MAR IVANIOS COLLEGE (AUTONOMOUS)

Mar Ivanios Vidya Nagar, Nalanchira
Thiruvananthapuram-695015

B.Sc. Computer Science Major Project Report

HOUSE PRICE PREDICTION SYSTEM USING MACHINE LEARNING

Submitted in partial fulfillment of the requirement for the Sixth Semester
B.Sc. Computer Science

SUBMITTED BY

B AMALA FRANCIS	**(Regno: 2200802)**
REHNA M	**(Regno: 2200808)**
SHYLESH KUMAR S	**(Regno: 2200831)**

Under the guidance of
Ms. Tinu C Phillip.

DEPARTMENT OF COMPUTER SCIENCE
B.Sc Computer Science

2023

MAR IVANIOS COLLEGE (AUTONOMOUS)

Mar Ivanios Vidya Nagar, Nalanchira

Thiruvananthapuram-695015

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the project entitled "HOUSE PRICE PREDICTION SYSTEM USING MACHINE LEARNING" is a bonafide record of the work done by B AMALA FRANCIS (Reg no: 2200802), REHNA M (Reg no: 2200808), SHYLESH KUMAR S (Reg no: 2200831), in partial fulfillment of the requirements for the award of Bachelor of Science Degree in Computer Science by the University of Kerala.

INTERNAL GUIDE **HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINERS

1.

2.

ACKNOWLEDGEMENT

We would like to express our genuine and heartfelt gratitude to everyone who has assisted us in this quest. Without their active advice, help, collaboration and encouragement, we would not have achieved headway in the project.

First and foremost, we thank God for keeping us safe and well so that we could successfully finish the first part of our project.

We would like to convey our heartfelt gratitude and respect to our Principal **Dr. JIJIMON K. THOMAS** and Director **Dr. K. OOMMACHAN** for providing all essential facilities.

We would like to express our gratitude to **Ms. TINU C. PHILIP**, Head of the Department, Computer Science, for all of her assistance and support in carrying out this project.

We are really grateful to our project guide, **Ms. TINU C. PHILIP**, for her important direction and assistance in completing the project in its current state.

We also express our profound thanks to faculty members, our parents and family members, who have always provided spiritual as well as financial assistance.

Last but not least, we would like to express our appreciation to all of our friends who directly or indirectly assisted us in completing this project report.

Any omissions in this brief acknowledgement do not imply a lack of gratitude.

ABSTRACT

The sales of homes are based on a number of variables, including location, neighborhood, population, and certain data used to estimate the price of each individual home. In addition to these housing costs, forecasting housing costs can be very helpful in predicting future housing costs for real estate. Every year, housing costs increase, necessitating the development of a real estate forecasting system. A developer can decide the selling price of a property with the aid of a price estimate, and clients can set a fair timeline for home purchases. In order to construct a housing price prediction, this study employs machine learning algorithms and technology as a research approach. We will be using Regression Models in Machine Learning.

Finally, we can state that the House Price Prediction System will be highly beneficial in identifying house prices and maintaining a record of the high and low of prices. As a result, it will assist the user in understanding the true value of the item and cannot be used in any fraudulent manner.

Keywords: House price prediction, Machine Learning, Linear regression, Random Forest.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Prior to now, it was highly normal and popular to value a property without properly assessing the land, infrastructure, etc. We require accurate home market predictions since we can identify a common mechanism running through all of the properties. buying and selling for the majority of people, owning a home is a lifelong dream. However, in India, many people make mistakes by purchasing real estate from strangers after hearing about it in the media. People in India purchase properties that are overpriced yet are not worthwhile. The average price of a house sold in India in 2021 was around 80 lakhs, however the real price for the area and size was closer to 60 lakhs.

The economic collapse that occurred earlier in the year served as a warning of the impending catastrophe. This situation is currently taking place, and house prices are rising relative to the state of the economy in our nation. However, because the Indian government does not provide data on house prices, it has been very challenging for people to purchase real estate. People then looked for evidence of house pricing on the internet.

We are attempting to anticipate the real estate price for the future using machine learning approaches with the aid of the prior works. Many methods, such as advance regression, have been applied in price prediction. To forecast the price of a house, we used multiple regression and additional algorithms with various tools. The Project's primary objective is to forecast the effective home pricing for real estate clients taking into account their priorities and budgets.

2. LITERATURE REVIEW

Paper:1

Paper Name: Comparison of Ensemble Methods for Real Estate Appraisal

Author: Prathamesh Kumar, Ishan Madan, Ashutosh Kale

In this paper, four ensemble methods, namely Bagging, Random Forest, Gradient Boosting, and Extreme Gradient Boosting were analyzed and compared in terms of their efficiency in the appraisal of real estate in Mumbai. The property listings available on the real estate website 99acres were used as the data source for this study. The analysis showed that Extreme Gradient Boosting (XGBoost) model performed the best as compared to the rest of the ensemble models. The results confirm that ensemble models can be useful for estimating real estate prices.

Paper:2

Paper Name: Prediction of House Pricing Using Machine Learning with Python

Author: Mansi Jain, Himani Rajput, Neha Garg

This paper provides an overview of how to predict house costs utilizing different regression methods with the assistance of python libraries. The proposed technique considered the more refined aspects used for the calculation of house prices and provide a more accurate prediction. It also provides a brief about various graphical and numerical techniques which will be required to predict the price of a house. This paper contains what and how the house pricing model works with the help of machine learning.

Paper:3

Paper Name: House Price Forecasting using Data Mining Techniques

Author: Atharva Chogle, Priyanka Khaire, Akshata Gaud, Jinal Jain

This project uses data mining algorithm to predict prices by analyzing current house prices, thereby forecasting the future prices according to the user's requirements. It uses Naïve Bayes Algorithm for finding the house price. Nave Bayesian is a statistical learning algorithm based on Bayes rule to compute joint probability. This paper contains the administrator and user phase, the administrator will add property details into the system based on the details the system will predict the estimated price, and the user searches property the list of property will be displayed to the user along with the predicted price the user can sell his property by adding his details onto the system.

Paper:4

Paper Name: Prediction of Real Estate Price based on Economic parameters

Author: Li, Li, Kai-Hsuan Chu.

The hedonic model fundamentally involves regression technique that believes various parameters such as property area, age, bedrooms number and so on. The Neural Network is trained to begin with and the weights and biases of the edges and nodes in that order are measured using trial and error method. Training the Neural Network model is a black box technique. However, the RSquared value for Neural Network model was greater compared to hedonic model and the RMSE value of Neural Network model was reasonably lower.

3. SYSTEM ANALYSIS

3.1 IDENTIFICATION OF NEED

There are numerous real estates classified websites in India, including 99acres, no broker, housing, magic bricks, and many others, where properties are offered for sale, purchase, or rental. However, there are many pricing irregularities in each of these websites, and there are certain instances where comparable properties are priced differently, which results in a lack of transparency and accuracy. Customers may occasionally believe that a particular listed house's worth is not justified, but there is no means to verify or otherwise verify the accuracy of the information.

Since most consumers, particularly in India, find the transaction costs to be quite high, addressing this issue will benefit both the customers and the real estate industry in the long run. Proper evaluations and justified prices of properties can restore a great deal of transparency and trust to the real estate industry.

Disadvantages of Existing System:

1. HUMAN resource: - The current system has too much manual work from filling a form to filing a document, delivering manifesto. This increases burden on workers but does not yield the results it should.

2. THORNY Job: - In current system if any modification is to be made it increases manual work and is error prone.

3. ERROR: - As the system is managed and maintained by workers errors are some of the possibilities.

3.2 EXISTING SYSTEM

There is a notable amount of research done in the house price prediction department but very research has come up to any real-life solutions. There is very little evidence of a working house price predictor set up by a company. For now, very few digital solutions exist for such a huge market and most of the methods used by people and companies are as follows:

Buyers/Customers:

1. When people first think of buying a house/Real estate they tend to go online and try to study trends and other related stuff. People do this so they can look for a house which contains everything they need. While doing these people make a note of the price which goes with these houses. However, the average person doesn't have detailed knowledge and accurate information about what the actual price should be. This can lead to misinformation as they believe the prices mentioned on the internet to be authentic.
2. The second thing that comes to mind while searching for a property is to contact various Estate agents. The problem with this is these agents need to be paid a fraction of the amount just for searching a house and setting a price tag for you. In most cases, this price tag is blindly believed by people because they have no other options. There might be cases that the agents and sellers may have a secret dealing and the customer might be sold an overpriced house without his/her knowledge.

Seller/Agencies:

1. When an individual thinks of selling his/her property they compare their property with hundreds and thousands of other properties which are posted all around the world. Determining the price by comparing it with multiple estates is highly time-consuming and has a potential risk of incorrect pricing.
2. Large Real estate companies have various products they need to sell and they have to assign people to handle each of these products. This again bases the prediction of a price tag on a human hence there is room for human error. Additionally, these assigned individuals need to be paid. However, having a computer do this work for you by crunching the heavy numbers can save a lot of time money and provide accuracy which a human cannot achieve.

3.3 PROPOSED SYSTEM

E-learning and e-education are heavily affected today. Automation is replacing manual systems everywhere. This project's goal is to forecast house prices in order to lessen the difficulties the consumer would experience. The customer currently consults a real estate agent to manage his or her investments and make proper for his investments, estates. However, this approach carries some risk because the agent could make a mistaken estate prediction and lose the customers' capital as a result. The manual approach that is still prevalent in the market is risky and out-of- date. There is a need for an updated, automated system to address this flaw. ML algorithms can be used to guide investors toward making an acceptable real estate investment based on their stated needs. The new system will also save money and time. Its operations will be straightforward. The regression algorithm is used by the suggested system. The user will give property details, based on that the system will predict the price.

The purpose of this system is to determine the price of a house by looking at the various features which are given as input by the user. These features are given to the ML model and based on how these features affect the label it gives out a prediction. This will be done by first searching for an appropriate dataset that suits the needs of the developer as well as the user. Furthermore, after finalizing the dataset, the dataset will go through the process known as data cleaning where all the data which is not needed will be eliminated and the raw data will be turned into a .csv file. Moreover, the data will go through data preprocessing where missing data will be handled and if needed label encoding will be done. Moreover, this will go through data transformation where it will be converted into a NumPy array so that it can finally be sent for training the model. While training the model we are using regression model that is Linear regression model which is used to make a system which predicts the price of the house by taking the parameters that the user has been entered and by comparing with values which is present in the dataset that's previously available. If the linear regression has much error rate, we just use Random Forest regression. By comparing the error rate of these 2 algorithms we will choose the best one as the final model which can yield accurate predictions. For knowing whether our model is good or not, we are using evaluation matrices like Mean Square Error, Root Mean Square Error and R Square Error.

We use Html and CSS for creating a website interface. We use this method because it is easy for the user to communicate with the system. Inside the website we create 4 textboxes and one button. For entering the details of the house such as *total sq feet, no of rooms etc.* we are using the textbox and for submitting we use the button. We also include a textbox for printing the predicted price of the house by using the regression methods.

3.3.1 Proposed Methodology

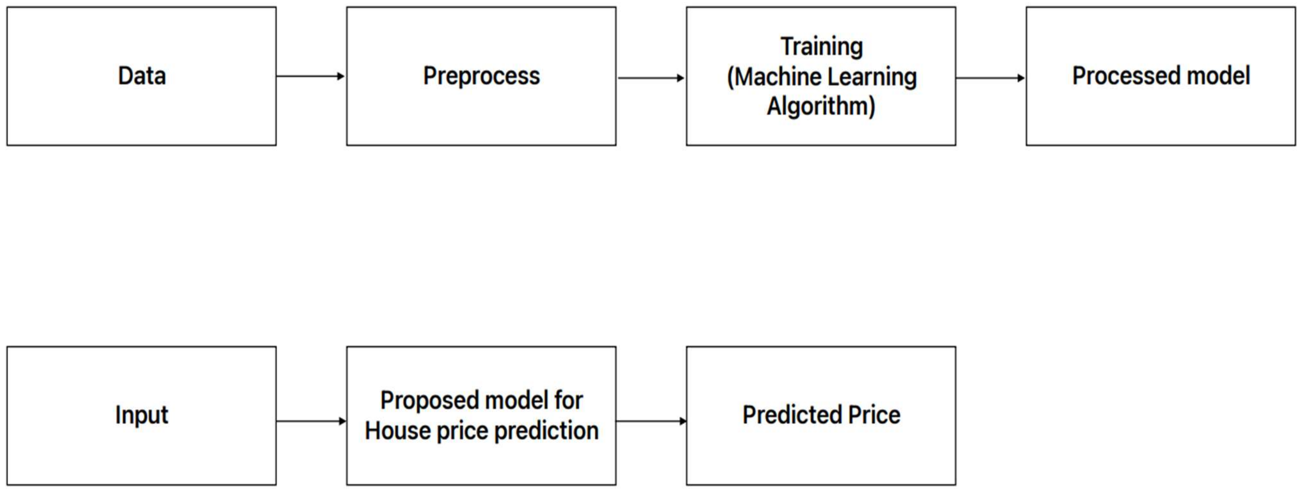


Figure 1:Proposed Methodology

3.3.2 State Chart Diagram of the System:

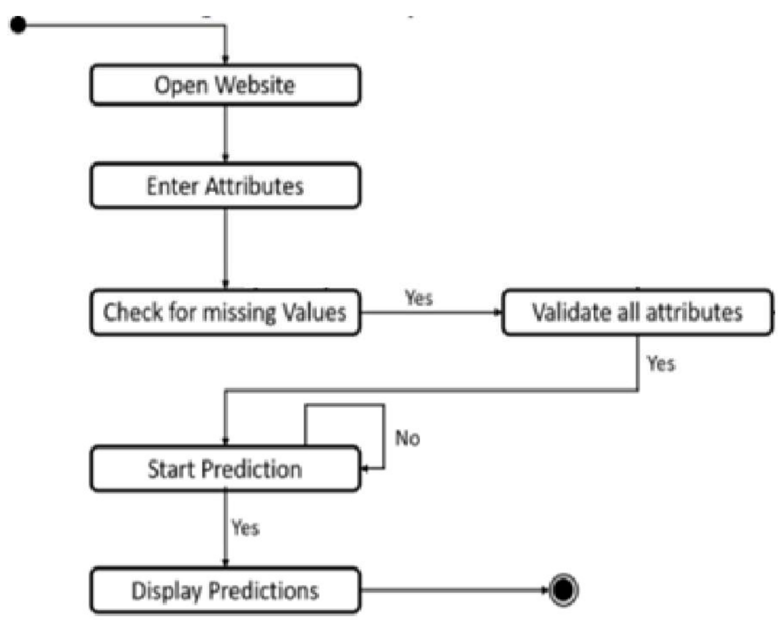


Figure 2:State Chart Diagram

4. FEASIBILITY STUDY

The very first phase in any system developing life cycle is preliminary investigation. The feasibility study is a major part of this phase. A measure of how beneficial or practical the development of any information system would be to the organization is the feasibility study.

4.1 OPERATIONAL FEASIBILITY

The System will reduce the time consumed to maintain manual records and is not tiresome and cumbersome to maintain the records. Hence operational feasibility is assured.

4.2 TECHNICAL FEASIBILITY

Minimum hardware requirements: - i3 Processor or Intel compatible processor. 1 GB RAM. Internet Connectivity. 2GB hard disk space.

4.3 ECONOMIC FEASIBILITY

Once the hardware and software requirements get fulfilled, there is no need for the user of our system to spend for any additional overhead. For the user, the Website will be economically feasible in the following aspects: The system will reduce a lot of labor work. Hence the Efforts will be reduced. Our application will reduce the time that is wasted in manual processes.

5. SYSTEM PROCESS

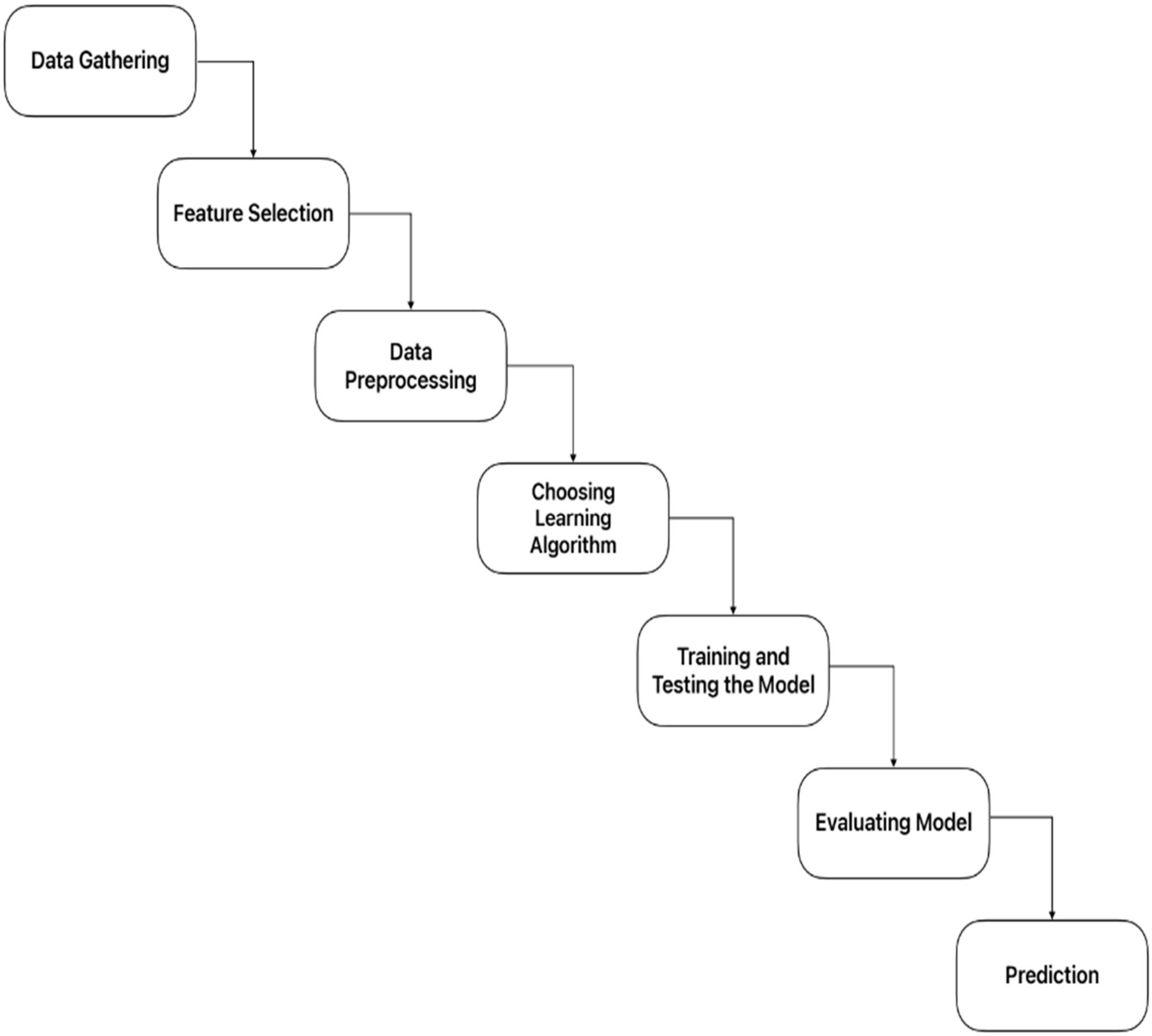


Figure 3:Flow of Architecture

5.1 DATA GATHERING

Data collection is the process of gathering and measuring information from countless different sources. These data can be numeric, categorical, or even free text. All these data will be converted to a structured format to use on the ML models.

5.1.1 The Dataset

Data preparation is the primary step for any machine learning problem. We will be using our own dataset for this problem. This dataset consists of a CSV file. The dataset consists of 1998 records and 6 attributes. The columns with nominal attributes are location, rooms, total_sqfeet, bath, district, price. These data are collected by using web scraping technology.

	A	B	C	D	E	F
1	location	size	total_sqft	bath	district	Price
2	Jawahar N	4 BHK	2700	5	Trivandrun	40000000
3	Vanchiyoo	5 BHK	3500	5	Trivandrun	27500000
4	Vanchiyoo	5 BHK	3700	5	Trivandrun	27500000
5	Kowdiar	3 BHK	1650	4	Trivandrun	8000000
6	Pattom	4 BHK	3800	4	Trivandrun	32500000
7	Kowdiar	3 BHK	1500	3	Trivandrun	5800000
8	Pattom	5 BHK	3000	5	Trivandrun	35000000
9	Kesavadas	4 BHK	3500	5	Trivandrun	21000000
10	Pattoor	5 BHK	3500	5	Trivandrun	27500000
11	Kowdiar	4 BHK	3500	4	Trivandrun	32500000
12	Kesavadas	4 BHK	1900	4	Trivandrun	8800000
13	Kowdiar	4 BHK	4000	4	Trivandrun	22500000
14	Kowdiar	4 BHK	3500	5	Trivandrun	21000000
15	Kowdiar	4 BHK	3500	4	Trivandrun	49000000
16	Pattom	4 BHK	2500	4	Trivandrun	13800000
17	Kowdiar	3 BHK	2100	3	Trivandrun	9800000
18	Thampanc	4 BHK	2200	4	Trivandrun	15500000
19	Kesavadas	4 BHK	1800	4	Trivandrun	7900000
20	Pattom	4 BHK	3500	4	Trivandrun	32500000
21	Kallara	5 BHK	6000	5	Trivandrun	5000000
22	Pattom	3 BHK	1600	3	Trivandrun	6900000
23	Vanchiyoo	5 BHK	3800	4	Trivandrun	27700000
24	Kowdiar	4 BHK	3600	5	Trivandrun	21000000
25	Pattom	4 BHK	3800	5	Trivandrun	32500000
26	Kowdiar	4 BHK	2000	4	Trivandrun	9500000
27	Kowdiar	2 BHK	1200	2	Trivandrun	4000000
28	Melarannc	2 BHK	450	1	Trivandrun	2300000
29	Vanchiyoo	4 BHK	2500	5	Trivandrun	20000000
30	Ulloor	5 BHK	5500	5	Trivandrun	22500000

Kerala_house_Shylesh_tvm

Figure 4:Dataset

In here we use different tools to extract the data from multiple websites. Mainly we used "olx.com" website to scrap the data. We used different tools for web scraping such as 'Parsehub', 'Instant Data Scraper' etc. In our project we only collect the details of the houses which are in the Trivandrum locality. So basically, our project determines the price of the houses which are in the TVM district.

5.1.2 Web Scraping and Tools

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, Stack Overflow, etc. have API's that allow you to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data.

Web scraping requires two parts, namely the **crawler** and the **scraper**. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website. The design of the scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data.
These automated tools are known as **Web Scrapers**.

In here we used 2 scrapers,

- Parsehub
- Instant Data Scraper

Parsehub

It is an application-based software which is used to scrap the data by giving the details of the website.

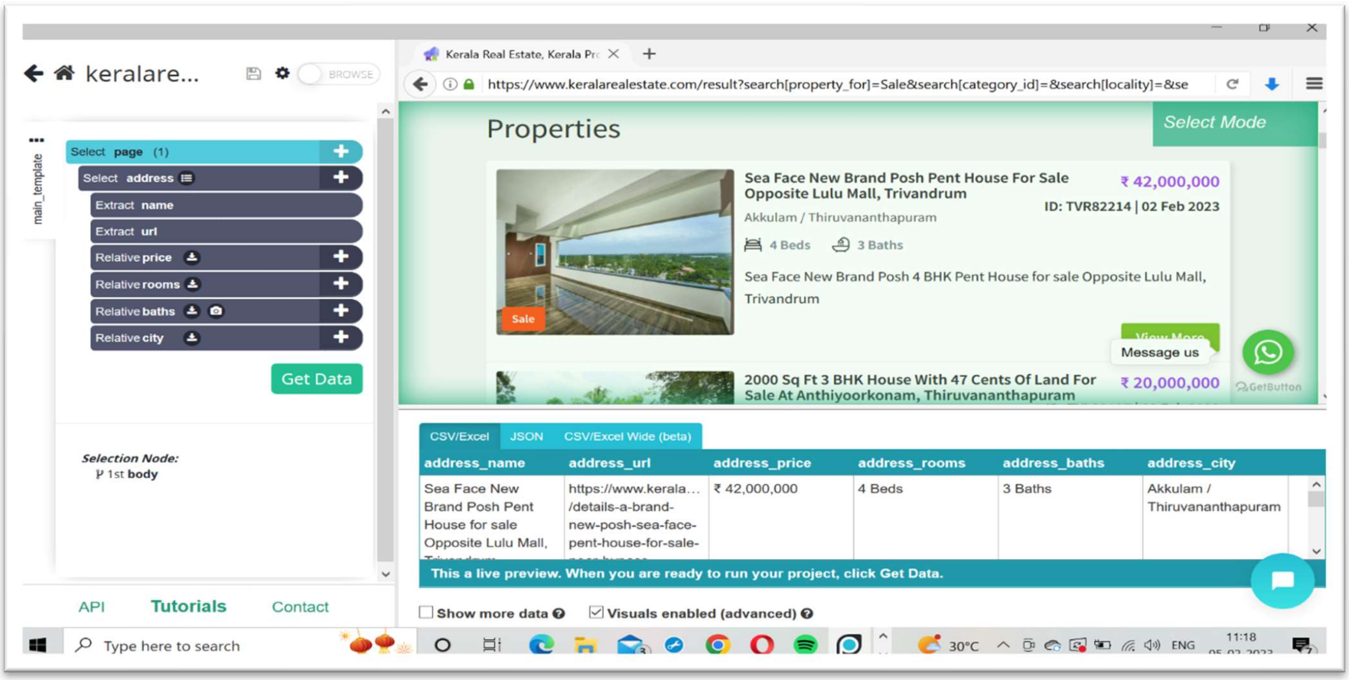


Figure 5:Parsehub

Instant Data Scraper

It is a chrome extension and it is an automated data extraction tool for any website.

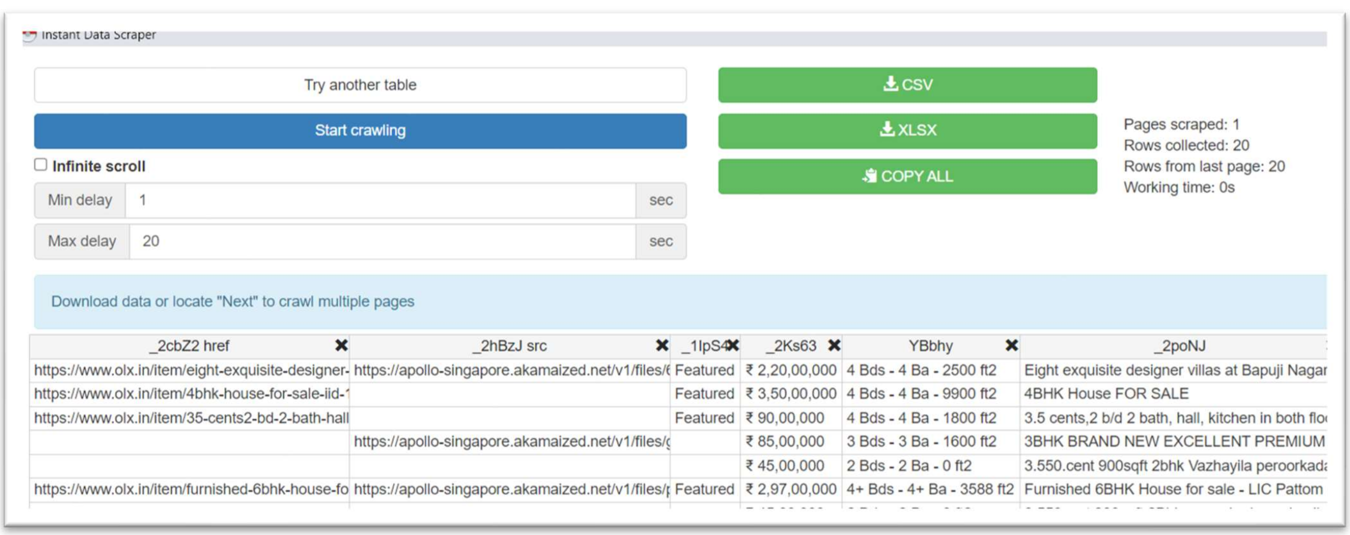


Figure 6:IDS

5.2 DATA CLEANING

Checking for missing values in any of the rows of the raw dataset is necessary to clean the data. The data we gather is typically erratic. It could have blank fields, inaccurate data, and outliers. The accuracy of the forecast made by the model may be negatively impacted by this type of data. Thus, it is crucial to eliminate any such noisy data.

5.3 DATA PREPROCESSING

In this phase, we need to convert our raw dataset into a structured form so that it is appropriate for training a machine learning model. it is necessary that all the independent variables are storing information in the form of numbers and not text.

- To convert this into numerical data so that ‘yes’ is represented by the number 1 and ‘no’ is represented by the number 0, we can use the ‘*LabelBinarizer*’ function available in the scikit-learn python library.
- To convert this text data into numerical data I used the concept of ‘*one hot encoding*’. After performing all the above tasks, all the information in the dataset is in the form of numerical data and it is qualified to train the model.

5.4 TRAINING AND TESTING THE MODEL

For training the model, 80% of the dataset will uses and for testing the model 20% of the dataset will use. The process of training an ML model involves providing an ML algorithm (that is, the *learning algorithm*) with training data to learn from. The term *ML model* refers to the model artifact that is created by the training process.

The training data must contain the correct answer, which is known as a *target* or *target attribute*. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

5.4.1 Algorithms Used

Linear Regression in Machine Learning

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables.

Consider the below image:

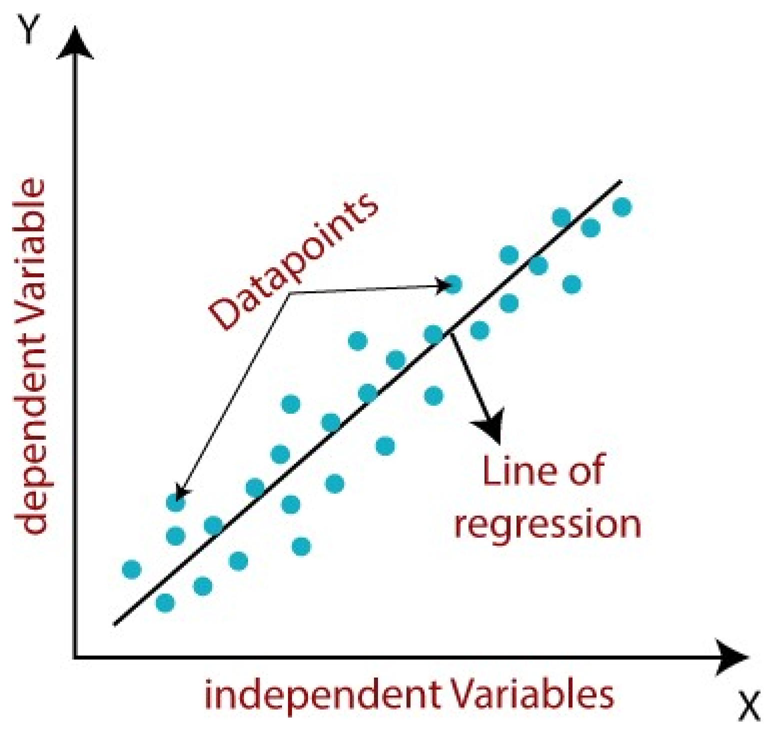


Figure 7: Linear Regression Graph

Mathematically, we can represent a linear regression as:

$$y = a_0 + a_1 x + \varepsilon$$

Here,

Y= Dependent Variable (Target Variable)

X= Independent Variable (predictor Variable)

a0= intercept of the line (Gives an additional degree of freedom)

$= (N\sum XY - (\sum X)(\sum Y) / (N\sum X^2 - (\sum X)^2)$

a1= Linear regression coefficient (scale factor to each input value).

$= (\sum Y - b \sum X) / N$

N= No of Observations

e= random error

The values for x and y variables are training datasets for Linear Regression model representation.

Types of linear regression:

Linear regression can be further divided into two types of the algorithm:

- Simple Linear Regression: -If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- Multiple Linear regression: -If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Cost function: When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

5.4.2 Working of Algorithm

In our project the linear regression is using as the model:

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables.

In here,

- The dependent variable(y): Prices of the house.
- The independent variable(x): Parameters such as sq feet, no of rooms, floors etc.

Let us consider the graph given below:

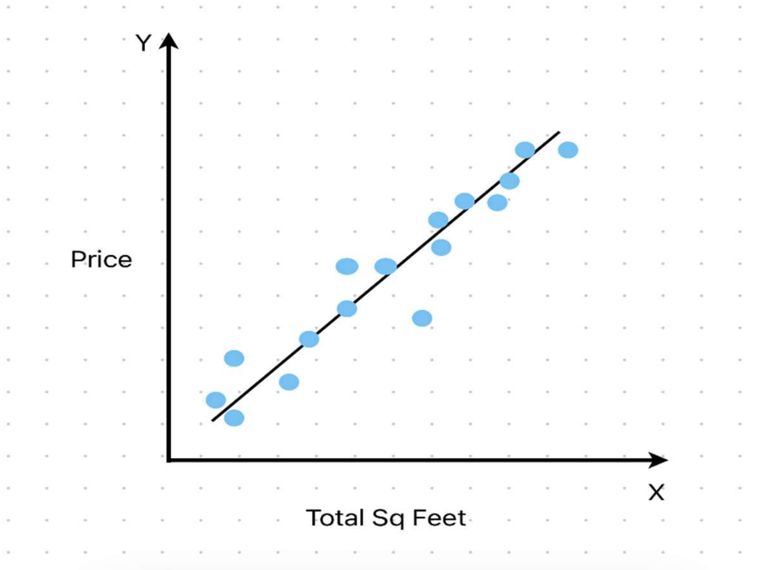


Figure 8:Linear regression graph showing relationship b/w Total sq feet and price

The graph is plot by using observations which shows the relationship between Sq feet and the price. A regression line depicts the relationship between these two variables. It is applied in scenarios where the change in the value of the independent variable causes changes in the value of the dependent variable. Consider the Eq:

$$y= a_0+a_1x+ \varepsilon$$

Assume the no of observations is taken as 5 ->N

From the above graph we can modify the equation as:

- y= Price of the house
- a_0, a_1= By using equation calculating a0 and a1
- x= Amount of Sq feet in graph
- ε= If any error present in the data

By using this method, we predict the price of the house. After the prediction the predicted value will be move for the Evaluation matrices to check the accuracy of the model.

- [If our dataset is not satisfied with the linear regression model, we just need to try another algorithm. Because of that, we are just referring to one more algorithm called the Random Forest Algorithm.]

Random Forest Regression

Random Forest is a supervised learning algorithm. The decision tree is contemplated as a base of the Random Forest algorithm. Random Forest can be used for regression as well as classification just like the Decision Tree algorithm. For the classification model it works with a categorical target, whereas in the regression model, it predicts the values of a continuous variable. Random Forest is a collection of several decision trees, just like a real forest that consists of trees. It is often also referred to an ensemble learning technique. For the processing, a set of samples of data are picked arbitrary which comprises different decision trees. Each tree gives the prediction, and the average of them is selected for the case in the regression model. In this paper, the regression algorithm of random forests is used. First of all, it divides the dataset into multiple sets, each set produces a decision tree based on different evaluation metrics used. Later the mean of all the predictions is considered as the final prediction. It is more clearly shown in Figure 4

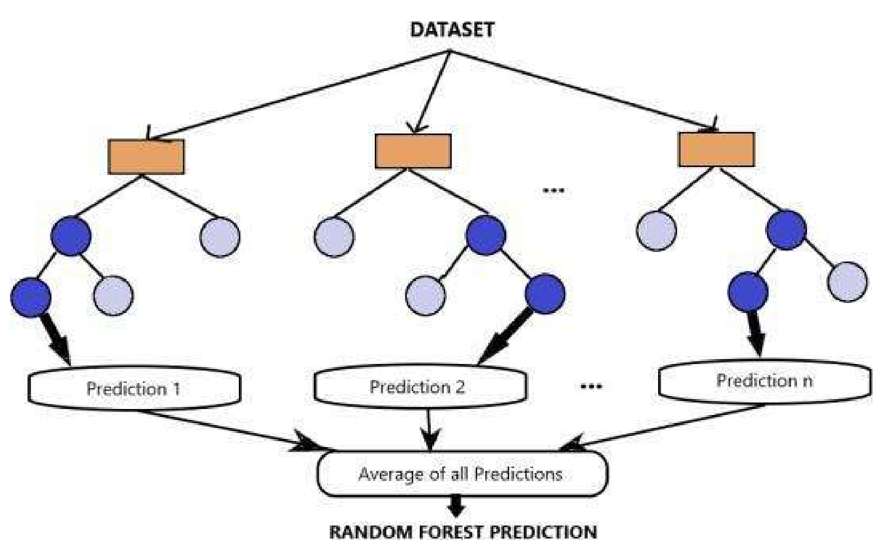


Figure 9: Random Forest

5.5 EVALUATION METRICES

Machine learning refers to the analysis of a certain dataset, which on processing with different algorithms, gives certain predictions. Accuracy prediction depends on the type of model. For a regression model, the evaluation metrics used are Root Mean Square Error, Mean Squared Error, Mean Squared Log Error, and R2 Score. The fallacy in the predictions is represented by the errors. The main conception includes that there will be a comparison between predicted and actual values to calculate evaluation metrics. However, in this study, the performance of the algorithms is measured by using MSE, RMSE and R-Squared.

5.5.1 Mean Square Error (MSE)

Is the average of squared error occurred between the predicted values and actual values. It can be written as: is the average of squared error occurred between the predicted values and actual values. It can be written as:

$$\text{MSE}=\frac{1}{k}\sum_{i=1}^{k}(xi - \hat{x}\text{i})^2$$

Where, k is the number of predictions, x refers to the actual values, and $\hat{x}$ the predicted values.

5.5.2 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is used as an evaluation metric in machine learning to measure the performance of the model. However, RMSE is similar to the Mean Square Error (MSE). Where all errors in MAE have the same weight, but RMSE penalizes the variance, which means it gives more weight to the errors that have large absolute values than that have small absolute values. Therefore, when RMSE and MAE are calculated, RMSE is always bigger than MAE. RMSE is more sensitive to the errors than MAE; therefore, using RMSE for measuring the performance is better than MAE. RMSE can be calculated as the square root of the sum of squared errors

$\sum (xi - \hat{x}i)^2$ over the sample size k.

Mathematically.

$$\text{RMSE} = \sqrt{MSE} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(xi - \hat{x}\text{i})^2}$$

It can be observed from the equation when the sum of squared errors is closer to zero, RMSE is closer to zero. Therefore, when RMSE is zero, it means there are no errors between the actual value x and the predicted value $\hat{x}$.

5.5.3 R² Error

R squared or the coefficient of determination is the measure of the closeness of predicted data with the actual data. It varies from 0 to 1, represented as a percentage. Higher the value of R squared, more accurate is the prediction model. In addition, R-Squared is calculated by measuring the deviations of the observations from their predicted values over the measurement of the deviations of the observations from their mean. It can also be explained as,

$$R^2 = 1 - \frac{\sum(xi - \hat{x}\text{i})^2}{\sum(xi - \bar{x}\text{i})^2}$$

Where $\bar{x}$i is the mean value of x.

6. SYSTEM SPECIFICATION

6.1 REQUIRED TOOLS

6.1.1 Hardware Requirements

- Processor – (minimum)i3
- Hard Disk – 2 GB
- Memory – 1GB RAM

6.1.2 Software Requirements

- Windows 7(ultimate, enterprise)
- Python
- Anaconda
- Jupyterlab

6.1.3 Language Description

Python:

Python is an interpreted, high-level, general-purpose for programming language. Created by Guido Van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is often described as a "batteries included" language due to its comprehension standard library.

Following are the important characteristics of Python Programming.

- It supports functional and structural programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA and Java.

NumPy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

Advantages

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be loaded.
- Easy handling of missing data (represented as Nan) in floating point as well as non-floating-point data
- Size mutability: columns can be inserted and deleted from Data Frame and higher dimensional objects
- Data set merging and joining.
- Flexible reshaping and pivoting of data sets
- Provides time-series functionality.
- Powerful group by functionality for performing split-apply-combine operations on data sets.

7. SYSTEM DESIGN

7.1 FRONT END

The front end is basically the structure or a build up for a website. In this to receive an information for predicting the price. It takes the form data entered by the user and executes the function which employees the prediction model to calculate the predicted price for the house**.**

7.1.1 Layout of Webpage Before Prediction

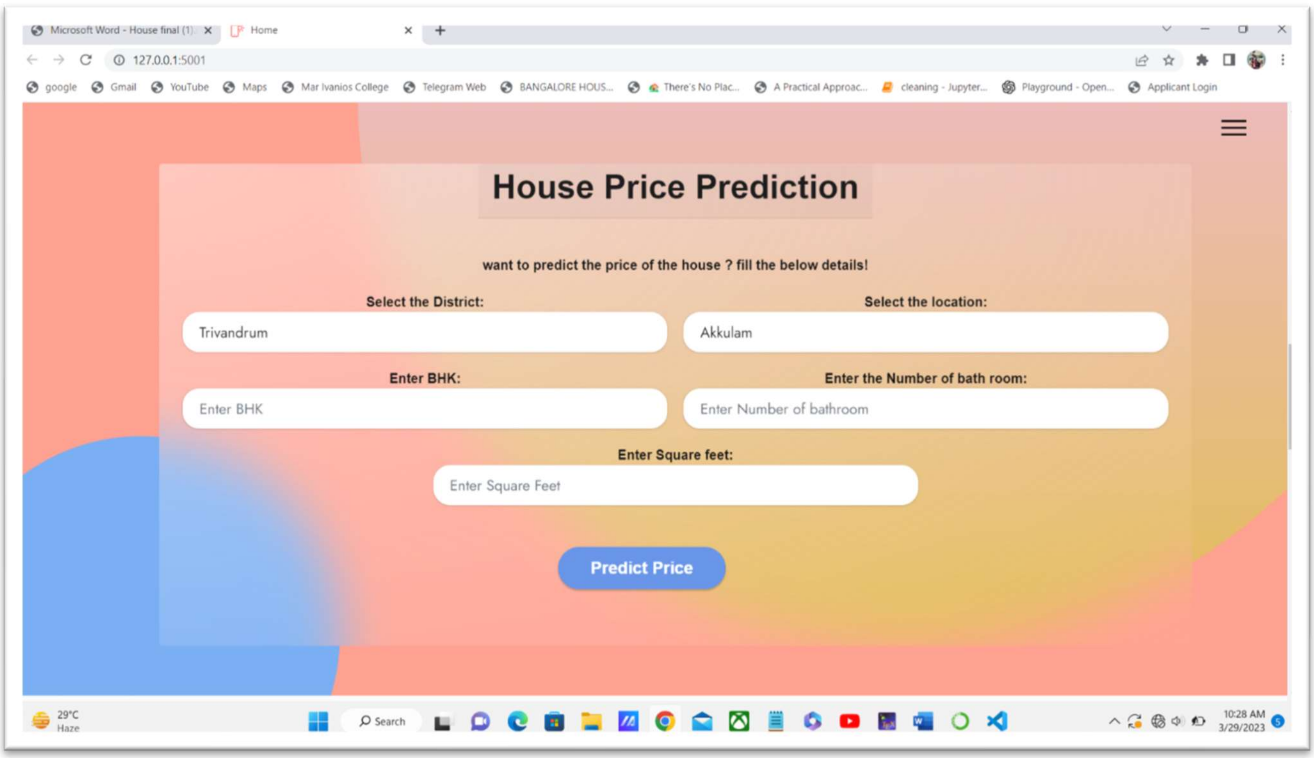


Figure 10: Front End

It consists of the parameter columns to enter the details of the house such as location, no of bedroom etc.

After entering all the details, the user can press the predict price by clicking on the button (Predict Price).

7.1.2 Layout of Webpage After Prediction

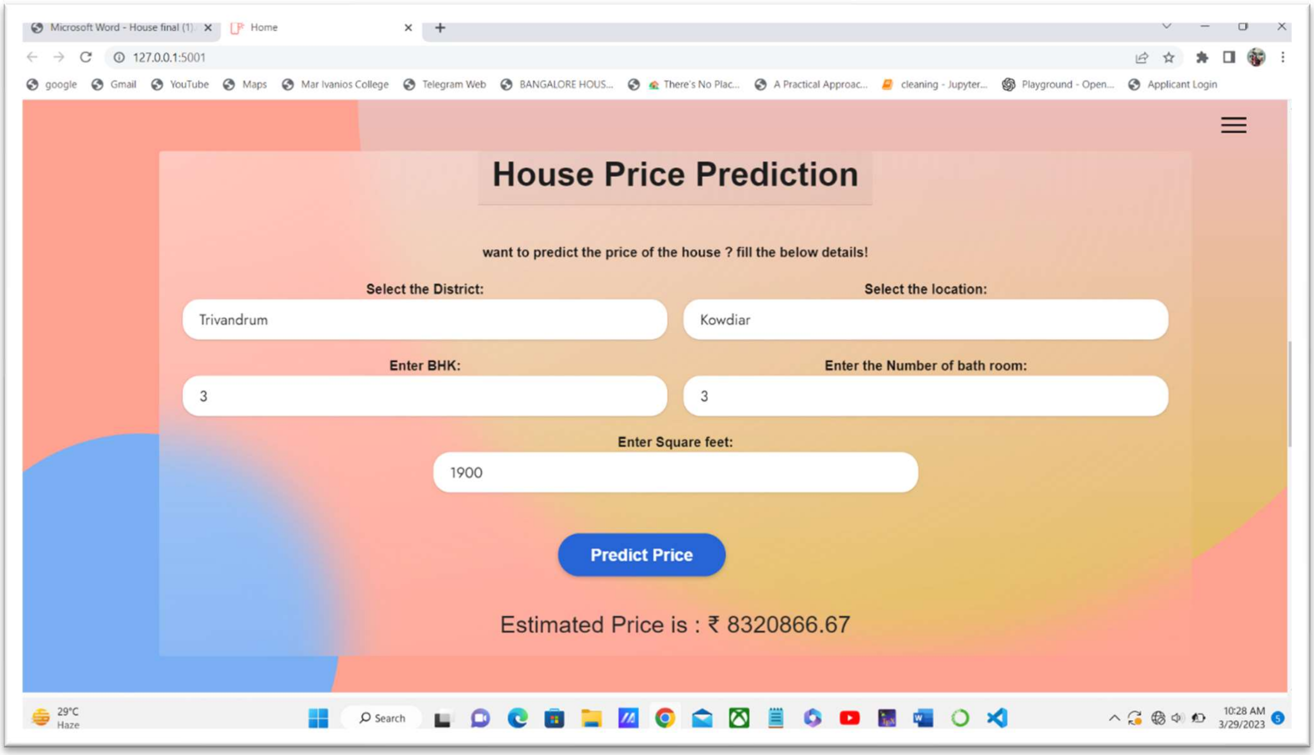


Figure 11: Front End After Prediction

After prediction, the resultant predicted value will be displayed on webpage as 'Estimated Price'.

7.2 USE CASE DIAGRAM (USER AND SYSTEM)

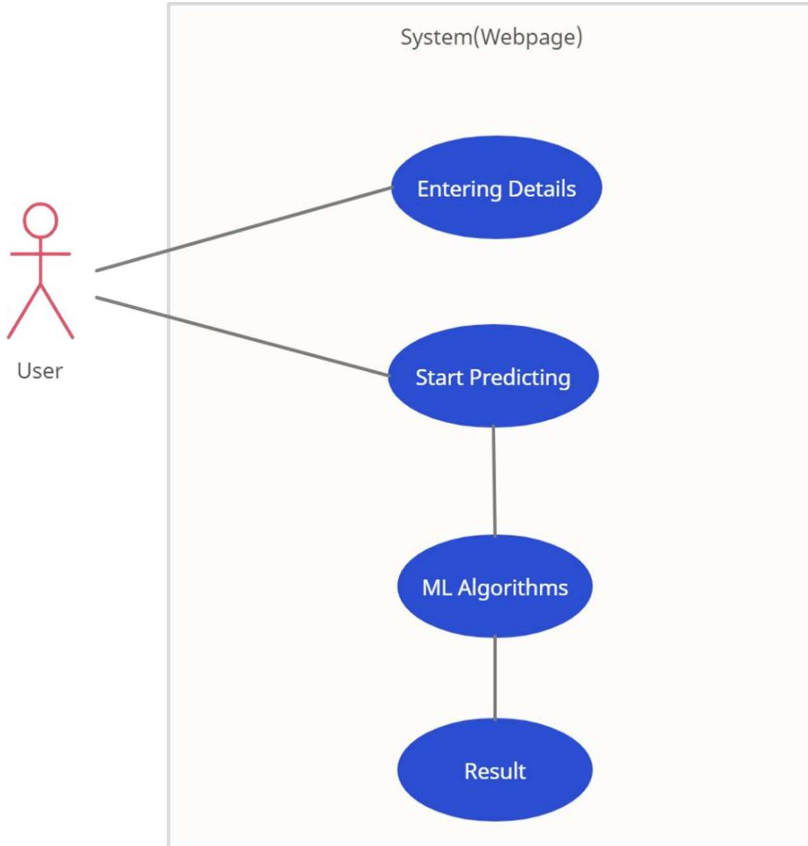


Figure 12: Use Case Diagram

Sequence: -User can enter the details of property in the webpage. After that user can move to the prediction. In the prediction stage the system uses ML algorithms and provides the result.

7.3 FLOWCHART

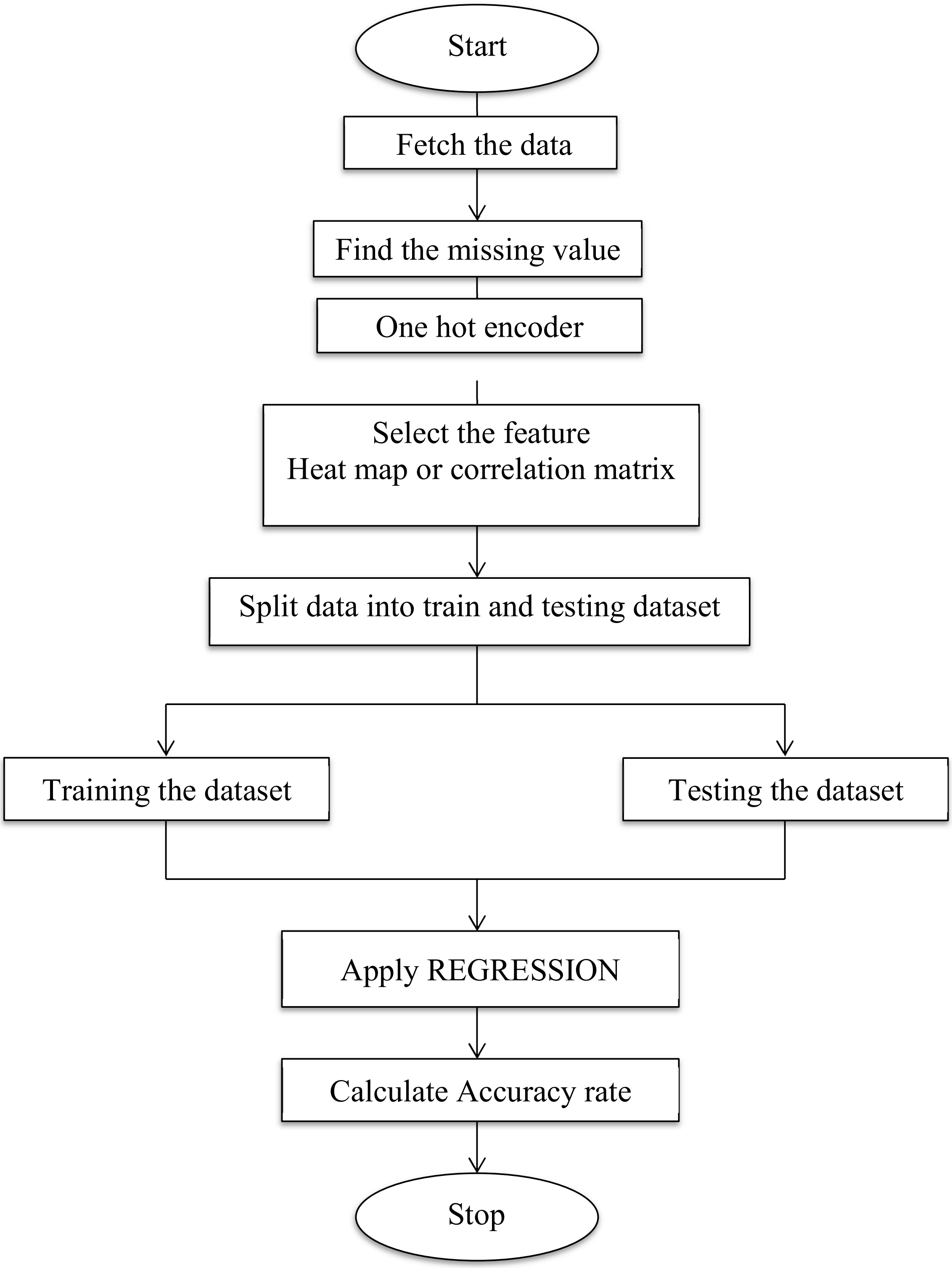


Figure 13:Flow chart

8.IMPLEMENTATION

ML MODEL:

Data Preparation:

The first step is to prepare the dataset for the analysis. The dataset should be cleaned, and missing values should be handled before proceeding with the analysis. The dataset should be split into training and testing sets.

Linear Regression Model:

The next step is to build a linear regression model to predict the house prices. The linear regression model uses a linear equation to predict the house price. We will use the Scikit-learn library in Python to build a linear regression model. We will train the model on the training dataset and evaluate it on the testing dataset.

Random Forest Regression Model:

The next step is to build a random forest regression model to predict the house prices. The random forest regression model uses a decision tree to predict the house price. We will use the Scikit-learn library in Python to build a random forest regression model. We will train the model on the training dataset and evaluate it on the testing dataset.

Model Evaluation:

The final step is to evaluate the performance of the linear and random forest regression models. We will use various evaluation metrics such as mean squared error, root mean squared error, mean absolute error, and R-squared value to evaluate the performance of the models.

Conclusion:

Based on the evaluation metrics, we can conclude which model is the best fit for predicting the house prices.

WEB PAGE:

Web Framework:

The web application can be built using a web framework such as Flask or Django. For this example, we will use Flask.

Front-End:

The front-end of the web page can be designed using HTML, CSS, and JavaScript. We can use a front-end framework such as Bootstrap to design the web page.

Back-End:

The back-end of the web application can be implemented using Python. We will use the Flask framework to build the back-end of the web application.

Machine Learning Model:

The machine learning model can be loaded in the Flask app and used to make the predictions. We can use the same linear regression and random forest regression models that we built earlier to make the predictions.

Deployment:

If we want to host our web application it can be deployed to a cloud hosting platform such as Heroku or AWS Elastic Beanstalk.

8.1 ML CODE

[434] :
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20, 10)
```

[435] :
```
df = pd.read_csv('Kerala_house_Shylesh_tvm.csv')
df.head()
```

[435] :

	location	size	total_sqft	bath	district	Price
0	Jawahar Nagar	4 BHK	2700	5	Trivandrum	40000000
1	Vanchiyoor	5 BHK	3500	5	Trivandrum	27500000
2	Vanchiyoor	5 BHK	3700	5	Trivandrum	27500000
3	Kowdiar	3 BHK	1650	4	Trivandrum	8000000
4	Pattom	4 BHK	3800	4	Trivandrum	32500000

[436] :
```
df.tail()
```

[436] :

	location	size	total_sqft	bath	district	Price
1192	Thirumala	5 BHK	2800	5	Trivandrum	14500000
1193	Thirumala	4 BHK	2100	4	Trivandrum	8500000
1194	Vayalilkkada	4 BHK	3300	5	Trivandrum	24500000
1195	Kariavattom	5 BHK	3795	5	Trivandrum	21000000
1196	Kazhakoottam	3 BHK	1500	3	Trivandrum	6900000

[437] :
```
df1=df['location'].value_counts()
df1
df1.to_csv("count1.csv")
```

[438] :
```
'''df1 = pd.read_csv('Count.csv')
# selecting rows based on condition
```

```
rslt_df = df1[df1['Count'] < 10]

print('\nResult dataframe :\n', rslt_df['location'])
'''
```

[438]:
```
"df1 = pd.read_csv('Count.csv')\n# selecting rows based on condition \nrslt_df =
df1[df1['Count'] < 10] \n    \nprint('\nResult dataframe :\n',
rslt_df['location'])\n"
```

[ ]:
```

```

[439]:
```
df.shape
```

[439]:
```
(1197, 6)
```

[440]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   location     1197 non-null   object
 1   size         1197 non-null   object
 2   total_sqft   1197 non-null   int64
 3   bath         1197 non-null   int64
 4   district     1197 non-null   object
 5   Price        1197 non-null   int64
dtypes: int64(3), object(3)
memory usage: 56.2+ KB
```

[441]:
```
#Cleaning
```

[442]:
```
backup=df.copy()
```

[443]:
```
df.isnull().sum()
```

[443]:
```
location      0
size          0
total_sqft    0
bath          0
district      0
Price         0
dtype: int64
```

[444]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
```

```
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   location    1197 non-null   object
 1   size        1197 non-null   object
 2   total_sqft  1197 non-null   int64
 3   bath        1197 non-null   int64
 4   district    1197 non-null   object
 5   Price       1197 non-null   int64
dtypes: int64(3), object(3)
memory usage: 56.2+ KB
```

[445]:
```
for column in df.columns:
    print(df[column].value_counts())
```

```
Kowdiar            91
Pattom             86
Thirumala          82
Kesavadasapuram    45
Karamana           34
                   ..
Sasthamangalam      1
Attingal            1
Ptp nagar           1
Mukkola             1
Kudappanakunnu      1
Name: location, Length: 165, dtype: int64
4 BHK    351
3 BHK    295
5 BHK    182
2 BHK    119
3 BHK    108
4 BHK     75
2 BHK     23
5 BHK     23
1 BHK     10
6 BHK      5
7 BHK      3
8 BHK      2
1 BHK      1
Name: size, dtype: int64
2000    62
2500    58
3500    53
1500    52
1800    51
        ..
4635     1
```

```
6120        1
3438        1
5946        1
1050        1
Name: total_sqft, Length: 215, dtype: int64
3     391
4     354
5     216
2     148
1      54
0      27
6       5
7       1
8       1
Name: bath, dtype: int64
Trivandrum     1197
Name: district, dtype: int64
7500000     48
7000000     46
9500000     41
8500000     38
6500000     29
            ..
450000       1
4430000      1
6350000      1
4350000      1
8400000      1
Name: Price, Length: 161, dtype: int64
```

[446]:

```
df.describe()
```

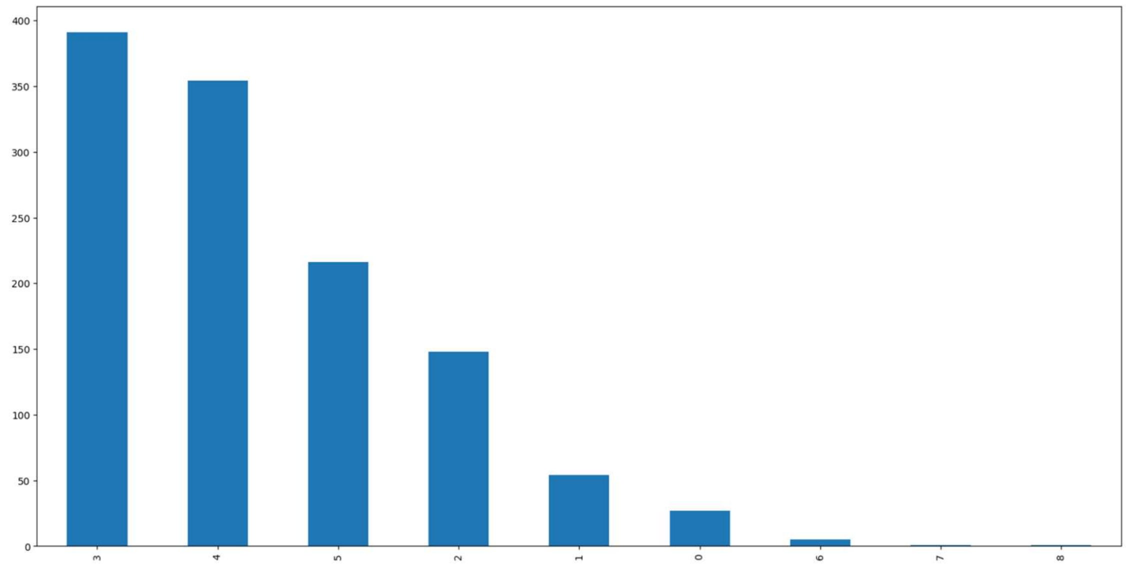
[446]:

	total_sqft	bath	Price
count	1197.000000	1197.000000	1.197000e+03
mean	2523.011696	3.395155	1.310208e+07
std	2163.581452	1.200645	2.283695e+07
min	0.000000	0.000000	1.400000e+05
25%	1500.000000	3.000000	6.000000e+06
50%	2000.000000	3.000000	8.300000e+06
75%	3000.000000	4.000000	1.500000e+07
max	35000.000000	8.000000	4.430000e+08

[447]:

```
df.bath.value_counts().plot(kind='bar')
```

[447]: <AxesSubplot:>



[448]:
```
df['location'].value_counts()
```

[448]:
```
Kowdiar              91
Pattom               86
Thirumala            82
Kesavadasapuram      45
Karamana             34
                     ..
Sasthamangalam        1
Attingal              1
Ptp nagar             1
Mukkola               1
Kudappanakunnu        1
Name: location, Length: 165, dtype: int64
```

[449]:
```
df['size'].value_counts()
```

[449]:
```
4 BHK     351
3 BHK     295
5 BHK     182
2 BHK     119
3 BHK     108
4 BHK      75
2 BHK      23
5 BHK      23
1 BHK      10
6 BHK       5
7 BHK       3
```

```
8 BHK        2
1 BHK        1
Name: size, dtype: int64
```

[450]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   location     1197 non-null   object
 1   size         1197 non-null   object
 2   total_sqft   1197 non-null   int64
 3   bath         1197 non-null   int64
 4   district     1197 non-null   object
 5   Price        1197 non-null   int64
dtypes: int64(3), object(3)
memory usage: 56.2+ KB
```

[451]:
```
df['Price']=df['Price'].astype(str)
```

[452]:
```
df=df[df['Price'].str.isnumeric()]
```

[453]:
```
df['Price']=df['Price'].str.replace(',','').astype(float)
```

[454]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1197 entries, 0 to 1196
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   location     1197 non-null   object
 1   size         1197 non-null   object
 2   total_sqft   1197 non-null   int64
 3   bath         1197 non-null   int64
 4   district     1197 non-null   object
 5   Price        1197 non-null   float64
dtypes: float64(1), int64(2), object(3)
memory usage: 65.5+ KB
```

[455]:
```
df['bhk']=df['size'].str.split().str.get(0).astype(int)
```

[456]:
```
df.describe()
```

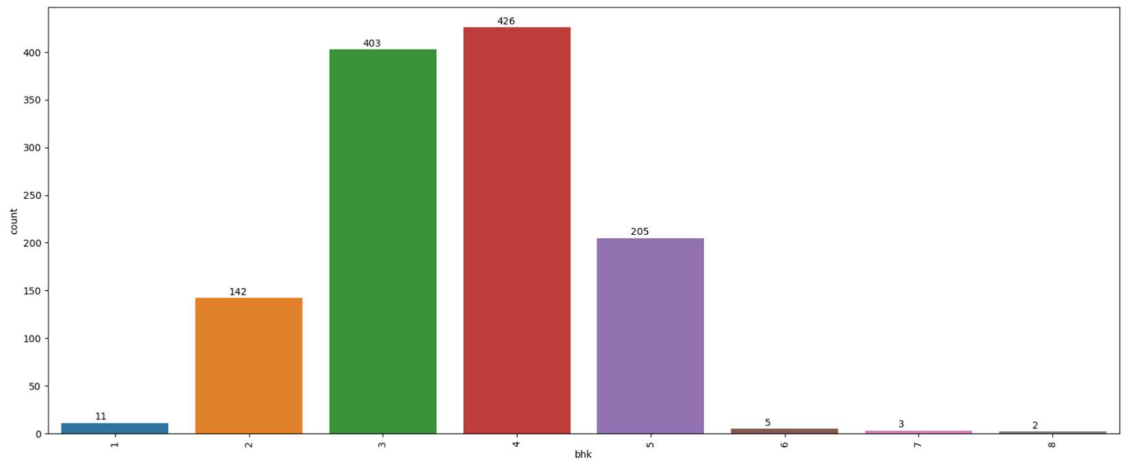
[456]:

	total_sqft	bath	Price	bhk
count	1197.000000	1197.000000	1.197000e+03	1197.000000

mean	2523.011696	3.395155	1.310208e+07	3.592314
std	2163.581452	1.200645	2.283695e+07	0.982731
min	0.000000	0.000000	1.400000e+05	1.000000
25%	1500.000000	3.000000	6.000000e+06	3.000000
50%	2000.000000	3.000000	8.300000e+06	4.000000
75%	3000.000000	4.000000	1.500000e+07	4.000000
max	35000.000000	8.000000	4.430000e+08	8.000000

[457]:

```
plt.figure(figsize = (20, 8))
ax=sns.countplot(x = 'bhk', data = df)
plt.xticks(rotation = 90)
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x()+0.25, p.get_height()+1), va =␣
↪'bottom', color = 'black')
```



[458]:

```
df
```

[458]:

	location	size	total_sqft	bath	district	Price	bhk
0	Jawahar Nagar	4 BHK	2700	5	Trivandrum	40000000.0	4
1	Vanchiyoor	5 BHK	3500	5	Trivandrum	27500000.0	5
2	Vanchiyoor	5 BHK	3700	5	Trivandrum	27500000.0	5
3	Kowdiar	3 BHK	1650	4	Trivandrum	8000000.0	3
4	Pattom	4 BHK	3800	4	Trivandrum	32500000.0	4
...	...	...	...	...	...	...	...
1192	Thirumala	5 BHK	2800	5	Trivandrum	14500000.0	5
1193	Thirumala	4 BHK	2100	4	Trivandrum	8500000.0	4
1194	Vayalilkkada	4 BHK	3300	5	Trivandrum	24500000.0	4
1195	Kariavattom	5 BHK	3795	5	Trivandrum	21000000.0	5
1196	Kazhakoottam	3 BHK	1500	3	Trivandrum	6900000.0	3

[1197 rows x 7 columns]

[459]:
```
df['location'].value_counts()
```

[459]:
```
Kowdiar              91
Pattom               86
Thirumala            82
Kesavadasapuram      45
Karamana             34
                     ..
Sasthamangalam        1
Attingal              1
Ptp nagar             1
Mukkola               1
Kudappanakunnu        1
Name: location, Length: 165, dtype: int64
```

[460]:
```
df['location'] = df['location'].apply(lambda x: x.strip())
location_counts = df ['location'].value_counts()
```

[461]:
```
location_counts
```

[461]:
```
Kowdiar              91
Pattom               86
Thirumala            82
Kesavadasapuram      45
Karamana             34
                     ..
Attingal              1
Ptp nagar             1
Muttada               1
Punnakulam            1
Kudappanakunnu        1
Name: location, Length: 163, dtype: int64
```

[462]:
```
location_count_less_1 = location_counts[location_counts<=10]
```

[463]:
```
df['location']=df['location'].apply(lambda x: 'other' if x in␣
  ↪location_count_less_1 else x)
df['location'].value_counts()
```

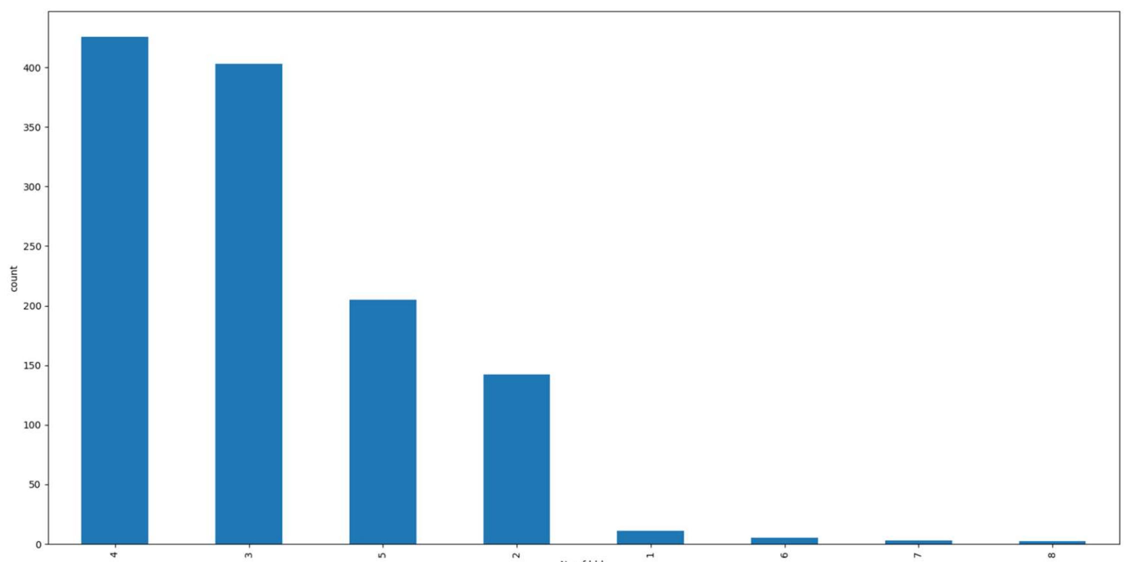
[463]:
```
other                438
Kowdiar               91
Pattom                86
Thirumala             82
Kesavadasapuram       45
Kazhakoottam          34
Karamana              34
Kumarapuram           34
```

```
Ulloor              33
Manacaud            28
Anayara             26
Thampanoor          26
Akkulam             25
Pongumoodu          20
Vattiyoorkavu       19
Thycaud             19
Vazhuthacaud        17
Vanchiyoor          17
Ambalamukku         17
Peroorkada          17
Karikkakam          14
Sreekaryam          14
Pattoor             14
Peyad               13
Attukal             12
Kuravankonam        11
Mannanthala         11
Name: location, dtype: int64
```

[464]:

```
df.bhk.value_counts().plot(kind='bar')
plt.xlabel('No of bhk')
plt.ylabel('count')
```

[464]: Text(0, 0.5, 'count')



[465]:

```
df.describe()
```

[465]:

	total_sqft	bath	Price	bhk
count	1197.000000	1197.000000	1.197000e+03	1197.000000
mean	2523.011696	3.395155	1.310208e+07	3.592314
std	2163.581452	1.200645	2.283695e+07	0.982731
min	0.000000	0.000000	1.400000e+05	1.000000
25%	1500.000000	3.000000	6.000000e+06	3.000000
50%	2000.000000	3.000000	8.300000e+06	4.000000
75%	3000.000000	4.000000	1.500000e+07	4.000000
max	35000.000000	8.000000	4.430000e+08	8.000000

[466]:
```
df['price_per_sqft']=df['Price']/df['total_sqft']
```

[467]:
```
df['price_per_sqft']
```

[467]:
```
0       14814.814815
1        7857.142857
2        7432.432432
3        4848.484848
4        8552.631579
            ...
1192     5178.571429
1193     4047.619048
1194     7424.242424
1195     5533.596838
1196     4600.000000
Name: price_per_sqft, Length: 1197, dtype: float64
```

[468]:
```
df.describe()
```

[468]:

	total_sqft	bath	Price	bhk	price_per_sqft
count	1197.000000	1197.000000	1.197000e+03	1197.000000	1197.000000
mean	2523.011696	3.395155	1.310208e+07	3.592314	inf
std	2163.581452	1.200645	2.283695e+07	0.982731	NaN
min	0.000000	0.000000	1.400000e+05	1.000000	56.000000
25%	1500.000000	3.000000	6.000000e+06	3.000000	3454.545455
50%	2000.000000	3.000000	8.300000e+06	4.000000	4640.000000
75%	3000.000000	4.000000	1.500000e+07	4.000000	6315.789474
max	35000.000000	8.000000	4.430000e+08	8.000000	inf

[469]:
```
df[df.total_sqft/df.bhk<300].head()
```

[469]:

	location	size	total_sqft	bath	district	Price	bhk	\
26	other	2 BHK	450	1	Trivandrum	2300000.0	2	
44	other	4 BHK	750	2	Trivandrum	5000000.0	4	
103	other	3 BHK	750	3	Trivandrum	7000000.0	3	
129	other	5 BHK	1100	5	Trivandrum	16000000.0	5	
167	other	4 BHK	800	2	Trivandrum	3200000.0	4	

```
     price_per_sqft
26      5111.111111
44      6666.666667
103     9333.333333
129    14545.454545
167     4000.000000
```

[470]:
```
df = df[~(df.total_sqft/df.bhk<300)]
df.shape
```

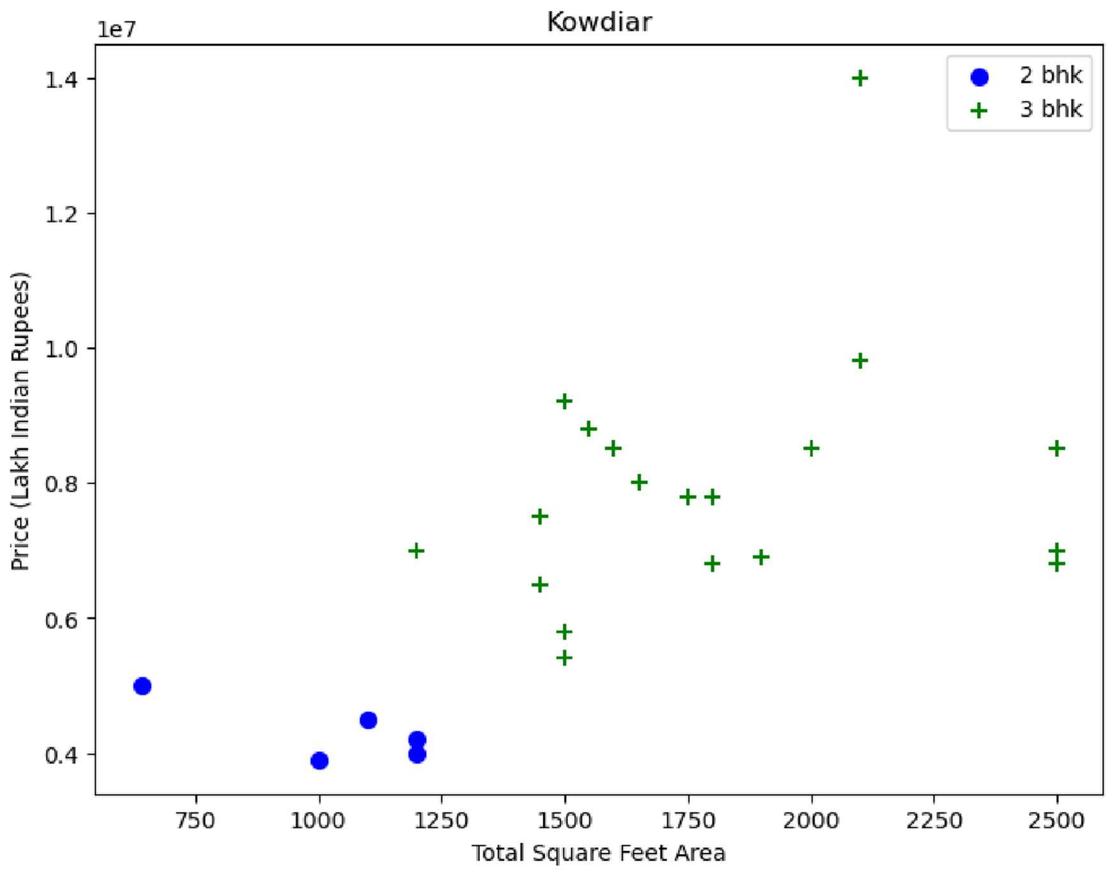
[470]:
```
(1157, 8)
```

[ ]:

[471]:
```
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m= np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.
↪price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out, reduced_df],ignore_index=True)
    return df_out
df = remove_pps_outliers(df)
df.shape
```

[471]:
```
(945, 8)
```

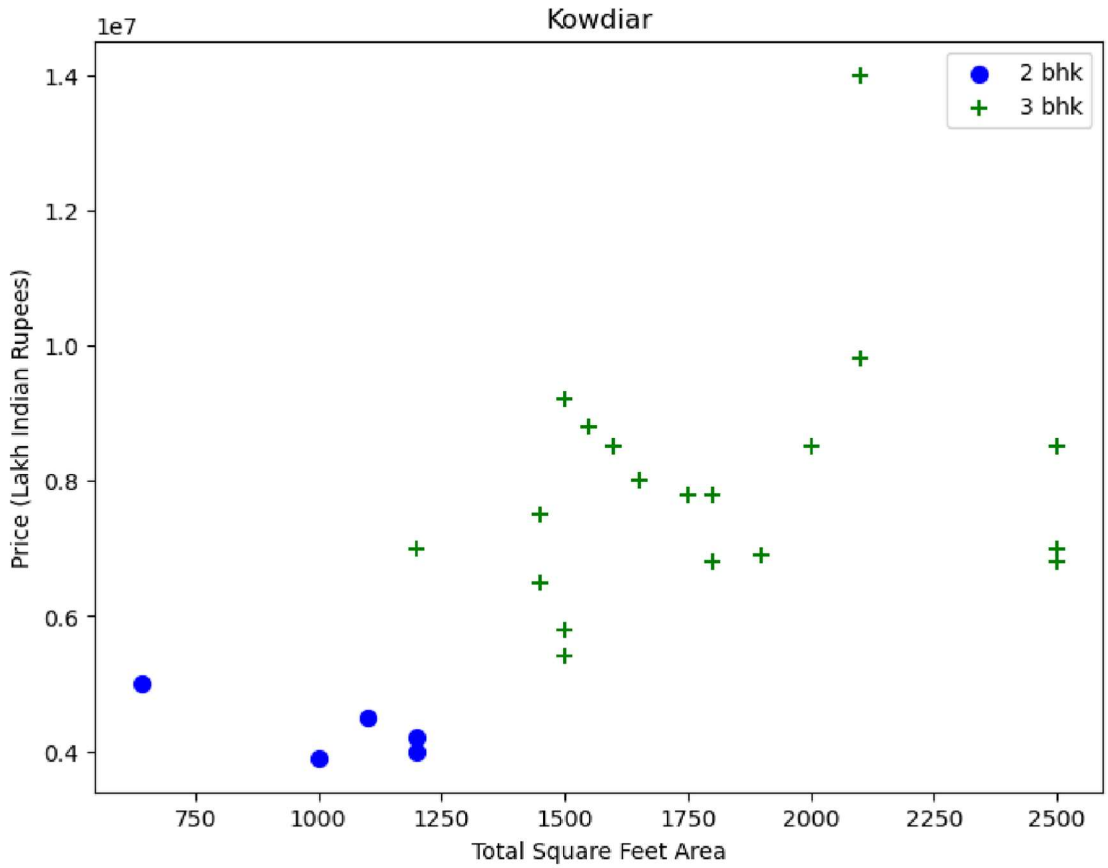
[472]:
```
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (8,6)
    plt.scatter(bhk2.total_sqft,bhk2.Price,color='blue',label='2 bhk', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.Price,marker='+', color='green',label='3␣
↪bhk', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()

plot_scatter_chart(df,"Kowdiar")
```



[473]:
```
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices, bhk_df[bhk_df.
↪price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')
df = remove_bhk_outliers(df)
df.shape
```

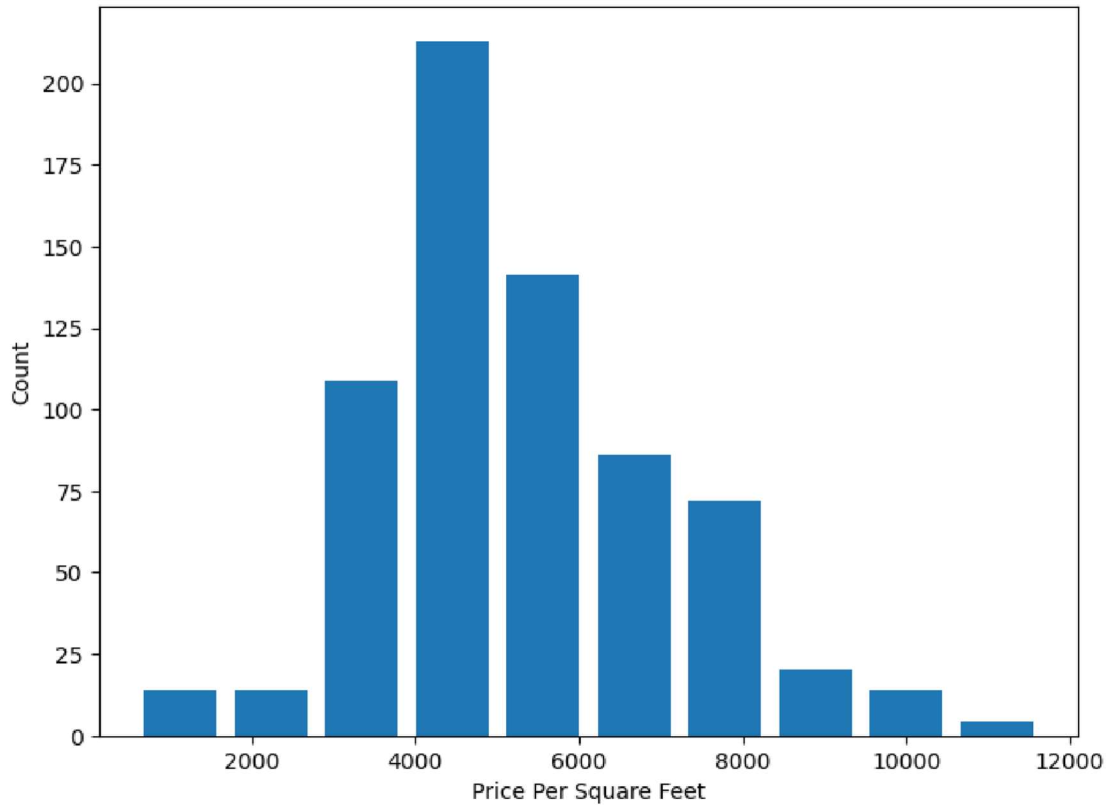
[473]: (687, 8)

[474]:
```
plot_scatter_chart(df,"Kowdiar")
```



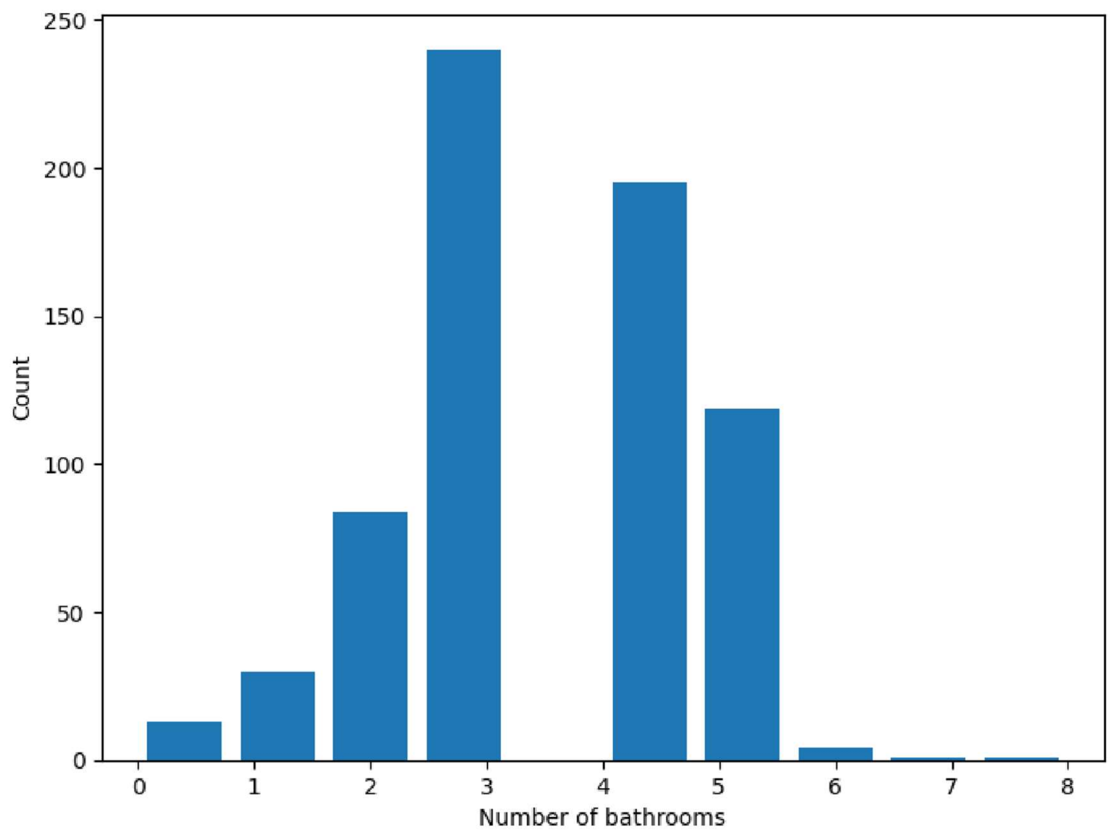
[475]:
```
plt.hist(df.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

[475]: Text(0, 0.5, 'Count')



[476]:
```
plt.hist(df.bath, rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

[476]: Text(0, 0.5, 'Count')

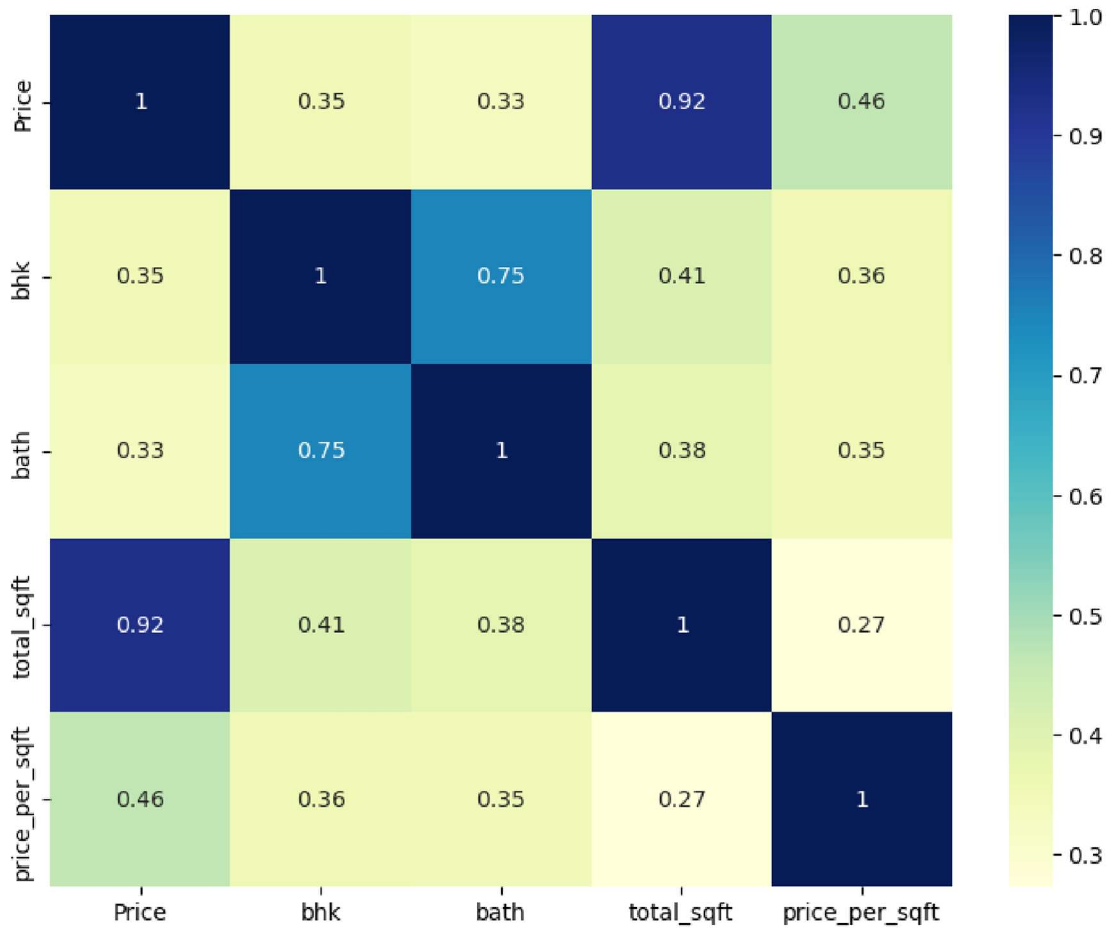


[477]:
```
df.price_per_sqft.describe()
```

[477]:
```
count        687.000000
mean        5313.998204
std         1802.704711
min          562.500000
25%         4076.704545
50%         5000.000000
75%         6363.636364
max        11666.666667
Name: price_per_sqft, dtype: float64
```

[478]:
```
plt.figure(figsize=(9, 7))
columns
 ↪=['Price','bhk','bath','total_sqft','price_per_sqft','district','location']
sns.heatmap(df[columns].corr(),cmap='YlGnBu',annot=True)
```

[478]: <AxesSubplot:>



[479]:
```
df. shape
```

[479]: (687, 8)

[480]:
```
df
```

[480]:

	location	size	total_sqft	bath	district	Price	bhk	\
0	Akkulam	2 BHK	950	2	Trivandrum	3150000. 0	2	
1	Akkulam	3 BHK	1375	3	Trivandrum	6700000. 0	3	
3	Akkulam	4 BHK	1800	4	Trivandrum	7000000. 0	4	
4	Akkulam	4 BHK	1750	4	Trivandrum	6300000. 0	4	
5	Akkulam	5 BHK	1960	5	Trivandrum	7900000. 0	5	
..	...	...	...	...	...	...	...	
938	other	4 BHK	2000	4	Trivandrum	7800000. 0	4	
939	other	4 BHK	3700	4	Trivandrum	30000000. 0	4	
941	other	4 BHK	2500	5	Trivandrum	11900000. 0	4	
943	other	4 BHK	3300	5	Trivandrum	24500000. 0	4	
944	other	5 BHK	3795	5	Trivandrum	21000000. 0	5	

```
     price_per_sqft
0       3315.789474
1       4872.727273
3       3888.888889
4       3600.000000
5       4030.612245
..              ...
938     3900.000000
939     8108.108108
941     4760.000000
943     7424.242424
944     5533.596838

[687 rows x 8 columns]
```

[481]:
```
df.drop(columns=['size','price_per_sqft'],inplace=True)
```

[482]:
```
## Cleaned data
```

[483]:
```
df.head()
```

[483]:

	location	total_sqft	bath	district	Price	bhk
0	Akkulam	950	2	Trivandrum	3150000.0	2
1	Akkulam	1375	3	Trivandrum	6700000.0	3
3	Akkulam	1800	4	Trivandrum	7000000.0	4
4	Akkulam	1750	4	Trivandrum	6300000.0	4
5	Akkulam	1960	5	Trivandrum	7900000.0	5

[484]:
```
df.to_csv("Cleaned_data_house1.csv")
```

[485]:
```
#model
```

[486]:
```
X=df.drop(columns=['Price'])
y=df['Price']
```

[487]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
↪25,random_state=93)
```

[488]:
```
print (X_train.shape)
print (X_test.shape)
```

```
(515, 5)
(172, 5)
```

[ ]:
```

```

[489]:
```
column_trans=make_column_transformer((OneHotEncoder(sparse=False),['location','district']),re
```

[490]:
```
scaler=StandardScaler()
```

[491]:
```
lr=LinearRegression()
```

[492]:
```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor(n_estimators = 500, random_state = 0)
```

[493]:
```
pipe1=make_pipeline(column_trans, scaler, lr)
pipe2=make_pipeline(column_trans, scaler, rfr)
```

[494]:
```
pipe1.fit(X_train, y_train)
```

[494]:
```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(sparse=False),
                                                  ['location', 'district'])])),
                ('standardscaler', StandardScaler()),
                ('linearregression', LinearRegression())])
```

[495]:
```
pipe2.fit(X_train, y_train)
```

[495]:
```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(sparse=False),
                                                  ['location', 'district'])])),
                ('standardscaler', StandardScaler()),
                ('randomforestregressor',
                 RandomForestRegressor(n_estimators=500, random_state=0))])
```

[496]:
```
y_pred_lr=pipe1.predict(X_test)
y_pred_lr
```

[496]:
```
array([ 9.29248270e+06, 3.21281947e+07, 6.89056270e+06, 1.42834587e+07,
        2.57535387e+07, 2.14278670e+06, 1.52211867e+07, 1.57592987e+07,
        8.84627470e+06, 1.08479387e+07, 1.08694427e+07, 2.76179867e+07,
        2.10625947e+07, 5.08640270e+06, 1.48214427e+07, 1.00670107e+07,
        1.12077467e+07, 2.52694427e+07, 6.55149070e+06, 4.50696376e+02,
        1.00668827e+07, 2.74907547e+07, 1.23237787e+07, 1.92316827e+07,
        1.97206427e+07, 1.08479387e+07, 4.47917070e+06, 1.42834587e+07,
        2.27734427e+07, 3.96461070e+06, 6.55609870e+06, 8.84627470e+06,
        2.27186587e+07, 6.26221070e+06, 2.86929307e+07, 6.97773070e+06,
        2.90469787e+07, 7.66867470e+06, 2.52537870e+06, 1.15396507e+07,
        5.86899470e+06, 8.60896270e+06, 5.90662670e+06, 2.17450907e+07,
        1.27382427e+07, 2.74683547e+07, 6.92883470e+06, 1.52631707e+07,
        5.38950670e+06, 1.13438107e+07, 2.46952347e+07, 3.09984270e+06,
        7.91443470e+06, 2.87686670e+06, 2.86929307e+07, 1.98340507e+07,
```

```
        1.12846747e+07,  2.65974427e+07,  1.92316827e+07,  3.49920270e+06,
        2.59995547e+07,  8.48966670e+06,  8.89440270e+06,  2.01624987e+07,
        1.08056987e+07,  1.42834587e+07,  6.84896270e+06,  1.61632667e+07,
        1.84851470e+06,  3.63358107e+07,  6.92883470e+06,  7.98381070e+06,
        6.55609870e+06,  1.42038427e+07,  2.70212507e+07,  1.08694427e+07,
        9.57702670e+06,  3.42393870e+06,  2.43523227e+07,  4.09593870e+06,
        7.80717070e+06,  1.19725070e+06,  1.45038747e+07,  1.20383387e+07,
        4.59219470e+06,  1.43152027e+07, -7.96861304e+05,  6.43270670e+06,
        6.92883470e+06,  2.96723470e+06,  3.49920270e+06,  1.66789787e+07,
        2.22768027e+07,  6.84653070e+06,  1.08479387e+07,  5.37248270e+06,
        1.27693467e+07,  6.06637070e+06,  1.27693467e+07,  1.63130267e+07,
        1.18279067e+07,  2.76179867e+07,  3.30690696e+05,  9.98765070e+06,
        5.86310670e+06,  1.23237787e+07,  8.80864270e+06,  8.00505870e+06,
        1.52631707e+07,  1.16281870e+06,  2.52537870e+06,  1.98081947e+07,
        1.03640987e+07, -3.87173036e+04,  7.52531470e+06,  1.49139470e+06,
        2.96723470e+06,  1.00883867e+07,  7.81305870e+06,  2.71338907e+07,
       -2.23333036e+04,  4.88928270e+06,  3.18821787e+07,  8.72851470e+06,
        8.40441870e+06,  1.73843470e+06,  2.46952347e+07,  9.46937870e+06,
       -5.98077304e+05,  2.61075867e+07,  1.53337870e+06,  1.63310747e+07,
        5.86899470e+06,  8.84627470e+06,  3.54118670e+06,  6.05561870e+06,
        3.94694670e+06,  9.89613070e+06,  4.50696376e+02, -4.69100530e+06,
        9.00115470e+06,  1.35139227e+07,  1.98340507e+07,  6.97773070e+06,
        1.43152027e+07,  1.37453467e+07,  2.99646107e+07,  3.11484827e+07,
        9.10227470e+06,  8.26809870e+06,  8.84627470e+06,  7.33280270e+06,
        2.02950670e+06,  3.60595470e+06,  2.45778587e+07,  9.88025870e+06,
        6.06189070e+06,  2.61075867e+07,  3.63358107e+07,  3.92467470e+06,
        2.96723470e+06,  1.65280270e+06,  9.92659470e+06,  1.55785627e+07,
       -2.38598130e+06,  1.23237787e+07,  7.64486670e+06,  1.49139470e+06,
        7.04608270e+06,  3.11597070e+06,  6.36826964e+04,  2.51949070e+06])
```

[497]:

```
y_pred_rfr=pipe2.predict(X_test)
y_pred_rfr
```

[497]:

```
array([ 8011918.57142857, 34259600.        ,  5263516.66666667,
       15188976.21933621, 22960987.3015873 ,  7555550.64935065,
       14213527.61904762, 24519830.        , 10303336.66666667,
       10738023.01587301,  9977890.47619048, 23461966.66666666,
       23045000.        ,  7406899.60317461, 14951042.38095238,
       10381743.80952381,  5189633.33333333, 27338300.        ,
        8370198.57142857,  7187586.66666667,  5672000.        ,
       27924400.        , 13855703.57142857, 15093649.04761904,
       19373157.61904762, 10738023.01587301,  6561711.11111111,
       15188976.21933621, 24231600.        ,  6748666.66666667,
        7596600.        , 10303336.66666667, 22398300.        ,
        6377515.        , 27952744.15584416,  7982800.        ,
       27034000.        , 11042746.66666667,  6289462.14285714,
       10146746.64502163,  7121992.96425797,  9065503.33333333,
```

```
        7377981.42857143, 26969000.        , 13317085.71428571,
       23934600.        ,  7650575.07936508, 13011629.2063492 ,
        6624150.        ,  9313731.11111111, 25943200.        ,
        7722315.        ,  7482836.66666667,  5237733.33333333,
       27952744.15584416, 22882500.        ,  7895754.76190476,
       27892000.        , 15093649.04761904,  8721195.71428571,
       25822200.        ,  8340135.71428571,  8314381.42857143,
       16431124.28571429,  9104822.14285715, 15188976.21933621,
        7935247.73448773, 14572933.33333333,  4857898.57142857,
       31872000.        ,  7650575.07936508,  7447849.52380952,
        7596600.        , 13776952.38095238, 27699128.57142857,
        9977890.47619048, 11135880.79365079,  6811043.80952381,
       22450900.        ,  7634730.95238095,  7217002.6984127 ,
        5606495.23809524, 13393593.33333333, 11167304.76190476,
        8148216.66666667, 15082475.95238095,  6661733.33333333,
        7655939.52380953,  7650575.07936508,  6283675.47619048,
        8721195.71428571, 13334222.38095238, 21901000.        ,
        5417543.35497835, 10738023.01587301,  5942666.66666667,
       13262985.71428572,  6936110.71428571, 13262985.71428572,
       14905041.9047619 , 12619850.        , 23461966.66666666,
        3014233.33333333, 13310119.52380952,  5630081.66666667,
       13855703.57142857,  8301671.42857143,  7050097.61904762,
       13011629.2063492 ,  7866010.23809524,  6289462.14285714,
       21761900.        ,  9659380.34188034,  6886876.66666667,
        8320866.66666667,  8428233.33333333,  6283675.47619048,
        8002370.        ,  7993100.        , 24394166.66666666,
        3987326.90476191,  7260451.06782107, 34810100.        ,
        8879453.33333333,  7768140.        ,  3895256.42857143,
       25943200.        ,  8962762.77777778,  4878476.73881674,
       26703300.        ,  4284768.0952381 , 14665426.66666667,
        7121992.96425797, 10303336.66666667, 10377844.84126984,
        7242973.80952381,  5915104.76190476,  9617000.        ,
        7187586.66666667,  2538746.66666667, 10110126.19047619,
       12653563.33333333, 22882500.        ,  7982800.        ,
       15082475.95238095, 14909316.66666667, 32992533.33333334,
       32888200.        ,  9861032.13564213,  4370398.33333333,
       10303336.66666667,  7575035.31746032,  5500241.9047619 ,
        7552926.19047619, 27247000.        , 11160158.04029305,
        9220815.71428571, 26703300.        , 31872000.        ,
        6492815.78643579,  6283675.47619048,  7575820.09379509,
       10200757.14285714, 12970405.71428571,  4106900.        ,
       13855703.57142857,  6876078.57142857,  8428233.33333333,
        7689106.66666667,  7352405.06493506,  3756613.33333333,
        5425441.9047619 ])
```

[498]:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
mae = mean_absolute_error(y_test, y_pred_lr)
```

```
mse = mean_squared_error(y_test, y_pred_lr)
rmse = np.sqrt(mse)

print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
```

```
MAE: 2995704.7672482585
MSE: 13686165428146.055
RMSE: 3699481.7783232904
```

[499]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error
mae = mean_absolute_error(y_test, y_pred_rfr)
mse = mean_squared_error(y_test, y_pred_rfr)
rmse = np.sqrt(mse)

print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
```

```
MAE: 1643355.8028889715
MSE: 6006877585036.224
RMSE: 2450893.2218756946
```

[500]:
```
score_lr=r2_score(y_test,y_pred_lr)
score_rfr=r2_score(y_test,y_pred_rfr)

print("Linear regression:",score_lr)
print("RandomForest regression",score_rfr)
```

```
Linear regression: 0.7951320632992682
RandomForest regression 0.9100831695100338
```

[501]:
```
new_score_lr=score_lr*100
new_score_rfr=score_rfr*100

print("Accuracy for Linear regression:",new_score_lr,"%")
print("Accuracy for RandomForest regression",new_score_rfr,"%")
```

```
Accuracy for Linear regression: 79.51320632992682 %
Accuracy for RandomForest regression 91.00831695100338 %
```

[502]:
```
'''scores=[]
for i in range(100):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
↪25,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,scaler,lr)
    pipe.fit(X_train,y_train)
```

```
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))'''
```

[502]:

```
'scores=[]\nfor i in range(100):\n    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=i)\n    lr=LinearRegression()\n    pipe=make_pipeline(column_trans,scaler,lr)\n    pipe.fit(X_train,y_train)\n    y_pred=pipe.predict(X_test)\n    scores.append(r2_score(y_test,y_pred))'
```

[503]:

```
#np.argmax(scores)
```

[504]:

```
#scores[np.argmax(scores)]
```

[505]:

```
import pickle
```

[506]:

```
pickle.dump (pipe2,open('RandomForestRegression1.pkl','wb'))
```

[507]:

```
pipe2.predict(pd.
↪DataFrame([['Kowdiar',1850,4,4,'Trivandrum']],columns=['location','total_sqft','bhk','bath'
```

[507]:

```
array([8370198.57142857])
```

[ ]:

[ ]:

8.2 FRONT END

8.2.1 Html and CSS

```
<!DOCTYPE html>
<html  >
<head>

  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="generator" content="Mobirise v5.8.0, mobirise.com">
  <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1">
  <link rel="shortcut icon" href="{{ url_for('static', filename='images/logo.png') }}" type="image/x-
icon">
  <meta name="description" content="">


  <title>Home</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='assets/bootstrap/css/bootstrap.min.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='assets/bootstrap/css/bootstrap-grid.min.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='assets/bootstrap/css/bootstrap-reboot.min.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='assets/parallax/jarallax.css')
}}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='assets/dropdown/css/style.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='assets/socicon/css/styles.css') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='assets/theme/css/style.css')
}}">


  <link rel="preload" href="{{ url_for('static', filename='styles.css') }}" as="style"
onload="this.onload=null;this.rel='stylesheet'">
  <noscript><link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}"></noscript>
  <link rel="preload" as="style" href="{{ url_for('static', filename='assets/mobirise/css/mbr-
additional.css') }}"><link rel="stylesheet" href="{{ url_for('static', filename='assets/mobirise/css/mbr-
additional.css') }}" type="text/css">


</head>
<body>

   <section data-bs-version="5.1" class="menu menu3 cid-twwIZxrcrx" once="menu" id="menu3-u">
```

```
        <nav class="navbar navbar-dropdown navbar-fixed-top collapsed">
            <div class="container-fluid">
                <div class="navbar-brand"></div>

                <button class="navbar-toggler" type="button" data-toggle="collapse" data-bs-
toggle="collapse" data-target="#navbarSupportedContent" data-bs-target="#navbarSupportedContent"
aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
                    <div class="hamburger">
                        <span></span>
                        <span></span>
                        <span></span>
                        <span></span>
                    </div>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav nav-dropdown nav-right navbar-nav-top-padding" data-app-modern-
menu="true"><li class="nav-item"><a class="nav-link link text-black display-4" href="#features3-x">
                            Features</a></li>
                    <li class="nav-item"><a class="nav-link link text-black display-4" href="#team1-q">
                            Our Team</a></li>
                    <li class="nav-item"><a class="nav-link link text-black display-4" href="#form1-
s">Contacts</a>
                    </li></ul>
                <div class="icons-menu">
                    <a class="iconfont-wrapper" href="https://www.facebook.com" target="_blank">
                        <span class="p-2 mbr-iconfont socicon-facebook socicon"></span>
                    </a>
                    <a class="iconfont-wrapper" href="https://www.twitter.com" target="_blank">
                        <span class="p-2 mbr-iconfont socicon-twitter socicon"></span>
                    </a>
                    <a class="iconfont-wrapper" href="https://www.instagram.com/shylesh_shylu_/"
target="_blank">
                        <span class="p-2 mbr-iconfont socicon-instagram socicon"></span>
                    </a>

                </div>
                </div>
            </div>
        </div>
    </nav>
</section>

<section data-bs-version="5.1" class="header13 cid-twwIopQIhn mbr-fullscreen" id="header13-t">
```

```
    <div class="mbr-overlay" style="opacity: 0.2; background-color: rgb(250, 250, 250);"></div>

    <div class="align-center container">
        <div class="row justify-content-center">
            <div class="col-12 col-lg-10">
                <h1 class="mbr-section-title mbr-fonts-style mb-3 display-1"><strong>House Price
Prediction</strong></h1>

                <p class="mbr-text mbr-fonts-style mb-3 display-7">
                    "Unlock the power of data to predict your dream home's price. Get accurate house price
predictions with our advanced algorithm, and make your home buying or selling journey a breeze."
                </p>
                <div class="mbr-section-btn mt-3"><a class="btn btn-primary display-4"
href="#abc">Predict Price</a></div>

            </div>
        </div>
    </div>
</section>

<section data-bs-version="5.1" class="features3 cid-twzqwjjhTZ" id="features3-x">


    <div class="container">
        <div class="mbr-section-head">
            <h4 class="mbr-section-title mbr-fonts-style align-center mb-0 display-2">
                <strong>Features</strong></h4>
            <h5 class="mbr-section-subtitle mbr-fonts-style align-center mb-0 mt-2 display-5"></h5>
        </div>
        <div class="row mt-4">
            <div class="item features-image col-12 col-md-6 col-lg-4">
                <div class="item-wrapper">
                    <div class="item-img">
                        <img src="{{ url_for('static', filename='images/mbr-815x543.jpg') }}" alt="Load
again">
                    </div>
                    <div class="item-content">
                        <h5 class="item-title mbr-fonts-style display-7"><strong>Easy to Predict
Price</strong></h5>

                        <p class="mbr-text mbr-fonts-style mt-3 display-7">Our website skillfully predicts
prices of houses according to the customer's needs in an efficient manner.</p>
                    </div>

                </div>
```

```
            </div>
            <div class="item features-image col-12 col-md-6 col-lg-4">
                <div class="item-wrapper">
                    <div class="item-img">
                        <img src="{{ url_for('static', filename='images/mbr-2-815x543.jpg') }}" alt="Load
again">
                    </div>
                    <div class="item-content">
                        <h5 class="item-title mbr-fonts-style display-7"><strong>Mobile
Friendly</strong></h5>

                        <p class="mbr-text mbr-fonts-style mt-3 display-7">Our website is also compatible
with any mobile devices.</p>
                    </div>

                </div>
            </div>
            <div class="item features-image col-12 col-md-6 col-lg-4">
                <div class="item-wrapper">
                    <div class="item-img">
                        <img src="{{ url_for('static', filename='images/istockphoto-583726750-815x815.jpg')
}}" alt="Load again">
                    </div>
                    <div class="item-content">
                        <h5 class="item-title mbr-fonts-style display-7"><strong>No Extra
Charges</strong></h5>

                        <p class="mbr-text mbr-fonts-style mt-3 display-7">We do not ask any extra fees or
any other broker fees of any kind.<br></p>
                    </div>

                </div>
            </div>
        </div>
    </div>
</section>
```

```
<!-- Bootstrap CSS -->

<section class="body1" id="abc">
<div class="containers">
  <div class="row">
    <div class="card" style="width:100%; height:100%; margin-top: 50px" >
      <div class="card-header" style="text-align:center">
        <h1><b style="color:rgb(24, 24, 24);">House  Price  Prediction</b></h1>
      </div>
        <br>
        <br>

        <h6><b style="color:rgb(18, 17, 17);">want to predict the price of the house ? fill the below
details!</b></h6>

        <div class="card-body">
          <form method="post" accept-charset="utf-8">

            <div class="row">
              <div class="col-md-6 form-group mx-auto" style="text-align: center" >
                  <label><b style="color:rgb(17, 16, 16);">Select the District:</b></label>
                  <select style="border-radius:20px" class="selectpicker form-control" id="district"
name="district"  required="1">
                    {% for district in districts %}
                    <option value="{{ district }}">{{ district }}</option>
                    {% endfor %}
                  </select>
                </div>

              <div class="col-md-6 form-group" style="text-align: center " >
                <label ><b style="color:rgb(15, 14, 14);">Select the location:</b></label>
                <select style="border-radius:20px" class="selectpicker form-control" id="location"
name="location"  required="1">
                    {% for location in locations %}
                      <option value="{{ location }}">{{ location }}</option>
                    {% endfor %}
                </select>
              </div>
              <div class="col-md-6 form-group" style="text-align: center" >
                <label><b style="color:rgb(17, 17, 17);" >Enter BHK:</b></label>
                <input style="border-radius: 20px;" type="text" class="form-control" id="bhk"
name="bhk" placeholder="Enter BHK">
              </div>

              <div class="col-md-6 form-group" style="text-align: center" >
                <label><b style="color:rgb(16, 15, 15);">Enter the Number of bath room:</b></label>
```

```
                <input style="border-radius: 20px;" type="text" class="form-control" id="bath"
name="bath" placeholder="Enter Number of bathroom">
              </div>
              <div class="col-md-6 form-group mx-auto" style="text-align: center" >
                <label><b style="color:rgb(17, 16, 16);">Enter Square feet:</b></label>
                <input style="border-radius: 20px;" type="text" class="form-control" id="total_sqft"
name="total_sqft" placeholder="Enter Square Feet">
              </div>
              <div class="col-md-12 form-group">
                <button style="margin-left:448px; margin-top:30px; width: 200px; border-radius: 30px;"
class="btn btn-primary btn-lg" onclick="send_data()">Predict Price</button>
              </div>

            </div>

          </form>
          <br>

          <div class="col-md-12 " style="text-align: center">
            <h3><span id="prediction"> </span></h3>
          </div>


        </div>

      </div>
</div>
</div>
</section>

<script>

      function form_handler(event)
      {
          event.preventDefault();
      }
      function send_data()
      {
            document.querySelector('form').addEventListener("submit",form_handler);

            var fd=new FormData(document.querySelector('form'));

            var xhr= new XMLHttpRequest();

            xhr.open('POST','/predict',true);
```

```
                document.getElementById("prediction").innerHTML = "Wait Predicting Price...!";
                xhr.onreadystatechange = function(){
                    if(xhr.readyState == XMLHttpRequest.DONE){
                        document.getElementById('prediction').innerHTML="Estimated Price is : ₹
"+xhr.responseText;
                    }
                };

                xhr.onload = function(){};
                xhr.send(fd);

        }
</script>
</div>

    </div>

  </div>

</div>

<section data-bs-version="5.1" class="team1 cid-twwyEt6c20" id="team1-q">

    <div class="container">
        <div class="row justify-content-center">
            <div class="col-12">
                <h3 class="mbr-section-title mbr-fonts-style align-center mb-4 display-2">
                    <strong>Our team</strong>
                </h3>

            </div>
            <div class="col-sm-4 col-lg-3">
                <div class="card-wrap">
                    <div class="image-wrap">
                        <img src="{{ url_for('static', filename='images/business-man-icon-free-vector.webp')
}}" alt="Load again">
                    </div>
                    <div class="content-wrap">
                        <h5 class="mbr-section-title card-title mbr-fonts-style align-center m-0 display-5">
                            <strong>Shylesh Kumar S</strong></h5>
                        <h6 class="mbr-role mbr-fonts-style align-center mb-3 display-4">
                            <strong>Programmer</strong>
                        </h6>

                        <div class="social-row display-7">
                            <div class="soc-item">
```

```
                    <a href="https://www.instagram.com/shylesh_shylu_/" target="_blank">
                        <span class="mbr-iconfont socicon-instagram socicon"></span>
                    </a>
                </div>
                <div class="soc-item">
                    <a href="https://telegram.com" target="_blank">
                        <span class="mbr-iconfont socicon-telegram socicon"></span>
                    </a>
                </div>

            </div>

        </div>
    </div>
</div>

<div class="col-sm-4 col-lg-3">
    <div class="card-wrap">
        <div class="image-wrap">
            <img src="{{ url_for('static', filename='images/black-10-580x386.jpeg') }}" alt="Load
again">
        </div>
        <div class="content-wrap">
            <h5 class="mbr-section-title card-title mbr-fonts-style align-center m-0 display-5">
                <strong>B Amala Francis </strong></h5>
            <h6 class="mbr-role mbr-fonts-style align-center mb-3 display-4">2nd Group
Member</h6>

            <div class="social-row display-7">
                <div class="soc-item">
                    <a href="https://www.instagram.com" target="_blank">
                        <span class="mbr-iconfont socicon-instagram socicon"></span>
                    </a>
                </div>
                <div class="soc-item">
                    <a href="https://telegram.com" target="_blank">
                        <span class="mbr-iconfont socicon-telegram socicon"></span>
                    </a>
                </div>

            </div>

        </div>
    </div>
</div>
```

```
            <div class="col-sm-4 col-lg-3">
                <div class="card-wrap">
                    <div class="image-wrap">
                        <img src="{{ url_for('static', filename='images/black-10-580x386.jpeg') }}" alt="Load
again">
                    </div>
                    <div class="content-wrap">
                        <h5 class="mbr-section-title card-title mbr-fonts-style align-center m-3 display-
5"><strong>Rehna M</strong></h5>
                        <h6 class="mbr-role mbr-fonts-style align-center mb-3 display-4">3rd Group
Member</h6>

                        <div class="social-row display-7">
                            <div class="soc-item">
                                <a href="https://www.instagram.com" target="_blank">
                                    <span class="mbr-iconfont socicon-instagram socicon"></span>
                                </a>
                            </div>
                            <div class="soc-item">
                                <a href="https://telegram.com" target="_blank">
                                    <span class="mbr-iconfont socicon-telegram socicon"></span>
                                </a>
                            </div>
                        </div>

                    </div>
                </div>
            </div>


        </div>
    </div>
</section>

<section data-bs-version="5.1" class="form1 cid-twwyYtsvJ7 mbr-fullscreen mbr-parallax-
background" id="form1-s">

    <div class="mbr-overlay" style="opacity: 0.5; background-color: rgb(255, 255, 255);"></div>
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-lg-8 mx-auto mbr-form" data-form-type="formoid">
                <form action="send.php" method="POST" class="mbr-form form-with-styler" data-form-
title="Form Name"><input type="hidden" name="email" data-form-email="true"
value="dI9iw2+JyZ9FzJ1G+cPz+rFY4bgE3nikyro8efoUpvLLUbu/8JCnrKLeKTZ9kuwzCTtBTSTIz
CcsViUs7M0FF0V4CtZoyoCuQHS78jSEScU642xJFwNKrLP2xUkjKkdo">
                    <div class="row">
```

```
                        <div hidden="hidden" data-form-alert="" class="alert alert-success col-12">Thanks for
filling out the form!</div>
                        <div hidden="hidden" data-form-alert-danger="" class="alert alert-danger col-12">
                            Oops...! some problem!
                        </div>
                    </div>
                    <div class="dragArea row">
                        <div class="col-12">
                            <h1 class="mbr-section-title mb-4 mbr-fonts-style align-center display-2">
                                <strong>Contact us!</strong>
                            </h1>
                        </div>
                        <div class="col-12">
                            <p class="mbr-text mbr-fonts-style mb-5 align-center display-7">Subscribe and our
team will reply you soon.</p>
                        </div>
                        <div class="col-md col-12 form-group mb-3" data-for="name">
                            <input type="text" name="name" placeholder="Name" data-form-field="Name"
class="form-control" id="name-form1-s">
                        </div>
                        <div class="col-md col-12 form-group mb-3" data-for="email">
                            <input type="email" name="email" placeholder="Email" data-form-field="Email"
class="form-control" id="email-form1-s">
                        </div>
                        <div class="mbr-section-btn col-12 col-md-auto"><button type="submit" class="btn
btn-primary display-4">Subscribe</button></div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</section>

<section data-bs-version="5.1" class="footer3 cid-twBNqdugpe" once="footers" id="footer3-y">


    <div class="container">
        <div class="media-container-row align-center mbr-white">
            <div class="row row-links">
                <ul class="foot-menu">
                <li class="foot-menu-item mbr-fonts-style display-7">
                        <a class="text-white" href="#" target="_blank">About us</a>
                    </li><li class="foot-menu-item mbr-fonts-style display-7">
                        <a class="text-white" href="#" target="_blank">Services</a>
                    </li><li class="foot-menu-item mbr-fonts-style display-7">
                        <a class="text-white" href="#" target="_blank">Contact Us</a>
```

```
                </li><li class="foot-menu-item mbr-fonts-style display-7">
                    <a class="text-white" href="#" target="_blank">Careers</a>
                </li><li class="foot-menu-item mbr-fonts-style display-7">
                    <a class="text-white" href="#" target="_blank">Work</a>
                </li></ul>
            </div>
            <div class="row social-row">
                <div class="social-list align-right pb-2">

                <div class="soc-item">
                        <a href="https://twitter.com" target="_blank">
                            <span class="socicon-twitter socicon mbr-iconfont mbr-iconfont-social"></span>
                        </a>
                    </div><div class="soc-item">
                        <a href="https://www.facebook.com" target="_blank">
                            <span class="socicon-facebook socicon mbr-iconfont mbr-iconfont-social"></span>
                        </a>
                    </div><div class="soc-item">
                        <a href="https://www.instagram.com" target="_blank">
                            <span class="mbr-iconfont mbr-iconfont-social socicon-instagram socicon"></span>
                        </a>
                    </div></div>
            </div>
            <div class="row row-copirayt">
                <p class="mbr-text mb-0 mbr-fonts-style mbr-white align-center display-
7">@Shylesh_Kumar_S</p>
            </div>
        </div>
    </div>
</section>
<script type="text/javascript" src="{{ url_for('static',
filename='assets/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<script type="text/javascript" src="{{ url_for('static', filename='assets/parallax/jarallax.js')
}}"></script>
<script type="text/javascript" src="{{ url_for('static', filename='assets/smoothscroll/smooth-scroll.js')
}}"></script>
<script type="text/javascript" src="{{ url_for('static', filename='assets/ytplayer/index.js')
}}"></script>
<script type="text/javascript" src="{{ url_for('static', filename='assets/dropdown/js/navbar-
dropdown.js') }}"></script>
<script type="text/javascript" src="{{ url_for('static', filename='assets/theme/js/script.js')
}}"></script>
<script type="text/javascript" src="{{ url_for('static', filename='assets/formoid/formoid.min.js')
}}"></script>
</body>
</html>
```

8.2.2 Flask

```
import numpy as np
import pandas as pd
from flask import Flask, render_template, request
import pickle


app = Flask(__name__, static_url_path='/static')
data = pd.read_csv('Cleaned_data_house1.csv')
pipe = pickle.load(open("RandomForestRegression1.pkl", "rb"))


@app.route('/')
def index():
    locations = sorted(data['location'].unique())
    districts = sorted(data['district'].unique())
    return render_template('House_Shylesh.html', locations=locations, districts=districts)


@app.route('/predict', methods=['POST'])
def predict():
    location = request.form.get('location')
    bhk = request.form.get('bhk')
    bath = request.form.get('bath')
    sqft = request.form.get('total_sqft')
    dist = request.form.get('district')
    input = pd.DataFrame([[location, sqft, bath, bhk, dist]], columns=['location', 'total_sqft', 'bath', 'bhk',
'district'])
    prediction = pipe.predict(input)[0]
    return str(np.round(prediction,2))


if __name__ == "__main__":
    app.run(debug=True, port=5001)
```

8.2.3 PHP code

```
<?php
// Check if form was submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    // Get form data
    $name = $_POST['name'];
    $email = $_POST['email'];

    // Set recipient email address
    $to = 'youremail@example.com';

    // Set email subject
    $subject = 'New Subscription from ' . $name;

    // Set email message
    $message = 'Hello, ' . $name . '!<br><br>' . 'You have a new subscription request from ' . $email . '.';

    // Set headers
    $headers = "From: " . $email . "\r\n";
    $headers .= "Reply-To: " . $email . "\r\n";
    $headers .= "Content-type: text/html\r\n";

    // Send email
    if (mail($to, $subject, $message, $headers)) {
        // Email sent successfully
        echo '<div class="alert alert-success">Thanks for subscribing!</div>';
    } else {
        // Error sending email
        echo '<div class="alert alert-danger">Oops! There was an error.</div>';
    }
}
?>
```

8.3 SCREENSHOTS

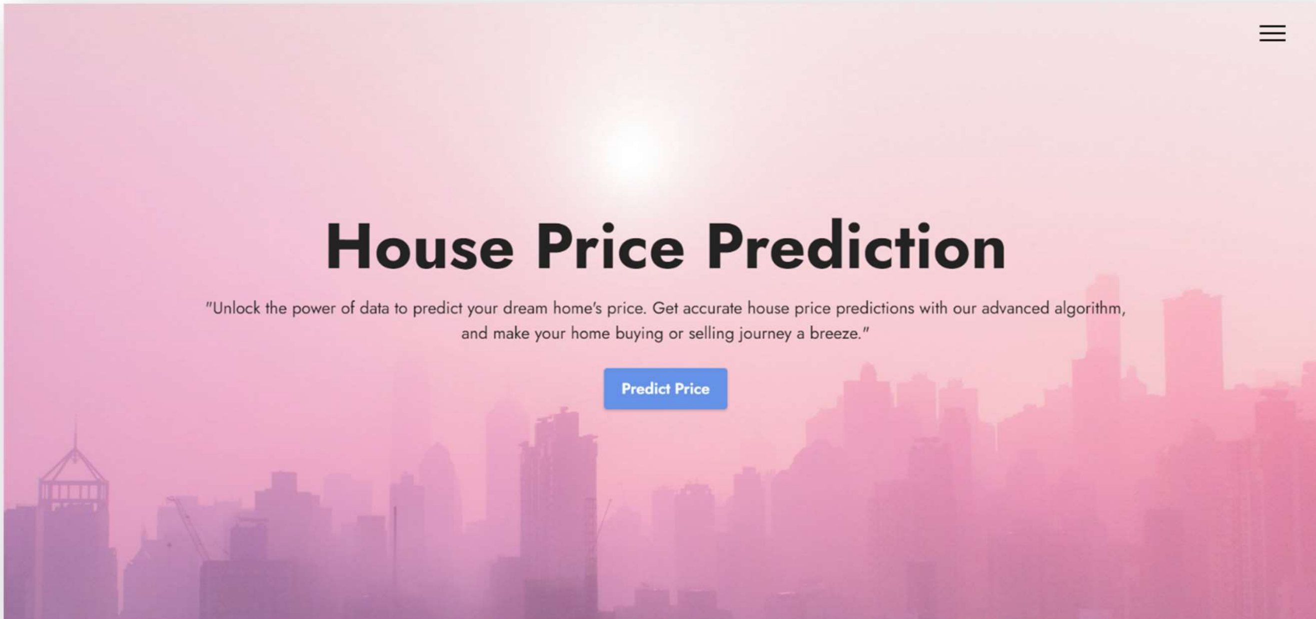


Figure 14: Main Page

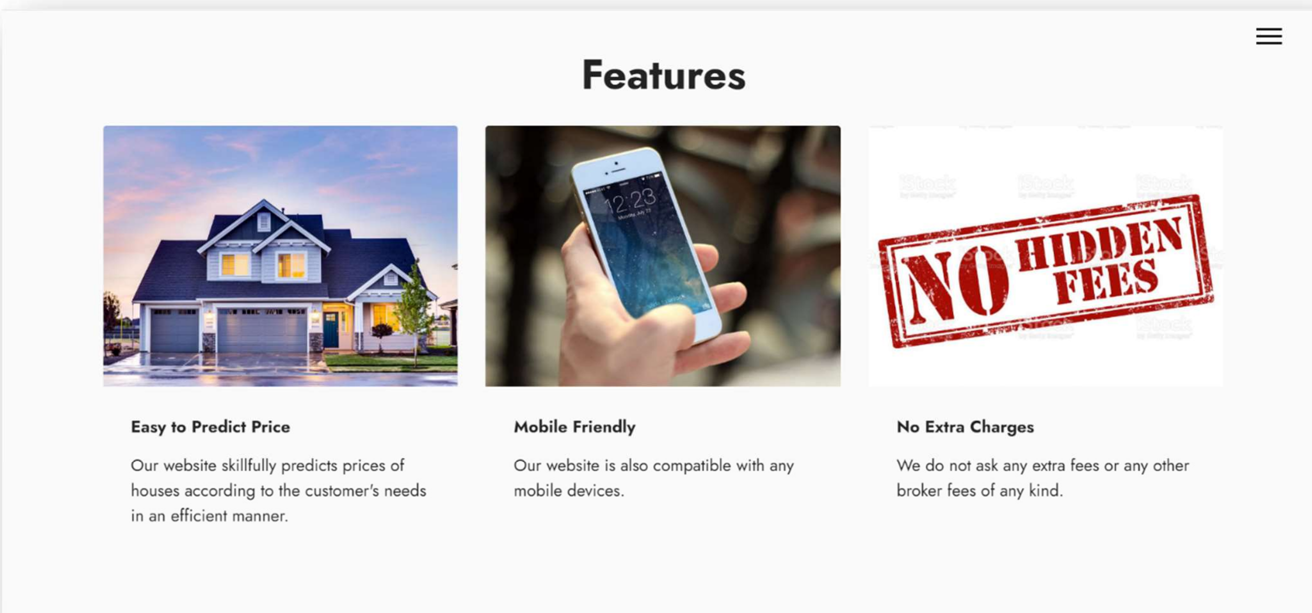


Figure 15:Features Page

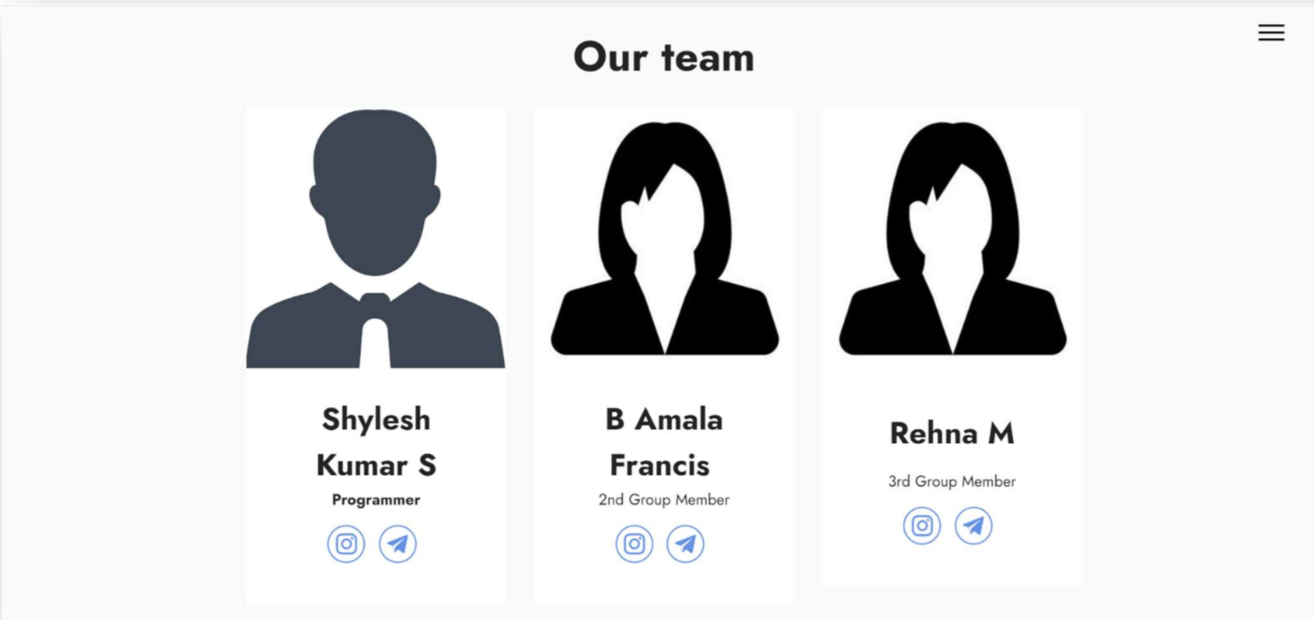


Figure 16:Team Page

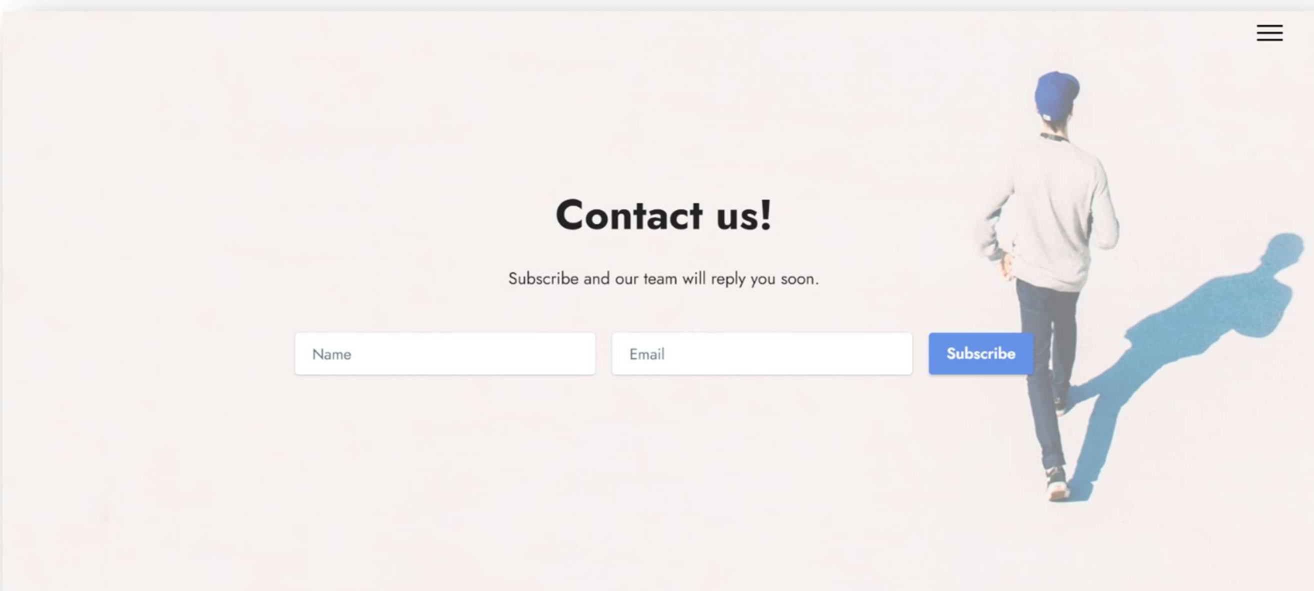


Figure 17:Contact Page

9. TESTING

The testing phase is a critical component of any project. It involves testing the project's functionality, performance, and usability to ensure that it meets the project's requirements and specifications. The testing phase typically follows the development phase and precedes the deployment phase.

It involves several steps to ensure that the system is functioning correctly and providing accurate predictions to the user. Here are some steps that can be taken to test our system:

1. Unit testing: Testing individual modules of the system to ensure their proper functioning.

2. Integration testing: Testing the integration between different modules of the system to ensure their proper coordination.

3. Regression testing: Testing the existing functionality of the system after any changes have been made to it.

4. Load testing: Testing the system's performance under heavy load to ensure its proper functioning.

5. Accuracy testing: Testing the accuracy of the system's predictions by comparing them with actual house prices in the same location and with the same attributes.

6. Usability testing: Testing the user interface to ensure that it is easy to use and intuitive.

7. Input validation testing: The system should validate all user inputs to ensure that they are correct and appropriate.

8. Security testing: Testing the system's security measures to ensure that user data is protected.

9. Cross-browser compatibility testing: Testing the website on different browsers and devices to ensure that it is working correctly on all platforms.

10. User experience testing: Testing the website to ensure that it is user-friendly and easy to navigate.

11. Performance testing: Testing the website's response time and load time under different network conditions to ensure its proper functioning.

12. Accessibility testing: Testing the website's accessibility for users with disabilities to ensure its compatibility with assistive technologies.

13. SEO testing: Testing the website's ranking on search engines and ensuring that it is optimized for search engines.

14. Content testing: Testing the website's content for accuracy, relevance, and up-to-date information.

9.1 TEST CASES

TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT
Enter districts in drop-down list	District should be letters only	Click predict price button	Given district that contains letters are analysed	Should display price that contains district with letters only
Enter location in the drop-down list	Location that is specified	Click predict price button	Given location that contains specified places are analysed	Should display price that contains location with specified place
Enter loaction that is not in the drop-down list	Location that is specified	Click predict price button	Given location that selected "other" option	It will display an alert message
Enter BHK that contains numbers	BHK should be numbers only	Click predict price button	Give BHK that contains numbers are analyed	Should display price that contains BHK with numbers
Enter BHK that doesnot contains numbers	BHK contains numbers only	Click predict price button	Given BHK that doesnot contain numbers are analyed	An alert display will pop up
Enter number of bathrooms that contains numbers	Bathroom contains numbers only	Click predict price button	Given bathroom that contain numbers are analyed	Should display price that contains bathrooms with numbers
Enter number of bathrooms that doesnot contains numbers	Bathroom contains numbers only	Click predict price button	Given bathroom that doenot contain numbers are analyed	An alert display will pop up
Enter squarefeet that contains numbers	Sqaurefeet contains numbers only	Click predict price button	Given sqaurefeet that contains numbers are analysed	Should display price that contains sqarefeet with numbers
Enter squarefeet that doesnot contains numbers	Squarefeet contains numbers only	Click predict price button	Given squarefeet that not contains numbers are analysed	An alert display will pop up

Figure 18:Test Cases

10. CONCLUSION

Real estate is a rapidly growing business. Each year, more and more people are buying houses. People take into consideration several features before buying a house. Buying your own house is what every human wish for. Using this proposed model, we want people to buy houses and real estate at their rightful prices and want to ensure that they don't get tricked by sketchy agents who just are after their money. Additionally, this model will also help big companies by giving accurate predictions for them to set the pricing and save them from a lot of hassle and save a lot of precious time and money. Correct real estate prices are the essence of the market and we want to ensure that by using this model. House price prediction predicts house pricing based on different features. In this paper, multiple features are used to predict the sale price by using linear regression. Many evaluation techniques like Mean Squared Error, Root Mean Square Error and R-Squared Score are used on the machine learning algorithm. Finally, we are creating a webpage as front end which makes the system quite user friendly.

11. FUTURE ENHANCEMENT

One potential enhancement for the system could be to incorporate more advanced statistical models to improve the accuracy of the predictions. For instance, regression models such as linear, logistic, or polynomial regression, could be used to identify the relationships between the six selected attributes and the price of TVM houses. This would enable the system to make more accurate predictions and potentially provide additional insights into how each attribute affects the price.

Another enhancement could be to incorporate real-time market data to account for fluctuations in the housing market. For example, the system could be updated with the latest information about interest rates, employment rates, housing inventory, and other economic factors that impact housing prices. This would enable the system to provide more up-to-date predictions that reflect the current state of the market.

Additionally, the system could be enhanced by incorporating natural language processing (NLP) techniques to analyze unstructured data sources such as social media posts, online forums, and news articles. By analyzing these sources, the system could identify emerging trends and sentiments in the housing market that may affect the price of TVM houses. This would allow the system to provide more comprehensive and accurate predictions. If at all feasible, in the distant future, we can gather more information about the various districts in Kerala and expand the project.

Finally, the system could be enhanced by incorporating a user feedback loop to improve the accuracy of the predictions over time. Users could provide feedback on the accuracy of the predictions, which could be used to refine the system and improve the accuracy of future predictions.

In conclusion, these enhancements could significantly improve the accuracy and usefulness of the system for predicting the prices of TVM houses with only six attributes. By incorporating more advanced statistical models, real-time market data, NLP techniques, and a user feedback loop, the system could provide more accurate, up-to-date, and comprehensive predictions that better serve your needs.

12. BIBLIOGRAPHY

- Varma, A. Sharma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231.

- Fauria, Palak, and Anand Khandare. "Real Estate Price Prediction Using Machine Learning Algorithm." E conference on Data Science and Intelligent Computing. 2020.

- Musciano, Chuck, and Bill Kennedy. HTML & XHTML: The Definitive Guide: The Definitive Guide. " O'Reilly Media, Inc.", 2002.

- Aggarwal, Shalabh. Flask framework cookbook. Packet Publishing Ltd, 2014.

- Grinberg, Miguel. Flask web development: developing web applications with python. " O'Reilly Media, Inc.", 2018.

- Middleton, Neil, and Richard Schneemann. Heroku: up and running: effortless application deployment and scaling. " O'Reilly Media, Inc.", 2013.

- J Manasa, Radha Gupta, N S Narahari, “Machine Learning based Predicting House Prices Using Regression Techniques”, IEEE 2020.