**(E-Stated) Real Estate App**

Submitted in partial fulfillment of the requirements

of the syllabus of

Android Apps Development Lab

in

Information Technology

by

NACHIKET DHOPAVKAR 118A3012

MANISH JADHAV 118A3021

RITIK KARAYAT 118A3022

Under the Guidance of:

Ms. Bushra Shaikh



Department of Information Technology

SIES Graduate School of Technology

2021-22

**CERTIFICATE**

This is to certify that the project entitled **“**(E-Stated) Real Estate App**”** is a bonafide work of the following students, submitted to the University of Mumbai in partial fulfillment of the requirement of the syllabus of **Android Apps Development Lab** in **Information Technology.**

**NACHIKET DHOPAVKAR 118A3012**

**MANISH JADHAV 118A3021**

**RITIK KARAYAT 118A3022**

Ms. Bushra Shaikh Dr. Lakshmi Sudha Dr. Atul N Kemkar

Internal Guide Head of Department Principal

**PROJECT REPORT APPROVAL**

This project report entitled ***(E-Stated) Real Estate App*** by following students is approved for the requirement of the syllabus of ***Android Apps Development Lab*** in ***Information Technology.***

**NACHIKET DHOPAVKAR 118A3012**

**MANISH JADHAV 118A3021**

**RITIK KARAYAT 118A3022**

**Name of External Examiner: --------------------------------**

**Signature:--------------------------------**

**Name of Internal Examiner: --------------------------------**

**Signature:--------------------------------**

**Date:**

**Place:**

**DECLARATION**

I declare that this written submission represents my ideas in my own words and where others’ ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

NACHIKET DHOPAVKAR 118A3012 \_\_\_\_\_\_\_\_\_\_\_\_\_

MANISH JADHAV 118A3021 \_\_\_\_\_\_\_\_\_\_\_\_\_

RITIK KARAYAT 118A3022 \_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Date:

**ACKNOWLEDGEMENT**

It gives us immense pleasure to thank Dr. Atul N Kemkar, our Principal for extending his support to carry out and develop the project. We also thank our Head of Department Dr. Lakshmi Sudha for her support in completing the project. We wish to express our deep sense of gratitude and thank to our Internal Guide, Ms. Bushra Shaikh for her guidance, help and useful suggestions, which helped in completing our project work in time.

We would like to thank the entire faculty of Information Technology Department for their valuable ideas and timely assistance in this project, last but not least, we would like to thank our non-teaching staff members of our college for their support, in facilitating timely completion of this project.

**Project Team**

Nachiket Dhopavkar

Manish Jadhav

Ritik Karayat

**ABSTRACT**

With property prices reaching cloud-9 and online scams getting common , it’s getting difficult for people to even look for properties online. Choosing the offline option they have to spend extra money to brokers. E-stated is a platform designed for buyers to easily get in touch with sellers. It’s like ‘Tinder’ for people looking for houses.

The main objective of E-Stated is to ease the process of buying and selling properties and avoiding any sort of middle-man shenanigans and providing a secure environment. The buyers can directly contact the builders.

Buyers can view the properties and filter them accordingly on the basis of various parameters like Location & Price to meet their optimum requirements. The app also gives a functionality of predicting the average price of a property by filling the various parameters mentioned above.

**Contents**

|  |  | **Page No.** |
| --- | --- | --- |
| **Chapter 1** | **Introduction** | **8** |
| **Chapter 2** | **Survey of Existing Apps** | **9-10** |
| **Chapter 3** | **Report on Present Investigation** | **111** |
|  | 3.1 Problem Statement | **11** |
|  | 3.2 Source of Problem Statement | **11** |
| **Chapter 4** | **Design and Implementation of Android Apps Components** | **12-16** |
|  | 4.1 Layouts | **12** |
|  | 4.2 Intents | **12-13** |
|  | 4.3 Activity | **13-14** |
|  | 4.4 SQLite | **14** |
|  | 4.5 Camera | **14-15** |
|  | 4.6 Location API | **15-16** |
|  | 4.7 WebView | **16** |
|  | 4.8 Generate APK file | **17** |
|  |  |  |
| **Chapter 5** | **Report on Proposed System and its Implementation** | **18-20** |
|  | 5.1 Block Diagram | **18** |
|  | 5.2 Flowchart | **19** |
|  | 5.3 Hardware | **20** |
| **Chapter 6** | **Results and Discussions** | 21-32 |
|  | 6.1 Summary of Screenshots with Navigational Flow | 21-32 |
| **Chapter 7** | **Conclusions** | **33** |
| **References** |  | **34** |

**Introduction**

In recent time, Android based applications are getting wider popularity and applicability across range of problem domains. However, literature investigation shows that most existing Real Estate Management Solutions are either web based or cloud based, and very few designed for Android platform. The existing Android based Real Estate Management Systems lack capacity to display property on map, and lack navigational support which could guide client to physical location of property.

The demand for real estate management service is on the rise. Frequently, people want to buy, saleor lease property. This suggest the need for automation of the real estate management services. Currently, there are good number of real estate management systems which are cloud or web based, but very few Android base. For examples, there is corporate real estate

management software developed for the real estate industry . It tracks client’s interactions and automates workflows. It enables buying and selling agents to close more deals faster.

Some benefits of this software to real estate professionals include enhanced visibility, reliable data across all corporate real estate functions, reduced costs of application/infrastructure support and maintenance, decreased vacancy rates and improved personnel efficiencies through automation of manual and duplicate processes. However, notable short coming of the app is lack of advanced filter options, and inability to display all properties on map.

**Survey On Existing Apps**

**1. Apartment managerXp**

**Introduction**

It is a cloud base property management system that centers all the estate business in one system accessible anytime anywhere. It is lease management software that supports both landlord and tenant. It provides an effective way to track real estate for critical financial and reporting needs. It has many features that enable managers to update information dynamically and also allows owners and property managers to adapt to a wide range of contingencies that occur in managing property. It integrates with Accounts Receivable, and Accounts Payable system. Primary features of this system includes Recurring and manual tenant billing, processing of receipts, calculation of security deposits, Reporting of sales over-age billing, Processing revenue fees, assessing fees and interest, creating rent projections, among others

**2. Airbnb**

**Introduction**

Airbnb is a website that offers exclusive accommodation in a house, apartment, boat, or room. It is free for host to create listing. The host decides how much to charge per night, per week or per month. Each listing allows host to promote properties through titles, descriptions, photographs with captions and host’s profile where potential guests can get to know a bit about the host. While Agoda.com serves a variety of information about hotels needed by tourists, it only accepts payments by credit cards. Agoda shows names of hotels, their average users rating and prices for rooms. Clicking on a hotel takes a customer to a page where

customer can have the option to see the hotel’s policies and listing of its facilities as well as other useful information

**3. Zillow & HomeSnap**

**Introduction.**

Another popular real estate management app is Zillow, which is available for both Android and iOS devices. The app can display your house (or your name) on map, and allow people to access it with a single click. Notable advantage of the app is provision of new Video Walkthrough feature that provides buyers with more realistic view and allow sellers to create their own customized video walkthroughs. However, the app is labelled with a lots of fraudulent and fake listings.

HomeSnap , often called the Snapchat of real estate, is an app for real estate agents that sources information from the MLS listings, and allows agent to send private messages to clients, even when the clients’ do not have the app on their devices. Moreover, the app provides incredibly effective location tracking mechanism. However, notable short coming of the app is lack of advanced filter options, and inability to display all properties on map.

**Report on Present Investigation**

**3.1) Problem Statement:**

As smartphones have become common among the masses, old ways of physically going to the broker or real estate agent and looking for properties are getting outdated and becoming inefficient. In this modern world we need something fast-paced and efficient to replace this process online and on the tips of your fingers.

The user can login in the app after registering in the app. Once logged in they can add , view ,predict price of a property or see the location of properties in maps.

**3.2) Source of Problem Statement:**

The demand for real estate management services is on the rise. Frequently, people want to buy, sell or lease property. This suggests the need for automation of the real estate management services. Currently, there are a good number of real estate management systems which are cloud or web based, but very few Android base. Thus we decided to create this app for good reason.

**Design and Implementation of Android Apps Components**

**4.1) Layouts**

Layout basically refers to the arrangement of elements on a page these elements are likely to be images, texts or styles. These are a part of **Android Jetpack**. They define the structure of [android user interface](http://web.cs.wpi.edu/~emmanuel/courses/cs4518/C17/slides/lecture03.pdf) in the app, like in an activity. All elements in the layout are built with the help of Views and ViewGroups. These layouts can have various widgets like buttons, labels, textboxes, and many others.

Some of the Layouts in Android are

* Linear Layout
* Relative Layout
* Constraint Layout
* Table Layout
* Frame Layout
* Absolute Layout

You can declare a layout in two ways:

* **Declare UI elements in XML**. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.

You can also use Android Studio's [Layout Editor](https://developer.android.com/studio/write/layout-editor) to build your XML layout using a drag-and-drop interface.

* **Instantiate layout elements at runtime**. Your app can create View and ViewGroup objects (and manipulate their properties) programmatically.

**We have used Constraint Layout for the Login and registration page.**

**We have used Linear Layout for the profile page.**

**We have used Scroll view with Linear Layout for designing all the data structure and algorithm pages.**

**We have used Relative Layout and Card view in the quiz pag**e.

**We have used Card View for home pages of data structures and algorithms and also for the quiz.**

**4.2) Intents**

**Android Intent** is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.

It is generally used with startActivity() method to invoke activity, broadcast receivers etc.

There are two types of intents:

* **Explicit intents** specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start. For example, you might start a new activity within your app in response to a user action, or start a service to download a file in the background.

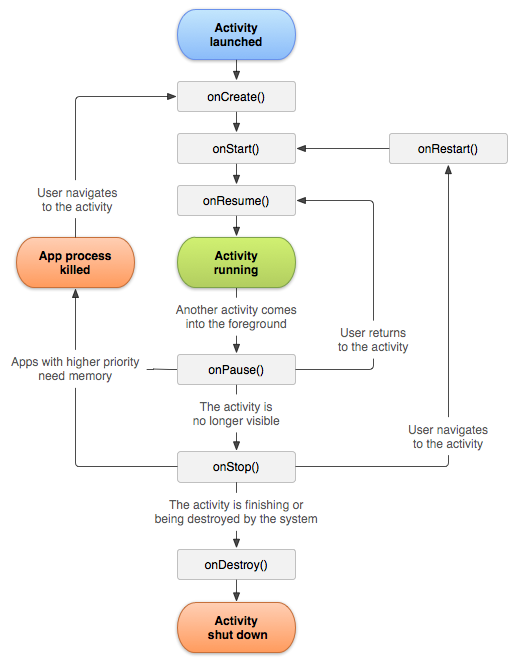
**We have used Explicit intent to connect various activities like going to Login page fromRegister Page.**

*startActivity(new Intent(RegisterActivity.this,LoginActivity.class));*

* **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.

**4.3) Activity**

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.



**Activities in our application are:**

**Login Page)**

**Sign In Page**

**Welcome Page**

**Add Properties**

**View Properties**

**Predict Price of the Properties**

**Maps**

**Detail Display Properties**

**4.4) SQLite**

**SQLite** is an **open-source relational database** i.e. used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database.

It is embedded in android by default. So, there is no need to perform any database setup or administration task.

Here, we are going to see the example of sqlite to store and fetch the data. Data is displayed in the logcat. For displaying data on the spinner or listview, move to the next page.

**SQLiteOpenHelper** class provides the functionality to use the SQLite database

**We have used SQLite to store the user details entered at the time of registration and also the to store the property details provided by the user like image of the house Square foot,BHK, and description of the house.**

**4.5) Camera**

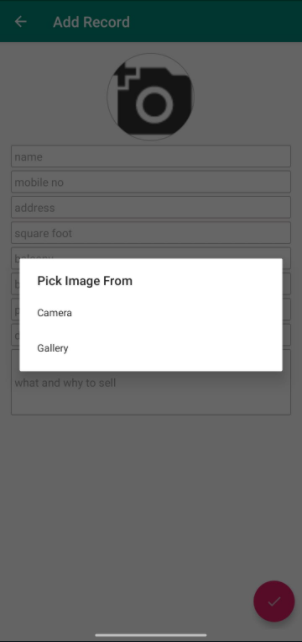
**Camera** is mainly used to capture picture of the house and we also added the functionality to crop the image of the house.

Android provides the facility to work on camera by 2 ways:

1. By Camera Intent
2. By Camera API

**We have used camera to allow user take a picture for the profile image. We used the Camera Intent**

Intent cameraIntent=new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

****

**4.6) Location API**

The location APIs available in Google Play services facilitate **adding location awareness to** your app with automated location tracking.

**We have used location api to take the Current location of the user so he can sees the near by properties of which are listed.**

****

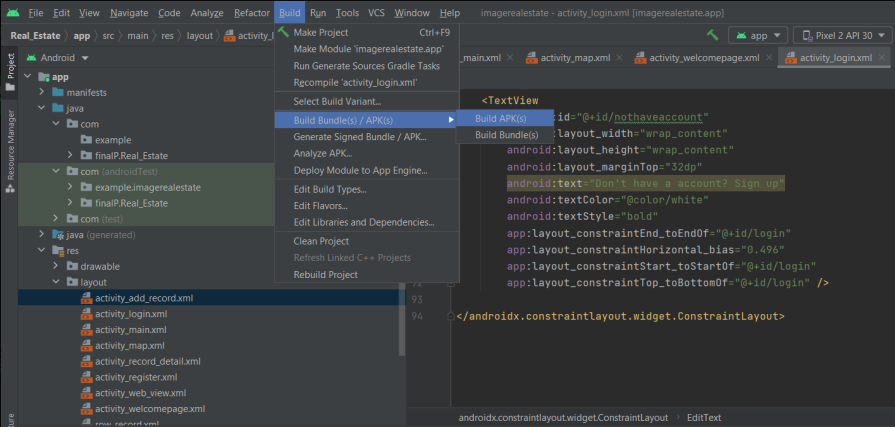
**4.7) Webview**

WebView is a view that display web pages inside your application. You can also specify HTML string and can show it inside your application using WebView. WebView makes turns your application to a web application.

**We have used Webview to connect our app with the website which is hosted on heroku cloud to get the approximate value of the house on the basis of users needs.**

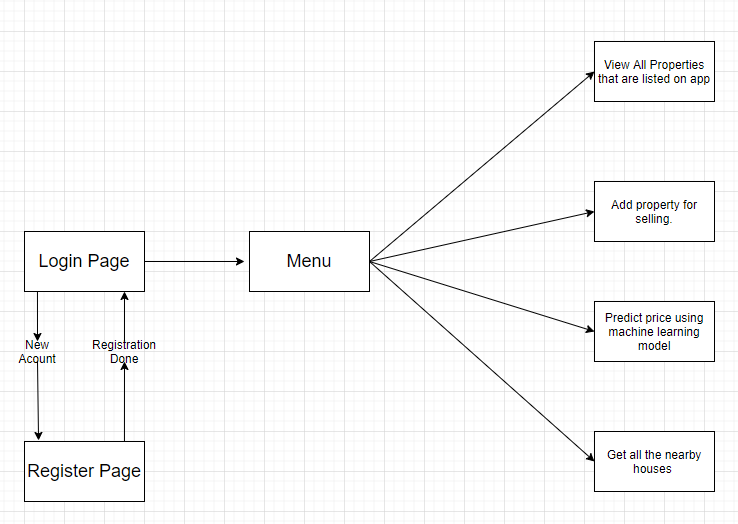
**4.8) Generate APK**

1. In the Android menu, go to Build > Build Bundle(s) / APK (s) > Build APK(s).
2. Android Studio will start building the APK for you. Once done, a pop-up on the bottom right will notify you of its completion. Click the ‘locate’ button in this dialog.
3. The ‘locate’ button should open File Explorer with the debug folder open that contains a file called “app-debug.apk”.
4. That’s it. Rename this file and share!

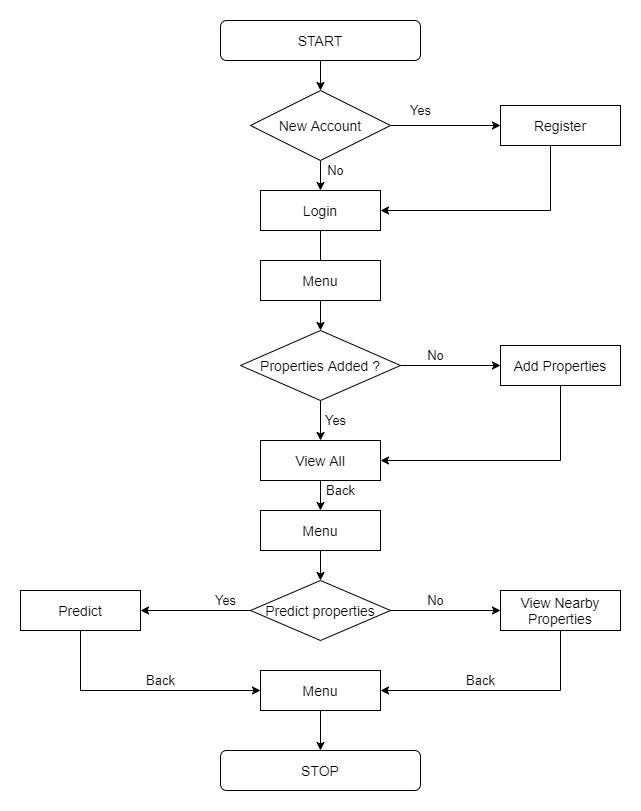


**Report on Proposed System and its Implementation**

**Block Diagram:-**

****

**Flowchart:**

****

**Hardware –**

* Android Device
* GPS
* Internet
* Camera

**Software / External Libraries used with description –**

* Android Studio

Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

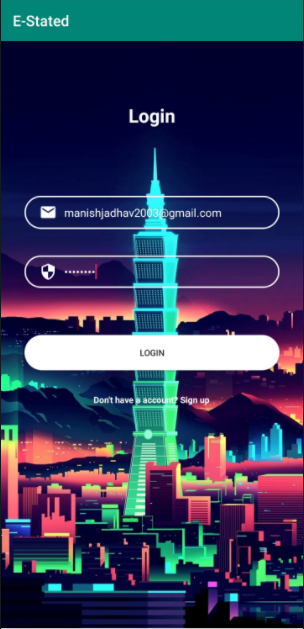
**Results and Discussions:**

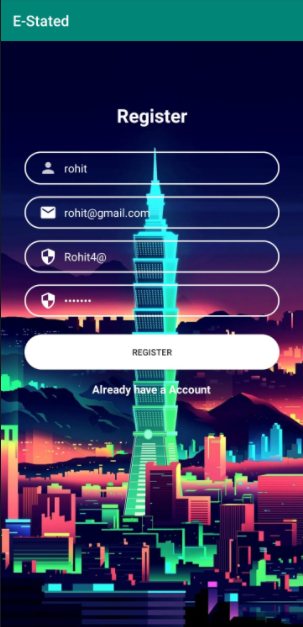
**Module A:**

**Login and Sign In**

The user needs to register filling basic details like name, email, phone number, address and create a password. Password must be 8 characters long. None of these fields can be kept blank.

Once registered user can login using the registered email id and password anytime he/se wants to access the app.





**Module B:**

**The Menu Page**

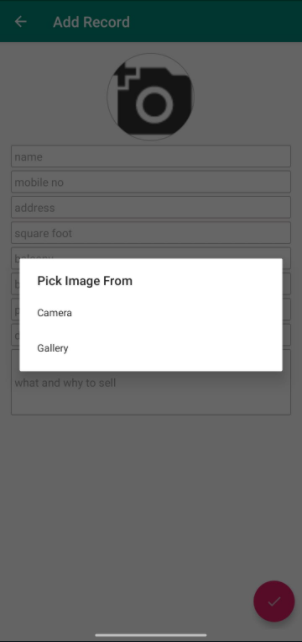
User can view all the features available to them in this page like adding , viewing properties etc.

****

**Module C:**

**Add Property**

Users can add properties filling the form and uploading the image so that it gets listed on the app for everyone to view.

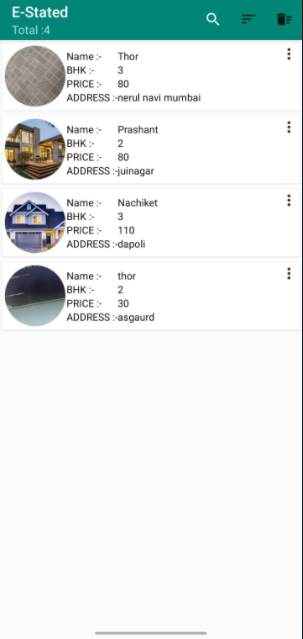
****

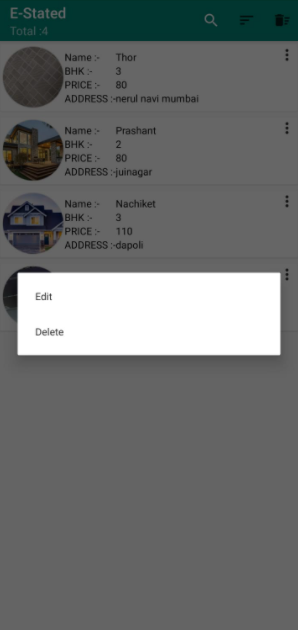
****

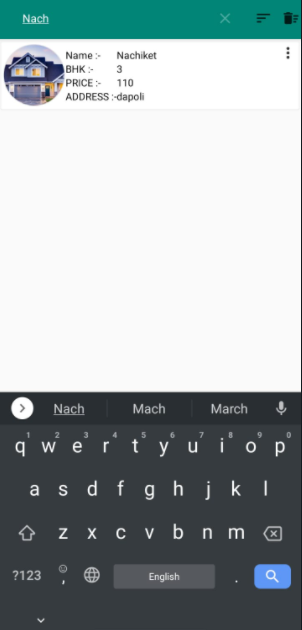
**Module D:**

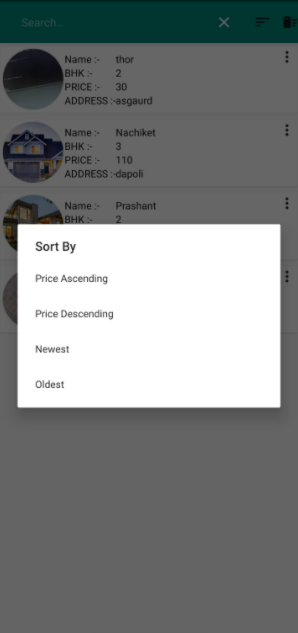
**View All Properties**

Users can view and edit all the listed properties in this page and filter them based on location and price or sort according to various parameters.

****

****

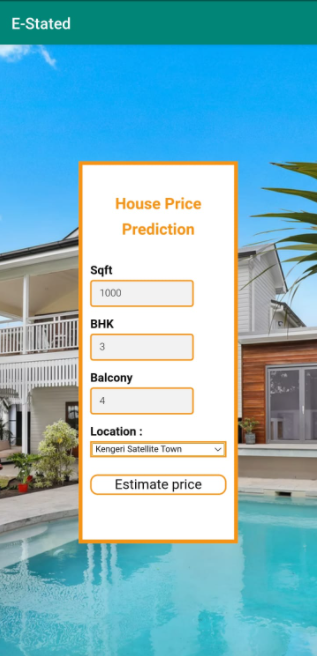
****

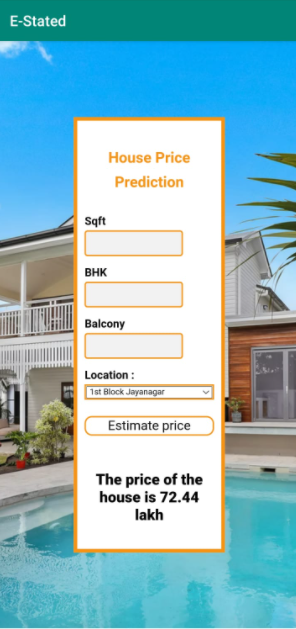
****

**Module E:**

**Predict Price**

Users can predict the price of properties as per their requirements. The model is trained on RidgeCV algorithm which takes 4 parameters i.e Sqft , BHK ,Balcony & Location.

****

****

**Module F:**

**Maps**

Users can view the nearby properties on maps in which their current location is shown by green marker and other properties are shown by red marker.

****

**Conclusion**

All in all this E-Stated makes life easier for brokers who are looking to list their properties online for selling them comfortably with ease. Users can view properties at once both in listed format and in maps showing nearby properties by fetching the live location. Price Prediction feature is also good for users to gain a sense of the market and buy accordingly.

Future Scope:

Buyer-Seller Module(Different Logins) & Dynamic Location Rendering via Add feature.

**References**

1. https://developer.android.com/training/data-storage/sqlite
2. https://developers.google.com/maps/documentation/android-sdk/start

[3] https://developer.android.com/guide/topics/media/camera