# Web app for buying and selling properties along with price predicting feature using ML

**Team**
Rishit - 191IT141
Sarthak Jain - 191IT145
Yash Gupta - 191IT158

# Agenda

I. Introduction
II. Literature Survey
III. Outcome of Literature Survey
IV. Motivation
V. Problem Statement
VI. Objectives
VII. Proposed Work
VIII. Conclusion
IX. Timeline of Project
X. Individual Contribution
XI. References

# Introduction

In this developing global village with the increasing human activities such as industrialization the land has become a very highly priced resource.The soaring land prices and middlemen being involved has made it a possible requirement to predict real estate prices. Similarly finding an ideal property for one is a troublesome task and utilises a lot of time and resources.

So there is the need of a platform where users can easily find their ideal home and sellers can advertise and list their properties to find a buyer fast instead of paying heavily to  advertise and giving money to middlemen to sell their house.

By this project we have developed a web app where sellers can list their properties and buyers can go through a number of houses and find their ideal match.

A user friendly UI coupled with user authentication, a web app was designed to serve the purpose.

Another distinguishing feature is the price prediction of a property based on its attributes such as location, size, etc.

We have thus employed machine learning to make predictions on the real estate prices based on the dataset of Bangalore.

We tried various  Machine Learning Algorithms  and picked one which gave us the highest accuracy.

The Algorithms we tried are XGBoost,lasso,decision tree for regression.

We used Grid Search to pick the best algorithm.

Users can make predictions by selecting location,area in square feet,No. of BHK's and No.of bathrooms.

Result will be an estimated price in Lakhs.

# Literature Review

| Authors | Methodology | Advantages | Limitations |
|---|---|---|---|
| G. Naga Satish, Ch. V. Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu | Building Machine Learning Model using Jupyter notebook and training on several Regression algorithms | Decision tree ,Linear regression | XGBoost not covered |
| | | | |
| Lianne | Various techniques and methods in python for efficient data cleaning | Removing faulty data | Dataset was not adequately large |
| | | | |
| Mr.Prakash P Lokande | Using Python to create server for backend and creating utility using pickle to make predictions | Flask server setup | Deployment of flask server not covered |
| | | | |
| Enes Yigitbas ,Ivan Jovanovikj | Component and modular structure for easier management | Component based development of adaptive user interfaces | No Mobile first approach |

# Outcome of Literature Review

By referring to various sources mentioned in Literature survey section we were able to cover all important modern day trends in field of web development like usage of framework like react to build advanced web applications,usage of API's(unsplash and firebase),integrating web applications with Machine Learning models using flask,integrating web applications with a database.

Mobile first approach was also adopted as stats suggest that mobile users access the web more than pc users due to increasing number of mobile phones.

We also learnt how to deploy our applications on web so that they are available for normal users.

A user friendly, responsive, easy to use and efficient web application with side drawer for proper and easy navigation.

Fast and efficient Mongo DB database was used to serve the purpose.

# Issues and Challenges

- Major issues and challenges that we faced during this project are selection of good dataset and cleaning of dataset-there were many faulty entries in our dataset so proper cleaning was important.
- Another challenge that we faced was during deployment since our backend is python (flask server) and to design frontend we used react framework.
- To overcome this challenge we deployed our backend on heroku and frontend separately on github pages.
- This led to another issue namely cors(cross-origin resource sharing) because request is sent to a server hosted on a different url.
- It was solved by modifying the headers of the request and using Flask_cors package.
- Frontend sends request to heroku served backend and then the server sends response based on the request.

# Motivation

Housing prices are an important reflection of the  country's economy, and housing price ranges are of great interest for both buyers and sellers and a need for the common citizens of the country.
Buying and selling house is a headache as buyers struggle to find their ideal house and seller don’t find the desired buyer.
Also house price predictions have become a trending topic today and has been globally employing a large section of the society.This has been increasingly demanding in order to remove the middlemen like brokers as their brokerage costs are an overhead to bear.
Thus we have used machine learning, which can predict property prices based on certain features such as location, size in square feet , the number of bathrooms and the number of bedrooms and linked the model to a web app where buyers and sellers can come together and buy and sell real estate.

# Problem Statement

- Web app for buying and selling properties along with price predicting feature using ML.

# Research Objectives

Major research objectives are :

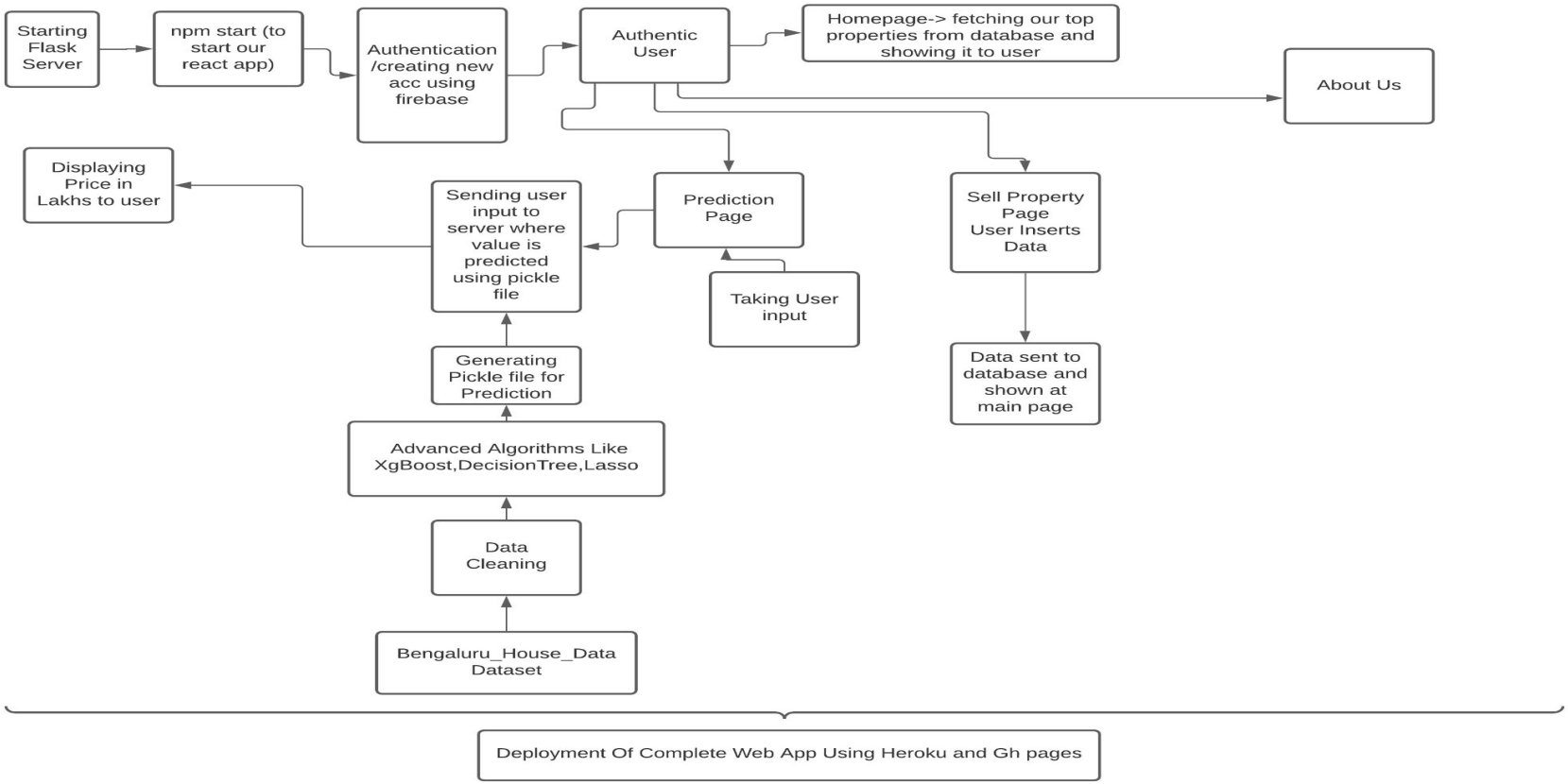
Creation of complex UI by combining various isolated components (increases reusability and modifications can be made easily.)

To build an online platform which is user friendly,secure and can run on any device and where user can predict best and most accurate price based on location,area,bhk,and no. of bathrooms.

To provide user with a platform where he can sell his property easily and in a secured manner.

Efficient deployment (scalable) which can handle many users. (JSON format is used to send responses since they are light-weight and easy to handle)

# Proposed Model



# Methodology

Data cleaning: The dataset contains innumerable columns which needs to be refined for accurate predictions.Thus dataset is cleaned to extract important features and remove erroneous and misleading data for easier training.

Training data:The dataset is then converted to a csv file and then rigorously trained upon by machine learning models and the final results are then dumped into a pickle file.

# Methodology

UI: For better user experience the UI is built along with the frontend framework react which produced a fast and beautiful web app, with easy navigation with side drawer, where the main content is dynamically inserted.

The flask server: The flask server handles the requests made by the front-end and provide appropriate response. It also uses the pickle file to make predictions based on the request data and provides a helping hand in routing to different pages.

# Methodology

Providing authentication: In order to ensure no malicious attacks can pollute the project and maintain user isolation we look to provide authentication using the firebase API.

Database: Mongo DB is the database used. It's speed, easy operations and document structure suited our requirement. Mongo DB atlas (cloud database) is used in the hosted version. The flask servers adds and retrieves data from the database and then send it as a response to the client.

# Methodology

Connecting the server and the UI: All the requests from front-end is made to the server and then useful data is extracted from response body and then shown to the user dynamically. The server in turn relies on the database for the data. Based on the request made, it reads or adds new information to the database.

Deployment:  The build version of react client is deployed on gh-pages and flask server is deployed on heroku.

# Proposed modifications

We look to widen our reaches by integrating datasets of many more cities across the country.

We look into integrating a transaction API for making purchases successful.

Developing an in app chat service for buyers and sellers.

Include email verification so that only serious users add properties.

Developing a checking mechanism as to user enter realistic size and price of property.

# Work Done

The dataset was cleaned by removing ambiguous values like too small areas(ex 1 sqft) and too large areas like 10000 sqft and also the corresponding null values were filtered out to remove any ambiguity in data.

This data was then trained rigorously on various machine learning algorithms like linear regression , lasso , decision trees  and xgboost which resulted in a stiff competition and xgboost turned out to be the winner with an accuracy of 80%.

The results were then dumped into a pickle file which was later integrated with our flask server setup.

# Work Done

The front-end was developed using the react framework.
The complete app was mapped to various different components and then React Router was used for proper routing and speed.
Various npm packages such as React-Router-Dom, firebase, unsplash-js were used.

Server was written in python(for easy use of ML Model) using the Flask framework. Proper routing was done to handle all the requests.

Mongo DB database was used. The cloud variant, Mongo DB atlas was set up and then linked with the Flask server using Flask_Pymongo package.

# 1) Cleaning of database

Before data cleaning

## Data Cleaning

[2]

```
df=pd.read_csv('Bengaluru_House_Data.csv')
df.head(10)
```

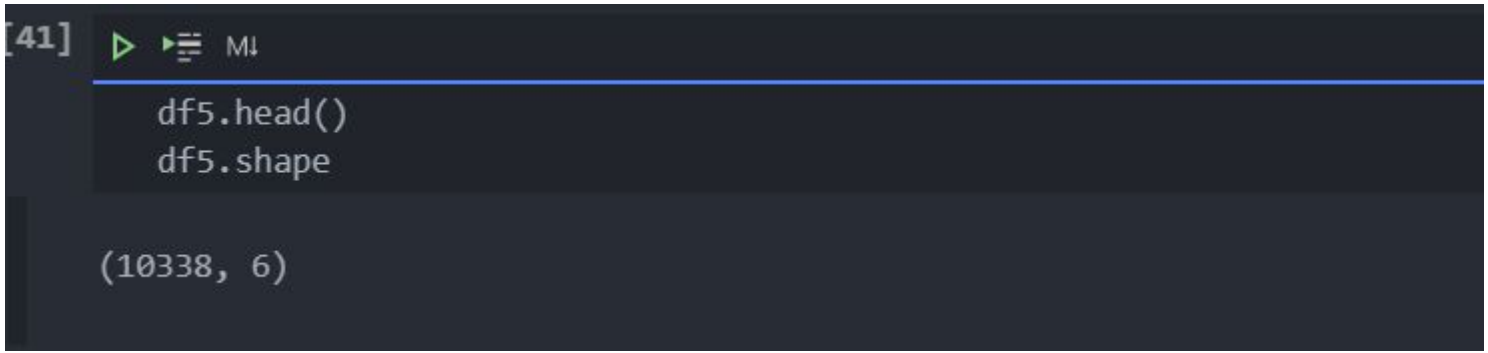
| | area_type | availability | location | size | society | total_sqft | bath | balcony | price |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Super built-up Area | 19-Dec | Electronic City Phase II | 2 BHK | Coomee | 1056 | 2.0 | 1.0 | 39.07 |
| 1 | Plot Area | Ready To Move | Chikka Tirupathi | 4 Bedroom | Theanmp | 2600 | 5.0 | 3.0 | 120.00 |
| 2 | Built-up Area | Ready To Move | Uttarahalli | 3 BHK | NaN | 1440 | 2.0 | 3.0 | 62.00 |
| 3 | Super built-up Area | Ready To Move | Lingadheeranahalli | 3 BHK | Soiewre | 1521 | 3.0 | 1.0 | 95.00 |
| 4 | Super built-up Area | Ready To Move | Kothanur | 2 BHK | NaN | 1200 | 2.0 | 1.0 | 51.00 |
| 5 | Super built-up Area | Ready To Move | Whitefield | 2 BHK | DuenaTa | 1170 | 2.0 | 1.0 | 38.00 |
| 6 | Super built-up Area | 18-May | Old Airport Road | 4 BHK | Jaades | 2732 | 4.0 | NaN | 204.00 |
| 7 | Super built-up Area | Ready To Move | Rajaji Nagar | 4 BHK | Brway G | 3300 | 4.0 | NaN | 600.00 |
| 8 | Super built-up Area | Ready To Move | Marathahalli | 3 BHK | NaN | 1310 | 3.0 | 1.0 | 63.25 |
| 9 | Plot Area | Ready To Move | Gandhi Bazar | 6 Bedroom | NaN | 1020 | 6.0 | NaN | 370.00 |

[3]

```
df.shape
```

```
(13320, 9)
```

# After data cleaning:



# 2)trying various algorithms using grid search cv (decision tree is giving best accuracy here)

| | model | best_score | best_params |
|---|---|---|---|
| 0 | linear_regression | 0.587175 | {'normalize': False} |
| 1 | lasso | 0.499658 | {'alpha': 1, 'selection': 'random'} |
| 2 | decision_tree | 0.682873 | {'criterion': 'mse', 'splitter': 'random'} |

# Improving accuracy by changing parameters of decision tree alogrithm

[42]

```
dr=DecisionTreeRegressor(criterion='mae', splitter='best',)
dr.fit(X_train,y_train)

dr.score(X_test,y_test)
```

```
0.7317486261201342
```

# 3)using advanced algorithm XGBoost to improve accuracy further

43]

```
import xgboost as xg
xgb_r = xg.XGBRegressor(objective ='reg:squarederror',n_estimators = 150, seed = 7)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
train_X=X_train
train_y=y_train
xgb_r.fit(train_X.values, train_y.values)
xgb_r.score(X_test.values,y_test.values)
```

```
0.8036933367025929
```

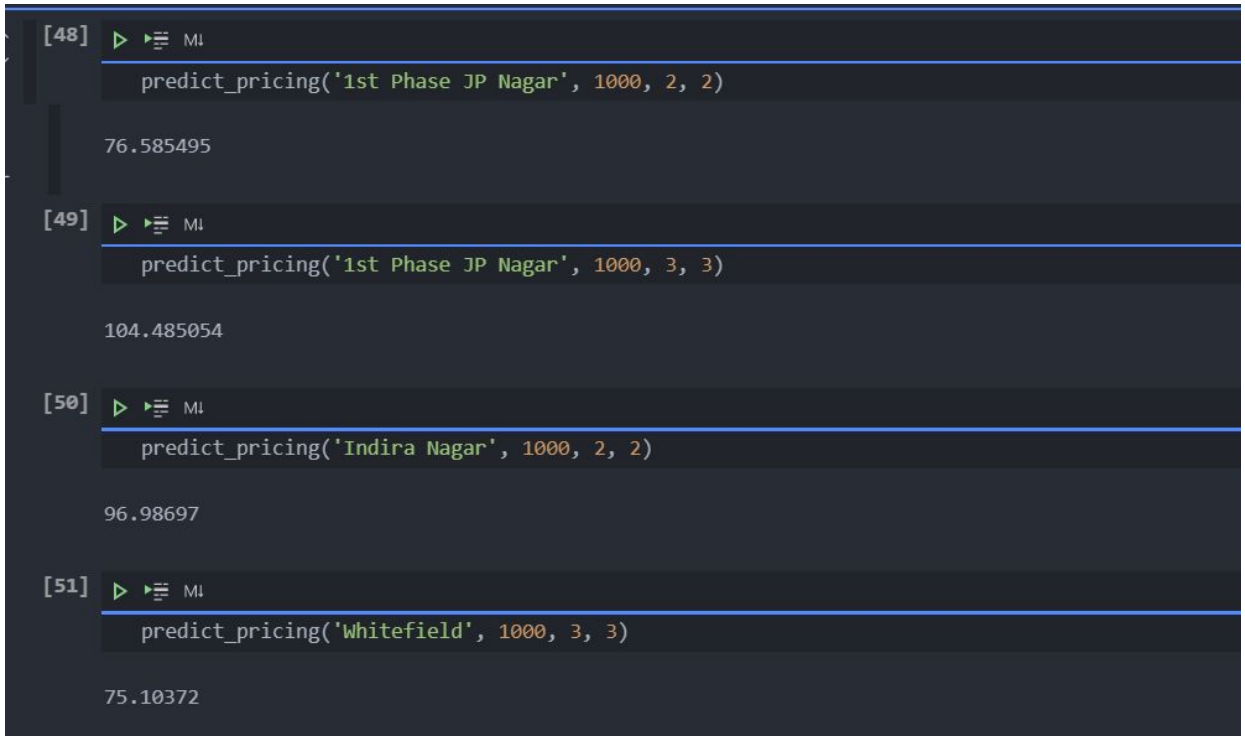
# Creation of pickle file which flask server will use to make predictions

[52]
```
import pickle
with open('Bengaluru_House_Data.pickle', 'wb') as f:
    pickle.dump(xgb_r, f)
```

[53]
```
import json
columns = {
    'data_columns': [col.lower() for col in X.columns]
}
with open('columns.json', 'w') as f:
    f.write(json.dumps(columns))
```

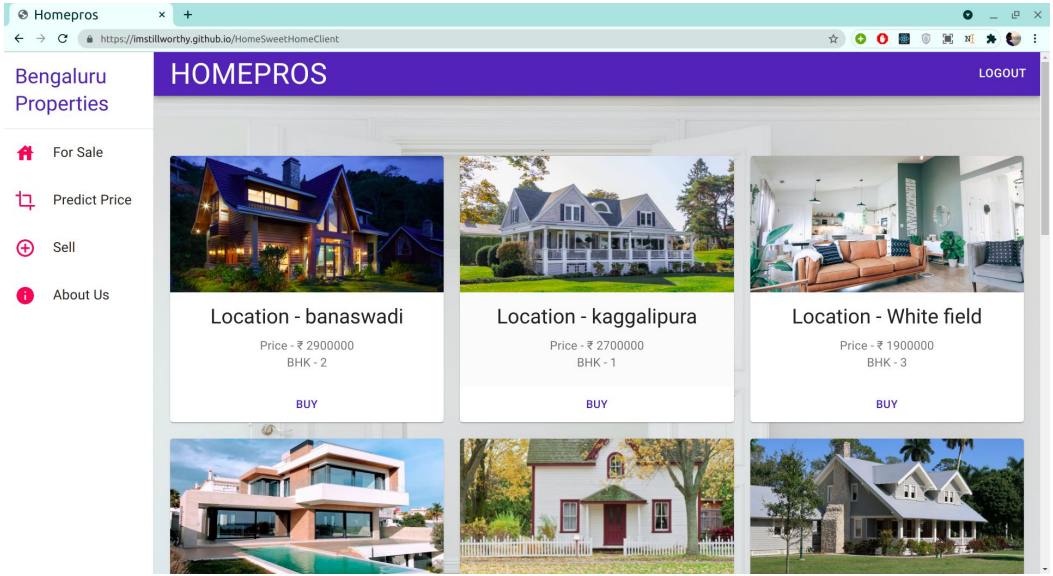
[-]

# Result and Analysis

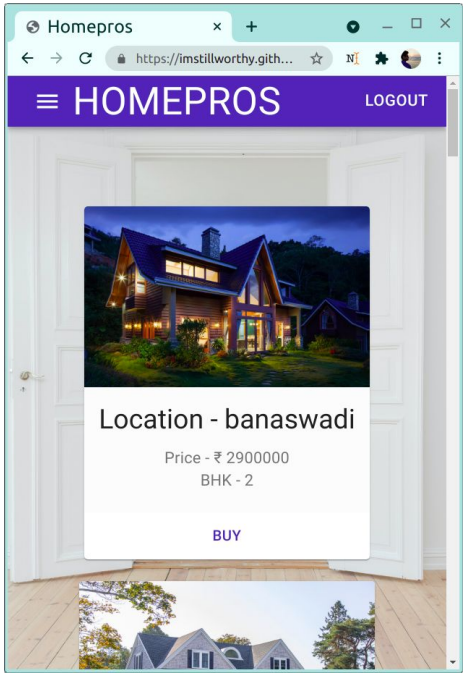



Prediction of price in Lakhs

# Result and Analysis

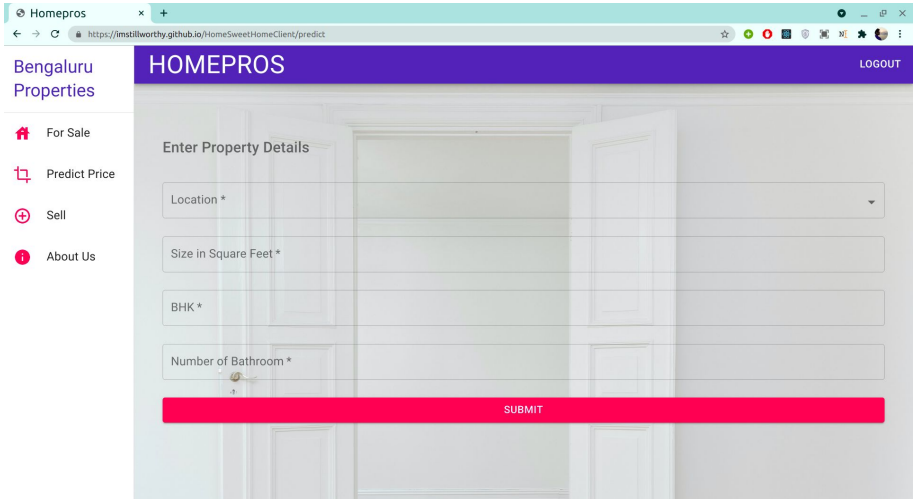



Home Page

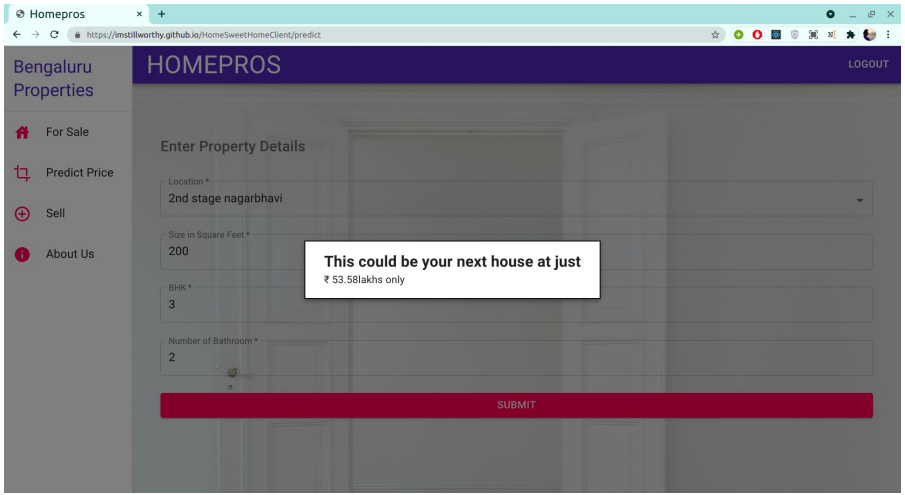



Fully Responsive

# Results and Analysis

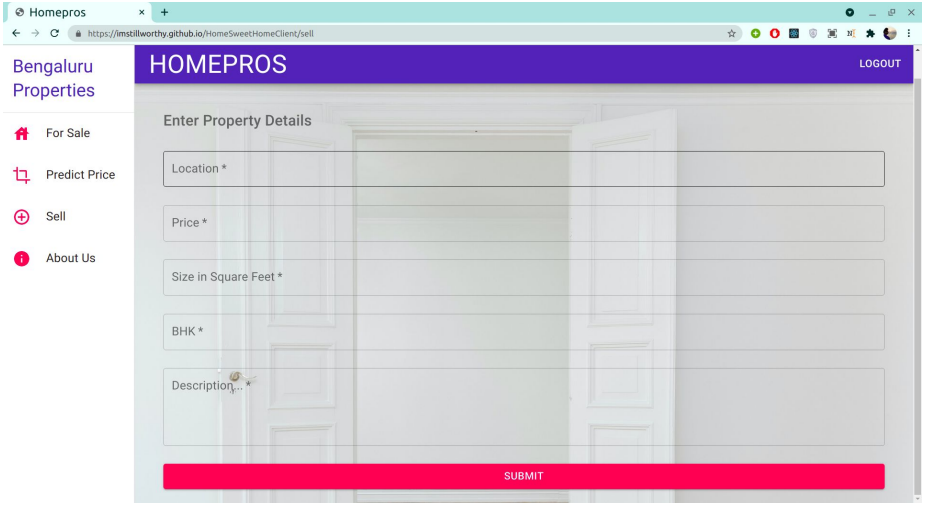



Predict Page

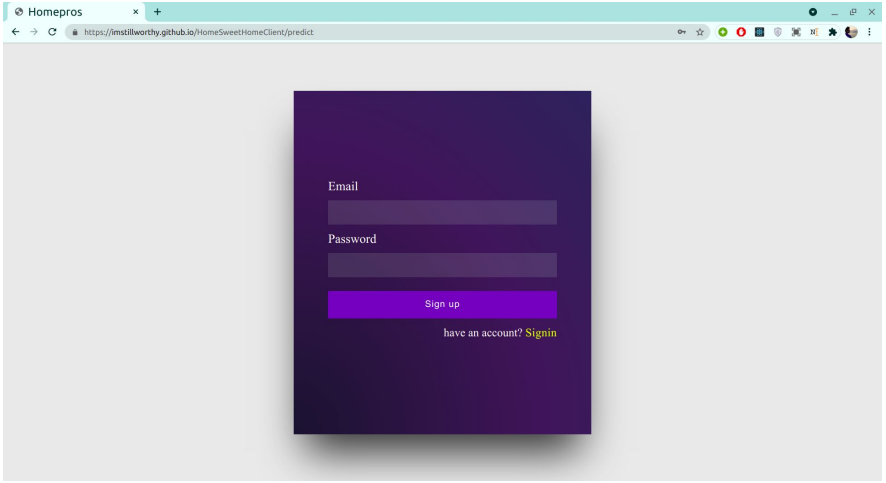



After prediction

# Result and Analysis




Sell a property



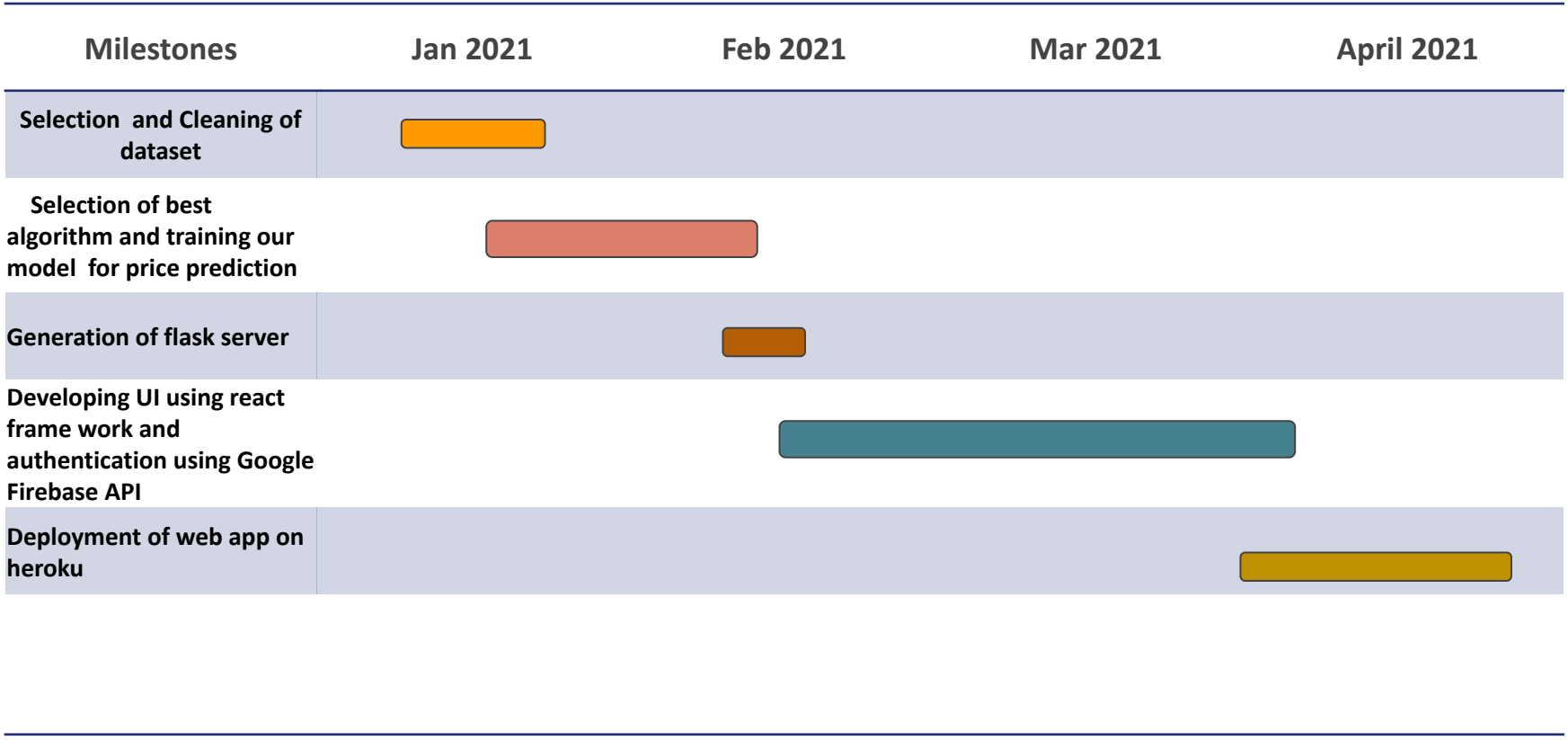

Signup and signin page

# Results and Analysis

Our Site is hosted at (deployment) :

https://imstillworthy.github.io/HomeSweetHomeClient

# Time-Line of Project



# Individual Contribution

**Selection and cleaning of dataset**: Rishit,Sarthak,Yash Gupta

**Selection of best Algorithm using Grid Search CV** : Rishit ,Sarthak

**Generation of flask server and pickle file**: Yash Gupta, Rishit

**Designing Authentication page using react and Firebase API** : Sarthak

**Designing Home Page and integration with database** : Yash Gupta

**Designing Sell Property Page and About us page** :Rishit

**Deployment of backend server on heroku** : Rishit,Sarthak,Yash Gupta

**Deployment of front end on Github pages** : Rishit,Sarthak,Yash Gupta

# References


- V. Limsombunchai, —House price prediction: Hedonic price model vs artificial neural network,‖ Am. J. …, 2004.
- M. Risdal, “Predicting House Prices Playground Competition: Winning Kernels”, 2017.
- http://blog.kaggle.com/2017/03/29/predicting-house-prices-playgroundcompetition-winning-kernels/ [Accessed: 25-Mar-2019].
- De Cook, Dean. “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project.” Journal of Statistics Education, vol. 19, no. 3, 2011.


.