A
Project Report
on

HOUSE MARKETPLACE WEBSITE

developed at

CODSOFT

for the partial fulfilment of

MASTER OF COMPUTER APPLICATION

ACADEMIC YEAR 2022-2024

Under the supervision of
Mr. Sanjay Kr. Pandey
Asst. Professor at SRMCEM

Submitted by
Vinay Sharma
Roll No: 2201220140098

C.ID: 142d2c4

CodSoft

CERTIFICATE

OF COMPLETION
PROUDLY PRESENTED TO

Vinay Sharma

has successfully completed 4 weeks of a virtual internship program in
Web Development
with wonderful remarks at **CODSOFT** from **05/09/2023 to 05/10/2023.**
We were truly amazed by his/her showcased skills and invaluable contributions to the tasks and projects throughout the internship.

Founder

contact@codsoft.in www.codsoft.in Date: 07/10/2023

SHRI RAMSWAROOP MEMORIAL COLLEGE OF ENGINEERING & MANAGEMENT

DEPARTMENT OF COMPUTER APPLICATION

SRMCEM

CERTIFICATE

This is to certify that the project entitled "House Marketplace Website" submitted by **Vinay Sharma [2201220140098]**and in thepartial fulfillment of the requirement for the award of the degree of Master of Computer Application of Dr. A.P.J. Abdul Kalam Technical University, is a record of student's own work carried under our supervision and guidance. The project report embodies result of original work and the study carried out by the student and the contents do not form the basis for the award of any other degree to the candidate or to anybody else.

Mr. Sanjay Kr. Pandey
(Project Guide)

Prof. Ajay Kr. Srivastava
(Head of Department)

SHRI RAMSWAROOP MEMORIAL COLLEGE OF ENGINEERING & MANAGEMENT

DEPARTMENT OF COMPUTER APPLICATION

SRMCEM

DECLARATION

I hereby declare that the project entitled "House Marketplace Website" submitted by me in the partial fulfillment of the requirement for the award of the degree of Master of Computer Application of Dr. A.P.J. Abdul Kalam Technical University, is a record of my own work carried under the supervision and guidance of Mr. Sanjay Kr. Pandey.

To the best of my knowledge this project has not been submitted to Dr. A.P.J. Abdul Kalam Technical University, or any other University or Institute for the award of any degree.

\Name- Vinay Sharma
Roll. No. 2201220140098

ACKNOWLEDGEMENT

I am indebted to my HOD **Dr. Ajay Kumar Srivastava** for giving me this platform for sharing my major project experience.

I am also grateful to my major project guide faculty member **Mr. Sanjay Kr. Pandey** for giving me the guidelines to properly present the topics learnt which also helped me in doing a lot of research and I came to know about so many valuable resources and a stress-free environment. I am very thankful to almighty for giving me such opportunity to share my experience.

Last but not least I am thankful to all my friends who always encouraged me and supported me.

Vinay Sharma
(2201220140098)

SHRI RAMSWAROOP MEMORIAL COLLEGE OF ENGINEERING &MANAGEMENT, LUCKNOW

PREFACE

The successful completion of the "House Marketplace website" project marks a significant milestone in our academic journey. This project has been a comprehensive exercise in applying the theoretical knowledge acquired during our coursework to a practical, real-world problem. It aims to address the myriad administrative and operational challenges faced by educational institutions through an integrated, user-friendly, and efficient software solution.

The primary objective of this project was to design and develop a robust system that can manage and streamline various activities within a college, ranging from student enrollment and attendance tracking to faculty management and timetable scheduling. By automating these processes, we sought to reduce the administrative burden on staff and enhance the overall efficiency of college operations.

In undertaking this project, we were driven by a commitment to innovation and excellence. The development process involved extensive research, meticulous planning, and rigorous testing to ensure the system meets the high standards required for academic institutions. We employed modern software engineering practices and tools, drawing on our skills in programming, database management, and user interface design

INDEX

HOUSE MARKETPLACE WEBSITE

INTRODUCTION

This project revolves around the development of a House Marketplace Website using React JS, aiming to provide users with a cutting-edge platform for buying, selling, and renting properties. The endeavour encompasses various aspects of web development, including front-end design, user experience optimization, and efficient data management. This report will delve into the project's objectives, methodologies, challenges encountered, and the ultimate outcomes achieved during the development process.

In the ever-evolving landscape of real estate, finding your perfect home shouldn't be a daunting task; it should be an exciting journey. Enter House Marketplace, where we redefine the house hunting experience with innovation and user-centric design, all powered by the robust capabilities of React dive into an immersive exploration of diverse homes, from cozy apartments to luxurious estates. Our intuitive interface, powered by React, ensures a smooth and responsive journey as you browse through a collection of properties.

At House Marketplace, we understand that real estate transactions are significant milestones in your life. Whether you're a first-time buyer, a seasoned investor, or looking to sell your property, our platform is tailored to meet your unique needs. Powered by React, our website offers an unparalleled user experience, ensuring that you have access to the latest listings, market trends, and expert advice at your fingertips.

In this increasingly demanding scenario for a platform that could help buyers to have a look at the available property listings with all its photographs, necessary specification and description, this application is an effort to provide such a web interface with some attractive and innovative features using the latest technologies.

The most significant feature in these websites is the interactive search criterion which lets the buyer specify their requirements to get the correct set of records from the database. The search tool should be strong enough to include all the required features which a buyer may desire. Some other new features like the drag and drop tools to save the listings, retrieving related results for each search, etc make this application more feature-rich.

IDENTIFICATION OF NEED:

Identifying and addressing these needs in the development of a House market place website project can significantly enhance the overall user experience and contribute to the success of the platform in meeting the evolving demands of the propertydealing.

- User-Friendly Interface
- Comprehensive Destination Information
- Security and Trust
- User Experience and Interface Design
- Personalized Travel Itineraries
- Local Expert Recommendations
- Mobile Responsiveness

FEASIBILITY STUDY:

A feasibility study is a crucial step in assessing the viability and potential success of a project. In the case of our mini project, the development of an online shopping platform, the feasibility study encompasses technical, economical, and operational aspects.

Technical Feasibility:

- The project requires a robust and scalable technology infrastructure. A thorough assessment of available technologies, development tools, and platforms indicates that the project can leverage widely adopted and well-supported technologies for web development.

- The chosen technologies allow for scalability and optimal performance. The system can handle increased user loads and transaction volumes, ensuring a responsive and reliable online shopping experience.

Economic Feasibility:

- A comprehensive cost-benefit analysis indicates that the benefits of the online shopping platform outweigh the associated costs. These benefits include increased sales opportunities, expanded market reach, and operational efficiencies that can lead to long-term economic gains.

Operational Feasibility:

- The operational impact on end-users is minimal due to the user-friendly interface and intuitive design. User training requirements are limited, contributing to operational feasibility.

AIMS AND OBJECTIVE

The foremost objective of this project was to give a different visualization style to the Real Estate Website which has more features, attractive animations and altogether a new look in contrast to the already existing websites. Usually, in a real estate website, the property search page consists of a traditional search style i.e. a set of textboxes / drop-downs to select a particular county for the property to search along with other web controls to specify the number of beds/baths or any additional features they are looking in the property.

The next objective of importance was to build a real estate website that not only reduces the annoying postbacks and loss of control focus but also gives a faster and more interactive user interface. Moreover, to make the website more features rich, features like a customized grid, drag and drop tool, accordion panels and sliding bars were added to the website.

- **Generate leads:** The website should be able to generate leads.
- **Legitimacy:** The website should define the broker as a legitimate real estate professional.
- **Authority:** The website should present the broker as an authority on the local market.
- **Display listings:** The website should help the real estate company display an unlimited number of property listings.
- Real estate websites are a digital space where realtors, home sellers, and homebuyers can get in touch to make a purchase or rent contract. Prospective buyers and sellers can browse listings, gather information, and submit inquiries anytime, even outside business hours.
- A well-designed website can make the broker and their business look more professional. A down-to-a-point bio can tell people what the broker does and who they are.
- It is to help the Real Estate Company to display unlimited number of property listings on the website.

Real estate investors may also have other goals, such as: Passive income, Capital growth, Stock diversification, Wealth preservation, Long-term wealth creation

PROJECT CATEGORY

The project category of a house selling project typically falls under "Real Estate" or "Property Management." This encompasses various aspects such as property listings, marketing strategies, client management, legal considerations, financial transactions, and potentially even aspects of interior design or home staging. Within these categories, specific project tasks may include creating property listings, organizing open houses, conducting market research, negotiating offers, and managing contracts and paperwork.

KEY COMPONENTS OF PROJECT CATEGORY:

- **E-Commerce:**

  The project revolves around the creation of ahouse selling platform, providing users with the ability to browse, selling, and services through a digital interface. House selling and renting solutions involve various functionalities such as selling home and checking for renting homes, booking, secure payment gateways.

- **User Interface Design:**

  A significant aspect of this project category is the design of an intuitive and user-friendly interface. The user interface is crucial for ensuring a positive and engaging home selling or renting, incorporating elements of responsive design and accessibility to cater to a diverse user base.

- **Mobile Commerce:**

  With the increasing prevalence of mobile devices, projects in this category often consider mobile commerce aspects. This includes optimizing the platform for mobile use, ensuring a seamless and responsive experience on smartphones and tablets.

- **User Engagement and Feedback:**

  Enhancing user engagement is a key focus of projects in this category. Features like personalized recommendations, Wishlist, and feedback mechanisms contribute to a more interactive and user-centric experience.

The "House Marketplace website and Web Development" category encompasses projects that leverage technology to create online platforms, providing a seamless and secure environment for rent, buy and sell home or services over the internet.

REQUIREMENT ANALYSIS

SOFTWARE REQUIREMENT:

MICROSOFT VISUAL CODE:

Visual Studio Code is a popular and powerful source code editor developed by Microsoft. It offers a wide range of features, including syntax highlighting, code completion, debugging capabilities, version control integration, and an extensive library of extensions. VS Code is widely used for web development, programming in various languages, and general text editing tasks.

FIREBASE:

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, ios, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

NODE JS:

Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser. Ryan Dahl developed it in 2009, and its latest iteration,version 15.14,was released in April 2021. Developers use Node.js to create server-side web applications, and it is perfect for data-intensive applications since it uses an asynchronous, event-driven model.

GOOGLE CHROME:

Chrome, a widely used web browser, offers an extension known as "Live Server" that aids developers in real-time web development. This extension enables instant preview and live reloading of web pages as code changes occur. By launching a local development server within the browser, Chrome's Live Server extension provides a dynamic environment for developers to view their changes immediately without manual refreshing. It supports a streamlined development process by providing a responsive and synchronized preview of HTML, CSS, and JavaScript alterations, enhancing productivity and facilitating rapid iteration during web development tasks.

GitHub:

GitHub functions as a prominent web-based platform built around Git, offering a collaborative space for version-controlled software development. It provides a centralized repository hosting service where developers can store, manage, and share code. GitHub facilitates seamless collaboration among teams through features like pull requests, issue tracking, and project management tools. It serves as a hub for open-source projects, enabling community contributions, code review, and integration with various development workflows.

HARDWARE REQUIREMENTS :

Computer:

- **Processor:** Dual-core or higher processor (Intel Core i3 or equivalent)
- **RAM:** Minimum 4GB (8GB or more recommended for smoother performance)
- **Storage:** At least 256GB SSD or HDD for storing project files and development tools.

Operating System:

Windows, macOS, or Linux-based system.

Internet Connection:

Stable internet connection for accessing online resources, repositories, and cloud services (like Firebase)

Additional Peripherals:

- Keyboard and mouse or other pointing devices for input.
- Headphones or speakers for audio feedback (optional)
- External devices for testing and debugging (smartphones, tablets)

Web Browser:

Chrome, Firefox, Safari, or any modern web browser for testing and development purposes.

.

PROBLEM DEFINITION

Project Objective

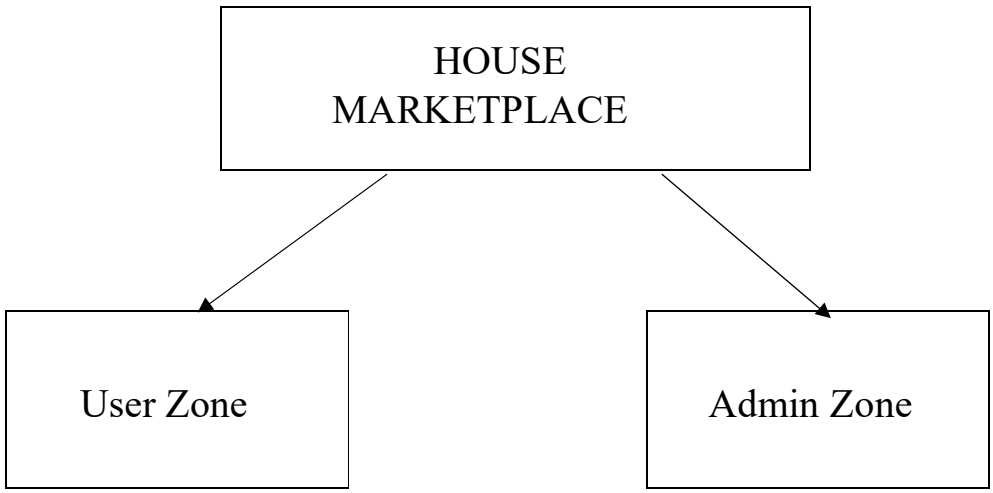
The objective of this project is to develop a user-friendly House Marketplace website using React JS. The platform aims to provide a seamless experience for users interested in buying, selling, or renting residential properties. The primary goal is to create a robust online marketplace that connects buyers, sellers, and renters in a secure, efficient, and visually appealing manner.

Proposed Methodology

Small to medium-sized commercial systems and personal computer devices are best suited for agile approaches.

This includes:

- Requirement Gathering
- Design Phase
- Technology Stack Selection
- Feature Implementation
- User Authentication & Security
- Testing & Quality Assurance
- Deployment & Maintenance



PROJECT PLANNING &PROJECT SCHEDULING

HOUSE MARKETPLACE						
S.NO	PHASE	START DATE	END DATE	DURATION	TOTAL DAYS	TOTAL DURATION
1	PLANNING AND ANALYSIS	01-8-23	08-8-23	3	8	24
2	DESIGN AND PROTOTYPING	09-8-23	14-8-23	3	6	18
3	APPLICATION DEVELOPMENT	15-8-23	01-9-23	5	16	80
4	PROJECT TESTING	02-9-23	08-9-23	4	7	28
5	IMPLEMENTATION	O9-9-23	15-9-23	2	6	12
6	OPERATIONS AND MANAGEMENT	16-9-23	19-9-23	2	4	8
				TOTAL	47	170

TOOLS DESCRIPTION

HTML (HYPERTEXT MARKUP LANGUAGE):

HTML is the standard markup language used to create the structure and content of web pages. It provides the foundation for organizing and presenting text, images, links, forms, and other elements on the web. HTML is essential for defining the layout and semantics of web documents, making it a fundamental building block of web development.

CSS (CASCADING STYLE SHEETS):

CSS is a style sheet language that complements HTML by controlling the visual presentation of web pages. It allows developers to define the appearance, layout, and design of elements within a web page, including colors, fonts, spacing, and responsive layouts. CSS plays a critical role in enhancing the aesthetic appeal and user experience of web content.

BOOTSTRAP 5 CSS:

Utilized for designing the website's user interface, Bootstrap 5 CSS framework offers a robust set of tools and components for creating responsive, visually appealing layouts. It aids in establishing consistent styling, grid systems, and responsive design elements across various devices, ensuring a modern and user-friendly interface.

REACT JS:

React is a framework that employs Webpack to automatically compile React, JSX, and ES6 code while handling CSS file prefixes. React is aJavaScript-based UI development library.

Although react is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development. React offers various extensions for entire application architectural support, such as Flux and React Native, beyond mere UI.

REACT ROUTER DOM V6:

Employed for Routing in Single Page Application (SPA) architecture, React Router Dom V6 facilitates navigation and enables the creation of dynamic, client-side routes. It helps in managing different views or pages within the application without the need for full-page reloads, providing a smoother and faster user experience.

NODE JS:

Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser. Ryan Dahl developed it in 2009, and its latest iterationversion 15.14,was released in April 2021. Developers use Node.js to create server-side web applications, and it is perfect for data-intensive applications since it uses an asynchronous, event-driven model.

FIREBASE V9 (Latest from Google Firebase):

Chosen as the backend technology, Firebase V9 by Google serves as a comprehensive backend-as-a-service (BaaS) platform. It provides functionalities such as database management, user authentication, cloud storage, and hosting. Firebase V9 offers real-time database capabilities, secure user authentication, and seamless integration with frontend technologies like React JS.

SCOPE OF SOLUTION

The scope and solution for a real estate project can vary depending on the specific goals and requirements. Here's a general outline:

SCOPE:

1. User Requirements Analysis: Gather requirements from stakeholders including buyers, sellers, agents, and administrators.
2. Market Research: Analyze the target market, competitor websites, and industry trends.
3. Feature Set Definition: Determine the features and functionality of the website including property search, listings management, user authentication, messaging, etc.
4. Design and User Experience: Develop wireframes and prototypes to visualize the website layout and user flow.
5. Development: Build the website using appropriate technologies and frameworks.
6. Testing: Conduct thorough testing to ensure functionality, performance, and security.
7. Deployment: Deploy the website to a hosting platform and ensure it is accessible to users.
8. Maintenance and Support: Provide ongoing maintenance, updates, and support to address any issues and keep the website running smoothly.

The scope of this project “Property Listing Web Application” is to enable the buyers to search for property listings online. The motive of developing this application is to design a feature-rich search engine that can make the search of commercial land/properties an easy task.

SOLUTION:

1. **User-Friendly Interface**: Design an intuitive interface that allows users to easily search for properties, view listings, and contact agents.
2. **Advanced Search Filters:** Implement advanced search filters such as location, price range, property type, amenities, etc., to help users find their desired properties quickly.
3. **Property Listings Management:** Develop a backend system for agents or administrators to manage property listings including adding, editing, and removing listings.
4. **User Authentication and Profiles**: Create user accounts for buyers, sellers, and agents, allowing them to save searches, favorite properties, and track their activity.
5. **Messaging System:** Integrate a messaging system that enables communication between users and agents directly through the website.
6. **Mobile Responsiveness:** Ensure the website is responsive and optimized for mobile devices to provide a seamless experience across different screen sizes.
7. **Security Features:** Implement security measures such as encryption, secure authentication, and data validation to protect user information and transactions.
8. **SEO Optimization:** Optimize the website for search engines to improve visibility and attract organic traffic.
9. **Analytics and Reporting**: Integrate analytics tools to track user behavior, monitor website performance, and generate reports for stakeholders.
10. **Scalability:** Design the website architecture to be scalable, allowing it to handle a large volume of users and listings as the platform grows.

    By defining the scope and implementing the proposed solution, the real estate project aims to address the challenges in the industry and provide a comprehensive platform that meets the needs of buyers, sellers, and agents alike.

MODULE DESCRIPTION

USER-END MODULES:

Explore Module:

Home page contains menu bar with logo, a slider and footer. It is the first and foremost page which will help user to interact with the platform.

- **To Rent Module:**

  Creating a rental module for a real estate website involves managing and displaying rental property listings, providing details, and potentially allowing users to inquire about or book properties.

- **To Sale Module:**

  This module helps in a binding contract that establishes the agreed-upon terms for the sale of a property. For example, the purchase price, the closing date, the type of property being sold and any contingencies or conditions that must be met before the sale is finalized.

Offers Module:

Offers module collectively form the backbone of a real estate project, providing users with a seamless and feature-rich experience while addressing the specific needs of buyers, sellers, and agents in the real estate industry.

Profile Module:

This section page contains various services provided by house marketplace.Agents or sellers should be able to create new property listings. This involves entering details such as property type (house, apartment, etc.), location, price, size, number of bedrooms and bathrooms, amenities, photos, and a description of the property.

- **Sign Up Module:**

  It facilitates the user to register on the portal by just providing some details like name, contact number, email, address, etc.

- **Login Module:**

  In this module, the user will be able to login into the app. And it is required for all users to login into the system without that user will not be able to place an order or enter app.

Contact Us Module:

Contact Us module helps the user to share their views and problem through some contact related details.

ADMIN-END MODULES:

Login Module:

In this module, the admin will be able to login into the system by providing some login details.

Registered User Details:

In registered details module the admin will be able to see the consumer who have registered themselves by providing their details.

Subscribed Module:

In subscribed details module the admin will be able to see the users who have subscribed the newsletter.

Query Module:

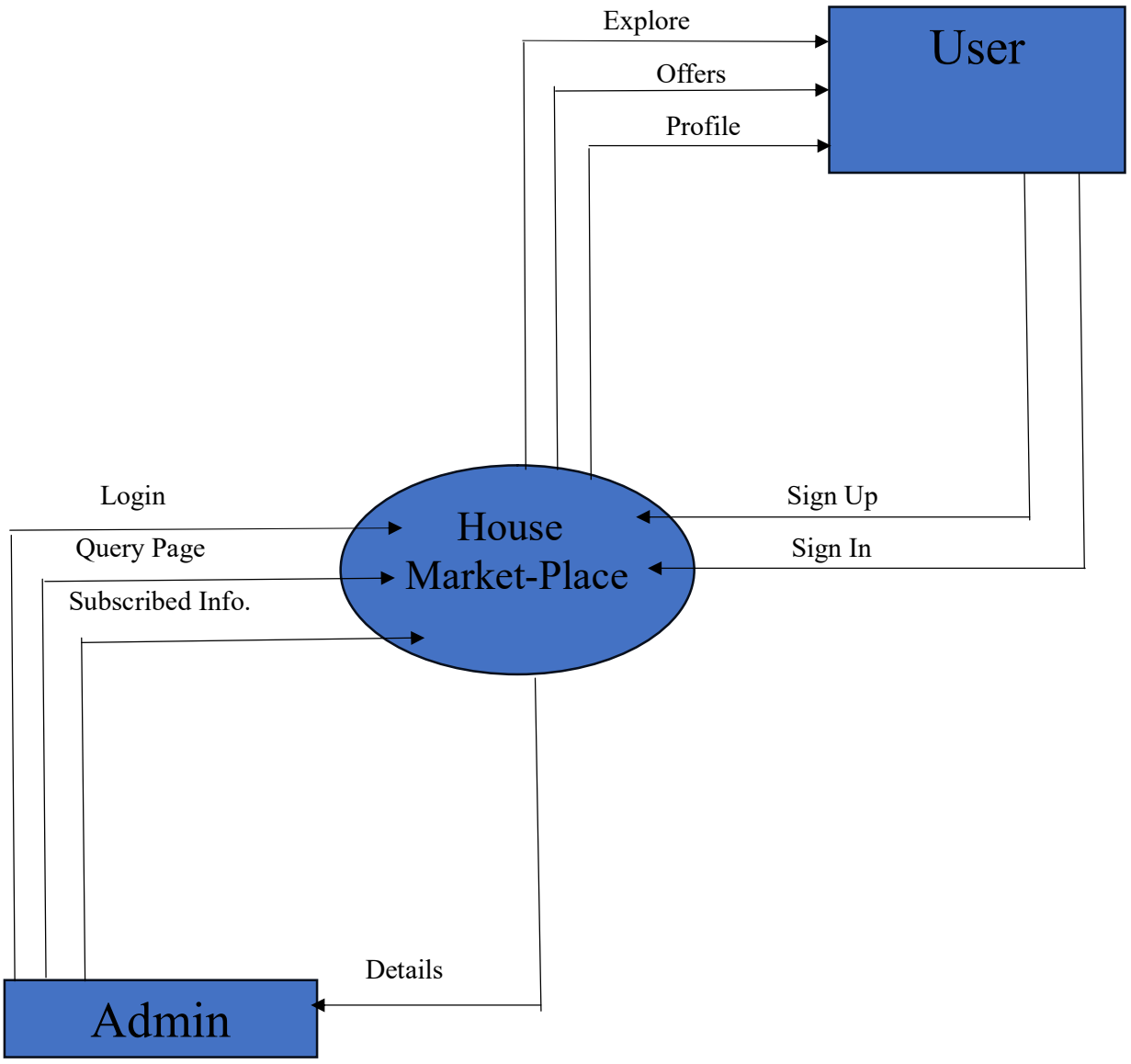
In query module admin will be able to see the users who have contacted for any concern and then act accordingly.

ANALYSIS DIAGRAM

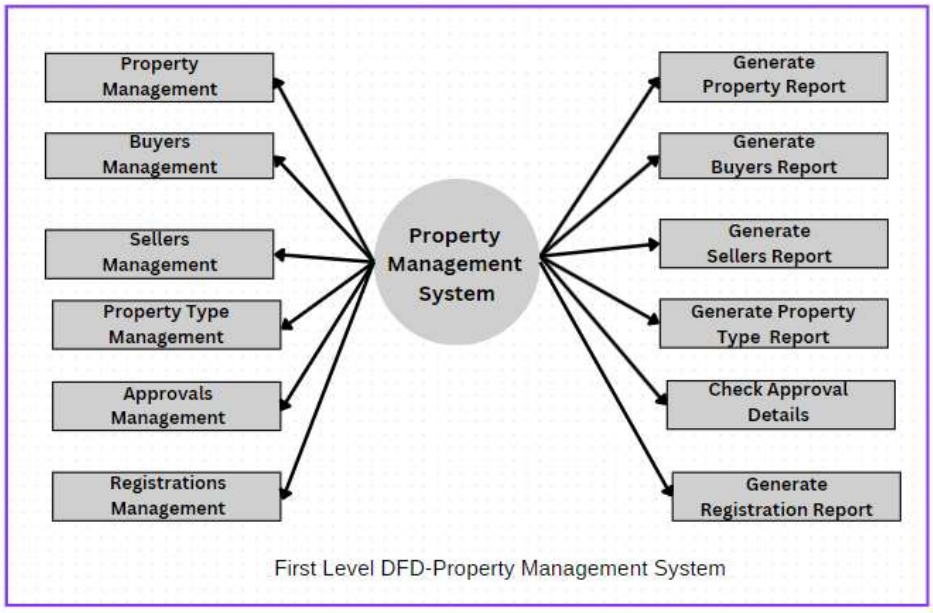
DATA FLOW DIAGRAM:

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM).

0 Level DFD:

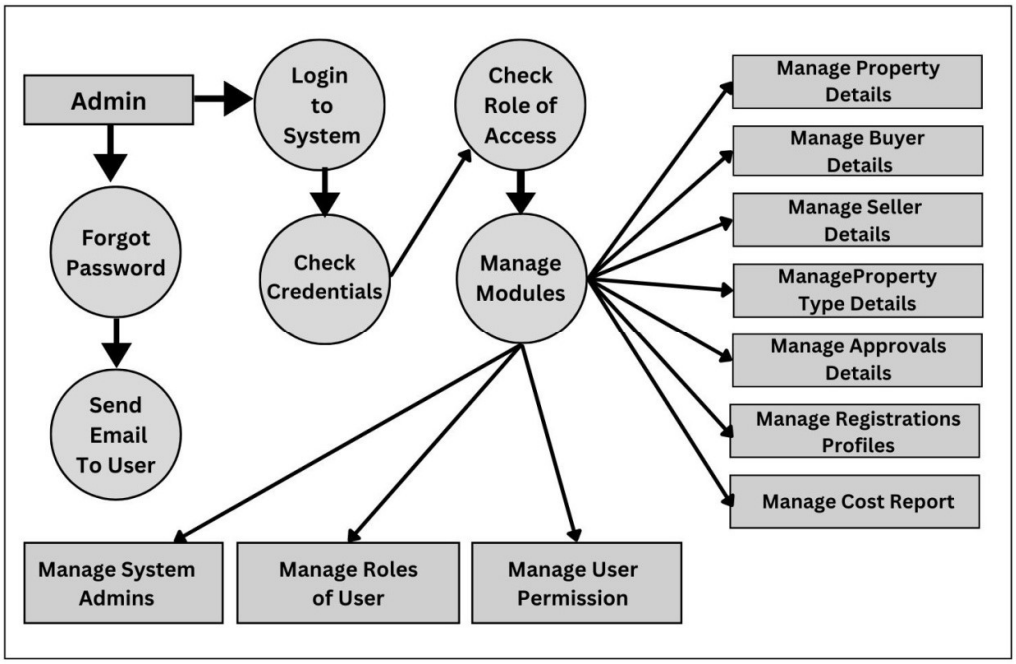


1st Level DFD:



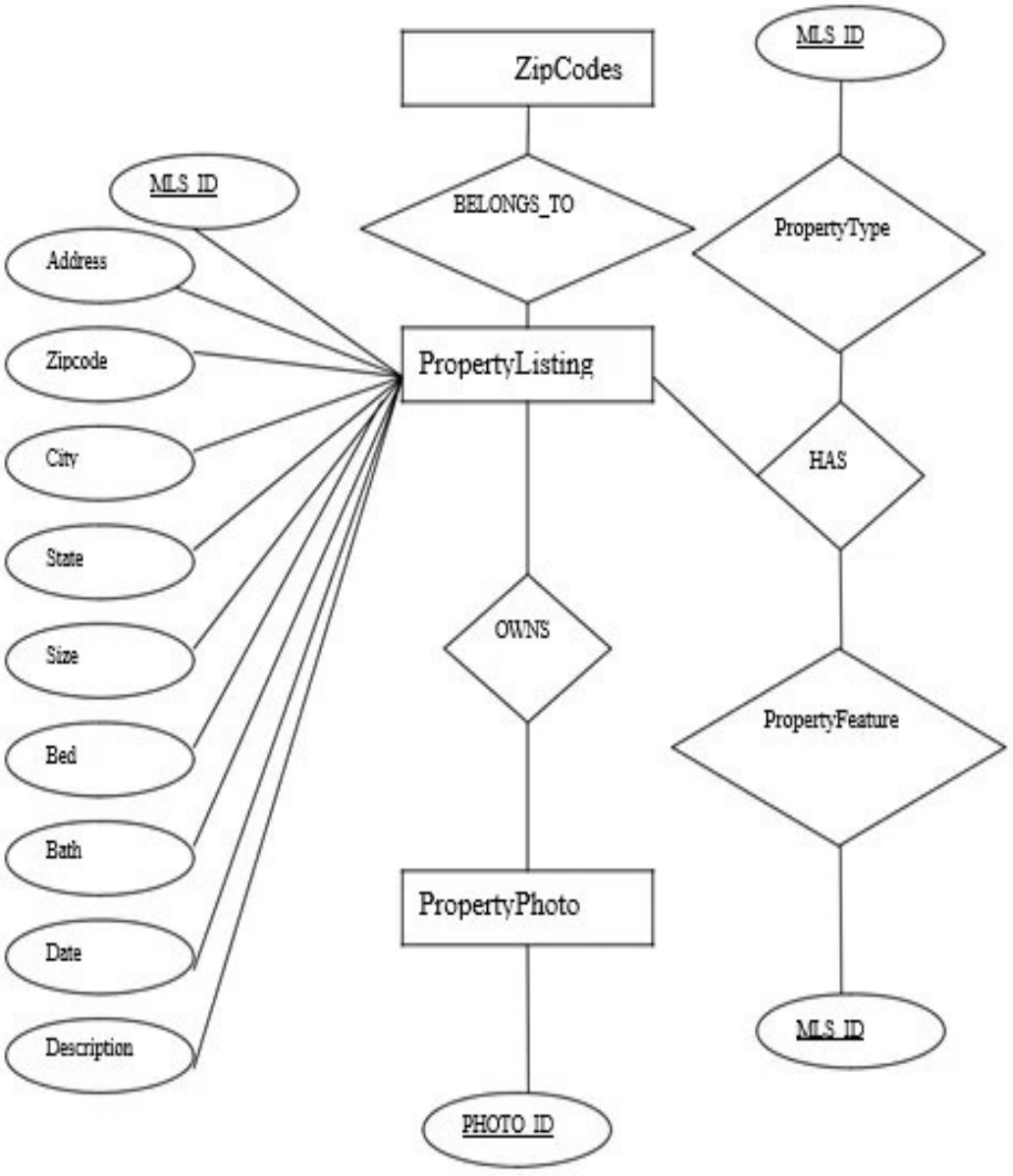
First Level DFD-Property Management System

2ndLevelDFD :



E-R DIAGRAM

An entity relationship (ER) diagram is a flowchart that shows how entities, like people, objects, or concepts, relate to each other within a system. ER diagrams are often used to design or debug relational databases for software engineering, business information systems, education, and research.



SYSTEM SECURITY AND MAINTENANCE

Security Testing of the Project Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. second phase is after the completion of software, this is system testing which verifies that the whole set of programs hanged together.

WHITE BOX TESTING:

Security Testing of the Project Black Box Testing: Testing is vital for the success of any software. No system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. Second phase is after the completion of software. This is system testing which verifies that the whole set of programs hanged together.

BLACK BOX TESTING:

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. Black box testing tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

ALPHA TESTING:

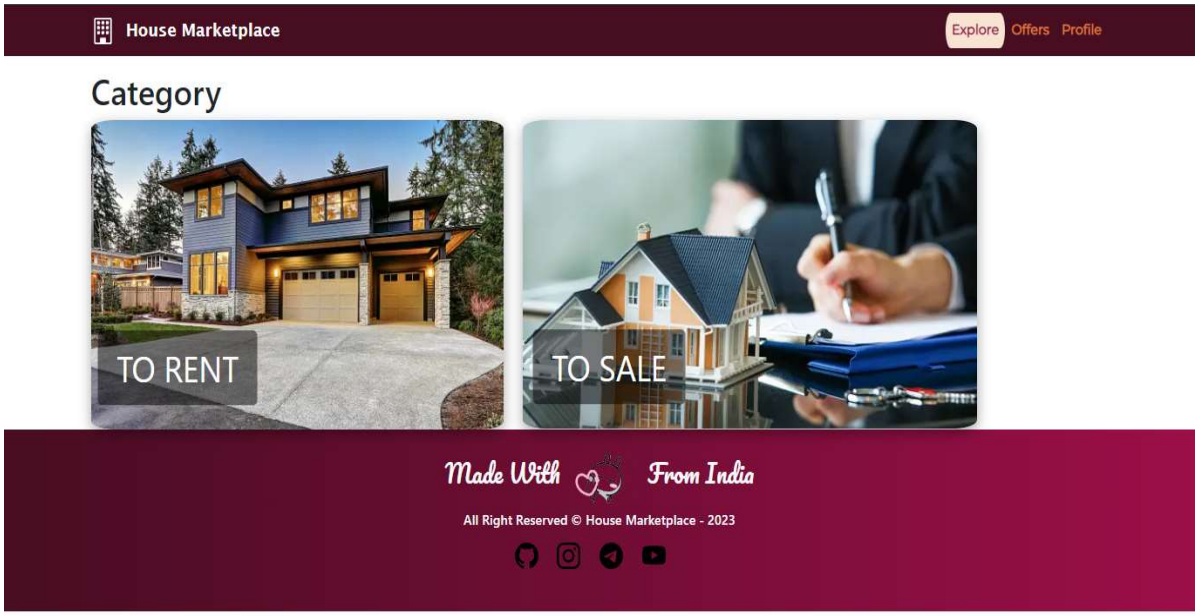
Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

BETA TESTING:

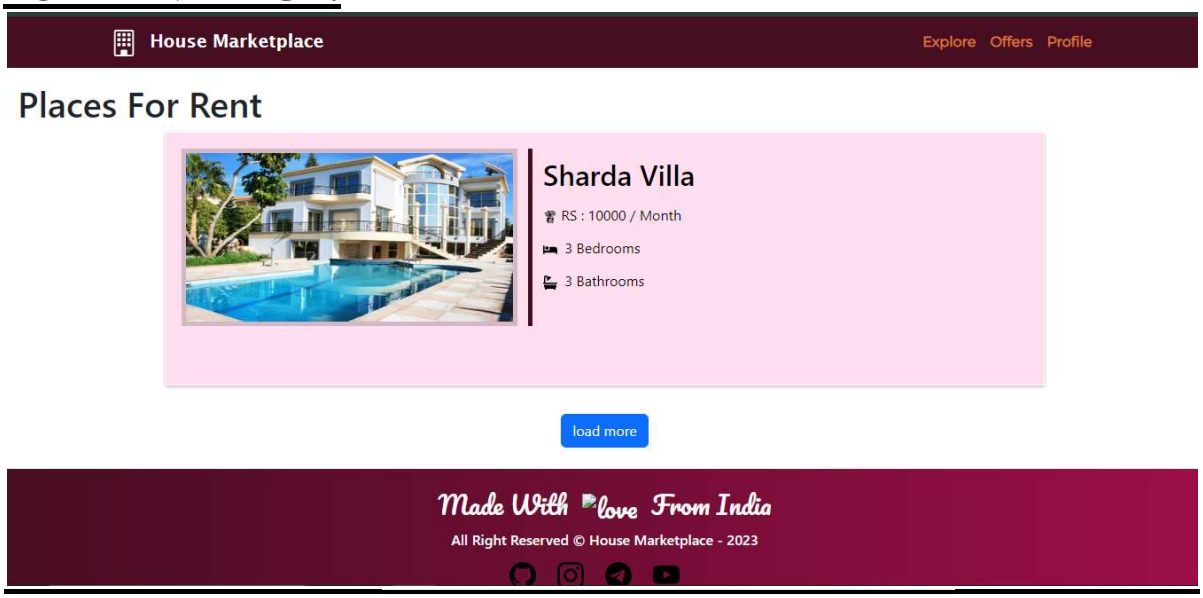
On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers.

SCREENSHOT OF HOUSE MARKETPLACE WEBSITE

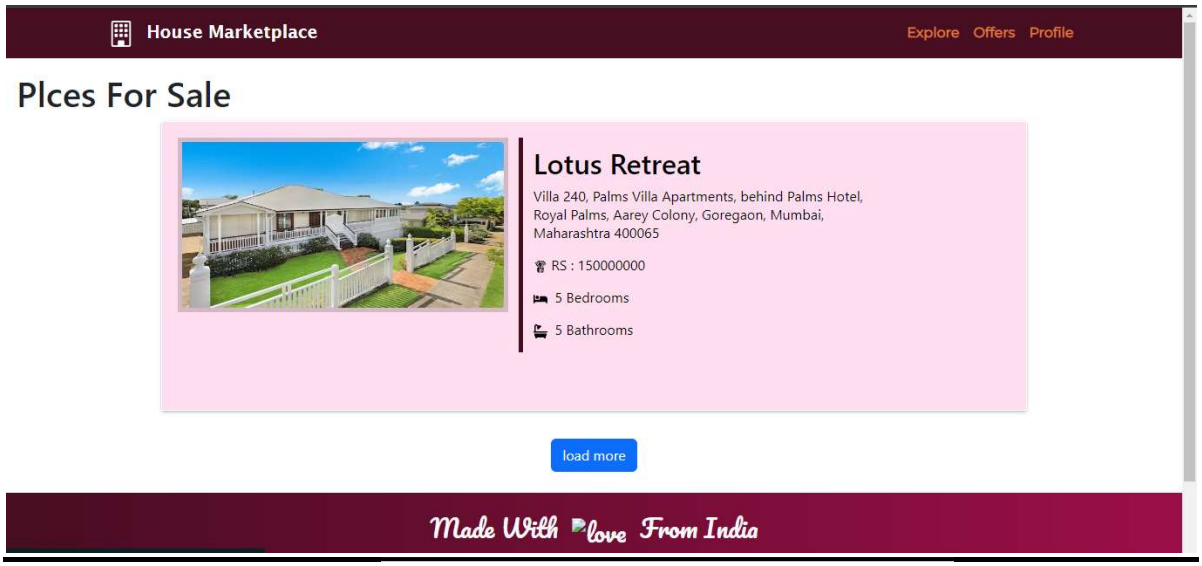
EXPLORE PAGE:



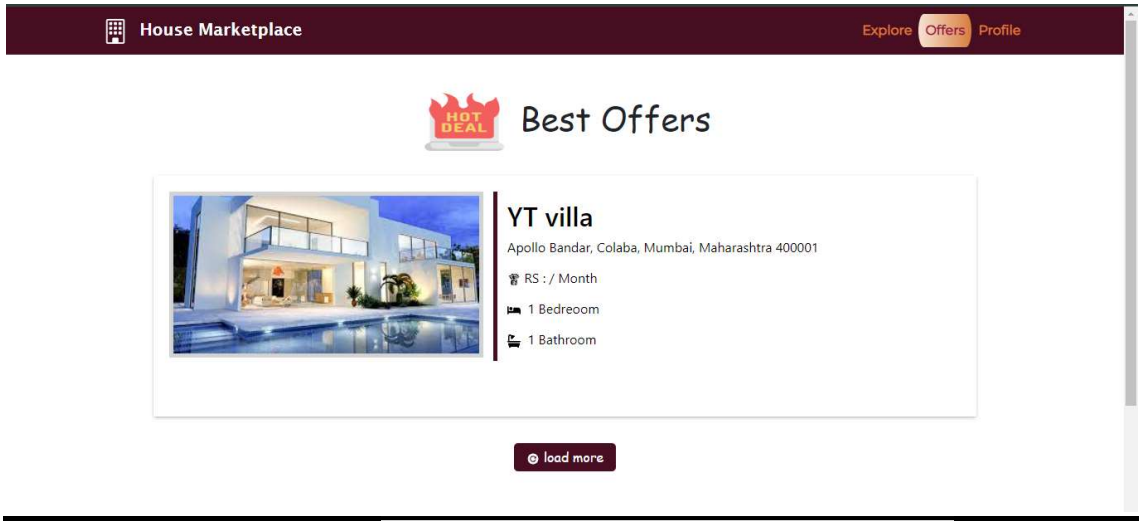
FOR RENT PAGE:



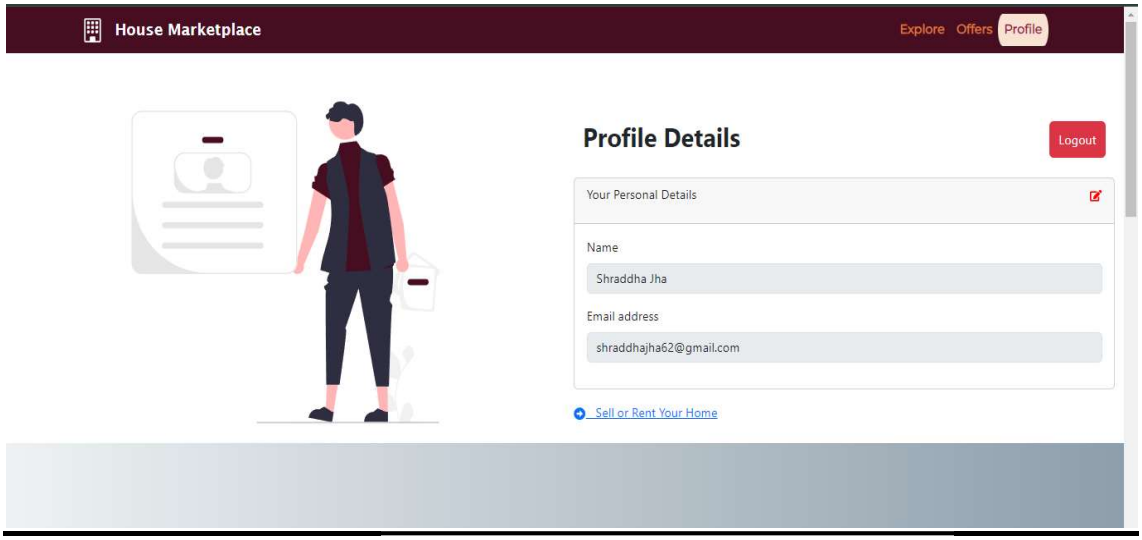
FOR SALE PAGE:



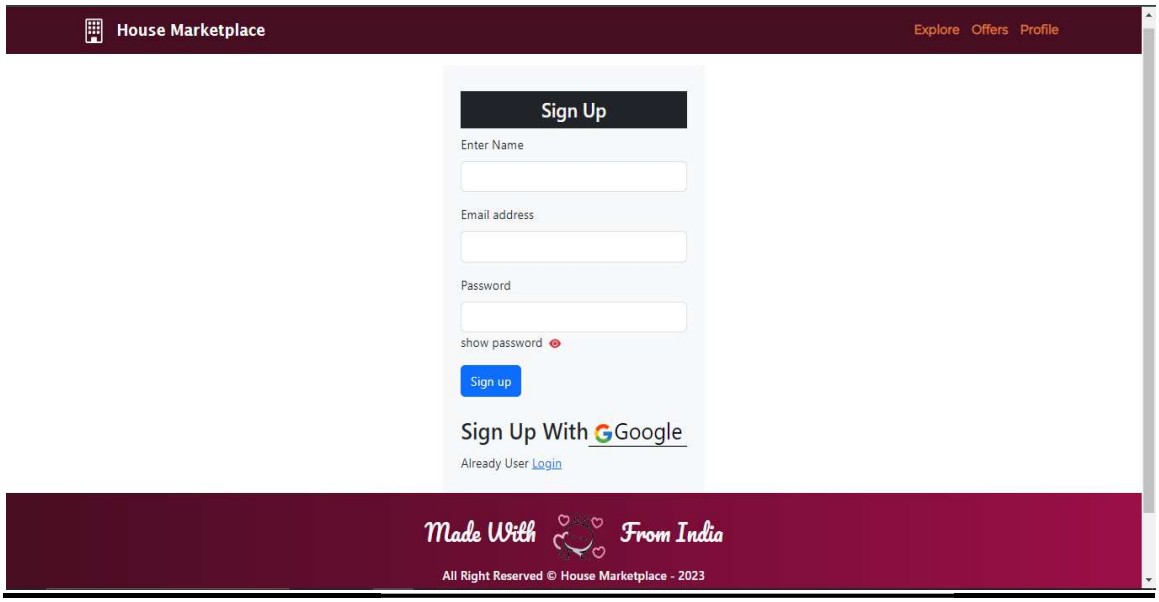
OFFER PAGE:



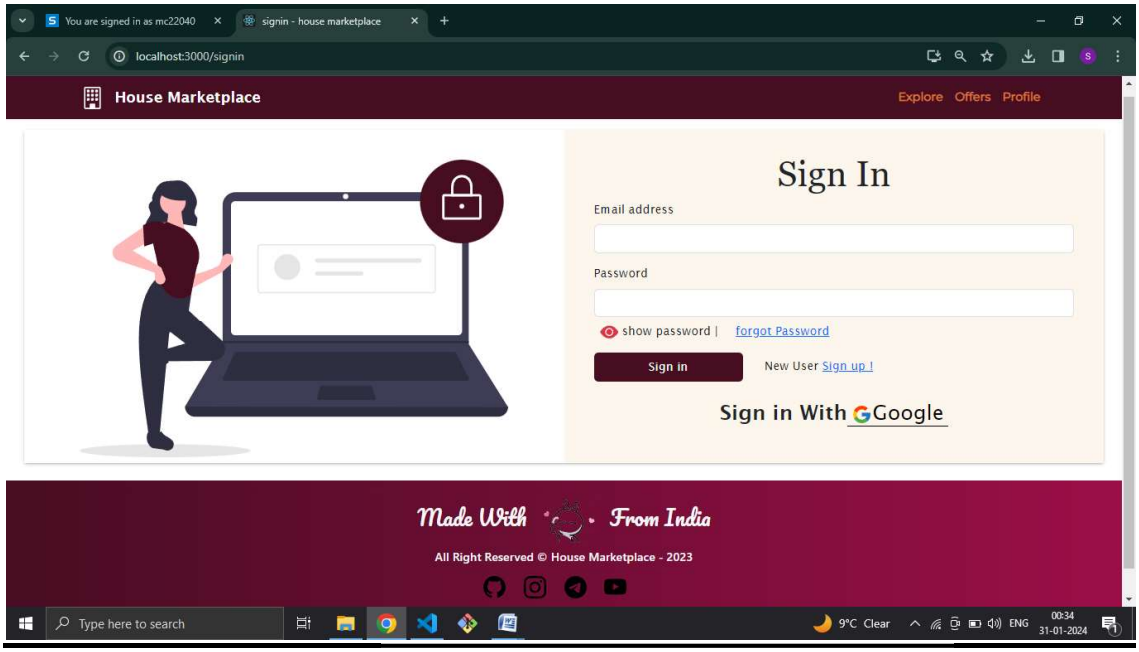
PROFILE PAGE:



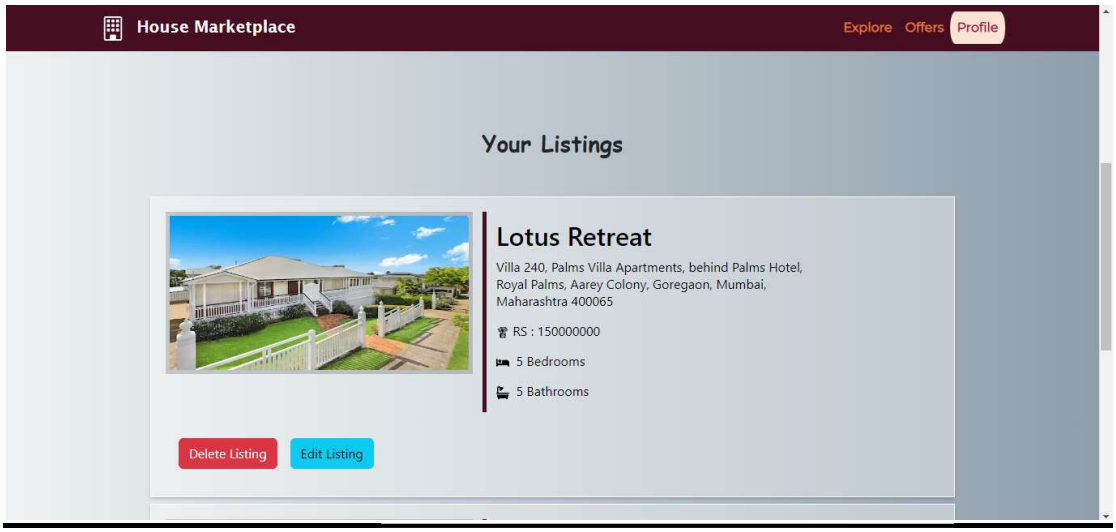
SIGN UP PAGE:



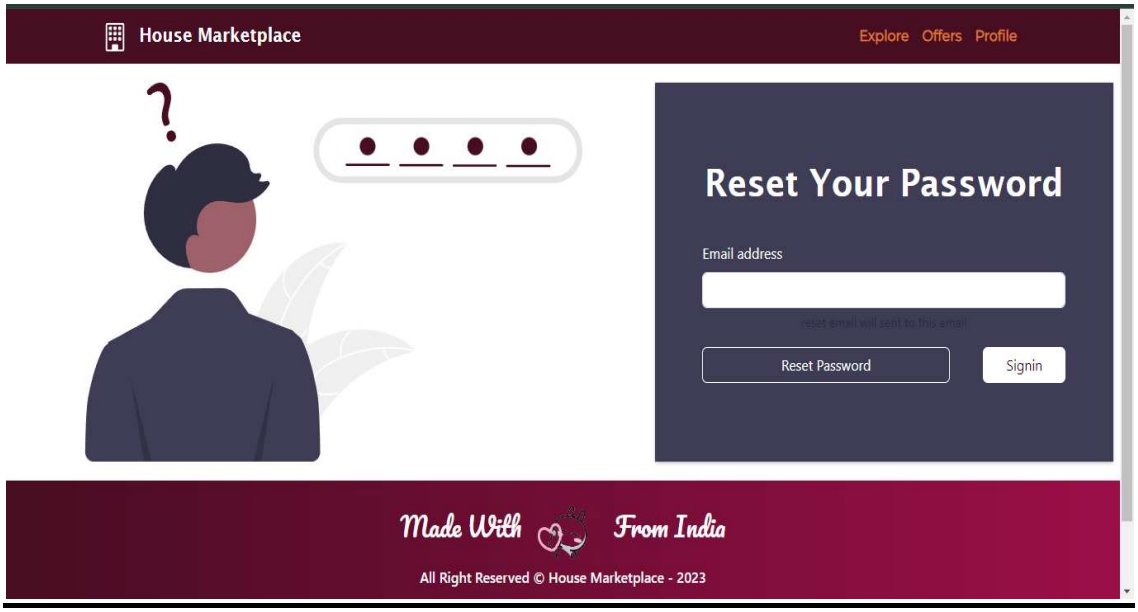
SIGN IN PAGE:



LISTING PAGE:



PASSWORD RESET PAGE:



IMPLEMENTATION

CLIENT-SIDE CODE(ROUTE PAGES):-

HOMEPAGE ROUTE-

```
import React from "react";
import {useNavigate}from "react-router-dom";
import Layout from "../components/Layout/Layout";


const HomePage = () => {
  const navigate=useNavigate();
  const img1="https://media.istockphoto.com/id/876864896/photo/luxurious-new-construction-home-in-bellevue-wa.webp?b=1&s=170667a&w=0&k=20&c=HHYUMi4AIyjPq-JZAmzz_HY37rvQV1hWfNEMnzCmYr0=";
  const img2="https://media.istockphoto.com/id/899471458/photo/purchase-agreement-for-new-house.webp?b=1&s=170667a&w=0&k=20&c=YuYOh4uAK_BcOloalnWIqzRRvJC1H9CGzutgWeTmrWg=";
return(
<Layout>
<div className="container mt-4">
<div className="row">
<h1>Category</h1>
<div className="col-md-5">
<div className="Imagecontainer">
<imgsrc={img1} alt="Rent" style={{width: "100%"}} />
<button className="btn"
onClick={()=>navigate("/category/rent")}>
   TO RENT</button>
</div>
</div>
```

```
<div className="col-md-5">
<div className="Imagecontainer">
<imgsrc={img2} alt="Rent" style={{width: "110%"}} />
<button className="btn"
onClick={()=>navigate("/category/sale")}>
   TO SALE</button>
</div>
</div>
</div>
</div>
</Layout>
 );
};
export default HomePage;
```

PROFILE ROUTE-

```
import React, { useState, useEffect } from "react";
import { toast } from "react-toastify";
import { useNavigate, Link } from "react-router-dom";
import Layout from "../components/Layout/Layout";
import { getAuth, updateProfile } from "firebase/auth";
import { db } from "../firebase.config";
import { FaEdit, FaArrowAltCircleRight } from "react-icons/fa";
import { MdDoneOutline } from "react-icons/md";
import {
 doc,
updateDoc,
 collection,
getDocs,
 query,
where,
```

```
orderBy,
deleteDoc,
} from "firebase/firestore";
import ListingItem from "../components/ListingItem";
import "../styles/profile.css";

const Profile = () => {
  const auth = getAuth();
  const navigate = useNavigate();
  // eslint-disable-next-line
  const [loading, setLoading] = useState(true);
  const [listings, setListings] = useState(null);

  //useeffect for getting data
useEffect(() => {
    const fetchUserListings = async () => {
      const listingRef = collection(db, "listings");
      const q = query(
listingRef,
where("useRef", "==", auth.currentUser.uid),
orderBy("timestamp", "desc")
      );
      const querySnap = await getDocs(q);
      console.log(querySnap);
      let listings = [];
querySnap.forEach((doc) => {
        return listings.push({
          id: doc.id,
          data: doc.data(),
        });
```

```
      });
      console.log(listings);
setListings(listings);
setLoading(false);
    };
fetchUserListings();
  }, [auth.currentUser.uid]);
  const [changeDetails, setChangeDetails] = useState(false);
  const [formData, setFormData] = useState({
    name: auth.currentUser.displayName,
    email: auth.currentUser.email,
  });
  const { name, email } = formData;

  const logoutHandler = () => {
auth.signOut();
toast.success("Successfully Logout");
    navigate("/signin");
  };
//onChange
  const onChange = (e) => {
setFormData((prevState) => ({
      ...prevState,
      [e.target.id]: e.target.value,
    }));
  };
  //submit handler
  const onSubmit = async () => {
    try {
      if (auth.currentUser.displayName !== name) {
```

```
        await updateProfile(auth.currentUser, {
displayName: name,
        });
        const userRef = doc(db, "users", auth.currentUser.uid);
        await updateDoc(userRef, { name });
toast.success("User Updated!");
      }
    } catch (error) {
      console.log(error);
toast("Something Went Wrong");
    }
  };

  //delete handler
  const onDelete = async (listingId) => {
    if (window.confirm("Are You Sure  want to delete ?")) {
      // await deleteDoc(doc, (db, "listings", listingId));
      await deleteDoc(doc(db, "listings", listingId));
      const updatedListings = listings.filter(
        (listing) =>listing.id !== listingId
      );
setListings(updatedListings);
toast.success("Listing Deleted Successfully");
    }
  }

  //edit handler
  const onEdit = (listingId) => {
    navigate(`/editlisting/${listingId}`);
  };
```

```
 return (
<Layout>
<div className="row profile-container">
<div className="col-md-6 profile-container-col1">
<imgsrc="./assets/profile.svg" alt="profile" />
</div>
<div className="col-md-6 profile-container-col2">
<div className="container mt-4  d-flex justify-content-between">
<h2>Profile Details</h2>
<button classNamebtnbtn-danger" onClick={logoutHandler}>
            Logout
</button>
</div>
<div className="   mt-4 card">
<div className="card-header">
<div className="d-flex justify-content-between ">
<p>Your Personal Details </p>
<span
                style={{ cursor: "pointer" }}
onClick={() => {
changeDetails&&onSubmit();
setChangeDetails((prevState) => !prevState);
                }}
>
                {changeDetails ? (
<MdDoneOutline color="green" />
                ) : (
<FaEdit color="red" />
                )}
</span>
```

```
</div>
</div>
<div className="card-body">
<form>
<div className="mb-3">
<label htmlFor="exampleInputPassword1" className="form-label">
                Name
</label>
<input
                type="text"
className="form-control"
                id="name"
                value={name}
onChange={onChange}
                disabled={!changeDetails}
               />
</div>
<div className="mb-3">
<label htmlFor="exampleInputEmail1" className="form-label">
                Email address
</label>
<input
                type="email"
                value={email}
className="form-control"
                id="email"
                aria-describedby="emailHelp"
onChange={onChange}
                disabled={!changeDetails}
               />
```

```
</div>
</form>
</div>
</div>
<div className="mt-3 create-listing">
<Link to="/create-listing">
<FaArrowAltCircleRight color="primary" />  Sell or Rent Your
        Home
</Link>
</div>
</div>
</div>

<div className="container-fluid mt-4 your-listings">
    {listings &&listings?.length> 0 && (
<>
<h3 className="mt-4">Your Listings</h3>
<div>
        {listings.map((listing) => (
<ListingItem
className="profile-listing"
          key={listing.id}
          listing={listing.data}
          id={listing.id}
onDelete={() =>onDelete(listing.id)}
onEdit={() =>onEdit(listing.id)}
          />
        ))}
</div>
</>
```

```
        )}
      </div>
      </Layout>
       );
      };
      export default Profile;
```

SIGNUP_ROUTE-

```
import React, { useState } from "react";
import { Link, useNavigate } from "react-router-dom";
import { toast } from 'react-toastify';
import Layout from "./../components/Layout/Layout";
import { BsFillEyeFill } from "react-icons/bs";
import OAuth from "../components/OAuth";
import {
getAuth,
createUserWithEmailAndPassword,
updateProfile,
} from "firebase/auth";
import { db } from "../firebase.config";
import { doc, setDoc, serverTimestamp } from "firebase/firestore";

const Signup = () => {
  const [showPassword, setShowPassword] = useState(false);
  const [formData, setFormData] = useState({
    email: "",
    name: "",
    password: "",
  });
  const { name, email, password } = formData;
  const navigate = useNavigate();

  const onChange = (e) => {
setFormData((prevState) => ({
      ...prevState,
      [e.target.id]: e.target.value,
    }));
  };

  const onSubmitHndler = async (e) => {
e.preventDefault();
    try {
      const auth = getAuth();
      const userCredential = await createUserWithEmailAndPassword(
```

```
        auth,
        email,
        password
      );
      const user = userCredential.user;
updateProfile(auth.currentUser, { displayName: name });
      const formDataCopy = { ...formData };
      delete formDataCopy.password;
formDataCopy.timestamp = serverTimestamp();
      await setDoc(doc(db, "users", user.uid), formDataCopy);
toast.success("Signup Successfully !");
navigate("/");

    } catch (error) {
      console.log(error);
toast.error("Something Went Wrong");
    }
  };
  return (
<Layout>
<div className="d-flex  align-items-center justify-content-center w-100 mt-4">
<form className="bg-light p-4" onSubmit={onSubmitHndler}>
<h4 className="bg-dark p-2 mt-2 text-light text-center">Sign Up </h4>
<div className="mb-3">
<label htmlFor="exampleInputEmail1" className="form-label">
            Enter Name
</label>
<input
            type="text"
            value={name}
className="form-control"
            id="name"
onChange={onChange}
            aria-describedby="nameHelp"
           />
</div>
<div className="mb-3">
<label htmlFor="exampleInputEmail1" className="form-label">
            Email address
</label>
<input
            type="email"
            value={email}
onChange={onChange}
className="form-control"
            id="email"
```

```
            aria-describedby="emailHelp"
          />
</div>
<div className="mb-3">
<label htmlFor="exampleInputPassword1" className="form-label">
            Password
</label>
<input
            type={showPassword ? "text" : "password"}
            value={password}
onChange={onChange}
className="form-control"
            id="password"
          />
<span>
            show password
<BsFillEyeFill
className="text-danger ms-2"
              style={{ cursor: "pointer" }}
onClick={() => {
setShowPassword((prevState) => !prevState);
              }}
            />
</span>
</div>
<button type="submit" className="btnbtn-primary">
          Sign up
</button>
<div>
<OAuth/>
<span>Already User</span><Link to="/signin">Login</Link>
</div>
</form>
</div>
</Layout>
  );
};

export default Signup;
```

SIGNIN ROUTE-

```
import React, { useState } from "react";
import { getAuth, signInWithEmailAndPassword } from "firebase/auth";
import { Link, useNavigate } from "react-router-dom";
import { toast } from "react-toastify";
import { BsFillEyeFill } from "react-icons/bs";
import Layout from "./../components/Layout/Layout";
import OAuth from "../components/OAuth";
import "../styles/signin.css";

const Signin = () => {
  const [showPassword, setShowPassword] = useState(false);
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  });
  const { email, password } = formData;
  const navigate = useNavigate();

  const onChange = (e) => {
setFormData((prevState) => ({
      ...prevState,
      [e.target.id]: e.target.value,
    }));
  };

  //loginHandler
  const loginHandler = async (e) => {
e.preventDefault();
    try {
      const auth = getAuth();
      const userCredential = await signInWithEmailAndPassword(
        auth,
        email,
        password
      );
      if (userCredential.user) {
toast.success("Login Success");
navigate("/");
      }
    } catch (error) {
      console.log(error);
toast.error("Invalid Email Or Password");
    }
  };
  return (
```

```
<Layout title="signin - house marketplace">
<div className="row m-4 signin-container ">
<div className="col-md-6">
<imgsrc="./assets/loginpage.svg" alt="login" />
</div>
<div className="col-md-6 signin-container-col2">
<form onSubmit={loginHandler}>
<h4 className=" text-center">Sign In</h4>
<div className="mb-3">
<label htmlFor="exampleInputEmail1" className="form-label">
        Email address
</label>
<input
        type="email"
        value={email}
onChange={onChange}
className="form-control"
        id="email"
        aria-describedby="emailHelp"
       />
</div>
<div className="mb-2 ">
<label htmlFor="exampleInputPassword1" className="form-label">
        Password
</label>
<input
        type={showPassword ? "text" : "password"}
        value={password}
onChange={onChange}
className="form-control"
        id="password"
       />
</div>
<div className="mb-3 show-pass-forgot">
<span>
<BsFillEyeFill
className="text-danger ms-2 "
         size={25}
         style={{ cursor: "pointer" }}
onClick={() => {
setShowPassword((prevState) => !prevState);
         }}
        />{" "}
        show password
</span>{" "}
       |
```

```
<Link to="/forgot-password" className="ms-4">
            forgot Password
</Link>
</div>
<button type="submit" className="btnsigninbutton">
          Sign in
</button>
<span className="ms-4 new-user"> New User</span>{" "}
<Link to="/signup">Sign up !</Link>
<OAuth />
</form>
</div>
</div>
</Layout>
  );
};

export default Signin;
```

OFFERS ROUTE-

```
import React, { useEffect, useState } from "react";
import Layout from "./../components/Layout/Layout";
import { IoReloadCircle } from "react-icons/io5";
import "../styles/offers.css";
import { db } from "./../firebase.config";
import { toast } from "react-toastify";
import {
  collection,
getDocs,
  query,
where,
orderBy,
  limit,
startAfter,
} from "firebase/firestore";
import Spinner from "../components/Spinner";
import ListingItem from "../components/ListingItem";

const Offers = () => {
  const [listing, setListing] = useState("");
  const [loading, setLoading] = useState(true);
  const [lastFetchListing, setLastFetchListing] = useState(null);

  //fetch listing
useEffect(() => {
   const fetchListing = async () => {
```

```
      try {
        //refrence
        const listingsRef = collection(db, "listings");
        //query
        const q = query(
listingsRef,
where("offer", "==", true),
orderBy("timestamp", "desc"),
limit(10)
        );
        //execute query
        const querySnap = await getDocs(q);
        const lastVisible = querySnap.docs[querySnap.docs.length - 1];
setLastFetchListing(lastVisible);
        const listings = [];
querySnap.forEach((doc) => {
          return listings.push({
            id: doc.id,
            data: doc.data(),
          });
        });
setListing(listings);
setLoading(false);
      } catch (error) {
        console.log(error);
toast.error("Unble to fetch data");
      }
    };
    //func call
fetchListing();
  }, []);

  //loadmore pagination func
  const fetchLoadMoreListing = async () => {
    try {
      //refrence
      const listingsRef = collection(db, "listings");
      //query
      const q = query(
listingsRef,
where("offer", "==", true),
orderBy("timestamp", "desc"),
startAfter(lastFetchListing),
limit(10)
      );
      //execute query
```

```
      const querySnap = await getDocs(q);
      const lastVisible = querySnap.docs[querySnap.docs.length - 1];
setLastFetchListing(lastVisible);
      const listings = [];
querySnap.forEach((doc) => {
        return listings.push({
          id: doc.id,
          data: doc.data(),
        });
      });
setListing((prevState) => [...prevState, ...listings]);
setLoading(false);
    } catch (error) {
      console.log(error);
toast.error("Unble to fetch data");
    }
  };
  return (
<Layout title="best offer on house">
<div className="offers pt-3 container-fluid">
<h1>
          {" "}
<img
src="/assets/offer.png"
            alt="offers"
className="offer-img"
          />{" "}
          Best Offers
</h1>
        {loading ? (
<Spinner />
) : listing &&listing.length> 0 ? (
<>
<div>
              {listing.map((list) => (
<ListingItem listing={list.data} id={list.id} key={list.id} />
            ))}
</div>
</>
        ) : (
<p>There Are No Current Offers </p>
        )}
<div className="d-flex align-items-center justify-content-center pb-4 mt-4">
          {lastFetchListing&& (
<button className="load-btn" onClick={fetchLoadMoreListing}>
<IoReloadCircle /> load more
```

```
</button>
        )}
</div>
</div>
</Layout>
  );
};

export default Offers;
```

CONTACT ROUTE-

```
import React, { useState, useEffect } from "react";
import Layout from "../components/Layout/Layout";
import { doc, getDoc } from "firebase/firestore";
import { db } from "../firebase.config";
import { useParams, useSearchParams } from "react-router-dom";
import { toast } from "react-toastify";
import "../styles/contact.css";

const Contact = () => {
  const [message, setMessage] = useState("");
  const [landlord, setLandlord] = useState("");
  const [searchParams, setSearchParams] = useSearchParams(); //eslint-disable-line
  const params = useParams();

useEffect(() => {
    const getLandlord = async () => {
      const docRef = doc(db, "users", params.landlordId);
      const docSnap = await getDoc(docRef);
      if (docSnap.exists()) {
setLandlord(docSnap.data());
      } else {
toast.error("Unble to ftech data");
      }
    };
getLandlord();
  }, [params.landlordId]);
  return (
<Layout title="contact details - house marketplace">
<div className="row contact-container">
<div className="col-md-6 contact-container-col-1">
<imgsrc="/assets/contact.svg" alt="contact" />
</div>
<div className="col-md-6 contact-container-col-2">
<h1>Contact Details</h1>
```

```
<div>
        {landlord !== "" && (
<main>
<h3 className="mb-4">
              Person Name :{" "}
<span style={{ color: "#470d21" }}>
               {" "}
               " {landlord?.name} "{" "}
</span>
</h3>

<div className="form-floating">
<textarea
className="form-control"
               placeholder="Leave a comment here"
               value={message}
               id="message"
onChange={(e) => {
setMessage(e.target.value);
               }}
              />
<label
htmlFor="floatingTextarea"
               style={{ color: "lightgray" }}
>
               Type your message here
</label>
</div>
<a
href={`mailto:${landlord.email}?Subject=${searchParams.get(
               "listingName"
)}&body=${message}`}
>
<button className="btn mt-2">Send Message</button>
</a>
</main>
        )}
</div>
</div>
</div>
</Layout>
  );
};

export default Contact;
```

CATEGORY ROUTE:

```
import React, { useEffect, useState } from "react";

import Layout from "./../components/Layout/Layout";
import { useParams } from "react-router-dom";
import { db } from "./../firebase.config";
import { toast } from "react-toastify";
import {
  collection,
getDocs,
  query,
where,
orderBy,
  limit,
startAfter,
} from "firebase/firestore";
import Spinner from "../components/Spinner";
import ListingItem from "../components/ListingItem";

const Category = () => {
  const [listing, setListing] = useState("");
  const [lastFetchListing, setLastFetchListing] = useState(null);
  const [loading, setLoading] = useState(true);
  const params = useParams();

  //fetch listing
useEffect(() => {
   const fetchListing = async () => {
     try {
       //refrence
       const listingsRef = collection(db, "listings");
       //query
       const q = query(
listingsRef,
where("type", "==", params.categoryName),
orderBy("timestamp", "desc"),
limit(1)
       );
       //execute query
       const querySnap = await getDocs(q);
       const lastVisible = querySnap.docs[querySnap.docs.length - 1];
setLastFetchListing(lastVisible);
       const listings = [];
querySnap.forEach((doc) => {
         return listings.push({
           id: doc.id,
```

```
            data: doc.data(),
          });
        });
setListing(listings);
setLoading(false);
      } catch (error) {
        console.log(error);
toast.error("Unble to fetch data");
      }
    };
    //func call
fetchListing();
  }, [params.categoryName]);

  //loadmore pagination func
  const fetchLoadMoreListing = async () => {
    try {
      //refrence
      const listingsRef = collection(db, "listings");
      //query
      const q = query(
listingsRef,
where("type", "==", params.categoryName),
orderBy("timestamp", "desc"),
startAfter(lastFetchListing),
limit(10)
      );
      //execute query
      const querySnap = await getDocs(q);
      const lastVisible = querySnap.docs[querySnap.docs.length - 1];
setLastFetchListing(lastVisible);
      const listings = [];
querySnap.forEach((doc) => {
        return listings.push({
          id: doc.id,
          data: doc.data(),
        });
      });
setListing((prevState) => [...prevState, ...listings]);
setLoading(false);
    } catch (error) {
      console.log(error);
toast.error("Unble to fetch data");
    }
  };
```

```
  return (
<Layout
    title={
params.categoryName === "rent" ? "Places For Rent" : "Plces For Sale"
    }
>
<div className="mt-3 container-fluid">
<h1>
        {params.categoryName === "rent"
         ? "Places For Rent"
         : "Places For Sale"}
</h1>
      {loading ? (
<Spinner />
) : listing &&listing.length> 0 ? (
<>
<div>
           {listing.map((list) => (
<ListingItem listing={list.data} id={list.id} key={list.id} />
          ))}
</div>
</>
      ) : (
<p>No Listing For {params.categoryName} </p>
      )}
</div>
<div className="d-flex align-items-center justify-content-center mb-4 mt-4">
      {lastFetchListing&& (
<button
className="btnbtn-primary text-center"
onClick={fetchLoadMoreListing}
>
         load more
</button>
      )}
</div>
</Layout>
  );
};

export default Category;
```

CREATE LISTING ROUTE:

```
import React, { useState, useEffect, useRef } from "react";
import { v4 as uuidv4 } from "uuid";
import { useNavigate } from "react-router-dom";
```

```
import Layout from "./../components/Layout/Layout";
import { getAuth, onAuthStateChanged } from "firebase/auth";
import Spinner from "../components/Spinner";
import { AiOutlineFileAdd } from "react-icons/ai";
import { toast } from "react-toastify";
import {
getStorage,
  ref,
uploadBytesResumable,
getDownloadURL,
} from "firebase/storage";
import { db } from "../firebase.config";
import { addDoc, collection, serverTimestamp } from "firebase/firestore";

const CreateListing = () => {
  const [loading, setLoading] = useState(false);
  const [geoLoactionEnable, setGeoLocationEnable] = useState(false);
  const [formData, setFormData] = useState({
    type: "rent",
    name: "",
    bedrooms: 1,
    bathrooms: 1,
    parking: false,
    furnished: false,
    address: "",
    offer: false,
regularPrice: 0,
discountedPrice: 0,
    images: {},
    latitude: 0,
    longitude: 0,
  });

  const {
    type,
    name,
    bedrooms,
    bathrooms,
    parking,
    furnished,
    address,
    offer,
regularPrice,
discountedPrice,
    images,
    latitude,
```

```
    longitude,
  } = formData;

  const auth = getAuth();
  const navigate = useNavigate();
  const isMounted = useRef(true);

useEffect(() => {
    if (isMounted) {
onAuthStateChanged(auth, (user) => {
setFormData({
          ...formData,
useRef: user.uid,
        });
      });
    } else {
      navigate("/signin");
    }

    // eslint-disable-next-line
  }, []);

  if (loading) {
    return <Spinner />;
  }

  //mutate func
  const onChangeHandler = (e) => {
    let boolean = null;
    if (e.target.value === "true") {
boolean = true;
    }
    if (e.target.value === "false") {
boolean = false;
    }
    //files
    if (e.target.files) {
setFormData((prevState) => ({
        ...prevState,
        images: e.target.files,
      }));
    }
    //text/booleans/number
    if (!e.target.files) {
setFormData((prevState) => ({
        ...prevState,
```

```
        [e.target.id]: boolean ?? e.target.value,
      }));
    }
  };

  //form submit
  const onSubmit = async (e) => {
e.preventDefault();
    // console.log(formData);
    if (discountedPrice>= regularPrice) {
setLoading(false);
toast.error("Discount Price should be less than Regular Price");
      return;
    }
    if (images > 6) {
setLoading(false);
toast.error("Max 6 Images can be selected");
      return;
    }
    let geoLocation = {};
    let location;
    if (geoLoactionEnable) {
      const response = await fetch(

`https://maps.googleapis.com/maps/api/geocode/json?address=${address}&key=AIza
SyBywlow6ReyPu1sX2KbS38EbTklYcDFMsI`
      );
      const data = await response.json();
      console.log(data);
    } else {
geoLocation.lat = latitude;
geoLocation.lng = longitude;
      // location = address;
    }

    //store images to firebase storage
    const storeImage = async (image) => {
      return new Promise((resolve, reject) => {
        const storage = getStorage();
        const fileName = `${auth.currentUser.uid}-${image.name}-${uuidv4()}`;
        const storageRef = ref(storage, "images/" + fileName);
        const uploadTask = uploadBytesResumable(storageRef, image);
uploadTask.on(
          "state_changed",
          (snapshot) => {
            const progress =
```

```
            (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
console.log("uplloas is" + progress + "% done");
          switch (snapshot.state) {
            case "paused":
console.log("upload is paused");
              break;
            case "running":
console.log("upload is runnning");
          }
        },
        (error) => {
          reject(error);
        },
        //success
        () => {
getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
            resolve(downloadURL);
          });
        }
      );
    });
  };
  const imgUrls = await Promise.all(
    [...images].map((image) =>storeImage(image))
).catch(() => {
setLoading(false);
toast.error("Images not uploaded");
    return;
  });
  console.log(imgUrls);

  //save form data
  const formDataCopy = {
    ...formData,
imgUrls,
geoLocation,
    timestamp: serverTimestamp(),
  };
formData.location = address;
  delete formDataCopy.images;
!formDataCopy.offer&& delete formDataCopy.discountedPrice;
  const docRef = await addDoc(collection(db, "listings"), formDataCopy);
toast.success("Listing Created!");
setLoading(false);
  navigate(`/category/${formDataCopy.type}/${docRef.id}`);
 };
```

```
 return (
<Layout>
<div className="container d-flex flex-column align-items-center justify-content-
center mb-4">
<h3 className="mt-3 w-50 bg-dark text-light p-2 text-center">
        Create Listing  
<AiOutlineFileAdd />
</h3>
      {/* sell rent button */}
<form className="w-50 bg-light p-4" onSubmit={onSubmit}>
<div className="d-flex flex-row mt-4">
<div className="form-check">
<input
className="form-check-input"
            type="radio"
            value="rent"
onChange={onChangeHandler}
defaultChecked
            name="type"
            id="type"
           />
<label className="form-check-label" htmlFor="rent">
            Rent
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
            type="radio"
            name="type"
            value="sale"
onChange={onChangeHandler}
            id="type"
           />
<label className="form-check-label" htmlFor="sale">
            Sale
</label>
</div>
</div>
       {/* name */}
<div className="mb-3 mt-4">
<label htmlFor="name" className="form-label">
           Name
</label>
<input
           type="text"
```

```
className="form-control"
            id="name"
            value={name}
onChange={onChangeHandler}
            required
           />
</div>
         {/* bedrooms */}
<div className="mb-3 mt-4">
<label htmlFor="bedrooms" className="form-label">
            Bedrooms
</label>
<input
            type="number"
className="form-control"
            id="bedrooms"
            value={bedrooms}
onChange={onChangeHandler}
            required
           />
</div>
         {/* bathrroms */}
<div className="mb-3 mt-4">
<label htmlFor="bathrooms" className="form-label">
            Bathrooms
</label>
<input
            type="number"
className="form-control"
            id="bathrooms"
            value={bathrooms}
onChange={onChangeHandler}
            required
           />
</div>
         {/* parking */}
<div className="mb-3 ">
<label htmlFor="parking" className="form-label">
Parking :
</label>
<div className="d-flex flex-row ">
<div className="form-check">
<input
className="form-check-input"
                type="radio"
                value={true}
```

```
onChange={onChangeHandler}
                name="parking"
                id="parking"
              />
<label className="form-check-label" htmlFor="yes">
                Yes
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
                type="radio"
                name="parking"
                value={false}
defaultChecked
onChange={onChangeHandler}
                id="parking"
              />
<label className="form-check-label" htmlFor="no">
                No
</label>
</div>
</div>
</div>
        {/* furnished */}
<div className="mb-3 ">
<label htmlFor="furnished" className="form-label">
Furnished :
</label>
<div className="d-flex flex-row ">
<div className="form-check">
<input
className="form-check-input"
                type="radio"
                value={true}
onChange={onChangeHandler}
                name="furnished"
                id="furnished"
              />
<label className="form-check-label" htmlFor="yes">
                Yes
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
```

```
                type="radio"
                name="furnished"
                value={false}
defaultChecked
onChange={onChangeHandler}
                id="furnished"
               />
<label className="form-check-label" htmlFor="no">
                No
</label>
</div>
</div>
</div>
        {/* address */}
<div className="mb-3">
<label htmlFor="address">Address :</label>
<textarea
className="form-control"
            placeholder="Enter Your Address"
            id="address"
            value={address}
onChange={onChangeHandler}
            required
           />
</div>
        {/* geoLoaction */}
{!geoLoactionEnable&& (
<div className="mb-3 ">
<div className="d-flex flex-row ">
<div className="form-check">
<label className="form-check-label" htmlFor="yes">
                Latitude
</label>
<input
className="form-control"
                type="number"
                value={latitude}
onChange={onChangeHandler}
                name="latitude"
                id="latitude"
               />
</div>
<div className="form-check ms-3">
<label className="form-check-label" htmlFor="no">
                Longitude
</label>
```

```
<input
className="form-control"
                type="number"
                name="longitude"
                value={longitude}
onChange={onChangeHandler}
                id="longitude"
              />
</div>
</div>
</div>
        )}
        {/* offers  */}
<div className="mb-3 ">
<label htmlFor="offer" className="form-label">
Offer :
</label>
<div className="d-flex flex-row ">
<div className="form-check">
<input
className="form-check-input"
              type="radio"
              value={true}
onChange={onChangeHandler}
              name="offer"
              id="offer"
             />
<label className="form-check-label" htmlFor="yes">
              Yes
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
              type="radio"
              name="offer"
              value={false}
defaultChecked
onChange={onChangeHandler}
              id="offer"
             />
<label className="form-check-label" htmlFor="no">
              No
</label>
</div>
</div>
```

```
</div>
        {/* regular price */}
<div className="mb-3 mt-4">
<label htmlFor="name" className="form-label">
            Regular Price :
</label>
<div className=" d-flex flex-row ">
<input
              type="number"
className="form-control w-50 "
              id="regularPrice"
              name="regularPrice"
              value={regularPrice}
onChange={onChangeHandler}
              required
             />
             {type === "rent" &&<p className="ms-4 mt-2">$ / Month</p>}
</div>
</div>
        {/* offer */}
        {offer && (
<div className="mb-3 mt-4">
<label htmlFor="discountedPrice" className="form-label">
              Discounted Price :
</label>

<input
              type="number"
className="form-control w-50 "
              id="discountedPrice"
              name="discountedPrice"
              value={discountedPrice}
onChange={onChangeHandler}
              required
             />
</div>
        )}

        {/* files images etc */}
<div className="mb-3">
<label htmlFor="formFile" className="form-label">
            select images :
</label>
<input
className="form-control"
            type="file"
```

```
            id="images"
            name="images"
onChange={onChangeHandler}
            max="6"
            accept=".jpg,.png,.jpeg"
            multiple
            required
           />
</div>
        {/* submit button */}
<div className="mb-3">
<input
            disabled={!name || !address || !regularPrice || !images}
className="btnbtn-primary w-100"
            type="submit"
            value="Create Listing"
           />
</div>
</form>
</div>
</Layout>
  );
};

export default CreateListing;
```

EDIT LISTING ROUTE:

```
import React, { useState, useEffect, useRef } from "react";
import { v4 as uuidv4 } from "uuid";
import { useNavigate } from "react-router-dom";
import Layout from "./../components/Layout/Layout";
import { getAuth, onAuthStateChanged } from "firebase/auth";
import Spinner from "../components/Spinner";
import { AiOutlineFileAdd } from "react-icons/ai";
import { toast } from "react-toastify";
import {
getStorage,
  ref,
uploadBytesResumable,
getDownloadURL,
} from "firebase/storage";
import { db } from "../firebase.config";
import { addDoc, collection, serverTimestamp } from "firebase/firestore";

const CreateListing = () => {
```

```
  const [loading, setLoading] = useState(false);
  const [geoLoactionEnable, setGeoLocationEnable] = useState(false);
  const [formData, setFormData] = useState({
    type: "rent",
    name: "",
    bedrooms: 1,
    bathrooms: 1,
    parking: false,
    furnished: false,
    address: "",
    offer: false,
regularPrice: 0,
discountedPrice: 0,
    images: {},
    latitude: 0,
    longitude: 0,
  });

  const {
    type,
    name,
    bedrooms,
    bathrooms,
    parking,
    furnished,
    address,
    offer,
regularPrice,
discountedPrice,
    images,
    latitude,
    longitude,
  } = formData;

  const auth = getAuth();
  const navigate = useNavigate();
  const isMounted = useRef(true);

useEffect(() => {
    if (isMounted) {
onAuthStateChanged(auth, (user) => {
setFormData({
          ...formData,
useRef: user.uid,
        });
      });
```

```
    } else {
      navigate("/signin");
    }

    // eslint-disable-next-line
  }, []);

  if (loading) {
    return <Spinner />;
  }

  //mutate func
  const onChangeHandler = (e) => {
    let boolean = null;
    if (e.target.value === "true") {
boolean = true;
    }
    if (e.target.value === "false") {
boolean = false;
    }
    //files
    if (e.target.files) {
setFormData((prevState) => ({
        ...prevState,
        images: e.target.files,
      }));
    }
    //text/booleans/number
    if (!e.target.files) {
setFormData((prevState) => ({
        ...prevState,
        [e.target.id]: boolean ?? e.target.value,
      }));
    }
  };

  //form submit
  const onSubmit = async (e) => {
e.preventDefault();
    // console.log(formData);
    if (discountedPrice>= regularPrice) {
setLoading(false);
toast.error("Discount Price should be less than Regular Price");
      return;
    }
    if (images > 6) {
```

```
setLoading(false);
toast.error("Max 6 Images can be selected");
     return;
   }
   let geoLocation = {};
   let location;
   if (geoLoactionEnable) {
     const response = await fetch(

`https://maps.googleapis.com/maps/api/geocode/json?address=${address}&key=AIza
SyBywlow6ReyPu1sX2KbS38EbTklYcDFMsI`
     );
     const data = await response.json();
     console.log(data);
   } else {
geoLocation.lat = latitude;
geoLocation.lng = longitude;
     // location = address;
   }

   //store images to firebase storage
   const storeImage = async (image) => {
     return new Promise((resolve, reject) => {
       const storage = getStorage();
       const fileName = `${auth.currentUser.uid}-${image.name}-${uuidv4()}`;
       const storageRef = ref(storage, "images/" + fileName);
       const uploadTask = uploadBytesResumable(storageRef, image);
uploadTask.on(
         "state_changed",
         (snapshot) => {
           const progress =
             (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
console.log("uplloas is" + progress + "% done");
           switch (snapshot.state) {
             case "paused":
console.log("upload is paused");
               break;
             case "running":
console.log("upload is runnning");
           }
         },
         (error) => {
           reject(error);
         },
         //success
         () => {
```

```
getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
            resolve(downloadURL);
          });
        }
      );
    });
  };
  const imgUrls = await Promise.all(
    [...images].map((image) =>storeImage(image))
).catch(() => {
setLoading(false);
toast.error("Images not uploaded");
    return;
  });
  console.log(imgUrls);

  //save form data
  const formDataCopy = {
    ...formData,
imgUrls,
geoLocation,
    timestamp: serverTimestamp(),
  };
formData.location = address;
  delete formDataCopy.images;
!formDataCopy.offer&& delete formDataCopy.discountedPrice;
  const docRef = await addDoc(collection(db, "listings"), formDataCopy);
toast.success("Listing Created!");
setLoading(false);
  navigate(`/category/${formDataCopy.type}/${docRef.id}`);
 };
 return (
<Layout>
<div className="container d-flex flex-column align-items-center justify-content-
center mb-4">
<h3 className="mt-3 w-50 bg-dark text-light p-2 text-center">
       Create Listing  
<AiOutlineFileAdd />
</h3>
      {/* sell rent button */}
<form className="w-50 bg-light p-4" onSubmit={onSubmit}>
<div className="d-flex flex-row mt-4">
<div className="form-check">
<input
className="form-check-input"
            type="radio"
```

```
            value="rent"
onChange={onChangeHandler}
defaultChecked
            name="type"
            id="type"
           />
<label className="form-check-label" htmlFor="rent">
            Rent
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
            type="radio"
            name="type"
            value="sale"
onChange={onChangeHandler}
            id="type"
           />
<label className="form-check-label" htmlFor="sale">
            Sale
</label>
</div>
</div>
        {/* name */}
<div className="mb-3 mt-4">
<label htmlFor="name" className="form-label">
            Name
</label>
<input
            type="text"
className="form-control"
            id="name"
            value={name}
onChange={onChangeHandler}
            required
          />
</div>
        {/* bedrooms */}
<div className="mb-3 mt-4">
<label htmlFor="bedrooms" className="form-label">
            Bedrooms
</label>
<input
            type="number"
className="form-control"
```

```
            id="bedrooms"
            value={bedrooms}
onChange={onChangeHandler}
            required
           />
</div>
         {/* bathrroms */}
<div className="mb-3 mt-4">
<label htmlFor="bathrooms" className="form-label">
            Bathrooms
</label>
<input
            type="number"
className="form-control"
            id="bathrooms"
            value={bathrooms}
onChange={onChangeHandler}
            required
           />
</div>
         {/* parking */}
<div className="mb-3 ">
<label htmlFor="parking" className="form-label">
Parking :
</label>
<div className="d-flex flex-row ">
<div className="form-check">
<input
className="form-check-input"
               type="radio"
               value={true}
onChange={onChangeHandler}
               name="parking"
               id="parking"
              />
<label className="form-check-label" htmlFor="yes">
               Yes
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
               type="radio"
               name="parking"
               value={false}
defaultChecked
```

```
onChange={onChangeHandler}
              id="parking"
             />
<label className="form-check-label" htmlFor="no">
              No
</label>
</div>
</div>
</div>
        {/* furnished */}
<div className="mb-3 ">
<label htmlFor="furnished" className="form-label">
Furnished :
</label>
<div className="d-flex flex-row ">
<div className="form-check">
<input
className="form-check-input"
              type="radio"
              value={true}
onChange={onChangeHandler}
              name="furnished"
              id="furnished"
             />
<label className="form-check-label" htmlFor="yes">
              Yes
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
              type="radio"
              name="furnished"
              value={false}
defaultChecked
onChange={onChangeHandler}
              id="furnished"
             />
<label className="form-check-label" htmlFor="no">
              No
</label>
</div>
</div>
</div>
        {/* address */}
<div className="mb-3">
```

```
<label htmlFor="address">Address :</label>
<textarea
className="form-control"
          placeholder="Enter Your Address"
          id="address"
          value={address}
onChange={onChangeHandler}
          required
         />
</div>
       {/* geoLoaction */}
{!geoLoactionEnable&& (
<div className="mb-3 ">
<div className="d-flex flex-row ">
<div className="form-check">
<label className="form-check-label" htmlFor="yes">
              Latitude
</label>
<input
className="form-control"
              type="number"
              value={latitude}
onChange={onChangeHandler}
              name="latitude"
              id="latitude"
             />
</div>
<div className="form-check ms-3">
<label className="form-check-label" htmlFor="no">
              Longitude
</label>
<input
className="form-control"
              type="number"
              name="longitude"
              value={longitude}
onChange={onChangeHandler}
              id="longitude"
             />
</div>
</div>
</div>
       )}
       {/* offers  */}
<div className="mb-3 ">
<label htmlFor="offer" className="form-label">
```

```
Offer :
</label>
<div className="d-flex flex-row ">
<div className="form-check">
<input
className="form-check-input"
              type="radio"
              value={true}
onChange={onChangeHandler}
              name="offer"
              id="offer"
             />
<label className="form-check-label" htmlFor="yes">
              Yes
</label>
</div>
<div className="form-check ms-3">
<input
className="form-check-input"
              type="radio"
              name="offer"
              value={false}
defaultChecked
onChange={onChangeHandler}
              id="offer"
             />
<label className="form-check-label" htmlFor="no">
              No
</label>
</div>
</div>
</div>
        {/* regular price */}
<div className="mb-3 mt-4">
<label htmlFor="name" className="form-label">
            Regular Price :
</label>
<div className=" d-flex flex-row ">
<input
              type="number"
className="form-control w-50 "
              id="regularPrice"
              name="regularPrice"
              value={regularPrice}
onChange={onChangeHandler}
              required
```

```
            />
            {type === "rent" &&<p className="ms-4 mt-2">$ / Month</p>}
</div>
</div>
        {/* offer */}
        {offer && (
<div className="mb-3 mt-4">
<label htmlFor="discountedPrice" className="form-label">
            Discounted Price :
</label>

<input
            type="number"
className="form-control w-50 "
            id="discountedPrice"
            name="discountedPrice"
            value={discountedPrice}
onChange={onChangeHandler}
            required
           />
</div>
        )}

        {/* files images etc */}
<div className="mb-3">
<label htmlFor="formFile" className="form-label">
           select images :
</label>
<input
className="form-control"
            type="file"
            id="images"
            name="images"
onChange={onChangeHandler}
            max="6"
            accept=".jpg,.png,.jpeg"
            multiple
            required
          />
</div>
        {/* submit button */}
<div className="mb-3">
<input
            disabled={!name || !address || !regularPrice || !images}
className="btnbtn-primary w-100"
            type="submit"
```

```
            value="Create Listing"
          />
</div>
</form>
</div>
</Layout>
  );
};

export default CreateListing;
```

COMPONENTS:-

HEADER.JS:

```
import React from "react";
import { Link, NavLink } from "react-router-dom";
import { BsFillHouseHeartFill } from "react-icons/bs";
import { FaBars } from "react-icons/fa";
import "../../styles/Header.css";

const Header = () => {
```

```
 return (
<>
<nav className="navbar navbar-expand-lg navbar-light bg-light sticky-sm-top">
<div className="container">
<Link className="navbar-brand" to="/">
<BsFillHouseHeartFill size={30} className="me-2" /> HOUSE MARKETPLACE
</Link>
<button
className="navbar-toggler"
        type="button"
        data-bs-toggle="collapse"
        data-bs-target="#navbarTogglerDemo02"
        aria-controls="navbarTogglerDemo02"
        aria-expanded="false"
        aria-label="Toggle navigation"
>
<span className="navbar-toggler-icon">
<FaBars />
</span>
</button>
<div className="collapse navbar-collapse" id="navbarTogglerDemo02">
<ulclassName="navbar-nav ms-auto mb-2 mb-lg-0">
<li className="nav-item">
<NavLink
className={`nav-link ${({ isActive }) =>
isActive ? "active" : "inactive"}`}
            aria-current="page"
            to="/"
>
            Explore
```

```
</NavLink>
</li>
<li className="nav-item">
<NavLink
className={`nav-link ${({ isActive }) =>
isActive ? "active" : "inactive"}`}
        to="/offers"
>
        Offers
</NavLink>
</li>
<li className="nav-item">
<NavLink
className={`nav-link ${({ isActive }) =>
isActive ? "active" : "inactive"}`}
        to="/profile"
>
        Profile
</NavLink>
</li>
</ul>
</div>
</div>
</nav>
</>
  ); };
export default Header;
```

FOOTER.JS:

```
import React from "react";
import { BsGithub, BsInstagram, BsLinkedin, BsTelegram} from "react-icons/bs";
```

```
import { MdEmail } from "react-icons/md";
import { Link } from "react-router-dom";
import "../../styles/footer.css";


const Footer = () => {
  return (
<div className="footer pt-4 d-flex flex-column align-items-center justify-content-center bg-dark text-light p-4">
<h3>
      Made With
<img
src="./assets/love.gif"
       alt="love"
       height={60}
       width={80}
className="mx-3 footer-gif"
      />
      From India
</h3>
<h6>All Right Reserved &copy; House Marketplace - 2023</h6>
<div className="d-flex flex-row p-2">
<p className="me-4" title="Github">
<Link to="https://github.com/Shraddha87Jha">
<BsGithub color="black" size={30} />
</Link>
</p>
<p className="me-4" title="Instagram">
<Link to="https://www.instagram.com/accounts/login/">
<BsInstagram color="black" size={30} />
</Link>
</p>
```

```
<p className="me-4" title="Linked In">
<Link to="https://www.linkedin.com/in/shraddha-jha-07957b217/">
<BsLinkedin color="black" size={30} />
</Link>
</p>
<p className="me-4" title="Email">
<Link to="">
<MdEmail  color="black" size={30} />
</Link>
</p>
<p className="me-4" title="Telegram">
<Link to="/">
<BsTelegram color="black" size={30} />
</Link>
</p>


</div>
</div>
  );};
export default Footer;
```

LAYOUT.JS:

```
import React from "react";
import Footer from "./Footer";
import Header from "./Header";
import { Helmet } from "react-helmet";
const Layout = ({ children, title }) => {
  return (
<>
<Helmet>
<meta charSet="utf-8" />
```

```
<title>{title}</title>
</Helmet>
<Header />
<main>{children}</main>
<Footer />
</>
 );};
Layout.defaultProps = {
 title: "House Markteplace - search best home near you  ",
};
export default Layout;
```

LISTINGITEMS.JS:

```
import React from "react";
import { Link } from "react-router-dom";
import { FaBed, FaBath } from "react-icons/fa";
import { GiTakeMyMoney } from "react-icons/gi";
import "../styles/listingitem.css";
const ListingItem = ({ listing, id, onDelete, onEdit }) => {
 return (
<>
<div className="card-item-parent d-flex align-items-center justify-content-center">
<div className="item-card category-link mb-2 w-75  ">
<Link to={`/category/${listing.type}/${id}`}>
<div className="row  p-2">
```

```
<div className="col-md-5 item-card-continer1">
<imgsrc={listing.imgUrls[0]} alt={listing.name} />
</div>
<div className="col-md-5 item-card-continer2">
<h2>{listing.name}</h2>
<p>{listing.location}</p>
<p>
<GiTakeMyMoney />RS :{" "}
                {listing.offer
                 ? listing.discountedPrice
                 : listing.regularPrice}{" "}
                {listing.type === "rent" && " / Month"}
</p>
<p>
<FaBed /> 
                {listing.bedrooms> 1
                 ? `${listing.bedrooms} Bedrooms`
                 : "1 Bedreoom"}
</p>
<p>
<FaBath /> 
                {listing.bathrooms> 1
                 ? `${listing.bathrooms} Bathrooms`
                 : "1 Bathroom"}
</p>
</div>
</div>
</Link>
<div className="m-2 p-3">
          {onDelete&& (
```

```
<button
className="btnbtn-danger"
onClick={() =>onDelete(listing.id)}
>
            Delete Listing
</button>
          )}
          {onEdit&& (
<button
className="btnbtn-info ms-3"
onClick={() =>onEdit(listing.id)}
>
            Edit Listing
</button>
          )}
</div>
</div>
</div>
</>
 );};
export default ListingItem;
```

SLIDER.JS:

```
import React, { useState, useEffect } from "react";
import { ImLocation2 } from "react-icons/im";
import { db } from "../firebase.config";
import "../styles/slider.css";
import {
  collection,
getDoc,
```

```
  query,
orderBy,
  limit,
getDocs,
} from "firebase/firestore";
import { useNavigate } from "react-router-dom";
import Swiper from 'swiper';
import SwipeCore, { EffectCoverflow, Navigation, Pagination } from "swiper";
import { Swiper, SwiperSlide } from "swiper/react";
import "swiper/swiper-bundle.min.css";
import "swiper/swiper.min.css";
import Spinner from "./Spinner";

//config
SwipeCore.use([EffectCoverflow, Pagination]);

const Slider = () => {
  const [listings, setListings] = useState(null);
  const [loading, setLoading] = useState(true);
  const navigat = useNavigate();
  const userPic =
    "https://openclipart.org/download/247319/abstract-user-flat-3.svg";

useEffect(() => {
    const fetchListings = async () => {
      const listingRef = collection(db, "listings");
      const q = query(listingRef, orderBy("timestamp", "desc"), limit(5));
      const querySnap = await getDocs(q);
      let listings = [];
querySnap.forEach((doc) => {
```

```
        return listings.push({
          id: doc.id,
          data: doc.data(),
        });
      });
setLoading(false);
setListings(listings);
    };
fetchListings();
console.log(listings === null ? "loading" : listings);
    // eslint-disable-next-line
  }, []);

  if (loading) {
    return <Spinner />;
  }
  return (
<>
<div style={{ width: "100%" }}>
        {listings === null ? (
<Spinner />
        ) : (
<Swiper
            effect={"coverflow"}
grabCursor={true}
centeredSlides={true}
slidesPerView={1}
coverflowEffect={{
              rotate: 50,
              stretch: 0,
```

```
            depth: 100,
            modifier: 1,
slideShadows: true,
          }}
          pagination={true}
className="mySwipe"
>
          {listings.map(({ data, id }) => (
<SwiperSlide
            key={id}
onClick={() => {
navigat(`/category/${data.type}/${id}`);
            }}
>
<img
src={data.imgUrls[0]}
              alt={data.name}
className="slider-img"
            />
<h4 className=" text-light p-4 m-0 ">
              {/* <img alt="user pic" src={userPic} height={35} width={35} /> */}
<ImLocation2 size={20} className="ms-2" /> Recently Added :{" "}
<br />
<span className="ms-4 mt-2"> {data.name}</span>
<span className="ms-2">
                | Price ( $ {data.regularPrice} )
</span>
</h4>
</SwiperSlide>
          ))}
```

```
</Swiper>
      )}
</div>
</>
 );};
export default Slider;
```

SPINNER.JS:

```
import React from "react";
import "../styles/spinner.css";
const Spinner = () => {
 return (
  // <div className="spinner-border text-primary" role="status">
  //   <span className="visually-hidden">Loading...</span>
  // </div>
<div className="loader"></div>
 );};
export default Spinner;
```

OAUTH.JS:

```
import React from "react";
import { useLocation, useNavigate } from "react-router-dom";
import { getAuth, signInWithPopup, GoogleAuthProvider } from "firebase/auth";
import { doc, setDoc, getDoc, serverTimestamp } from "firebase/firestore";
import { db } from "../firebase.config";
import { toast } from "react-toastify";
import { FcGoogle } from "react-icons/fc";

const OAuth = () => {
  const navigate = useNavigate();
```

```
  const location = useLocation();

  const onGoolgleAuthHandler = async () => {
   try {
     const auth = getAuth();
     const provider = new GoogleAuthProvider();
     const result = await signInWithPopup(auth, provider);
     const user = result.user;
     const docRef = doc(db, "users", user.uid);
     const docSnap = await getDoc(docRef);
     if (!docSnap.exists()) {
       await setDoc(doc(db, "users", user.uid), {
         name: user.displayName,
         email: user.email,
         timestamp: serverTimestamp(),
       });
     }
navigate("/");
   } catch (error) {
toast.error("Problem With Google Auth ");
   }
  };

  return (
<div>
<h3 className="mt-4 text-center ">
       Sign {location.pathname === "/signup" ? "Up" : "in"} With
<button
onClick={onGoolgleAuthHandler}
         style={{
```

```
            outline: "none",
        backgroundColor: "transparent",
            border: "none",
        borderBottom: "1px solid black",
          }}
        >
        <span>
        <FcGoogle />
            Google
        </span>
        </button>
        </h3>
        </div>
          );
        };

        export default OAuth;
```

PRIVATE ROUTE.JS

```
        import { Navigate,Outlet } from "react-router-dom";
        import React from "react";
        import useAuthState from './../hooks/useAuthState';
        import Spinner from "./Spinner";

        const PrivateRoute = () =>{
        const {loggedIn,checkState}=useAuthState()

        if(checkState){
           return <Spinner/>
        }
```

```
return loggedIn?<Outlet /> : <Navigate to="/signin"/>;
};
export default PrivateRoute;
```

CSS styles code:

CONTACT.CSS

```
.contact-container {
   margin: 40px 0px;
   display: flex;
   align-items: center;
   justify-content: center;
 }
 .contact-container-col-1 img {
   width: 100%;
   height: 400px;
 }
 .contact-container-col-2 h1 {
   font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
   font-weight: 700;
 }
 .contact-container-col-2 main {
   margin-top: 30px;
 }
  .contact-container-col-2 textarea {
   height: 120px !important;
   box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px,
     rgba(0, 0, 0, 0.3) 0px 1px 3px -1px !important;
   color: #470d21;
   font-size: 1.5rem !important;
 }
```

```
.contact-container-col-2 button {
  background-color: #470d21;
  color: white;
  width: 250px;
  margin-top: 20px !important;
}
.contact-container-col-2 button:hover {
  background-color: #a94266;
  color: white;
  width: 250px;
  margin-top: 20px !important;
}

/* media query */
@media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2) {
  .contact-container-col-1 img {
    width: 100%;
    height: 200px;
  }
  .contact-container-col-2 {
    margin-top: 20px;
  }
  .contact-container-col-2 button {
    width: 350px !important;
  }
  .contact-container-col-2 textarea {
    font-size: 1rem !important;
  }
}
```

FOOTER.CSS

```
@import url("https://fonts.googleapis.com/css2?family=Pacifico&display=swap");
.footer {
  background: rgb(71, 13, 33) !important;
  background: linear-gradient(
    90deg,
    rgb(68, 10, 30) 0%,
    rgb(23, 32, 124) 100%
  ) !important;
}
.footer h3 {
  font-family: "Pacifico", cursive;
}

.footer p:hover {
  background-color: #f9e4d4 !important;
  color: #f9e4d4 !important;
  padding: 2px;
}

@media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2) {
  .footer h3 {
    font-size: 1rem !important;
  }
  .footer-gif {
    height: 40px !important;
    width: 40px !important;
  }
}
```

FORGOT PASSWORD.CSS

```
.forgot-password-col1 img {
  height: 400px;
  width: 100%;
 }

 .forgot-password-container {
  padding: 20px;
 }

 .forgot-password-col2 {
  text-align: center;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  background-color: #3f3d56;
  box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px,
   rgba(0, 0, 0, 0.3) 0px 1px 3px -1px !important;
 }

 .forgot-password-col2 h1 {
  font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
   "Lucida Sans Unicode", Geneva, Verdana, sans-serif;
  font-weight: 700;
  color: white;
 }
 .forgot-password-col2 label {
  color: white;
  margin-top: 30px;
```

```
  display: flex;
  align-items: flex-start;
  font-size: 1rem;
}
.forgot-password-col2 input {
  width: 440px;
}
.forgot-password-col2 button {
  width: 300px;
  border: 1px solid white;
  background-color: #3f3d56;
  color: white;
}
.forgot-password-col2 button:hover {
  color: white;
  border: 1px solid transparent !important;
  background-color: #470d21;
}
.forgot-password-col2 .signin {
  width: 100px;
  text-decoration: none;
  color: #470d21;
  background-color: white;
}
.forgot-password-col2 .signin:hover {
  width: 100px;
  text-decoration: none;
  color: black;
  background-color: white;
  opacity: 0.7;
```

```
}

/* media query */
@media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2) {
  .forgot-password-col1 img {
    height: 200px;
    width: 100%;
  }
  .forgot-password-col2 {
    margin-top: 10px;
    padding: 20px 0px;
  }
  .forgot-password-col2 h1 {
    font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
      "Lucida Sans Unicode", Geneva, Verdana, sans-serif;
    font-weight: 700;
    color: white;
    margin-top: 20px;
  }
  .forgot-password-col2 label {
    color: white;
    margin-top: 10px;
    display: flex;
    font-size: 1.3rem;
  }
  .forgot-password-col2 input {
    width: 300px;
  }
  .forgot-password-col2 .btn-goup {
    display: flex !important;
```

```
    flex-direction: column !important;
    align-items: center;
    justify-content: center;
    width: 300px;
  }
  .forgot-password-col2 .signin {
    width: 300px;
    margin-top: 20px;
  }
 }
```

HEADER.CSS

```
@import url('https://fonts.googleapis.com/css2?family=Montserrat&display=swap');


.navbar {
  font-family: 'Montserrat', sans-serif !important;
  font-weight: bold;
  background-color: #470D21 !important;
}
.navbar .navbar-brand{
  color: white !important;
  font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif;
}
.navbar .navbar-toggler-icon{
 color: white !important;
}
.navbar button:focus{
 color: white !important;
}
.navbar .nav-link{
  color: #d1a138 !important;
```

```
}


.navbar .nav-item a:hover{
   border-radius: 10px/30px;
   color: #9C0F48 !important;
   background: rgb(249,228,212);
background: linear-gradient(90deg, rgba(249,228,212,1) 0%, rgba(214,125,62,1) 100%);
   /* background-color: #F9E4D4 !important ; */
   box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px, rgba(0, 0, 0, 0.3) 0px 1px 3px -1px
!important;
}
.navbar .active{
   border-radius: 10px/30px;
   background-color: #F9E4D4 !important ;
   color: #9C0F48 !important;
   box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px, rgba(0, 0, 0, 0.3) 0px 1px 3px -1px
!important;
}
```

HOMEPAGE.CSS

```
.home-cat {
   padding-top: 40px !important;
   padding-bottom: 40px !important;
   padding: 0;
   margin: 0;
   background: url("../../public/assets/back.jpg") no-repeat;
   background-size: 100% 100%;
 }


 .home-cat h1 {
   text-align: center;
   font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
```

```
    "Lucida Sans Unicode", Geneva, Verdana, sans-serif;
   margin-bottom: 5rem;
   margin-top: 2rem;
   font-weight: 700;
  }

  /* media quer */
  @media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-
webkit-min-device-pixel-ratio: 2) {
   .home-cat h1 {
    margin-bottom: 2rem;
    margin-top: 2rem;
   }
   .home-cat {
    padding-top: 40px !important;
    padding-bottom: 40px !important;
    padding: 0;
    margin: 0;
    background: url("../../public/assets/back.jpg") no-repeat;
    background-size: 100% 50%;
   }
  }
```

LISTING.CSS

```
.listing-container {
   margin: 60px 0px;
  }
  .listing-container-col1 img {
   width: 100%;
   height: 400px;
  }
```

```
.listing-container-col2 {
  border-left: 5px solid #470d21;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  justify-content: center;
}
.listing-container-col2 h3 {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  font-weight: 600;
  margin-bottom: 30px;
}

/* media query */
@media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2) {
  .listing-container {
    margin-top: 10px;
    padding: 0x;
  }
  .listing-container-col2 {
    border-left: none;
    margin-left: 10px;
    margin-top: 30px;
  }
}
```

LISTINGITEM.CSS

```
.item-card {
  border: 1px solid white;
  background-color: rgba(172, 229, 243, 0.527);
```

```
  box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px,
    rgba(0, 0, 0, 0.3) 0px 1px 3px -1px !important;
  padding: 10px;
  margin-bottom: 10px;
}
.item-card:hover {
  background-color: rgba(243, 85, 164, 0.213);
}

.item-card-continer1 {
  border-right: 5px solid #470d21;
}
.item-card-continer1 img {
  height: 200px;
  width: 100%;
  border: 5px solid rgba(0, 0, 0, 0.173);
}
.item-card-continer2 {
  margin-top: 10px;
}
.item-card-continer2 h2 {
  font-weight: 600;
}
/* media query */
 @media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-
webkit-min-device-pixel-ratio: 2) {
  .item-card-continer1 {
    border-right: none;
  }
  .item-card {
    padding: 0px;
```

```
  }
  .card-item-parent {
    margin: 0px -40px;
  }
  .item-card-continer2 {
    margin-left: 15px;
    margin-top: 15px;
  }
}
```

OFFERS.CSS

```
.offers {
  padding-bottom: 50px;
  background: url("../../public/assets/offer-bg.jpg") no-repeat;
  background-size: 100% 100%;
}

.offers h1 {
  font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
  text-align: center;
  margin: 30px 0px;
}

.offers .offer-img {
  height: 70px;
  width: 110px;
}

.load-btn {
  background-color: #470d21 !important;
  font-size: 1rem;
```

```
  font-family: cursive;
  padding: 5px 15px;
  color: white;
  outline: none;
  border: none;
  border-radius: 5px;
}
.load-btn:hover {
  color: #470d21 !important;
  font-size: 1rem;
  font-family: cursive;
  padding: 5px 15px;
  background-color: white !important;

  border-radius: none;
}
/* media query */
@media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2) {
  .offers h1 {
    font-family: cursive;
    font-size: 2rem;
    text-align: center;
    margin: 30px 0px;
  }
  .offers .offer-img {
    height: 70px;
    width: 90px;
  }
}
```

PROFILE.CSS

```
.profile-container {
  margin: 0px !important;
  margin-top: 60px !important;
 }
 .profile-container-col1 img {
  width: 100%;
  height: 400px;
 }
 .profile-container-col2 h2 {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  font-weight: 900;
 }
 .profile-container-col2 h3 {
  margin-bottom: 50px !important;
 }
 .your-listings {
  background: #8e9eab; /* fallback for old browsers */
  background: -webkit-linear-gradient(
    to right,
    #eef2f3,
    #8e9eab
  ); /* Chrome 10-25, Safari 5.1-6 */
  background: linear-gradient(
    to right,
    #eef2f3,
    #040124
  ); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */

  padding-top: 30px;
  padding-bottom: 30px;
```

```
  }
  .your-listings h3 {
    margin-top: 80px !important;
    font-weight: 700;
    font-family: cursive;
    text-align: center;
    margin-bottom: 50px !important;
  }
  .your-listings-mt-4 h3{
    color:black;
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    }
  /* media query */
  @media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-
webkit-min-device-pixel-ratio: 2) {
    .profile-container-col1 img {
      width: 100%;
      height: 200px;
    }
    .create-listing {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
    }
    .your-listings h3 {
      margin-top: 0px !important;
    }
  }
```

SIGNIN.CSS

```
.signin-container {
```

```
  box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px,
    rgba(0, 0, 0, 0.3) 0px 1px 3px -1px !important;
}

.signin-container img {
  height: 400px;
  width: 100%;
  margin-top: 40px;
}

.signin-container-col2 {
  background-color: rgba(250, 235, 215, 0.464);
  padding: 40px;
  padding-top: 30px;
  font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
    "Lucida Sans Unicode", Geneva, Verdana, sans-serif;
}

.signin-container-col2 h4 {
  font-family: Georgia, "Times New Roman", Times, serif;
  font-size: 3rem;
}
.signin-container-col2 input:focus {
  outline: 1px solid #470d21 !important;
  box-shadow: none !important;
  color: #470d21;
}

.signinbutton {
  background-color: #470d21;
```

```
  color: white;
  width: 200px !important;
 }
 .signinbutton:hover {
  background-color: #b12556;
  color: white;
 }

 /* media query */
 @media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2) {
  .new-user {
   display: inline-block;
   margin-top: 20px;
  }
  .show-pass-forgot {
   font-size: 10px;
  }
  .signin-container img {
   height: 200px;
   width: 100%;
  }
  .signinbutton {
   width: 260px !important;
  }
 }
```

SIGNUP.CSS

```
.signup-container {
  margin: 0px;
```

```
  margin-top: 20px;
  padding: 20px;
}
.signup-container-col-1 img {
  height: 600px;
  width: 100%;
}

.signup-container-col-2 {
  box-shadow: rgba(50, 50, 93, 0.25) 0px 2px 5px -1px,
    rgba(0, 0, 0, 0.3) 0px 1px 3px -1px !important;
  padding: 0px 40px;
  margin-bottom: 100px !important;
}
.signup-container-col-2 h3 {
  font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande",
    "Lucida Sans Unicode", Geneva, Verdana, sans-serif;
  font-weight: 600;
  margin-top: 30px !important;
}
.signup-container-col-2 .signup-button {
  background-color: #3f3d56;
  color: white;
  width: 250px;
}
.signup-container-col-2 .signup-button:hover {
  background-color: #470d21;
  color: white;
  border-radius: 5px;
}
```

```
 /* media query */
 @media only screen and (min-device-width: 320px) and (max-device-width: 480px) and (-
webkit-min-device-pixel-ratio: 2) {
   .signup-container-col-1 img {
     height: 200px;
     width: 100%;
   }
   .signup-container-col-2 .signup-button {
     margin-bottom: 10px;
   }
   .signup-container-col-2 span {
     margin-left: 0px !important;
   }
 }
```

SLIDER.CSS

```
.slider-img{
   width: 100% !important;
   height: 75vh !important;

}

.mySwipe h4{
   background-color: #F9E4D4 !important;
   color: #470D21 !important;
   font-size: 16px;
   border-left: 30px solid #470D21;
}

@media only screen
```

```
  and (min-device-width: 320px)
  and (max-device-width: 480px)
  and (-webkit-min-device-pixel-ratio: 2) {
    .slider-img{
        width: 100% !important;
        height: 52vh !important;
    }
    .mySwipe h4{
        background-color: #F9E4D4 !important;
        color: #470D21 !important;
        font-size: 14px;
        padding-left: 0px !important;
        padding-top: 10px !important;
    }
}
```

SPINNER.CSS

```
.loader {
    color: #8e0b51;
    font-size: 90px;
    text-indent: -9999em;
    overflow: hidden;
    width: 1em;
    height: 1em;
    border-radius: 50%;
    margin: 72px auto;
    position: relative;
    -webkit-transform: translateZ(0);
    -ms-transform: translateZ(0);
    transform: translateZ(0);
    -webkit-animation: load6 1.7s infinite ease, round 1.7s infinite ease;
```

```
  animation: load6 1.7s infinite ease, round 1.7s infinite ease;
}
@-webkit-keyframes load6 {
  0% {
    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,
      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;
  }
  5%,
  95% {
    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,
      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;
  }
  10%,
  59% {
    box-shadow: 0 -0.83em 0 -0.4em, -0.087em -0.825em 0 -0.42em,
      -0.173em -0.812em 0 -0.44em, -0.256em -0.789em 0 -0.46em,
      -0.297em -0.775em 0 -0.477em;
  }
  20% {
    box-shadow: 0 -0.83em 0 -0.4em, -0.338em -0.758em 0 -0.42em,
      -0.555em -0.617em 0 -0.44em, -0.671em -0.488em 0 -0.46em,
      -0.749em -0.34em 0 -0.477em;
  }
  38% {
    box-shadow: 0 -0.83em 0 -0.4em, -0.377em -0.74em 0 -0.42em,
      -0.645em -0.522em 0 -0.44em, -0.775em -0.297em 0 -0.46em,
      -0.82em -0.09em 0 -0.477em;
  }
  100% {
    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,
```

```
    0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;
  }
}
@keyframes load6 {
  0% {
    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,
      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;
  }
  5%,
  95% {
    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,
      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;
  }
  10%,
  59% {
    box-shadow: 0 -0.83em 0 -0.4em, -0.087em -0.825em 0 -0.42em,
      -0.173em -0.812em 0 -0.44em, -0.256em -0.789em 0 -0.46em,
      -0.297em -0.775em 0 -0.477em;
  }
  20% {
    box-shadow: 0 -0.83em 0 -0.4em, -0.338em -0.758em 0 -0.42em,
      -0.555em -0.617em 0 -0.44em, -0.671em -0.488em 0 -0.46em,
      -0.749em -0.34em 0 -0.477em;
  }
  38% {
    box-shadow: 0 -0.83em 0 -0.4em, -0.377em -0.74em 0 -0.42em,
      -0.645em -0.522em 0 -0.44em, -0.775em -0.297em 0 -0.46em,
      -0.82em -0.09em 0 -0.477em;
  }
  100% {
```

```
    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,
      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;
  }
}
@-webkit-keyframes round {
  0% {
    -webkit-transform: rotate(0deg);
    transform: rotate(0deg);
  }
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
}
@keyframes round {
  0% {
    -webkit-transform: rotate(0deg);
    transform: rotate(0deg);
  }
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
}
```

FUTURE SCOPE OF PROJECT

This project is developed as an internship project and still gives a lot of scope for its extension which could be made to the project if it is going to be developed as a commercial product. We can use a pure object-oriented domain model to deal with the database access tier using SQL Server. In doing this, we can get a better architecture design that will improve code efficiency and running performance. The feature of providing Google Maps within this application adds up to the functionality of the website. With the advancement of technology, dynamic maps can be generated using AJAX which can help the buyer locate a particular area where the property is located in the Google Map. The inclusion of all these features would make the application feature-rich. The advantages of putting these functionalities in the project are described in detail in the following sections.

The future scope of a real estate project depends on various factors such as location, market trends, economic conditions, and technological advancements. Here are some considerations for the future scope of a real estate project:

1.Market Trends: Understanding current and future market trends is crucial. Factors such as population growth, demographic shifts, urbanization, and changes in consumer preferences can influence the demand for different types of real estate properties.

2.Technological Advancements: Emerging technologies such as virtual reality (VR), augmented reality (AR), Building Information Modelling (BIM), Internet of Things (IoT), and green building practices are transforming the real estate industry. Integrating these technologies into the project can enhance efficiency, sustainability, and user experience, thus increasing the project's attractiveness and value.

3.Sustainability and Green Building: With increasing awareness of environmental issues, there's a growing demand for sustainable and eco-friendly buildings. Incorporating green building practices, energy-efficient designs, and renewable energy sources can not only reduce operational costs but also attract environmentally conscious tenants and buyers.

4.Mixed-Use Developments: Mixed-use developments, which combine residential, commercial, retail, and recreational spaces in a single project, are becoming increasingly popular. They offer convenience, diversity, and vibrant communities, thus presenting significant opportunities for real estate developers.

5.Infrastructure Development: Infrastructure projects such as new transportation networks, airports, highways, and public amenities can significantly impact the value and demand for real estate in certain areas. Keeping an eye on planned infrastructure developments can provide insights into future growth opportunities.

6.Flexible and Adaptive Spaces: Flexibility is becoming increasingly important in real estate projects, especially in the wake of changing work patterns and lifestyles. Designing spaces that can adapt to different uses and accommodate evolving needs can ensure long-term viability and relevance.

LIMITATION OF PROJECT

Real estate projects can face several limitations and challenges, which can impact their success and profitability. Some of the common limitations include:

1. **Market Volatility:** Real estate markets are subject to fluctuations influenced by economic conditions, interest rates, consumer sentiment, and geopolitical factors. Market downturns can lead to reduced demand, declining property values, and difficulties in securing financing.

2. **Financing Constraints:** Real estate development often requires substantial upfront capital investment. Securing financing from banks, investors, or other sources can be challenging, especially for large-scale projects or in periods of economic uncertainty. Tight lending standards, high interest rates, and stringent loan requirements can limit access to funding.

3. **Regulatory Hurdles:** Real estate projects are subject to various regulations, zoning laws, environmental restrictions, and building codes imposed by government authorities. Navigating complex regulatory frameworks and obtaining necessary permits and approvals can be time-consuming, costly, and fraught with uncertainties, leading to delays and additional expenses.

4. **Construction Risks:** Construction projects entail inherent risks such as cost overruns, delays, supply chain disruptions, labor shortages, and unforeseen technical issues. Poor project management, design flaws, contractor disputes, and adverse weather conditions can further exacerbate construction-related challenges, impacting project timelines and budgets.

5. **Market Saturation:** Oversupply of real estate properties in a particular market segment or location can lead to increased competition, downward pressure on prices, and longer vacancy periods. Failure to accurately assess market demand and competition levels can result in overbuilding and diminished returns on investment.

6. **Environmental and Sustainability Concerns:** Increasing awareness of environmental issues and sustainability considerations has raised expectations for green building practices and eco-friendly design standards. Failure to incorporate sustainable features, address environmental concerns, or comply with regulatory requirements related to energy efficiency and carbon emissions can undermine the attractiveness and long-term viability of real estate projects.

CONCLUSION

This Real Estate Web Application is a typical React Native web application using React and SQL in the Javascript programming language. It is developed in Microsoft's Visual code NodeJS programming environment. The buyer performs a search for the property listings by putting either Zipcode/City/State or MLS# in the search textbox. The business logic tier communicates with the database tier requesting the results of the query sent by it. The results obtained by the database are displayed on the data grid, by refreshing the grid rather than refreshing the entire web page.

The efficiency of the application is improved by the use of web methods that help in separating the Application Tier from the Presentation Tier. The performance of this application is evaluated by rigorously testing it against various test scenarios. The efficiency and correctness of the application are evaluated with the help of various test cases. Some ways in which this system could be enhanced with additional functionalities are discussed in the section.

FUTURE CONSIDERATIONS:

This chapter describes the future scope and extensions for the project. There is still a huge scope of implementing something new and more to the project which can make it to the level of a commercial product. This section also concludes by starting with the advantages and applications of this Real Estate Web Application.

BIBLIOGRAPHY

BOOK REFERENCES:

- A Complete Overview On : WEB DEVELOPMENT , by Ayush Mauryavanshi.
- HTML and CSS: Visual QuickStart Guide.
- React js: "Learning **React**" by Kirupa Chinnathambi;
- Mongo: MongoDB complete guide by Manu Sharma
- Node js: NODE. JS GUIDEBOOK: Dhruti Shah

WEB REFERENCES:

[1] React library Overview, https://reactjs.org/

[2] Microsoft SQL Server, http://en.wikipedia.org/wiki/Microsoft_SQL_Server

[3] 3-Tier System Architecture, http://en.wikipedia.org/wiki/Multitier_architecture

[4] Introduction to NodeJS Architecture, https://nodejs.org/en/