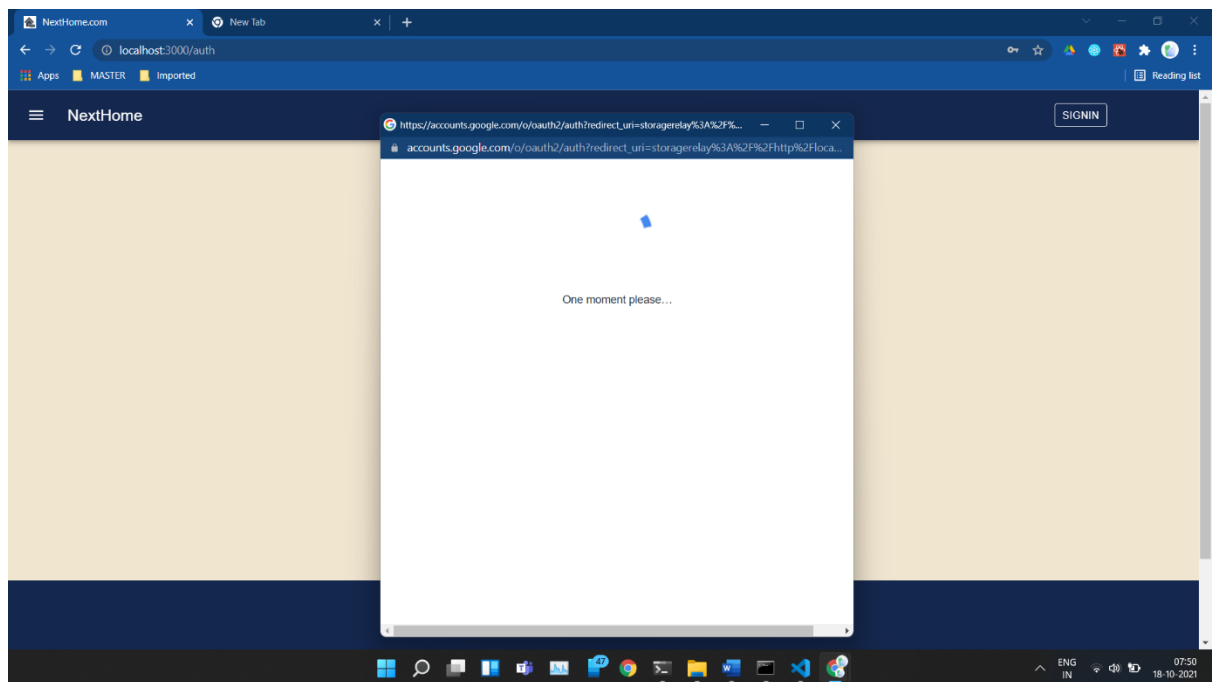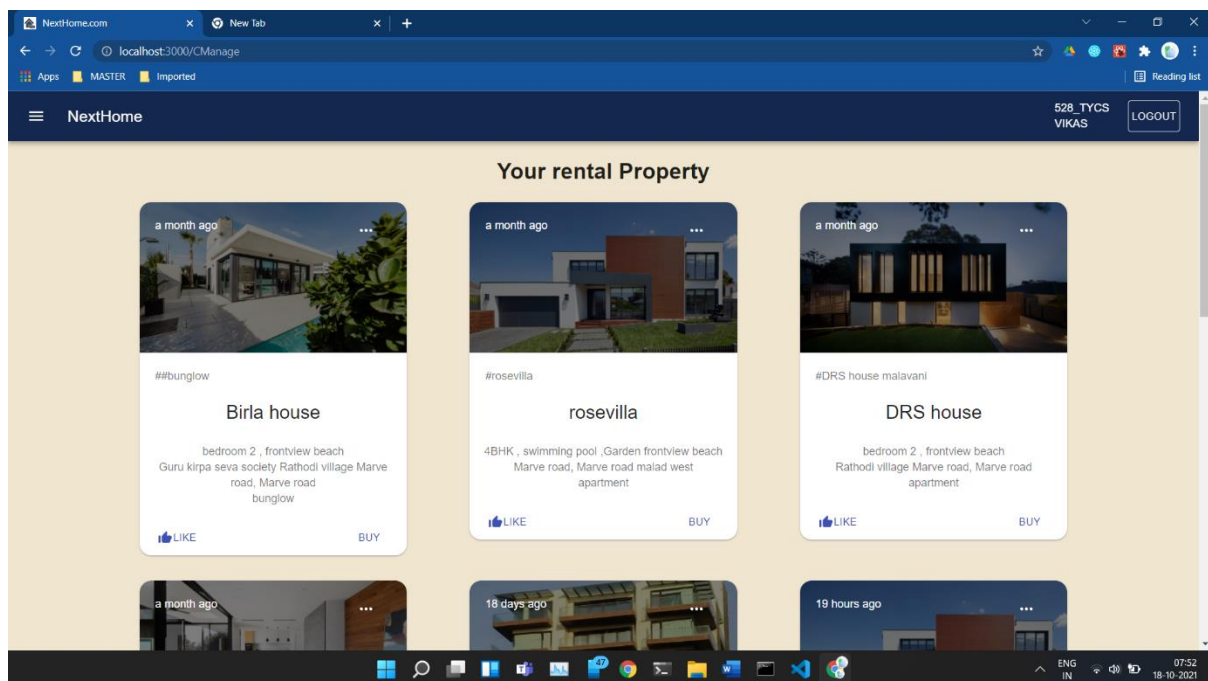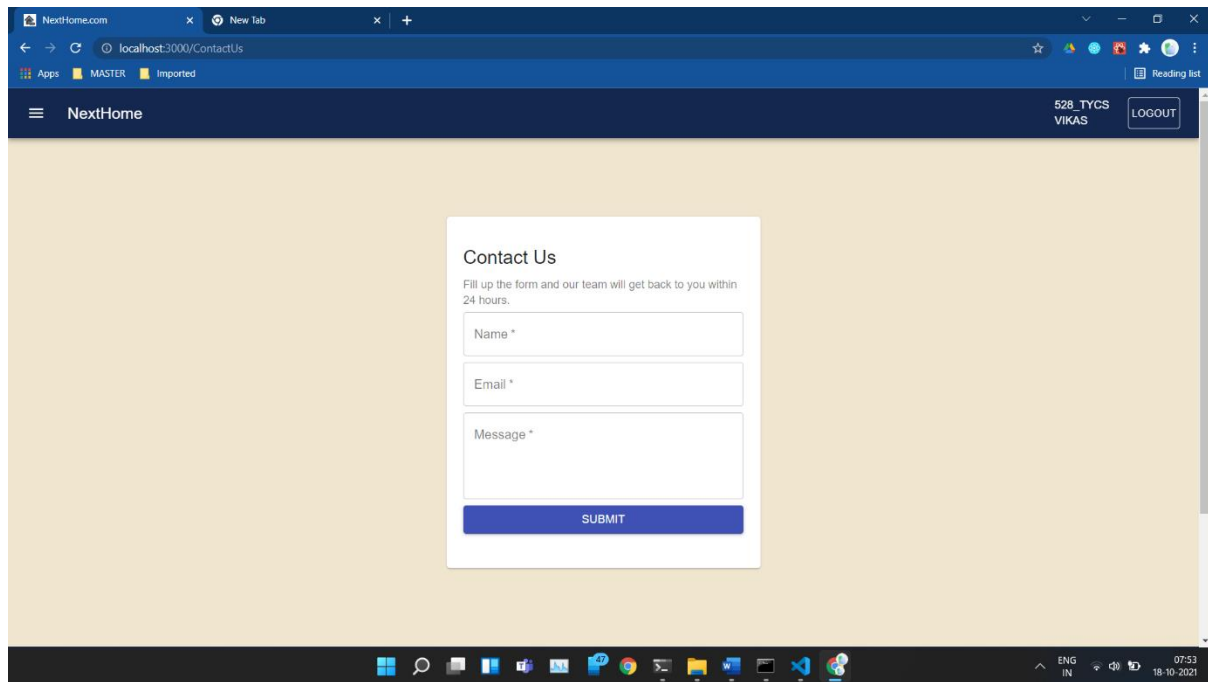
Sign In google
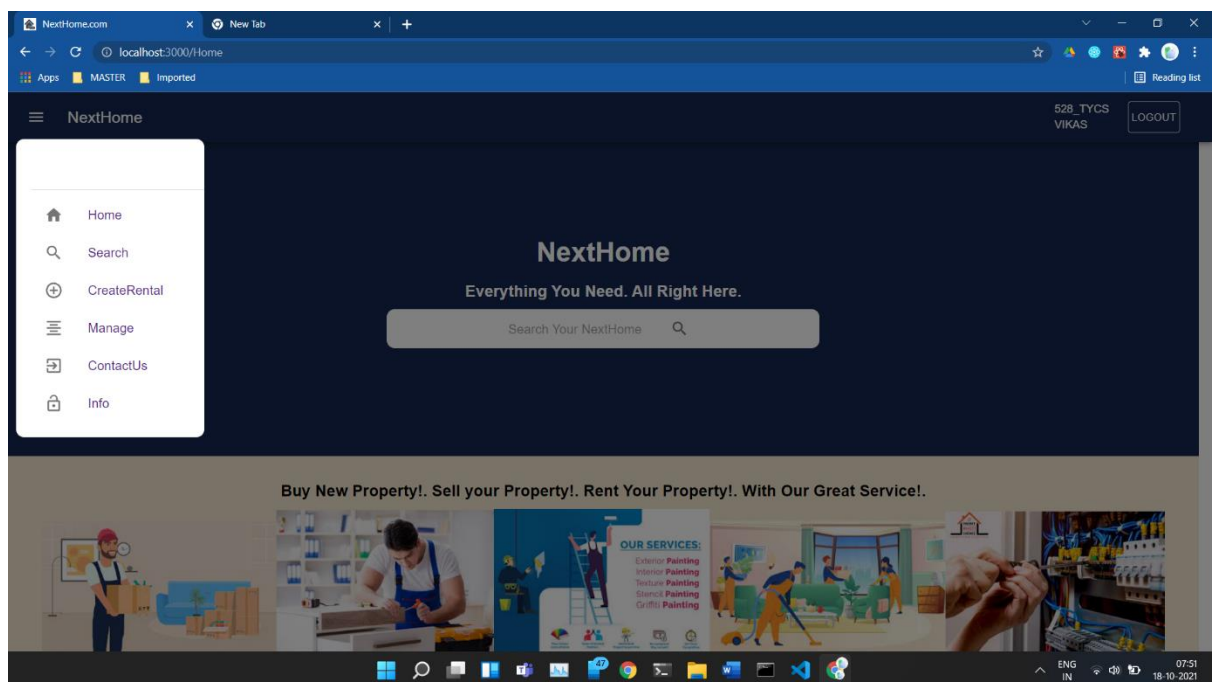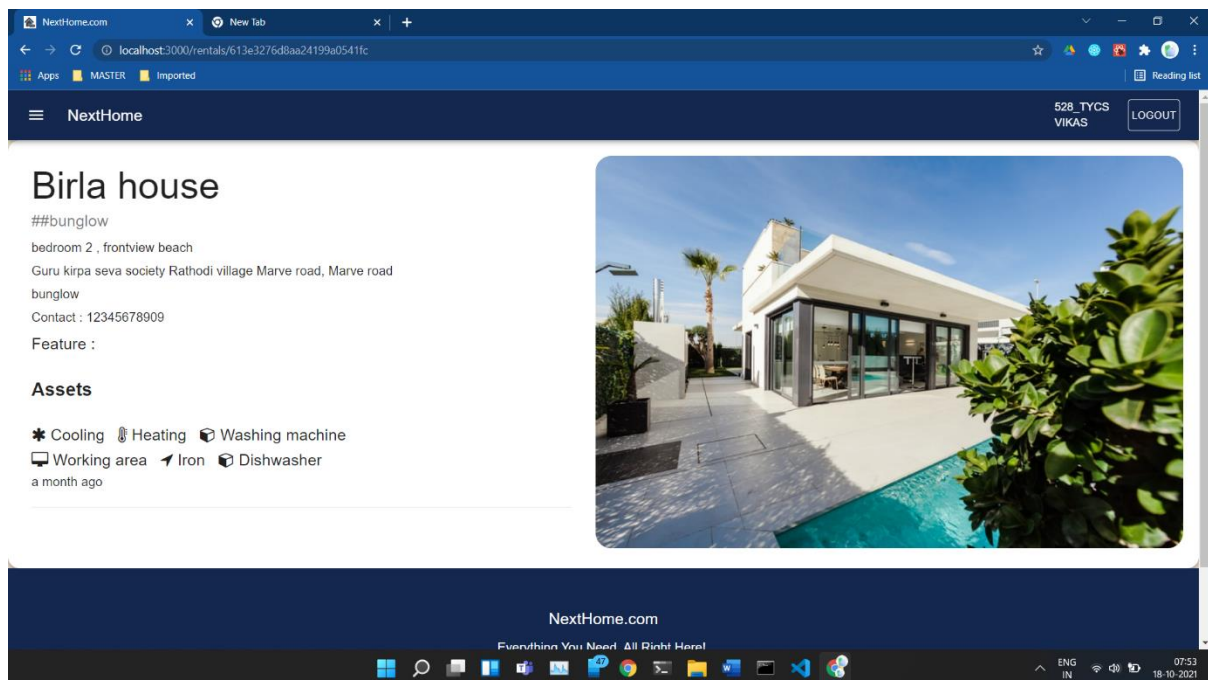
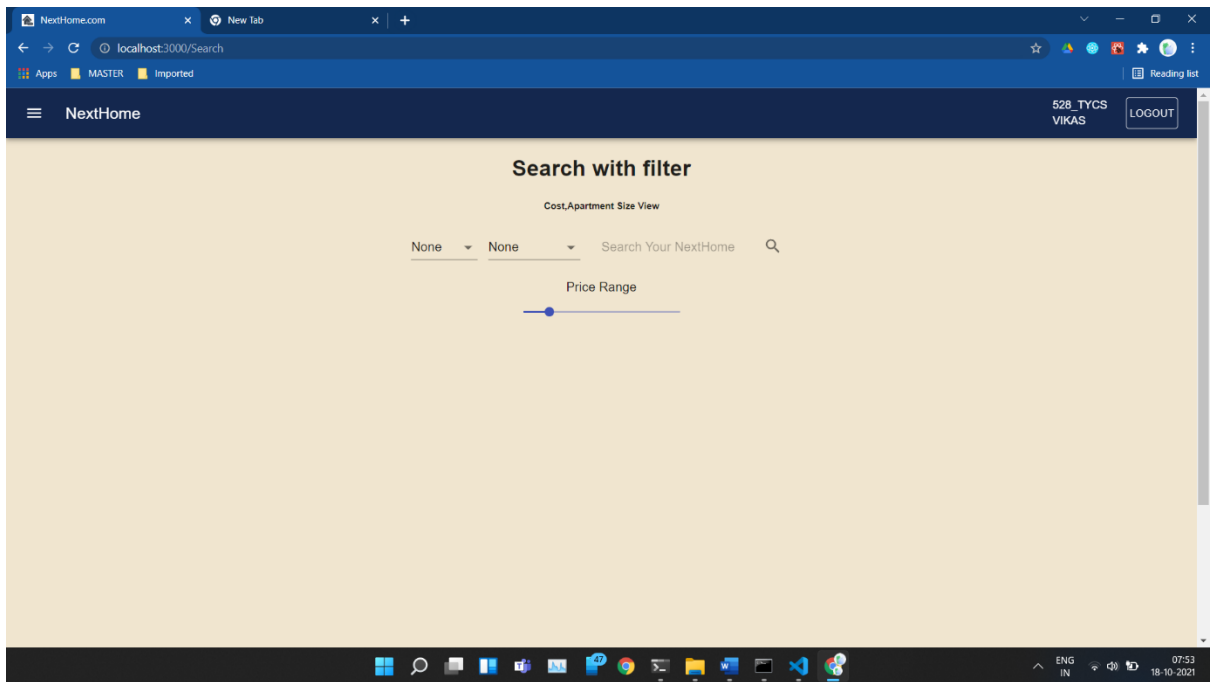

After Login page With Navbar
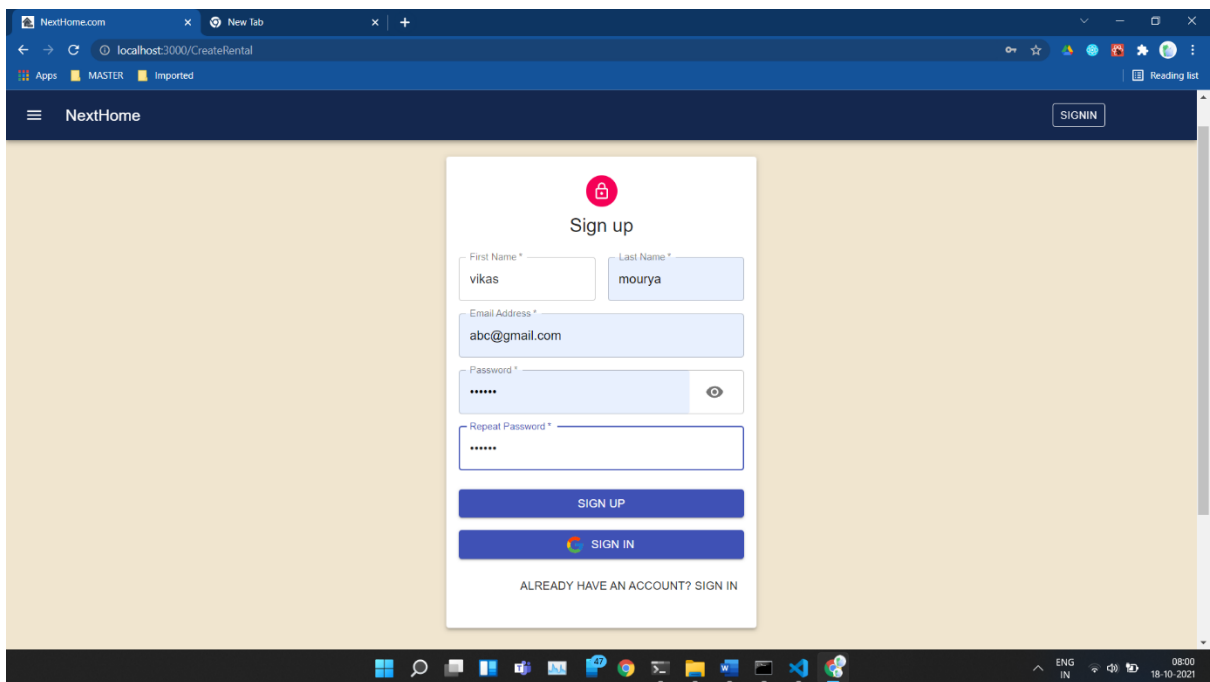
Add Property Page

Your Rental Property page

Contact Us page

Property Description page

Search with filter Page

Sign Up page

CHAPTER 7: CONCLUSIONS

7.1 Conclusion:

House Rental business has emerged with a new goodies compared to the past experience where every activity concerning House rental business is limited to a physical location only. Even though the physical location has not been totally eradicated; the nature of functions and how these functions are achieved has been reshaped by the power of internet. Nowadays, customers can reserve book/buy/sale House online, rent House online, and have the house contracted successfully without any sweat once the customer is a registered member of the House Rental Management System. The web based House rental system has offered an advantage to both Tenants as well as Landlords to efficiently and effectively manage the business and satisfies customers' need at the click of a button.

GUI makes the interface very much user friendly. The system is highly user friendly and is well efficient to interactions with the users of the system.

7.2 Significance of the System

To enhance company value: In today's era more than half of population is connected with internet. And people use to search the details and services online to increase their knowledge about the online trends and its services. In this busy world even they try to shop in the website online. Therefore, if a company doesn't have its website then it is losing a large part of customer.

Increase the goodwill of company: Whenever you are introducing your company to any user and he or she is interested in having a deal then if you have a website it definitely uplifts your company value.

Online promotion: Companies spend lot of money and efforts in promoting their products through pamphlets, brochure, advertisements, etc. but if you have website you can skip this expense by publishing information in your website.

Works 24X7: Website are always available all the time (24 hrs.) round the year for your clients even beyond the boundaries of your country.

7.3 Limitation:

Limitation of the present system:

● It requires an internet connection.

● It requires a large database
● System: The system at present does not take care of the money payment methods, as the consolidated constructs need SSL standards and are critically to be initiated in the first face, the application of the credit card transactions is applied as a developmental phase in the coming days. The system needs more elaborative technicality for its inception and evolution.

7.3 Future scope of the project:

The system 'Home Renting website ' will fulfil the entire requirement of the users. The system is developed as easy as possible for the sake of end users. In the present system payment is available only through cash on delivery but in future online payment ,admin panel will be made. The developed software for the organization is flexible and it can be made to run on all kinds of platforms. The system is error free and highly portable. It can be implemented in any servers in the Internet providing an easy access to the clients. It also has more options of the future developments.

REFERENCES

www.stackoverflow.com

www.w3school.com

www.google.com

www.reactjs.org

www.reserachgate.com