REAL ESTATE APPLICATION

A PROJECT REPORT

Submitted by

Register No: 730922632001 - ABIN BABU

Register No: 730922632002 - R.AMUDHEESWARAN

Register No: 730922632003 - P A.ARUN

Register No: 730922632004 - S.CHANDRU

Register No: 730922632005 - K.CHITRAARASU

in partial fulfillment of the requirement
for the award of the degree
of

MASTER OF COMPUTER APPLICATIONS

in

DEPARTMENT OF COMPUTER APPLICATIONS

EXCEL ENGINEERING COLLEGE

An Autonomous Institution, affiliated to Anna University Chennai

Approved by AICTE, New Delhi

KOMARAPALAYAM – 637303.

NOV/DEC - 2023

EXCEL ENGINEERING COLLEGE

KOMARAPALAYAM – 637303

BONAFIDE CERTIFICATE

Certified that this project report titled **"REAL ESTATE APPLICATION"** is the bonafide work of **ABIN BABU – (Register No: 730922632001) R.AMUDHEESWARAN – (Register No: 730922632002) P A.ARUN- (Register No:730922632003) S.CHANDRU – (Register No: 730922632004) K.CHITRAARASU- (Register No: 730922632005)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	***SIGNATURE***
Dr.P.S.VELUMANI,MCA.,Ph.D.,	*Mr.M.K.NAGESWARAN,MCA.,M.Phil.,*
HEAD OF THE DEPARTMENT	***SUPERVISOR***
Professor / Head	*Assistant professor*
Department of MCA	*Department of MCA*
Excel Engineering College	*Excel Engineering College*
Komarapalayam – 637303.	*Komarapalayam – 637303.*

Submitted for the viva-voce examination held on ________________.

Internal Examiner **External Examiner**

DECLARATION

I jointly declare that the project report on **“REAL ESTATE APPLICATION”** is the result of original work done by me and best of my knowledge, similar work has not been submitted to **“ANNA UNIVERSITY, CHENNAI”** for therequirement of Degree of **MASTER OF COMPUTER APPLICATIONS**. Thisproject report is submitted on the partial fulfilment of the requirement of the award of Degree of **MASTER OF COMPUTER APPLICATIONS**.

SIGNATURE

____

(730922632001 - Abin Babu)

SIGNATURE

____

(730922632002 - R.Amudheeswaran)

SIGNATURE

____

(730922632003 - P A.Arun)

SIGNATURE

____

(730922632004 - S.Chandru)

SIGNATURE

____

(730922632005 - K.Chitraarasu)

Place: Komarapalayam

Date: _____________.

ACKNOWLEDGEMET

We wish to express our sincere gratitude to our honorable chairman **Dr.A.K.NATESAN,M.Com., MBA., Ph.D., FTA.,** for providing immense facilities a tour institution.

We are very proudly rendering our thanks to our Principal **Dr. K. BOMMANNA RAJA, M.E., Ph.D.,** for the facilities and the encouragement given by him to the progress and completion of our project.

We proudly render our immense gratitude to the Head of the Department for his **Dr.P.S.VELUMANI,MCA.,Ph.D.,** effective leadership, encouragement and guidance in the project.

We are highly indebted to provide our heart full thanks to our supervisor **Mr. M.K. NAGESWARAN,MCA.,M.Phil.,** Assistant Professor for his valuable ideas, encouragement and supportive guidance throughout the project.

We wish to extend our sincere thanks to all faculty members of ours Computer Applications Department for their valuable suggestions, kind cooperation and constant encouragement for successful completion of this project.

We wish to acknowledge the help received from various Departments and various individuals during the preparation and editing stages of the manuscript.

TABLE OF CONTENT

ABSTRACT

The Real Estate Application for Buying, Selling, and Renting Homes is a versatile and user-centric digital platform that redefines the real estate experience. This application seamlessly connects buyers, sellers, and renters with the properties and services they need, offering a comprehensive set of features designed to streamline the entire real estate transaction process. A vast and regularly updated database of residential properties, available for purchase, sale, or rent. Advanced search options with filters, enabling users to find properties tailored to their preferences. High-resolution images and comprehensive property descriptions for informed decision-making. Personalized profiles for users, enabling them to manage their preferences and activities. Saved searches and favorite listings for quick reference. Notifications for updates on properties of interest. Tools for homeowners and property managers to list properties for sale or rent, complete with pricing and availability details. Integration with property evaluation and inspection services to ensure accurate property representation. Assistance with property promotion and marketing. In-app messaging and chat functionality for seamless interactions between buyers, sellers, and agents. Notifications for property updates and appointment scheduling. Video integration for virtual property tours and meetings. Stringent security measures to protect user data and transactions. Compliance with privacy regulations and data protection standards.

CHAPTER 1
INTRODUCTION

1.1 Introduction

The Real Estate Application for Buying, Selling, and Renting Homes represents a transformative leap in the realm of real estate technology. In a rapidly evolving industry, where the search for a dream home, the sale of a cherished property, or the quest to find the perfect rental can be an intricate and sometimes daunting process, this innovative application emerges as a beacon of efficiency and convenience.

The world of real estate, often characterized by its complexities, has been revitalized by this comprehensive digital platform, designed to cater to the diverse needs of buyers, sellers, and renters. This application serves as an invaluable resource, delivering a seamless and user-friendly experience from the initial property search to the final contract signing.

In an age where technology has revolutionized every facet of our lives, it is only fitting that the real estate industry undergoes a digital transformation of its own. The Real Estate Application for Buying, Selling, and Renting Homes brings this vision to life by incorporating a wide range of features and functionalities that are tailor-made to enhance and simplify the entire real estate process.

This introduction sets the stage for an exploration of the key features and capabilities of this application, demonstrating how it empowers individuals, real estate professionals, and the industry as a whole. With an emphasis on efficiency, security, and accessibility, this application redefines the way we approach buying, selling, and renting homes, making the process not just more manageable but also more enjoyable. Let us delve deeper into the realms of this real estate revolution, uncovering the myriad ways in which it elevates and streamlines real estate transactions, benefiting everyone involved.

The decision to buy a home is one of life's pivotal moments. It represents a vision of the future, a place to call your own, and an investment in a haven where you can create lasting memories. Whether you're a first-time buyer, looking to upgrade, or seeking an investment property, the process involves careful considerations. From defining your needs and budget to navigating the intricacies of mortgages and negotiations, the journey of buying a home is a thrilling adventure, and we're here to guide you every step of the way.

Selling a home is often a bittersweet endeavour, marked by nostalgia for cherished memories and anticipation for new beginnings. It's a journey of presenting your property in the best light, setting a competitive price, and connecting with the right buyer. Our comprehensive guide will help you navigate the nuances of home selling, from prepping your property and marketing it effectively to negotiating offers and handling the intricacies of closing the deal.

Problem Identified:

In the realm of real estate applications that cater to buying, selling, and renting homes, several common challenges and issues have been identified that can hinder the user experience and overall functionality of the platform. Here are some of the key problems:

Limited Property Listings: Many real estate applications suffer from limited property listings, which can lead to frustration for users who are unable to find a suitable property that matches their criteria. This is particularly problematic in highly competitive housing markets. Outdated Information: Stale or outdated property information and listings can lead to wasted time and missed opportunities for users. Keeping property listings current is a significant challenge for these applications. Lack of Transparency: Some applications lack transparency in terms of pricing, property history, and hidden fees, leading to a lack of trust and hesitation among buyers, sellers, and renters. Inadequate Search and Filter Options: Inefficient search and filtering options can make it challenging for users to find properties that meet their specific requirements, leading to a frustrating user experience.

Security and Privacy Concerns: Real estate transactions involve sensitive financial and personal information. Applications must ensure robust security measures to protect user data and privacy throughout the buying, selling, and renting processes. Ineffective Communication: Poor communication tools and a lack of real-time updates can result in missed opportunities, delays in the negotiation process, and overall dissatisfaction among users. Complex Legal Procedures: Real estate transactions often involve complex legal procedures and documentation. A lack of support or guidance in this area can create significant obstacles for users. Accessibility and Mobile Integration: Some real estate applications are not optimized for mobile use, limiting accessibility for users who prefer to search for properties on their smartphones or tablets.

Market Information: Users often require access to comprehensive market data and trends to make informed decisions about buying, selling, or renting properties. Insufficient market information can lead to missed opportunities and financial decisions based on incomplete data. Inadequate Support for Real Estate Professionals: Real estate agents and property managers need tools to manage their listings and client interactions effectively. A lack of features catering to their needs can hinder their success within the platform. Addressing these identified problems is crucial for enhancing the user experience and providing a more effective, efficient, and trustworthy real estate application for buying, selling, and renting homes. Solutions to these challenges can help users make informed decisions, save time, and have greater confidence in their real estate transactions.

Scope of the project:

Property Listings and Search: The application should provide an extensive database of residential properties available for sale or rent, covering various property types, locations, and price ranges. Advanced search and filter options should be available to help users find properties that meet their specific criteria.

User Accounts and Profiles: Users should be able to create and manage personal profiles with preferences, saved searches, and favorite listings. Notifications and alerts can keep users updated about new listings and price changes.

Property Management for Sellers: Property owners and sellers should have tools to list their properties, including the ability to add property details, images, and pricing information. Integration with property valuation and inspection services may be included for accurate property representation.

Financial Tools: Mortgage calculators and affordability estimators should help buyers assess their financial readiness and affordability. Real-time market data and trends should be provided to assist users in making informed decisions.

Communication Tools: In-app messaging and chat functionality should enable direct communication between buyers, sellers, renters, and real estate agents. Video integration can facilitate virtual property tours and remote meetings.

Real Estate Professionals: Real estate agents and professionals should have tools to manage their listings and client interactions effectively. Integration with Multiple Listing Services (MLS) may be included to broaden property exposure.

1.2 Objective of the Project

1. To Create an intuitive and user-friendly interface that allows users to easily navigate and search for properties.

2. To Ensure the application offers a vast and up-to-date database of properties, including homes for sale and rent, with detailed descriptions, high-quality images, and accurate information.

3. To Implement advanced search and filtering options, enabling users to refine property searches based on their specific criteria, such as location, price, size, and amenities.

4. To Create user profiles that allow buyers, sellers, and renters to save their preferences, track property listings.

5. To Provide messaging that facilitate communication between buyers, sellers, and real estate agents within the app.

6. To Integrate mapping and geolocation services to help users visualize property locations.

CHAPTER 2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

1. **Zillow:** Zillow is a popular real estate platform in the United States. It offers an extensive database of property listings, including homes for sale and rent. Users can search by location, price, property type, and more. The platform also provides home value estimates.
2. **Trulia:** Trulia is another well-known real estate platform, acquired by Zillow Group. It offers a user-friendly interface for home buyers, sellers, and renters to search for properties and gather information.
3. **Redfin:** Redfin is a real estate brokerage that provides an online platform for buying and selling homes. They offer a full-service experience, including home tours, online transaction management, and competitive commission rates.
4. **Realtor.com:** Realtor.com is a comprehensive platform that lists both residential and commercial properties for sale and rent. It offers in-depth property details, neighbourhood information, and a mobile app for users on the go.
5. **Apartments.com:** Apartments.com primarily focuses on rental listings, making it a popular choice for renters looking for apartments or houses to rent. It includes various search filters and resources for renters.
6. **LoopNet:** LoopNet specializes in commercial real estate, making it a valuable resource for businesses searching for office spaces, warehouses, retail locations, and more.

DISADVANTAGE:

1. **Limited Geographic Coverage**: Many real estate apps are region-specific or focused on a few key markets. This can be a disadvantage for users looking to buy, sell, or rent properties in areas where the platform has limited coverage.
2. **Inaccurate Property Information:** Inaccurate or outdated property information is a common issue. Listings may not reflect the current status of properties, leading to frustration for users.
3. **Lack of Transparency:** Users might find a lack of transparency regarding property history, pricing trends, and the negotiation process, which can create distrust and uncertainty.

4. **Privacy Concerns:** Sharing personal information, including contact details, with various parties in the real estate transaction process can raise privacy concerns for users.
5. **Limited Property Media:** Not all listings include high-quality images, videos, or virtual tours. This can make it challenging for users to visualize the property accurately.
6. **Inconsistent User Experience:** Some applications may have inconsistent user interfaces or features across different platforms (web, mobile, desktop), causing confusion and a less-than-optimal user experience.

2.2 PROPOSED SYSTEM

KEY FEATURES:

1. **User-Friendly Interface:** RealPro boasts an intuitive and user-friendly interface, making it accessible for users of all ages and tech-savviness.
2. **Listing Search:** Users can search for properties based on various criteria such as location, price range, property type, and more. They can also save their favourite listings for future reference.
3. **Advanced Search Filters:** RealPro provides advanced filters for users to narrow down their search results, including features like the number of bedrooms, bathrooms, and square footage.
4. **High-Quality Images:** Listings come with high-resolution images, enabling users to explore properties from the comfort of their homes.
5. **Agent Finder:** Users can find experienced real estate agents in their desired location to guide them through the buying, selling, or renting process.
6. **Secure Messaging:** RealPro offers a secure messaging system that allows users to communicate with agents, property owners, and other interested parties.
7. **Pricing Analytics:** RealPro employs data analytics to provide users with accurate market insights, including price trends and property value estimates.

ADVANTAGES:

1. **Comprehensive Property Listings:** Users can easily access a vast database of property listings, including high-quality images, 3D tours, and detailed information, enabling them to make well-informed decisions.

2. **Efficient Search and Filtering:** The system offers advanced search filters, allowing users to quickly find properties that match their specific criteria, such as location, price, property type, and more.
3. **Agent Support:** RealPro connects users with experienced real estate agents who can provide expert guidance and assistance throughout the buying, selling, or renting process.
4. **Secure Messaging:** The secure messaging system within the application ensures safe and efficient communication between users, agents, and property owners.
5. **Property Comparison:** Users can compare multiple properties side by side, simplifying the decision-making process.
6. **Mobile Accessibility:** RealPro is available on both iOS and Android platforms, catering to users who prefer to use the application on their smartphones and tablets.

2.3 FEASIBILITY STUDY

A feasibility study for a real estate application focused on buying, selling, and renting homes is an essential step to assess the viability of such a venture. It helps determine whether your idea is worth pursuing and what challenges you might face. Here's a guide on how to conduct a feasibility study for your real estate application:

1. **Project Description:** Start by providing a detailed description of your real estate application. Explain its core features, objectives, and target audience. Specify that it's focused on buying, selling, and renting homes.
2. **Market Analysis:** Research the real estate market in your target location. Understand the demand for such an application, and assess the competition. Look at existing real estate apps and identify gaps you can fill.
3. **Target Audience:** Define your primary and secondary target audiences. Who are the potential users of your application? Are you targeting home buyers, sellers, renters, or a combination?

2.4 TECHNICAL FEASIBILITY

The technical feasibility of a real estate application for buying, selling, and renting homes is a critical aspect to consider in your feasibility study. It involves evaluating whether the proposed technology and infrastructure are capable of supporting the application's development, maintenance, and scalability. Here are key factors to consider:

- Assess the technology stack needed to develop the application. This includes choosing the programming languages, frameworks, databases, and other tools. Ensure that the selected stack aligns with your project's requirements and is supported by a skilled development team.
- Consider the handling and storage of property listings, user data, images, and other information. Ensure that the application can efficiently manage and retrieve data, and that you have a strategy for database scaling as the user base grows.
- Evaluate the application's ability to scale with a growing user base. This includes considering server scalability, load balancing, and other infrastructure requirements. Cloud-based solutions like AWS, Azure, or Google Cloud can provide scalability benefits.
- Determine if the application will be available on multiple platforms, such as web and mobile (iOS and Android). Ensure that the development team can handle cross-platform development or, if needed, separate native app development.

2.5 ECONOMICAL FEASIBILITY

Economic feasibility is an important aspect to consider when developing a real estate application for buying, selling, and renting homes. Evaluating the economic feasibility of such an application involves assessing the potential costs, revenue streams, and market dynamics. Here are some key factors to consider:

- Conduct thorough market research to understand the demand for a real estate application in your target market. Analyse the size of the real estate industry, the level of competition, and the specific needs of homebuyers, sellers, and renters.
- Estimate the initial development costs, including software development, design, and infrastructure. Consider ongoing operational expenses such as maintenance, server costs, and customer support.
- Evaluate the technology infrastructure required to support the application, including hosting, scalability, and data security. Consider the costs of data storage and server maintenance.
- Calculate the expected return on investment for your real estate application. Assess the time it will take to break even and start generating profit.

2.6 OPERATIONAL FEASIBILITY

Operational feasibility is a critical aspect of assessing the viability of a real estate application for buying, selling, and renting homes. This feasibility study focuses on whether the application can be effectively developed, implemented, and operated. Here are key considerations for operational feasibility:

- Evaluate the technical infrastructure and requirements needed to develop and maintain the application. This includes hardware, software, databases, and other technology components. Ensure that you have access to or can acquire the necessary technical resources.
- Identify the sources of real estate data, including property listings, market trends, and user information. Ensure that you can access and integrate this data into your application effectively and legally.
- Define the user experience and interface design. Ensure that the application is user-friendly and provides an intuitive experience for both buyers, sellers, and renters. Conduct usability testing to refine the design.
- Assess the security measures required to protect user data, financial transactions, and sensitive information. Comply with data privacy regulations and implement robust security protocols.
- Plan for ongoing maintenance, updates, and customer support. Users will expect regular updates and responsive support, which are essential for the long-term success of the application.
- Estimate the ongoing operational costs, including server hosting, customer support, marketing, and software maintenance. Ensure that your revenue model can cover these expenses and generate profit.

CHAPTER 3
SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS:

❖ SYSTEM	:	INTEL(R) CORE(TM) 1.60GHZ
❖ RAM	:	8GB
❖ INTERNAL STORAGE	:	256GB

3.2 SOFTWARE REQUIREMENTS:

❖ OPERATING SYSTEM	:	Android 10
❖ SOFTWARE	:	ANDROID STUDIO
❖ FRAMEWORK	:	FLUTTER
❖ PROGRAMMING LANGUAGE	:	DART
❖ BACKEND	:	FIREBASE

CHAPTER 4
SOFTWARE DESCRIPTION

4.1 FRONT END

4.1.1 FLUTTER

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform application for Android, IOS, Linux, macOS, windows and the web from a single codebase.

The first version of Flutter was known as "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, the last major release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first stable version of the framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

On May 6, 2020, the Dart software development kit (SDK) version 2.8 and Flutter 1.17.0 were released, adding support for the Metal API which improves performance on iOS devices by approximately 50%, as well as new Material widgets and network tracking development tools.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with a new Canvas Kit renderer and web specific widgets, early-access desktop application support for Windows, macOS, and Linux and improved Add-to-App APIs. This release also utilized Dart 2.0 that featured sound null-safety, which caused many breaking changes and issues with many external packages; however, the Flutter team included instructions and tools to mitigate these issues.

Flutter is a simple and high-performance framework based on Dart language, provides high performance by rendering the UI directly in the operating system's canvas rather than through native frame work.

Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML.To be specific, Flutter application is itself a widget. Flutter widgets also supports animations and gestures. The application logic is based on reactive programming. Widget may optionally have a state. By changing the state of the widget, Flutter will automatically (reactive programming) compare the widget's state (old and new) and render the widget with only the necessary changes instead of re-rendering the whole widget.

Features of Flutter

- Modern and reactive framework.
- Uses Dart programming language and it is very easy to learn.
- Fast development.
- Beautiful and fluid user interfaces.
- Huge widget catalog.
- Runs same UI for multiple platforms.
- High performance application

DART

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

Languages are defined by their technical envelope-the choices made during development that shape the capabilities and strengths of a language. Dart is designed for a technical envelope that is particularly suited to client development, prioritizing both development (sub-second stateful hot reload) and high-quality production experiences across a wide variety of compilation targets (web, mobile, and desktop).

Dart also forms the foundation of Flutter. Dart provides the language and runtimes that power Flutter apps, but Dart also supports many core developer tasks like formatting, analysing, and testing code.

Dart: The language

The Dart language is type safe; it uses static type checking to ensure that a variable's value always matches the variable's static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during experimentation or for code that needs to be especially dynamic.

Dart offers sound null safety, meaning that values can't be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable is always non-nullable. If you inspect your running code in the debugger, you'll see that non-nullability is retained at runtime (hence sound null safety).

Dart: The libraries

Dart has a rich set of core libraries, providing essentials for many everyday programming tasks:

- Built-in types, collections, and other core functionality for every Dart program (dart:core)
- Richer collection types such as queues, linked lists, hashmaps, and binary trees (dart:collection)
- Encoders and decoders for converting between different data representations, including JSON and UTF-8 (dart:convert)
- Mathematical constants and functions, and random number generation (dart:math)
- File, socket, HTTP, and other I/O support for non-web applications (dart:io)
- Support for asynchronous programming, with classes such as Future and Stream (dart:async)
- Lists that efficiently handle fixed-sized data (for example, unsigned 8-byte integers) and SIMD numeric types (dart:typed_data)
- Foreign function interfaces for interoperability with other code that presents a C-style interface (dart:ffi)
- Concurrent programming using isolates—independent workers that are similar to threads but don't share memory, communicating only through messages (dart:isolate)
- HTML elements and other resources for web-based applications that need to interact with the browser and the Document Object Model (DOM) (dart:html)

Dart: The platforms

Dart's compiler technology lets you run code in different ways:

Native platform: For apps targeting mobile and desktop devices, Dart includes both a Dart VM with just-in-time (JIT) compilation and an ahead-of-time (AOT) compiler for producing machine code.

Web platform: For apps targeting the web, Dart includes both a development time compiler (dartdevc) and a production time compiler (dart2js). Both compilers translate Dart into JavaScript.

The Flutter framework is a popular, multi-platform UI toolkit that's powered by the Dart platform, and that provides tooling and UI libraries to build UI experiences that run on iOS, Android, macOS, Windows, Linux, and the web.

Dart Native (machine code JIT and AOT)

During development, a fast developer cycle is critical for iteration. The Dart VM offers a just-in-time compiler (JIT) with incremental recompilation (enabling hot reload), live metrics collections (powering DevTools), and rich debugging support.

When apps are ready to be deployed to production—whether you're publishing to an app store or deploying to a production backend—the Dart AOT compiler enables ahead- of-time compilation to native ARM or x64 machine code. Your AOT-compiled app launches with consistent, short startup time.

The AOT-compiled code runs inside an efficient Dart runtime that enforces the sound Dart type system and manages memory using fast object allocation and a generational garbage collector.

4.2 BACK END

4.2.1 FIREBASE

Cloud Functions for Firebase is a serverless framework that lets you automatically run backend code in response to events triggered by Firebase features and HTTPS requests. Your JavaScript or TypeScript code is stored in Google's cloud and runs in a managed environment. There's no need to manage and scale your own servers.

Firebase developers can easily integrate with external services for tasks like processing payments and sending SMS messages. Also, developers can include custom logic that is either too heavyweight for a mobile device, or which needs to be secured on a server.

Cloud Functions for Firebase is optimized for Firebase developers:

- Firebase SDK to configure your functions through code
- Integrated with Firebase Console and Firebase CLI
- The same triggers as Google Cloud Functions, plus Firebase Realtime Database, Firebase Authentication, and Firebase Analytics triggers

Build powerful apps. Spin up your backend without managing servers. Effortlessly scale to support millions of users with Firebase databases, machine learning infrastructure, hosting and storage solutions, and Cloud Functions.

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through real time listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

For developers that need a more full-featured backend, Cloud Functions provides a gateway to the powerful capabilities in Google Cloud Platform.

Flexibility

The Cloud Firestore data model supports flexible, hierarchical data structures. Store your data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.

Expressive querying

In Cloud Firestore, you can use queries to retrieve individual, specific documents or to retrieve all the documents in a collection that match your query parameters. Your queries can include multiple, chained filters and combine filtering and sorting. They're also indexed by default, so query performance is proportional to the size of your result set, not your data set.

Realtime updates

Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.

Offline support

Cloud Firestore caches data that your app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, Cloud Firestore synchronizes any local changes back to Cloud Firestore.

Designed to scale

Cloud Firestore brings you the best of Google Cloud's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations, and real transaction support. We've designed Cloud Firestore to handle the toughest database workloads from the world's biggest apps.

CHAPTER 5
PROJECT DESCRIPTION

5.1 OVERVIEW OF THE PROJECT

The Real Estate Application for Buying, Selling, and Renting Homes is a comprehensive software application designed to facilitate the real estate industry, helping individuals, agents, and agencies to buy, sell, or rent residential properties with ease. This application leverages technology to streamline the real estate process, making it more accessible and efficient for both buyers and sellers.

User Registration and Profiles:

- Users can create accounts and set up profiles.
- Profiles may include personal information, contact details, and preferences.

Property Listings:

- Sellers and real estate agents can create property listings with detailed information.
- Property listings include photos, descriptions, price, location, and property type.

Search and Filters:

- Users can search for properties based on various criteria like location, price range, property type, and more.
- Advanced search filters for more precise results.

Property Details:

- Detailed information about each property, including images, floor plans, features, and nearby amenities.

Messaging and Communication:

- In-app messaging system for users to communicate with sellers or agents.
- Notifications for messages and property updates.

Save and Compare:

- Users can save properties of interest and compare them side by side.
- A favourites section for easy access to saved listings.

Maps and Location Services:

- Integration with maps to display property locations.

- Nearby places and services (schools, hospitals, shopping, etc.) can be shown.

Reviews and Ratings:

- Users can leave reviews and ratings for properties they have interacted with.
- Provides valuable feedback for both buyers and sellers.

Mortgage Calculator:

- An integrated mortgage calculator to estimate monthly payments based on property price and interest rates.

User Authentication and Security:

- Secure login and authentication protocols to protect user data.

Admin Dashboard:

- An administrative dashboard for managing user accounts, property listings, and resolving disputes.

Property Verification:

- A system for verifying property listings to ensure accuracy and authenticity.

Payment Gateway:

- Integration with a payment gateway for online transactions and fees.

Mobile Compatibility:

- Compatibility with mobile devices to access listings and features on the go.

Legal Compliance:

- Ensuring the application complies with local real estate regulations and laws.

Benefits:

- Simplifies the real estate buying, selling, and renting process.
- Provides a centralized platform for property listings and transactions.
- Offers a more efficient and convenient way to find, explore, and connect with properties

5.2 DESCRIPTION OF MODULES

5.2.1 Onboarding page

This page show case the features of the app.

5.2.2 Log in module

This page is used to get the email id form the user.

5.2.3 Dashboard

Access to various administrative tools and features.

5.2.4 Your Places

A list of saved properties is displayed, including property details and images.

5.2.5 Profile

This page is used to showcase the user details like profile image, name etc.

5.2.6 Manage Places

This is the page where the user chat with their friends.

5.2.7 DCL View

It's challenging to provide a comprehensive explanation of its functionality.

5.2.8 Map View

Users can search for properties within a specific area by entering location names, ZIP codes, or drawing custom search boundaries on the map.

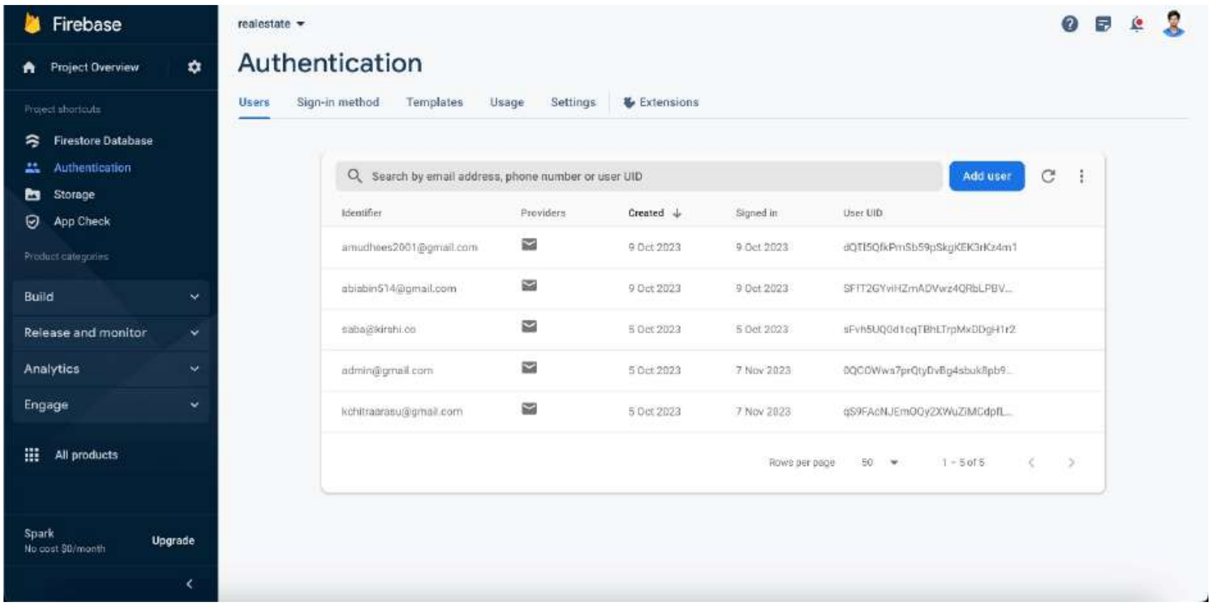
5.2.9 Filter Module

Users can set a price range to filter properties within their budget. Filters to specify the desired number of bedrooms and bathrooms.

DATA BASE

The database design plays a major role in the buildings of a project and also the table design is one of the important phases of a project

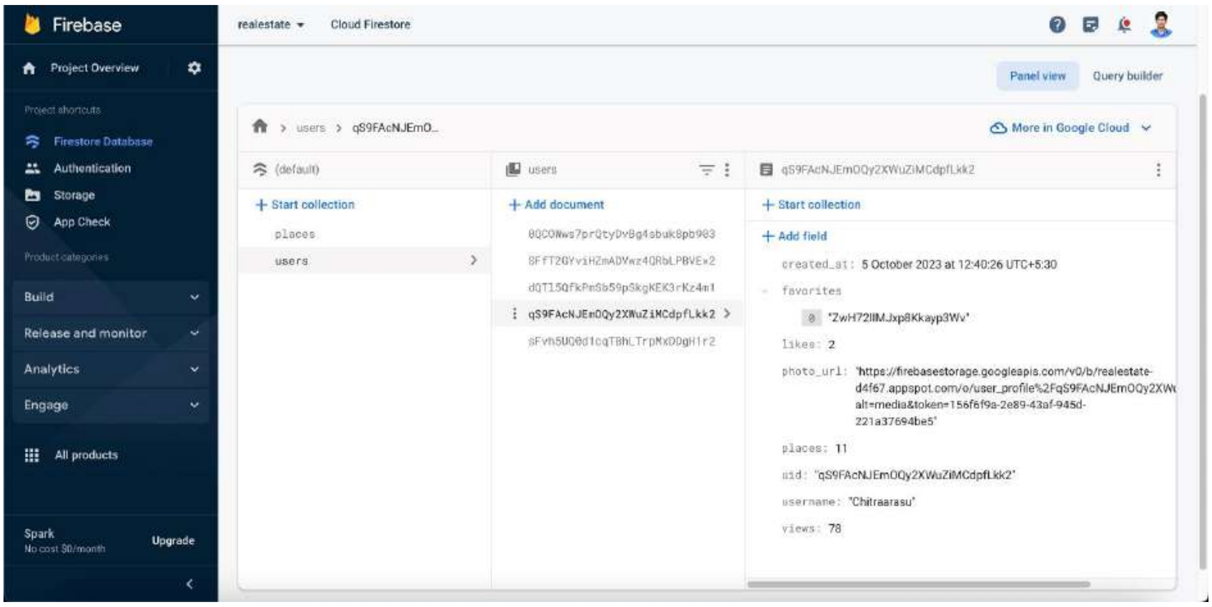
1. AUTHENTICATION



Identifier	Providers	Created	Signed in	User UID
amudhees2001@gmail.com	✉	9 Oct 2023	9 Oct 2023	dQTI5QfkPmSb59pSkgKEK3rKz4m1
abiabin514@gmail.com	✉	9 Oct 2023	9 Oct 2023	SFfT2GYviHZmADVwz4QRbLPBV...
saba@kirshi.co	✉	5 Oct 2023	5 Oct 2023	sFvh5UQ0d1cqTBhLTrpMxDDgH1r2
admin@gmail.com	✉	5 Oct 2023	7 Nov 2023	0QCOWws7prQtyDvBg4sbuk8pb9...
kchitraarasu@gmail.com	✉	5 Oct 2023	7 Nov 2023	qS9FAcNJEmOQy2XWuZiMCdpfL...

Rows per page 50 1 – 5 of 5

2.USER DATA



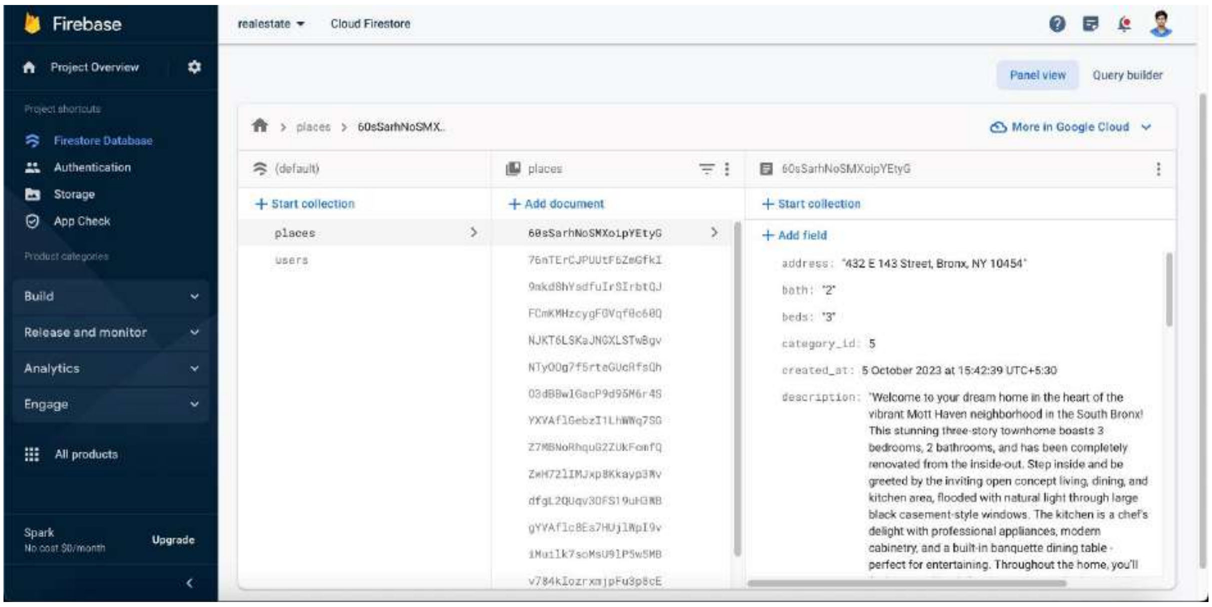
(default): places, users

users: 0QCOWws7prQtyDvBg4sbuk8pb903, SFfT2GYviHZmADVwz4QRbLPBVEx2, dQTI5QfkPmSb59pSkgKEK3rKz4m1, qS9FAcNJEmOQy2XWuZiMCdpfLkk2, sFvh5UQ0d1cqTBhLTrpMxDDgH1r2

qS9FAcNJEmOQy2XWuZiMCdpfLkk2

```
created_at: 5 October 2023 at 12:40:26 UTC+5:30
favorites
    0 "ZwH72lIMJxp8Kkayp3Wv"
likes: 2
photo_url: "https://firebasestorage.googleapis.com/v0/b/realestate-
           d4f67.appspot.com/o/user_profile%2FqS9FAcNJEmOQy2XW
           alt=media&token=156f6f9a-2e89-43af-945d-
           221a37694be5"
places: 11
uid: "qS9FAcNJEmOQy2XWuZiMCdpfLkk2"
username: "Chitraarasu"
views: 78
```

3.PLACES

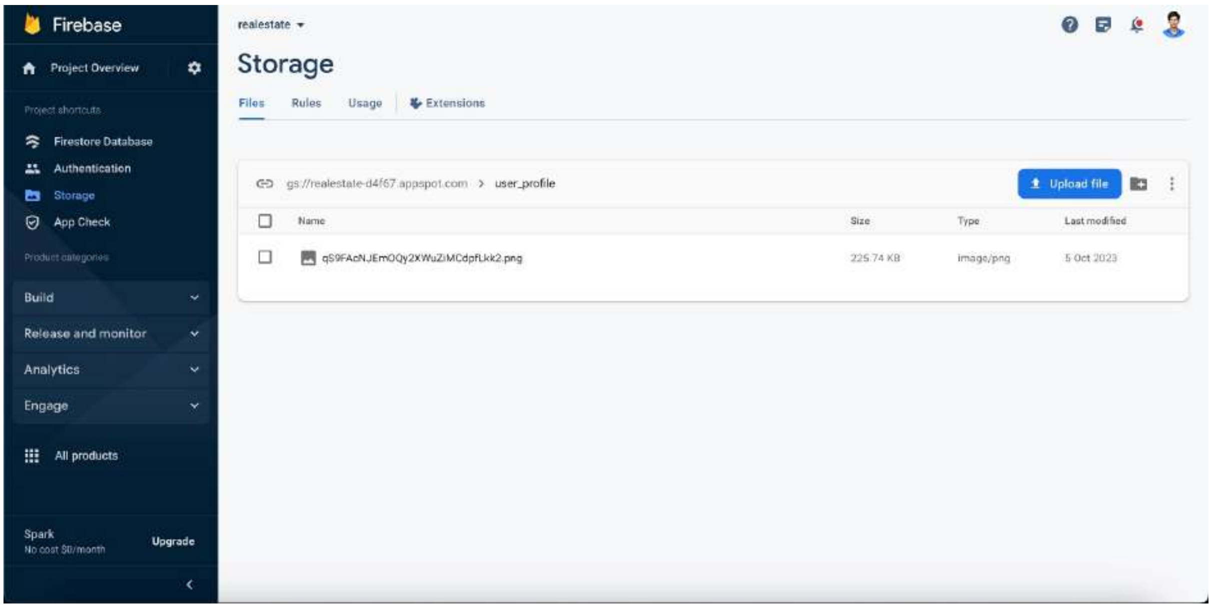


places
users

60sSarhNoSMXoipYEtyG
76nTErCJPUUtF6ZmGfkI
9mkd8hYsdfuIrSIrbtQJ
FCmKMHzcygFGVqf8c60Q
NJKT6LSKaJNGXLSTwBgv
NTyOOg7f5rteGUcRfsQh
O3dB8wlGacP9d95M6r4S
YXVAflGebzI1LhWWq7SG
Z7MBNoRhquG2ZUkFomfQ
ZwH72lIMJxp8Kkayp3Wv
dfgL2QUqv3OFS19uHGWB
gYVAflc8Es7HUjlWpI9v
iMuilk7soMsU9lP5w5MB
v784kIozrxmjpFu3p8cE

```
address:  "432 E 143 Street, Bronx, NY 10454"
bath:  "2"
beds:  "3"
category_id:  5
created_at:  5 October 2023 at 15:42:39 UTC+5:30
description:  "Welcome to your dream home in the heart of the
               vibrant Mott Haven neighborhood in the South Bronx!
               This stunning three-story townhome boasts 3
               bedrooms, 2 bathrooms, and has been completely
               renovated from the inside-out. Step inside and be
               greeted by the inviting open concept living, dining, and
               kitchen area, flooded with natural light through large
               black casement-style windows. The kitchen is a chef's
               delight with professional appliances, modern
               cabinetry, and a built-in banquette dining table -
               perfect for entertaining. Throughout the home, you'll
```

4.STORAGE

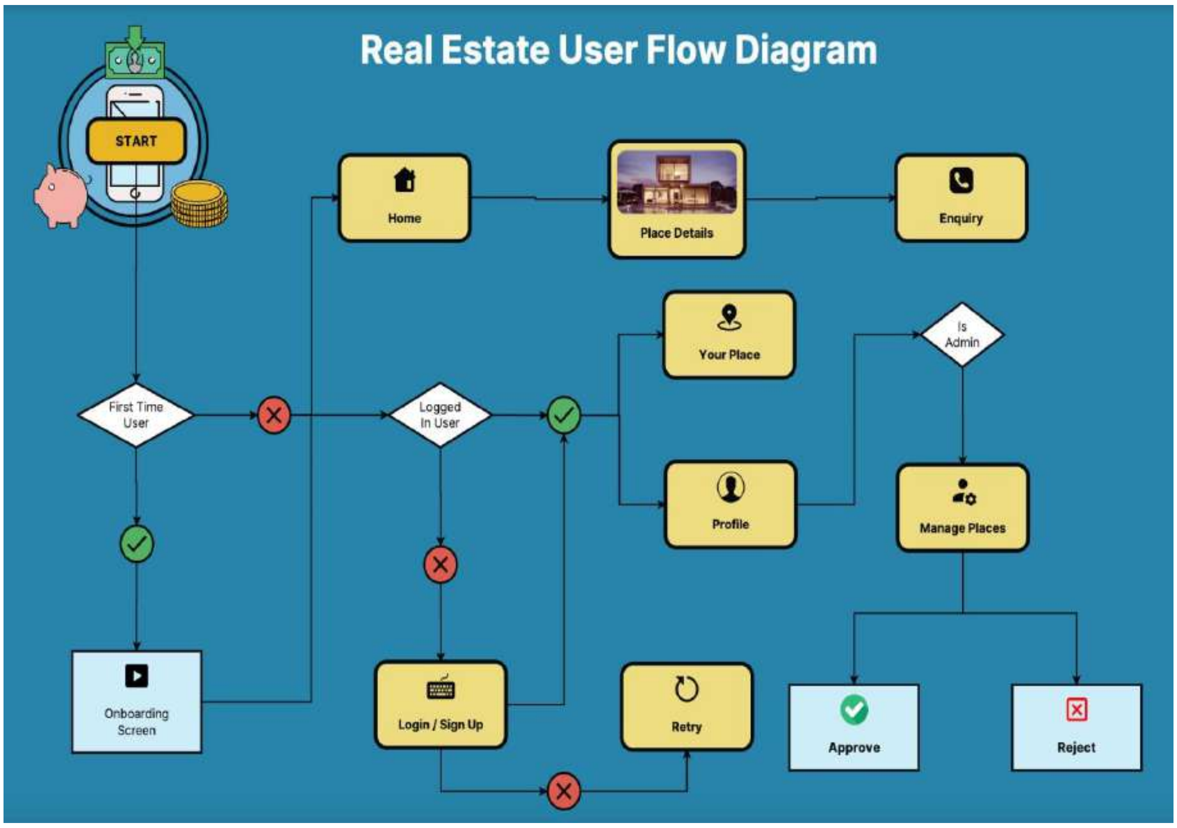


gs://realestate-d4f67.appspot.com > user_profile

Name	Size	Type	Last modified
qS9FAcNJEmOQy2XWuZiMCdpfLkk2.png	225.74 KB	image/png	5 Oct 2023

5.3 SYSTEM DESIGN

5.3.1 SYSTEM ARCHITECTURE



CHAPTER 6
SYSTEM TESTING

6.1 TESTING

Software testing is an important element of Software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of S/W as a system element and the costs associated with Software failure are motivating forces for well planned, through testing.

Though the test phase is often thought of as separate and distinct from the development effort--first develop, and then test--testing is a concurrent process that provides valuable information for the development team.

There are at least three options for integrating Project Builder into the test phase:

- Testers do not install Project Builder, use Project Builder functionality to compile and source-control the modules to be tested and hand them off to the testers, whose process remains unchanged.
- The testers import the same project or projects that the developers use.
- Create a project based on the development project but customized for the testers (for example, it does not include support documents, specs, or source), who import it.

6.1.1 TESTING OBJECTIVES

There are several rules that can serve as testing objectives. They are

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.
- If testing is conducted successfully according to the objectives stated above, it will uncover errors in the software.

6.1.2 UNIT TESTING

This is the first level of testing. In this different module are tested against the specifications produced during the design of the module. During this testing the number of arguments is compared to input parameters, matching of parameter and arguments etc. It is also ensured whether the file attributes are correct, whether the Files are opened before using, whether Input/output errors are handled etc. Unit Test is conducted using a Test Driver usually.

Before We performed the unit testing, we are satisfied about 80% after that We have satisfied with our project of 100%.

6.1.3 VALIDATION TESTING

This provides the final assurance that the software meets all functional, behavioural and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in which the user expects.

CHAPTER - 7
SYSTEM IMPLEMENTATION

The project is developed by using FLUTTER frame work as the frontend and FIREBASE as the backend. Once the system has been designed, the next step is to convert the designed one into actual code, so as to satisfy the user requirements as excepted. If the system is approved to be error free it can be implemented.

When the initial design was done for the system, the department was consulted for acceptance of the design so that further processing of the system development can be carried on. After the development of the system, a demonstration was given to them about working of the system. The aim of the system illustration was to identify any malfunctioning of the system.

Implementation includes proper training to end users. The implemented software should be maintained for prolonged running of the software.

Initially the system was run parallel with manual system. The system has been tested with data and has proved to error-free and user-friendly. Training was given to end user about the software and its features. Thus, the project is running without any errors and warnings.

CHAPTER - 8
CONCLUSION & FUTURE ENHANCEMENT

8.1 CONCLUSION

The Real Estate Application for Buying, Selling, and Renting Homes represents a significant advancement in the real estate industry, offering a user-friendly, efficient, and comprehensive platform for property transactions. This application has addressed the needs of various stakeholders, including homebuyers, sellers, real estate agents, and property managers, by streamlining the process of buying, selling, and renting homes. The key features and modules implemented have significantly improved the user experience and added value to the real estate market.

The application's user registration and profiles enable personalized experiences, while property listings provide comprehensive information about available homes. The search and filter functionalities simplify the property search process, helping users find the properties that align with their preferences. Features like messaging, reviews, and ratings promote communication and trust within the community.

The "Dashboard Module" allows for efficient administration and management of the application, ensuring data integrity, security, and compliance with regulations. Additionally, the "Your Places" and "Map View" modules provide users with tools to manage their property-related activities and visualize property locations on an interactive map.

The "Filter Module" enhances the search experience by allowing users to refine their property searches based on specific criteria and preferences, ultimately saving time and effort.

8.2 FUTURE ENHANCEMENT

To further improve the Real Estate Application and stay competitive in the evolving real estate industry, the following enhancements could be considered:

1. **Augmented Reality (AR) Integration:** Implement AR features to allow users to visualize properties and their interiors in real-time through their smartphones, providing an immersive experience.
2. **Machine Learning and AI:** Use machine learning algorithms to provide personalized property recommendations based on user behaviour, preferences, and historical data.

3. **Expanded Payment Options:** Offer additional payment methods and cryptocurrency options to enhance flexibility in transactions.
4. **Virtual Property Tours:** Develop virtual property tours or 3D property modelling to give users a comprehensive view of properties without physical visits.
5. **Rental Property Management:** Expand the application's capabilities to support property management for landlords and property managers.

CHAPTER 9

APPENDIX – A

9.1 SOURCE CODE

pubspec.yaml

```
name: real_estate
description: A new Flutter project.
publish_to: 'none'
version: 1.0.0+1

environment:
  sdk: '>=3.1.2 <4.0.0'

dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.2
  introduction_screen: ^3.0.0
  get: ^4.6.1
  lottie: ^2.4.0
  country_pickers: ^2.0.0
  get_storage: ^2.1.1
  firebase_core: ^2.13.1
  firebase_auth: ^4.6.2
  firebase_storage: ^11.2.7
  cloud_firestore: ^4.9.2
  fluttertoast: ^8.2.2
  flutter_zoom_drawer: ^3.1.1
  animated_toggle_switch: ^0.8.0
  flutter_cached_pdfview: ^0.4.2
  flutter_map: ^5.0.0
  latlong2: ^0.9.0
  file_picker: ^5.5.0
  geolocator: ^10.1.0
  loading_indicator: ^3.1.1
  carousel_slider: ^4.2.1
  maps_launcher: ^2.2.0
  url_launcher: ^6.1.14
  flutter_image_compress: ^2.0.4
  avatar_glow: ^2.0.2
  speech_to_text: ^6.3.0
  flutter_native_splash: ^2.3.3
  path_provider: ^2.1.1


dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^2.0.0
  flutter_launcher_icons: ^0.13.1
  package_rename: ^1.5.1

flutter:

  uses-material-design: true

  assets:
    - assets/
    - assets/images/
    - assets/animations/
    - assets/sounds/speech_to_text_listening.m4r
    - assets/sounds/speech_to_text_cancel.m4r
    - assets/sounds/speech_to_text_stop.m4r
```

```
fonts:
    - family: Nunito
      fonts:
        - asset: assets/fonts/Nunito-Black.ttf
        - asset: assets/fonts/Nunito-Bold.ttf
        - asset: assets/fonts/Nunito-ExtraBold.ttf
        - asset: assets/fonts/Nunito-ExtraLight.ttf
        - asset: assets/fonts/Nunito-Light.ttf
        - asset: assets/fonts/Nunito-Medium.ttf
        - asset: assets/fonts/Nunito-Regular.ttf
        - asset: assets/fonts/Nunito-SemiBold.ttf
```

Main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_native_splash/flutter_native_splash.dart';
import 'package:get/get.dart';
import 'package:get_storage/get_storage.dart';
import 'package:real_estate/screens/dashboard/dashboard.dart';
import 'package:real_estate/utils/manager/font_manager.dart';
import 'package:real_estate/utils/resizer/fetch_pixels.dart';

import 'controller/binder.dart';
import 'firebase_options.dart';
import 'screens/onboarding/onboarding_page.dart';

void main() async {
  final binding = WidgetsFlutterBinding.ensureInitialized();
  FlutterNativeSplash.preserve(widgetsBinding: binding);
  await GetStorage.init();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  SystemChrome.setPreferredOrientations(
    [
      DeviceOrientation.portraitDown,
      DeviceOrientation.portraitUp,
    ],
  );
  Future.delayed(const Duration(seconds: 1), () {
    FlutterNativeSplash.remove();
  });
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  GetStorage box = GetStorage();

  @override
  Widget build(BuildContext context) {
    FetchPixels(context);

    return GetMaterialApp(
      debugShowCheckedModeBanner: false,
      title: "Real Estate",
      initialBinding: Binder(),
      theme: ThemeData(
        primarySwatch: Colors.blue,
        fontFamily: fontNunito,
      ),
      home: (box.read("isSkipped") ?? false) ? Dashboard() : OnBoardingPage(),
    );
  }
}
```

OnboardingPage.dart

```
import ‘package:flutter/material.dart’;
import ‘package:get/get.dart’;
import ‘package:get_storage/get_storage.dart’;
import ‘package:introduction_screen/introduction_screen.dart’;
import ‘package:real_estate/screens/dashboard/dashboard.dart’;
import ‘package:real_estate/utils/c_extensions.dart’;

import ‘../../widget/widget_utils.dart’;

class OnBoardingPage extends StatelessWidget {
  OnBoardingPage({Key? Key}) : super(key: key);

  GetStorage box = GetStorage();

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: const BoxDecoration(
        color: Colors.white,
      ),
      child: SafeArea(
        child: IntroductionScreen(
          pages: [
            PageViewModel(
              title: ‘Sell a home’,
              body:
                  ‘Whether you choose traditional listing or explore innovative selling methods,
we\’re here to guide you towards a successful home sale.’,
              image: buildImage(‘sale’.png),
              decoration: getPageDecoration(),
            ),
            PageViewModel(
              title: ‘Buy a home’,
              body:
                  ‘Discover your perfect home with us, where comfort meets your vision of a
dream property.’,
              image: buildImage(‘buy’.png),
              decoration: getPageDecoration(),
            ),
            PageViewModel(
              title: ‘Rent a home’,
              body:
                  ‘Discover your next home to rent, where convenience and comfort await.’,
              image: buildImage(‘rent’.png),
              decoration: getPageDecoration(),
            ),
          ],
          done: Container(
            decoration: BoxDecoration(
              color: Colors.black,
              borderRadius: BorderRadius.circular(15),
            ),
            child: Padding(
              padding: EdgeInsets.all(14.0),
              child: getCustomFont(“Next”, 18, Colors.white, 1),
            ),
          ),
          onDone: () => goToPhoneNumberScreen(context),
          showSkipButton: true,
          skip: getCustomFont(“Skip”, 15, Colors.black, 1),
          onSkip: () => goToPhoneNumberScreen(context),
          skipStyle: ButtonStyle(
              foregroundColor: MaterialStateProperty.all(
            Colors.black,
          )),
          next: Container(
            decoration: BoxDecoration(
              color: Colors.black,
              borderRadius: BorderRadius.circular(15),
            ),
            child: const Padding(
              padding: EdgeInsets.all(14.0),
              child: Icon(
                Icons.arrow_forward_ios_rounded,
                color: Colors.white,
              ),
            ),
          ),
          dotsDecorator: getDotDecoration(),
          globalBackgroundColor: Colors.transparent,
        ),
      ),
    );
  }

  void goToPhoneNumberScreen(context) {
    box.write(“isSkipped”, true);
    Get.offAll(
      () => Dashboard(),
      transition: Transition.fadeIn,
    );
  }

  Widget buildImage(String path) {
    return Center(
      child: Image.asset(
        path,
        width: double.infinity,
      ),
    );
  }

  DotsDecorator getDotDecoration() => DotsDecorator(
        color: Colors.black87,
        activeColor: Colors.black,
        size: const Size(10, 10),
        activeSize: const Size(22, 10),
        activeShape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(24),
        ),
      );

  PageDecoration getPageDecoration() => PageDecoration(
        titleTextStyle: const TextStyle(
          fontSize: 28,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
        bodyTextStyle: const TextStyle(
          fontSize: 20,
          color: Colors.black,
        ),
        footerPadding: const EdgeInsets.all(16).copyWith(bottom: 0),
        imagePadding: const EdgeInsets.all(24),
      );
}
```

Binder.dart

```
import 'package:get/get.dart';
import 'package:real_estate/controller/route_controller.dart';
import 'package:real_estate/screens/home/vm_home.dart';
import 'package:real_estate/screens/your_places/vm_new_place.dart';
import '../screens/login/vm_login.dart';

class Binder extends Bindings {
  @override
  void dependencies() {
    Get.put(RouteController());

    Get.lazyPut(() => VMLogin(), fenix: true);
    Get.lazyPut(() => VMNewPlace(), fenix: true);
    Get.lazyPut(() => VMHome(), fenix: true);
  }
}
```

RouteController.dart

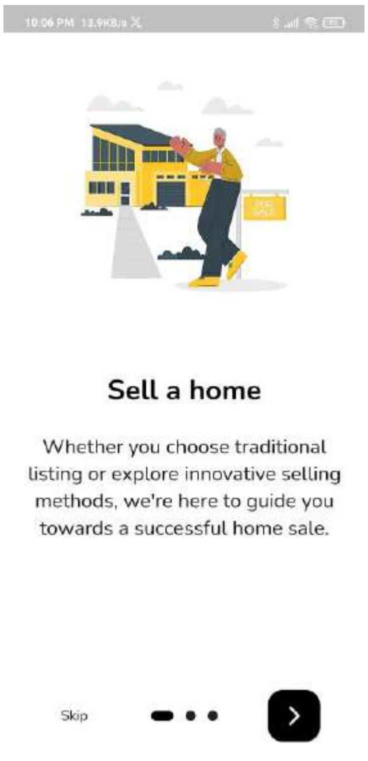
```
import 'package:flutter_zoom_drawer/flutter_zoom_drawer.dart';
import 'package:get/get.dart';

class RouteController extends GetxController {
  static RouteController to = Get.find();

  final zoomDrawerController = ZoomDrawerController();

  RxInt currentPos = RxInt(0);

  void toggleDrawer() {
    zoomDrawerController.open!();
    // update();
  }
}
```

Rest of the code available in this website : -

https://github.com/chitraarasu/real_estate_app

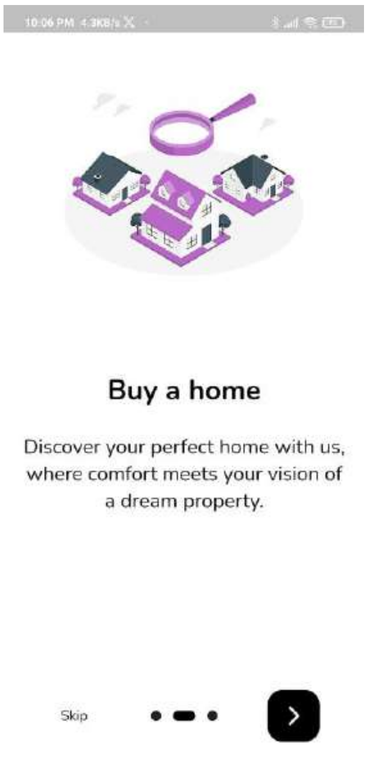
APPENDIX – B

9.2 OUTPUT



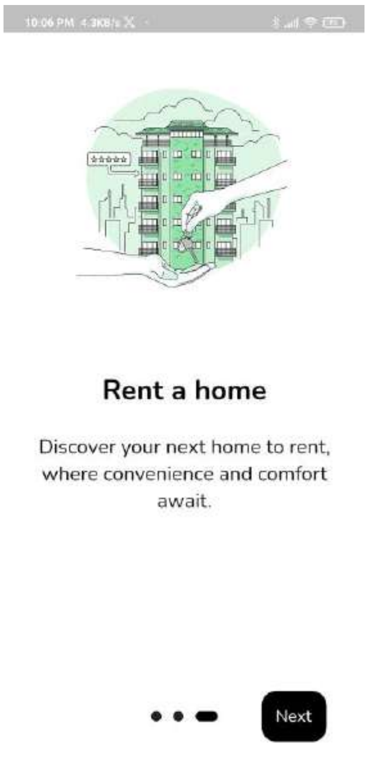
Sell a home

Whether you choose traditional listing or explore innovative selling methods, we're here to guide you towards a successful home sale.



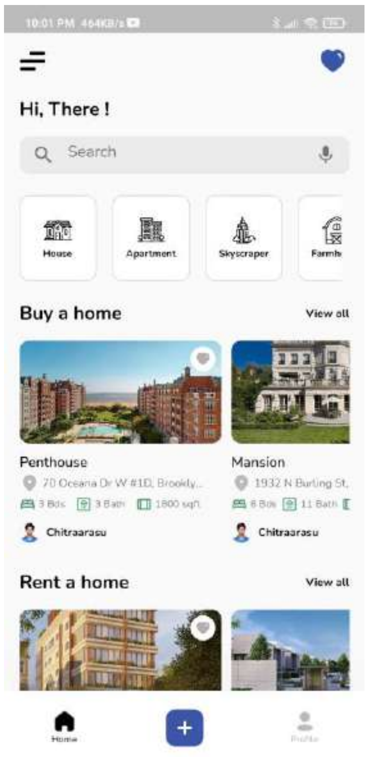
Buy a home

Discover your perfect home with us, where comfort meets your vision of a dream property.

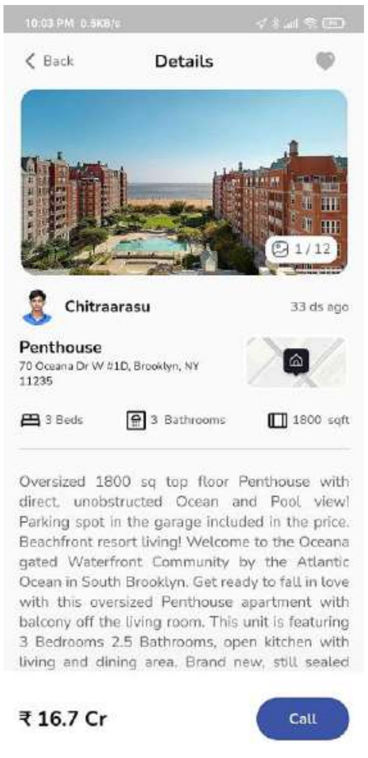


Rent a home

Discover your next home to rent, where convenience and comfort await.



Hi, There !



Details

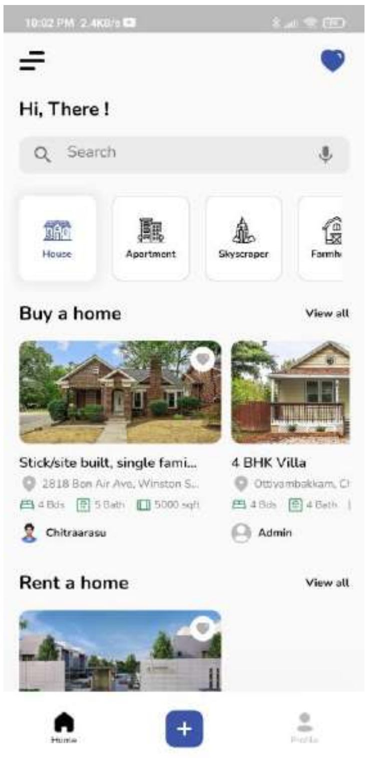
Penthouse

70 Oceana Dr W #1D, Brooklyn, NY 11235

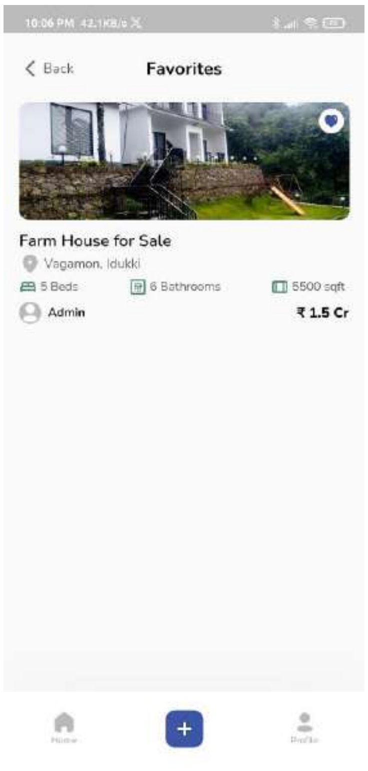
3 Beds 3 Bathrooms 1800 sqft

Oversized 1800 sq top floor Penthouse with direct unobstructed Ocean and Pool view! Parking spot in the garage included in the price. Beachfront resort living! Welcome to the Oceana gated Waterfront Community by the Atlantic Ocean in South Brooklyn. Get ready to fall in love with this oversized Penthouse apartment with balcony off the living room. This unit is featuring 3 Bedrooms 2.5 Bathrooms, open kitchen with living and dining area. Brand new, still sealed

₹ 16.7 Cr



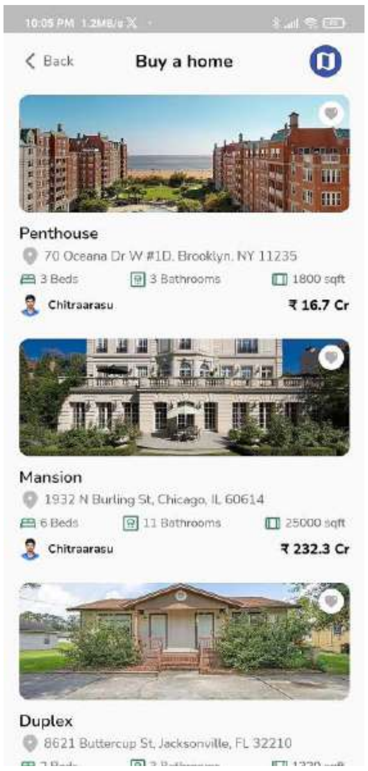
Hi, There !



10:06 PM 42.1KB/s

Back Favorites

Farm House for Sale
Vagamon, Idukki
5 Beds 6 Bathrooms 5500 sqft
Admin ₹ 1.5 Cr



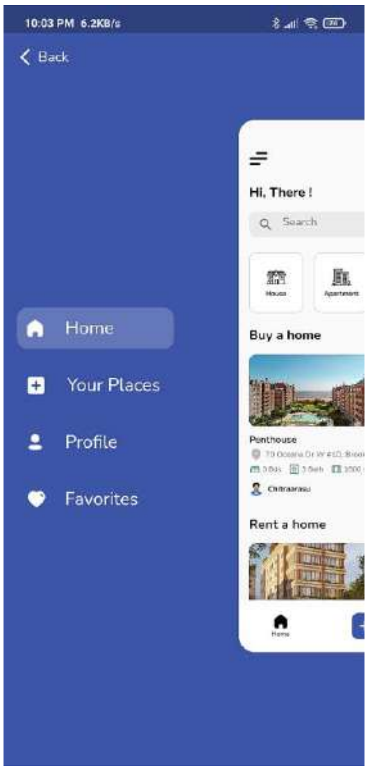
10:05 PM 1.2MB/s

Back Buy a home

Penthouse
70 Oceana Dr W #1D, Brooklyn, NY 11235
3 Beds 3 Bathrooms 1800 sqft
Chitraarasu ₹ 16.7 Cr

Mansion
1932 N Burling St, Chicago, IL 60614
6 Beds 11 Bathrooms 25000 sqft
Chitraarasu ₹ 232.3 Cr

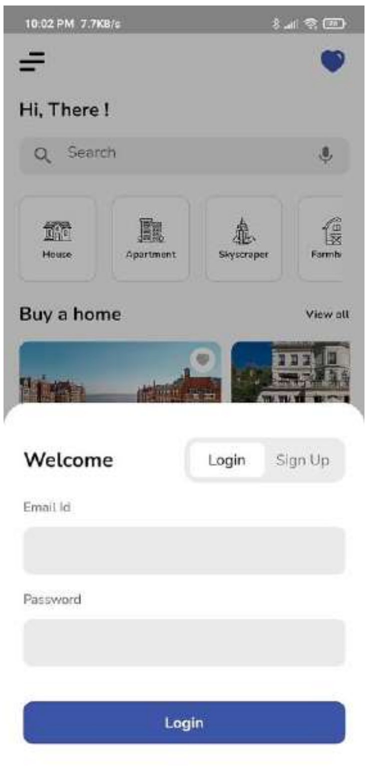
Duplex
8621 Buttercup St, Jacksonville, FL 32210



10:03 PM 6.2KB/s

Back

Home
Your Places
Profile
Favorites



10:02 PM 7.7KB/s

Hi, There !

Search

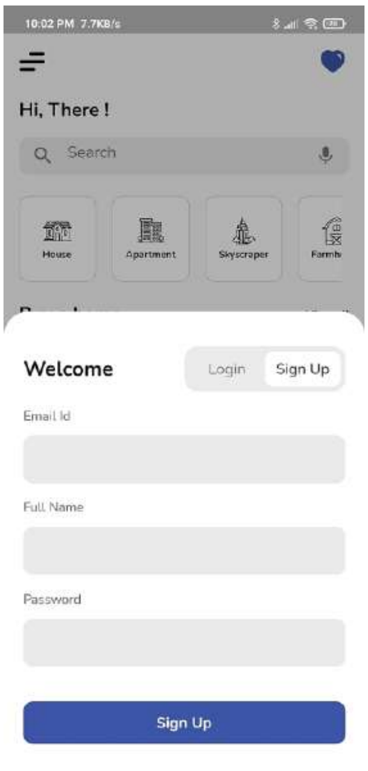
Buy a home View all

Welcome Login Sign Up

Email Id

Password

Login



10:02 PM 7.7KB/s

Hi, There !

Search

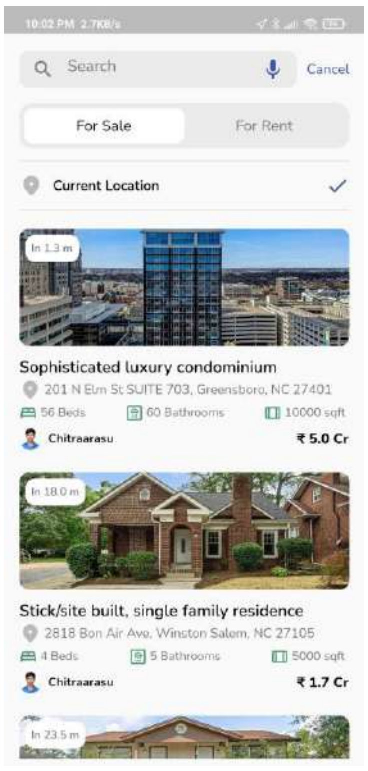
Welcome Login Sign Up

Email Id

Full Name

Password

Sign Up



10:02 PM 2.7KB/s

Search Cancel

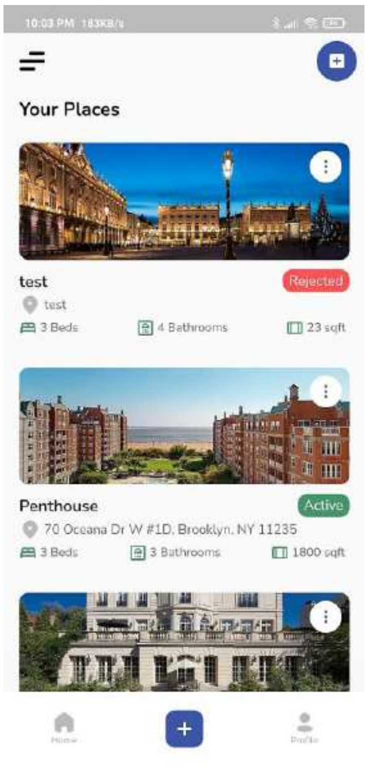
For Sale For Rent

Current Location

In 1.3 m
Sophisticated luxury condominium
201 N Elm St SUITE 703, Greensboro, NC 27401
56 Beds 60 Bathrooms 10000 sqft
Chitraarasu ₹ 5.0 Cr

In 18.0 m
Stick/site built, single family residence
2818 Bon Air Ave, Winston Salem, NC 27105
4 Beds 5 Bathrooms 5000 sqft
Chitraarasu ₹ 1.7 Cr

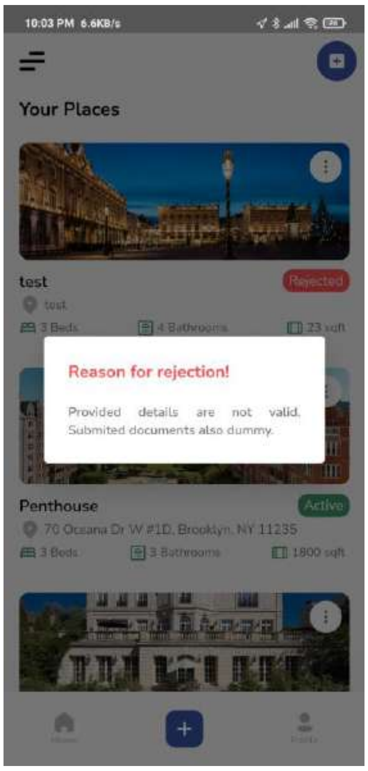
In 23.5 m


10:03 PM 183KB/s

Your Places

test Rejected
test
3 Beds 4 Bathrooms 23 sqft

Penthouse Active
70 Oceana Dr W #1D, Brooklyn, NY 11235
3 Beds 3 Bathrooms 1800 sqft


10:03 PM 6.6KB/s

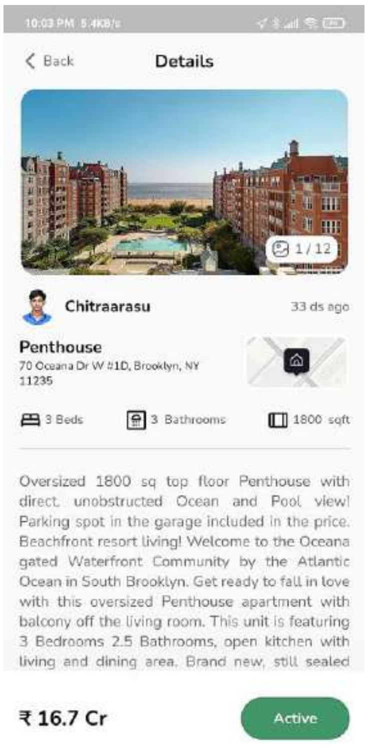
Your Places

test Rejected
test
3 Beds 4 Bathrooms 23 sqft

Reason for rejection!

Provided details are not valid. Submited documents also dummy.

Penthouse Active
70 Oceana Dr W #1D, Brooklyn, NY 11235
3 Beds 3 Bathrooms 1800 sqft


10:03 PM 5.4KB/s

Back Details

1/12

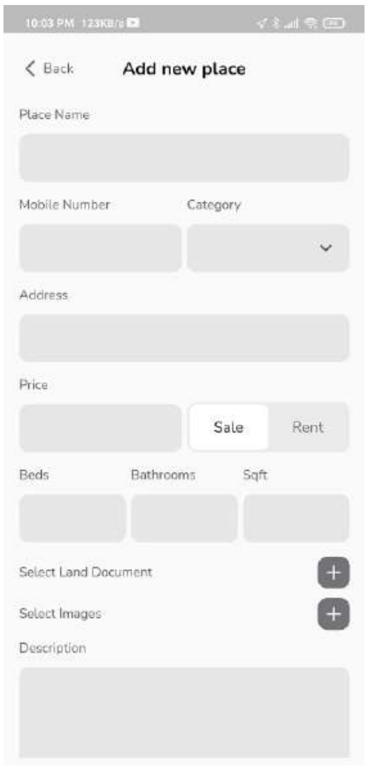
Chitraarasu 33 ds ago

Penthouse
70 Oceana Dr W #1D, Brooklyn, NY 11235

3 Beds 3 Bathrooms 1800 sqft

Oversized 1800 sq top floor Penthouse with direct, unobstructed Ocean and Pool view! Parking spot in the garage included in the price. Beachfront resort living! Welcome to the Oceana gated Waterfront Community by the Atlantic Ocean in South Brooklyn. Get ready to fall in love with this oversized Penthouse apartment with balcony off the living room. This unit is featuring 3 Bedrooms 2.5 Bathrooms, open kitchen with living and dining area. Brand new, still sealed

₹ 16.7 Cr Active


10:03 PM 123KB/s

Back Add new place

Place Name

Mobile Number Category

Address

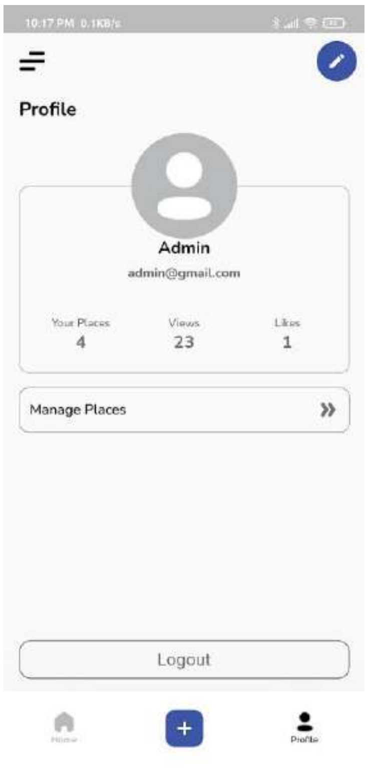
Price Sale Rent

Beds Bathrooms Sqft

Select Land Document

Select Images

Description


10:17 PM 0.1KB/s

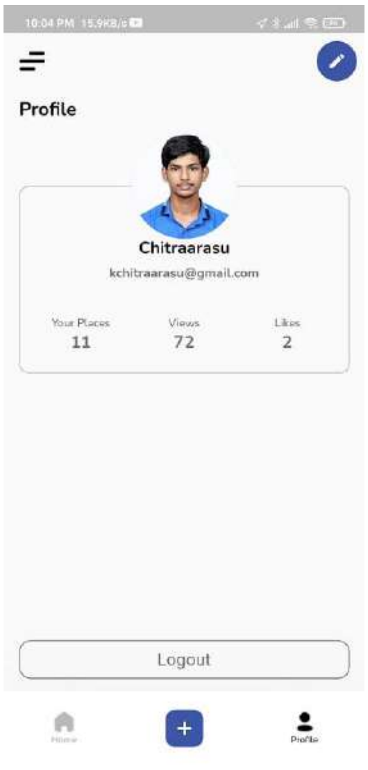
Profile

Admin
admin@gmail.com

Your Places	Views	Likes
4	23	1

Manage Places

Logout

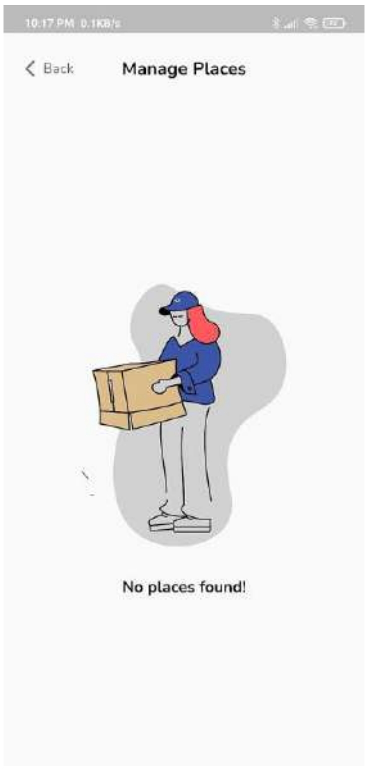

10:04 PM 15.9KB/s

Profile

Chitraarasu
kchitraarasu@gmail.com

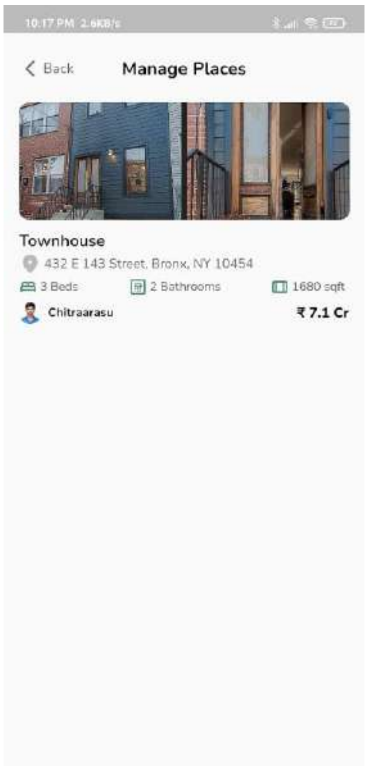
Your Places	Views	Likes
11	72	2

Logout


10:17 PM 0.1KB/s

Back Manage Places

No places found!


10:17 PM 2.6KB/s

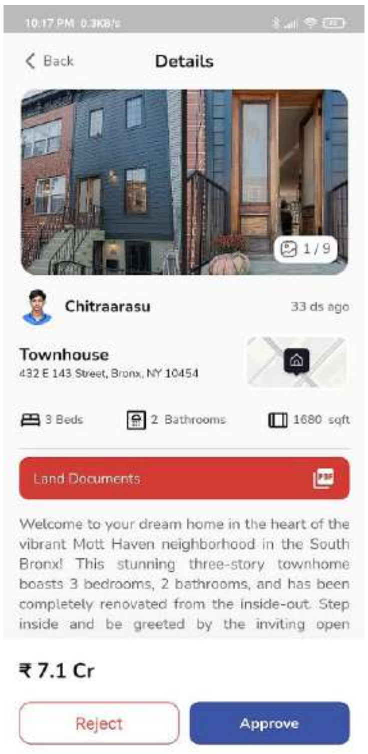
Back Manage Places

Townhouse

432 E 143 Street, Bronx, NY 10454

3 Beds 2 Bathrooms 1680 sqft

Chitraarasu ₹ 7.1 Cr


10:17 PM 0.3KB/s

Back Details

1 / 9

Chitraarasu 33 ds ago

Townhouse

432 E 143 Street, Bronx, NY 10454

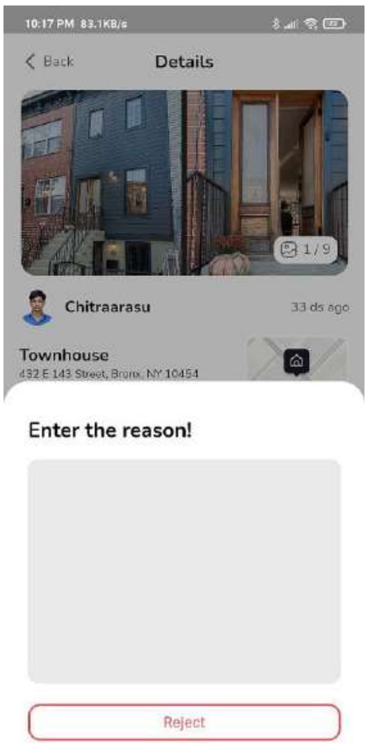
3 Beds 2 Bathrooms 1680 sqft

Land Documents

Welcome to your dream home in the heart of the vibrant Mott Haven neighborhood in the South Bronx! This stunning three-story townhome boasts 3 bedrooms, 2 bathrooms, and has been completely renovated from the inside-out. Step inside and be greeted by the inviting open

₹ 7.1 Cr

Reject Approve


10:17 PM 83.1KB/s

Back Details

1 / 9

Chitraarasu 33 ds ago

Townhouse

432 E 143 Street, Bronx, NY 10454

Enter the reason!

Reject

CHAPTER 10
REFERENCES

1. **Flutter Official Website:** Explore Flutter, an open-source UI toolkit by Google for building natively compiled applications for mobile, web, and desktop.
   Link: https://flutter.dev/
2. **Flutter Gallery:** Check out the Flutter Gallery, a showcase of interactive examples and code snippets to learn and explore Flutter's capabilities.
   Link: https://gallery.flutter.dev/
3. **Firebase Documentation:** Learn about Firebase, a powerful platform for mobile and web app development, offering tools like real-time databases, authentication, and more.
   Link: https://firebase.google.com/docs
4. **Dart Programming Language:** Discover Dart, a versatile programming language optimized for building mobile, desktop, server, and web applications.
   Link: https://dart.dev/
5. **Dart Package Repository (pub.dev):** Find and share Dart packages on pub.dev, the official repository for enhancing your Flutter and Dart projects.
   Link: https://pub.dev/
6. **Book - "Flutter UI succinctly" by Ed Freitas:** Learn about building user interfaces with Flutter in this succinct book by Ed Freitas.
7. **Book - "Managing State in Flutter Pragmatically" by Waleed Arshad:** Explore effective strategies and best practices for managing state in Flutter applications in Waleed Arshad's book.
8. **Book - "Flutter Cookbook" by Simone Alessandria:** Access practical solutions, recipes, and examples for common Flutter development tasks in Simone Alessandria's Flutter Cookbook.
9. **Firestore Quickstart Guide:** Quickstart guide for Cloud Firestore, Firebase's NoSQL document database.
   Link: https://firebase.google.com/docs/firestore/quickstart
10. **GitHub Repository by Chitraarasu:** Explore the GitHub repository by Chitraarasu, containing source code, projects, or documentation related to their work.
    Link: https://github.com/chitraarasu