

# **INTERNSHIP AT ENLIGHTEN INFOSYSTEMS**

## **A PROJECT REPORT**

*Submitted*

*by*

**Maulik S Raval**

**200050131090**

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

**BACHELOR OF ENGINEERING**

**In**

**Computer Science and Engineering**

**Babaria Institute of Technology, Vadodara**



**Gujarat Technological University, Ahmedabad**

**Academic year: 2023-24**



## Babaria Institute of Technology

Vadodara-Mumbai NH 8, Varnama, Vadodara, Gujarat

391243

## CERTIFICATE

This is to certify that the Internship report submitted along with the project entitled **Adaptive and Predictive Modelling** has been carried out by **Maulik Sanjaykumar Raval** under my guidance impartial fulfillment for the degree of Bachelor of Engineering in Computer and Science Engineering, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2023-24.

---

Ms. Jalpa P Bhatt

Internal Guide

---

Dr. Rock Upadhyay

Head of the Department

# INDUSTRY COMPLETION CERTIFICATE



2B, Neptune Tower, Near  
BPC House, BPC Road, Alkapuri,  
Vadodara - 390 007.  
+91 81402 31918  
[www.enlighteninfosystems.com](http://www.enlighteninfosystems.com)  
[contact@enlighteninfosystems.com](mailto:contact@enlighteninfosystems.com)



18<sup>th</sup> April, 2024

## Certificate

To Whomsoever It May Concern

This is to certify that **Mr. Maulik S. Raval** (Enrollment No.: 200050131090), student of **B.E (CSE), BIT, Vadodara** affiliated to Gujarat Technological University, Ahmedabad (Gujarat) has carried out **Internship** in the domain of **Data Science** at our organization from 16<sup>th</sup> January, 2024 to 18<sup>th</sup> April, 2024.

During this period he was found to be hardworking, honest, motivated and a committed individual. He took keen interest in the development assignments given to him, which met our expectations.

We wish him great success in all his future endeavors.

For, Enlighten Infosystems

*A.Z. Surti*  
Dr. Akil Z. Surti

Project Mentor



Product Development | Software Development | Mobile App. Development  
Website Development | Training & Consultancy | Domain & Hosting

# UNIVERSITY COMPLETION CERTIFICATE



## GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2022-2023

Date of certificate generation : 03 April 2024 (08:58:05)

This is to certify that, **Raval Maulik Sanjaykumar** ( Enrolment Number - 200050131090 ) working on project entitled with **Adeptive and Predictive Modeling** from **Computer Science & Engineering** department of **BABARIA INSTITUTE OF TECHNOLOGY, VARNAMA** had submitted following details at online project portal.

Internship Project Report

**Completed**

Name of Student : R a v a l M a u l i k  
Sanjaykumar

Name of Guide : Mrs. Jalpa Poojan Bhatt

Signature of Student : \_\_\_\_\_

\*Signature of Guide : \_\_\_\_\_

### Disclaimer :

This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

\*Guide has to sign the certificate, Only if all above activities has been Completed.



**Babaria Institute of Technology  
Mumbai NH 8, Varnama, Vadodara-  
391243,Gujarat**

**DECLARATION**

I hereby declare that the Internship / Project report submitted along with the Internship entitled **Data Science** submitted in partial fulfillment for the degree of Bachelor of Engineering in Computer and Science Engineering to Gujarat Technological University, Ahmedabad is a bonafide record of original project work carried out by me at Enlighten Infosystems under the supervision of Dr. Akil Z Surti and that no part of this report has been copied from any student's reports or taken from any other source, without providing due reference.

Name of the Student

Maulik Sanjaykumar Raval

Sign of Student

## **ACKNOWLEDGEMENT**

Any achievement doesn't depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. Several personalities have helped me. I would like to take this opportunity to thank all of them.

I would like to extend my deepest gratitude to Enlighten Infosystems and Dr. Akil Surti, who gave me the golden opportunity to do this wonderful internship.

I extend warm thanks to Dr. Rocky Upadhyay, H.O.D., Dept. of Computer and Science Engineering BITS, Vadodara for his constant encouragement, motivation, and guidance.

Further, I would like to thank my internal guide Ms. Jalpa P Bhatt for giving me constant guidance during the internship and helping me a lot in finalizing this internship within the limited time frame.

Thanking you

Maulik S Raval

## **ABSTRACT**

*This report encapsulates the experiences and learnings from a 12-week internship at Enlighten Infosystems in Vadodara, Gujarat, India, as part of the requirements for the completion of the Bachelor of Engineering degree in Computer Science and Engineering from Gujarat Technological University, Ahmedabad.*

*The internship focused on the theme of "Adaptive and Predictive Modelling," delving into the realms of data science and machine learning. It involved understanding the core concepts of these fields and their practical applications in real-world scenarios.*

*The report begins with an introduction to Enlighten Infosystems, the host organization for the internship, providing insights into its operations, services, and clientele. It further explores the significance of adaptive and predictive modeling, elucidating their roles in data analysis, forecasting, and pattern recognition.*

*The learning journey during the internship is divided into two phases. In the first phase, the emphasis was on grasping the fundamentals of data science and machine learning, including learning Python programming, exploring various libraries such as Numpy, Pandas, Matplotlib, and Seaborn, and gaining proficiency in data manipulation and visualization.*

*The second phase delved deeper into advanced concepts, including database exploration with MySQL and the exploration of machine learning algorithms like Histogram of Oriented Gradients (HOG), Region-based Convolutional Neural Networks (R-CNN), Fast R-CNN, Single Shot-Detector (SSD), and You Only Look Once (YOLO), Random Forest, linear Regression, LightGBM, etc. Each algorithm was examined in detail, along with its applications, advantages, and limitations.*

*Throughout the internship, practical projects and hands-on exercises were undertaken to apply the acquired knowledge and skills. The experience not only enhanced technical competencies but also fostered soft skills such as communication, teamwork, and problem-solving.*

*In conclusion, the internship at Enlighten Infosystems provided invaluable insights into the dynamic field of data science and machine learning. It equipped the intern with the requisite knowledge, skills, and practical experience, laying a strong foundation for future endeavors in the realm of technology and analytics.*

**-Maulik S Raval**

## **LIST OF FIGURES**

1.2 Adaptive and Predictive Modelling.....	03
1.3.2 Data Science and Machine Learning.....	04
2.2.1.1 NumPy.....	06
2.2.1.3 NumPy.....	06
2.2.2.1 Pandas.....	07
2.2.2.2 Pandas.....	08
2.2.2.3 Pandas.....	08
2.2.3.1 Matplotlib.....	09
2.2.3.2 Matplotlib.....	09
2.2.3.3 Matplotlib.....	10
2.2.3.4 Matplotlib.....	10
2.2.4.1 Seaborn.....	11
2.2.4.2 Seaborn.....	12
2.2.4.3 Seaborn.....	13
2.2.4.4 Seaborn.....	14
2.2.5.1 SciPy.....	14
2.2.5.2 SciPy.....	15
2.2.5.3 SciPy.....	15
2.2.5.4 SciPy.....	15

2.2.7.1 OpenCV.....	16
2.2.7.2 OpenCV.....	17
2.2.7.3 OpenCV.....	17
3.1.1 Hotel Management.....	18
3.1.2 Hotel Management.....	18
3.1.3 Hotel Management.....	19
3.1.4 Hotel Management.....	19
3.1.5 Hotel Management.....	20
3.1.6 Hotel Management.....	20
3.1.7 Hotel Management.....	21
5.1.1 Exploring Dataset.....	31
5.1.2 Exploring Dataset.....	32
5.2.1 Data Cleaning.....	33
5.2.2 Data Cleaning.....	33
5.2.3 Data Cleaning.....	34
5.2.4 Data Cleaning.....	34
5.2.5 Data Cleaning.....	35
5.2.6 Data Cleaning.....	35
5.2.7 Data Cleaning.....	36
5.2.8 Data Cleaning.....	36
5.2.9 Data Cleaning.....	37

5.2.10 Data Cleaning.....	37
5.3.1 EDA.....	38
5.3.2 EDA.....	39
5.3.3 EDA.....	39
5.3.4 EDA.....	40
5.3.5 EDA.....	40
5.3.6 EDA.....	41
5.3.7 EDA.....	41
5.3.8 EDA.....	42
5.3.9 EDA.....	42
5.4.1.1 Decision Tree.....	43
5.4.1.2 Decision Tree.....	43
5.4.1.3 Decision Tree.....	44
5.4.2.1 Random Forest.....	44
5.4.2.2 Random Forest.....	44
5.4.2.3 Random Forest.....	45
5.4.3.1 LightGBM.....	45
5.4.3.2 LightGBM.....	45
5.4.3.3 LightGBM.....	46
5.4.4.1 Linear Regression.....	46
5.4.4.2 Linear Regression.....	46

5.4.4.3 Linear Regression.....	47
5.4.5.1 Naïve Base.....	47
5.4.5.2 Naïve Base.....	47
5.5.1 Accuracy Comparison.....	48
5.6.1 Prediction Model.....	49
5.6.2 Prediction Model.....	49
5.6.3 Prediction Model.....	50
6.1.1 Setting-up Flask.....	51
6.2.1 Registration.....	52
6.2.2 Registration.....	52
6.2.3 Registration.....	53
6.3.1 Login.....	54
6.3.2 Login.....	54
6.3.3 Login.....	55
6.3.4 Login.....	55
6.4.1 Pikal File.....	56
6.4.2 Pikal File.....	56
6.5.1 index File.....	57
6.5.2 index File.....	57
6.5.3 index File.....	58
6.5.4 index File.....	58

6.5.5 index File.....	59
6.6.1 App.py File.....	59
6.6.2 App.py File.....	60
6.6.3 App.py File.....	60
6.6.4 App.py File.....	61
6.6.5 App.py File.....	61
6.7.1 GUI.....	62
6.7.2 GUI.....	63
6.7.3 GUI.....	63
6.7.4 GUI.....	64
6.7.5 GUI.....	64
6.7.6 GUI.....	65
6.7.7 GUI.....	65

## TABLE OF CONTENTS

<b>❖ DECLARATION.....</b>	<b>I</b>
<b>❖ ACKNOWLEDGEMENT.....</b>	<b>II</b>
<b>❖ ABSTRACT.....</b>	<b>III</b>
<b>❖ LIST OF FIGURES.....</b>	<b>V</b>
<b>1. INTRODUCTION.....</b>	<b>01</b>
1.1 OVERVIEW OF THE COMPANY.....	01
1.2 WHAT IS ADEPTIVE AND PREDICTIVE MODELLING?.....	02
1.3 DATA SCIENCE AND MACHINE LEARNING.....	03
<b>2. LEARNING PHASE-I.....</b>	<b>05</b>
2.1 LEARNING FUNDAMENTALS.....	05
2.2 GETTING STARTED WITH MACHINE LEARNING.....	05
<b>3. LEARNING PHASE-II.....</b>	<b>18</b>
3.1 EXPLORING DATABASE .....	18
3.2 EXPLORING MACHINE LEARNING ALGORITHM.....	21
<b>4. SYSTEM ANALYSIS.....</b>	<b>29</b>
4.1 INTRODUCTION.....	29
4.2 GOALS.....	29
4.3 FEASIBILITY STUDY.....	29
<b>5. SYSTEM DESIGN.....</b>	<b>31</b>
5.1 EXPLORING DATASET .....	31
5.2 DATA CLEANING.....	32
5.3 EXPLORATORY DATA ANALYSIS.....	38
5.4 ALGORITHMS.....	43

5.5 ACCURACY COMPARISION.....	48
5.6 REAL ESTATE PRICE PREDICTION MODEL.....	48
<b>6. IMPLEMENTATION.....</b>	<b>51</b>
6.1 SETTING-UP FLASK.....	51
6.2 REGISTRATION.....	51
6.3 LOGIN.....	53
6.4 PIKAL FILE.....	56
6.5 INDEX FILE.....	57
6.6 APP.PY.....	59
6.7 GRAFICAL USER INTERFACE.....	62
<b>7. CONCLUSION.....</b>	<b>66</b>
<b>8. REFERENCES.....</b>	<b>67</b>

## CHAPTER 1: INTRODUCTION

### 1.1 OVERVIEW OF THE COMPANY



**Company Name:** Enlighten Infosystems

**Address:** 2B, Neptune Tower, Near BPC House, BPC Road, Alkapuri,  
Vadodara - 390007

**Contact No:** +91 8140231918

**Email ID:** contact@enlighteninfosystems.com

Enlighten Infosystems, based in Vadodara, is a dynamic IT software development, training, professional service, and consulting firm that works with corporations, firms, and businesses. Enlighten Infosystems, a privately held company, provides software design, development, services, and support to the following areas and verticals: Education, Retail, Finance, Manufacturing and Engineering, Health Care Technology, and Travel and Hospitality. The company provides several IT software development services while also meeting the needs of its clientele. Its strength is delivering services on time and to the client's expectations. Enlighten Infosystems adheres to a few principles: Focus on quality and use cutting-edge technologies and technology development updates for our valued clients. The staff has demonstrated their abilities on medium- to large-scale projects with over 500 users. They also have specialized knowledge in the development of novel products. Enlighten Infosystems has worked on more than ten projects in the previous

years and has a customer base of more than 30 clients, providing satisfactory services. Aside from Vadodara, they have clientele from Gujarat, Maharashtra, Rajasthan, Delhi NCR, and Madhya Pradesh.

The internship aims to provide direction to the activities, help to focus on a result, and to assess the result achieved.

The two factors influencing my development were taken into account when formulating the goal before beginning the internship:

1. Understand what I have learned (theoretical and practical knowledge on my subject field) with actual work experience to complement my field-specific skills and gain new skills.
2. Apply and analyze at least one future skill.

## **1.2 WHAT IS ADEPTIVE AND PREDICTIVE MODELLING?**

Adaptive modeling is a technique used in data analysis, statistics, and machine learning where the model structure or parameters are adjusted dynamically based on incoming data or changing conditions. The primary goal of adaptive modeling is to create models that can continuously learn and improve over time without the need for manual intervention.

Predictive modeling is a statistical technique used to forecast future outcomes or trends based on historical data and patterns. It involves building a mathematical model that captures the relationships between input variables (also known as predictors or features) and the target variable (the variable to be predicted). The primary goal of predictive modeling is to make accurate predictions about future events or behaviors.

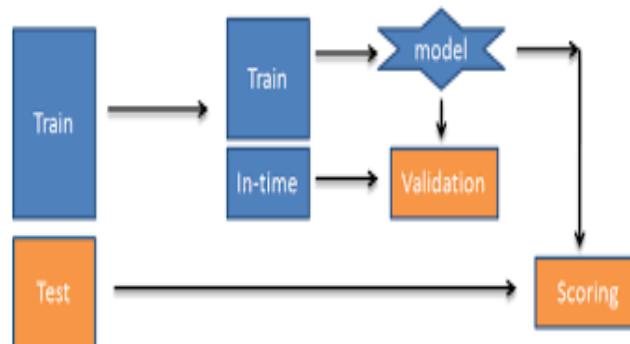


Figure 1.2 Adaptive and Predictive Modelling

## 1.3 DATA SCIENCE AND MACHINE LEARNING

### 1.3.1 What is Data Science?

It is the complicated analysis of enormous amounts of data in a company's or organization's repository. This study includes where the data came from, an analysis of its substance, and how this data may help the organization expand in the future. Organizational data is always in one of two forms: structured or unstructured. When we analyse this data, we gain significant insights into business or market patterns, which gives the company an advantage over its competitors because they have boosted their efficacy by spotting patterns in the data collection.

### 1.3.2 What is Machine Learning?

Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and techniques that enable computers to learn from data and improve their performance on a specific task without being explicitly programmed. In other words, machine learning algorithms learn patterns and relationships from data, allowing them to make predictions, decisions, and inferences based on new or unseen data.

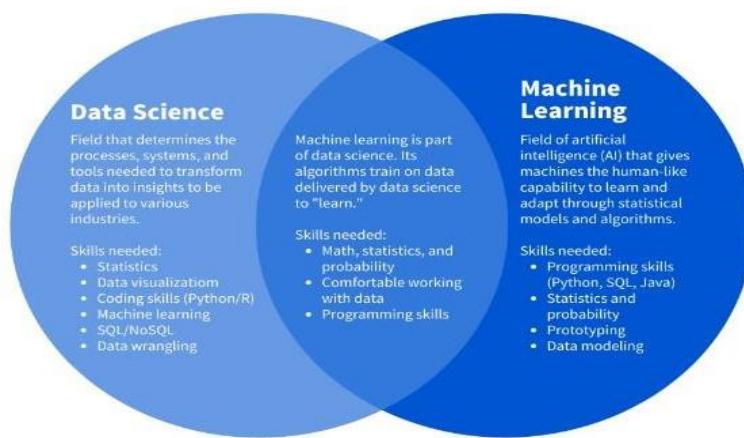


Figure 1.3.2 Data Science and Machine

## CHAPTER 2: LEARNING PHASE-I

### 2.1 LEARNING FUNDAMENTALS

At the start of my internship, I got to know the company better. An early stage of my life saw me go through the process of basic information transfer. I got my ideas about the Python language clear. From setting up the environment to learning the basics of any programming language, like data types, arrays, different kinds of looping, and so on. I learned a lot about data science and machine learning, such as the different kinds of data, datasets, methods, and different libraries.

### 2.2 GETTING STARTED WITH DATA SCIENCE AND MACHINE LEARNING

I learned more about what Data Science and Machine Learning are. In addition, I learned about the ideas behind machine learning (ML) and its two types, supervised learning and unsupervised learning. I then used PyCharm to put the different methods into action. I also learned a few more things, such as how to clean data, do exploratory data analysis, use graphics, prepare data for analysis, and do small amounts of statistics. As I worked on the newer jobs, I also learned the basics of NumPy, Pandas, Matplotlib, Seaborn, Data frames, and other tools. Finally, I worked on datasets from the real world to figure out how to use ML techniques on them.

#### 2.2.1 NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

```

1 import numpy as np
2
3 a = np.array([10, 20, 30])
4 b = np.array([40, 50, 60])
5 print(a)
6 print(b)
7
8 # Different mathematical Operations on two different arrays
9 print(np.add(a, b))
10 print(np.subtract(a, b))
11 print(np.multiply(a, b))
12 print(np.divide(a, b))
13 print(np.mod(a, b))
14 print(np.power(a, b))
15 print(np.sqrt(b))
16
17 # it is an N-dimentional Array
18 arr = np.array([[10, 20, 30], [20, 30, 40], [30, 40, 50]])
19 print(type(arr))
20 # output - <class 'numpy.ndarray'> here, nd stands for N dimentional array
21
22 #Slicing in array
23 arr1 = np.array([10, 20, 30, 40])
24 print(arr1[0:3])
25 print(arr1[0:])
26 print(arr1[:4])
27 print(arr1[3:])

```

Figure 2.2.1.1

```

29 #slicing for 2D array
30 arr2 = np.array([[10, 20, 30], [20, 30, 40]])
31 print(arr2[0:2,0:2])
32 print(arr2[0:2,0:3])
33 print(arr2[0,1:4])
34 print(arr2[1,1:3])
35 print(np.shape(arr2)) #Shape function will show the number of rows and columns(here output will 2 rows and 3 columns)
36 print(np.size(arr2)) # Size function will show you the total size of the array including all rows and columns
37 print(np.ndim(arr2)) # ndim function will show you the number of dimention of specific array
38 print(arr2.dtype) #dtype will give you the data type which is get used in specified array
39 print(arr2.astype(float)) #astype function wil used to change the datatype of specified array
40 print(len(arr2)) # number of nested values
41
42 # Combining and Splitting arrays
43     #methods to concatenate array
44 arr3 = np.array([10, 20, 30])
45 arr4 = np.array([40, 50, 60])
46 print(np.concatenate((arr3, arr4)))
47     #method to concanate according to axis
48 arr5 = np.array([[1,2,3],[4,5,6]])
49 arr6 = np.array([[7,8,9],[11,12,13]])
50 print(np.concatenate( arrays: (arr5, arr6),axis=1)) #vertically
51 print(np.concatenate( arrays: (arr5, arr6),axis=0)) #horizontally
52 print(np.hstack((arr5, arr6))) # hstack = horizontally concatenation
53 print(np.vstack((arr5, arr6))) # vstack = vertically concatenation

```

Figure 2.2.1.2

```
55     #methods to split array
56 arr7 = np.array([1,2,3,4,5,6,7,8,9])
57 print(np.array_split(arr7, indices_or_sections: 3))
58 c = np.array_split(arr7, indices_or_sections: 3)
59 print(c[0])
60 print(c[1])
61
62 #Adding and Removing elements from array
63 arr8 = np.array([1,2,3,4,5,6])
64 arr9 = np.array([10,11,12,13,14,15,16])
65 print(np.append(arr8, values: 10))
66 print(np.insert(arr8, obj: 1, values: 40))
67 print(np.insert(arr5, obj: 1, values: [100,200],axis=1))
68 print(np.insert(arr5, obj: 1, values: [100,200,300],axis=0))
69 print(np.delete(arr9, obj: 1))
70 print(np.delete(arr5, obj: 1, axis=1))
71
72 #sort,filter,search
73     #Sorting
74 arr10 = np.array([1,22,5,3,9,7,11,0])
75 print(np.sort(arr10))
76 print(np.sort(arr6))
77     #search
78 arr11 = np.array([1,222,45,2,6,8])
79 s = np.where(arr11 == 222)
80 print(s)
```

Figure 2.2.1.3

## 2.2.2 Pandas

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's Active Python.

```

1 import numpy.random
2 import pandas as pd
3 dict1 = {'Name': ['John', 'Carol', 'David', 'Floki']
4          , 'Age': [20, 30, 40, 50]
5          , 'Gender': ['M', 'F', 'M', 'M']
6          , 'Phone': ['023456789', '7895641230', '5731594560', '987654321']
7          }
8 df = pd.DataFrame(dict1)
9 print(df)
10
11 #to create a CSV file
12 df.to_csv('pandas_example.csv')
13 df.to_csv(path_or_buf='pandas_NoIndex_example.csv', index=False)
14 print("")
15 #to see the initial row
16 print(df.head(2))
17
18 print("")
19 #to see last few rows
20 print(df.tail(2))
21
22 print("")
23 # to Do all statistical operations
24 print(df.describe())

```

Figure 2.2.2.1

```

39 #dataframe
40 new_dataframe = pd.DataFrame(numpy.random.rand(334,5), index=numpy.arange(334))
41 print(new_dataframe)
42 print(type(new_dataframe))
43 print(new_dataframe.describe())
44 # new_dataframe[0][0] = "Maulik"
45 # print(new_dataframe.head())
46 print(new_dataframe.to_numpy())
47
48 #Transpose= row into column and column into row
49 print(new_dataframe.T)
50
51 # sorting index on row
52 print(new_dataframe.sort_index(axis=0, ascending=False))
53
54 #sorting index on columns
55 print(new_dataframe.sort_index(axis=1, ascending=False))
56
57 #view
58 new_dataframe2 = new_dataframe
59 new_dataframe2[0][1] = "Hello World"
60 print(new_dataframe)
61
62 #LOC function to edit dataframe
63 new_dataframe.loc[0,0] = 7980
64 print(new_dataframe)
65
66 new_dataframe.columns = list("ABCDE")
67 print(new_dataframe)
68 new_dataframe.loc[0,'B'] = 779
69 print(new_dataframe)

```

Figure 2.2.2.2

### 2.2.3 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, Python shells, web application servers, and various graphical user interface toolkits.

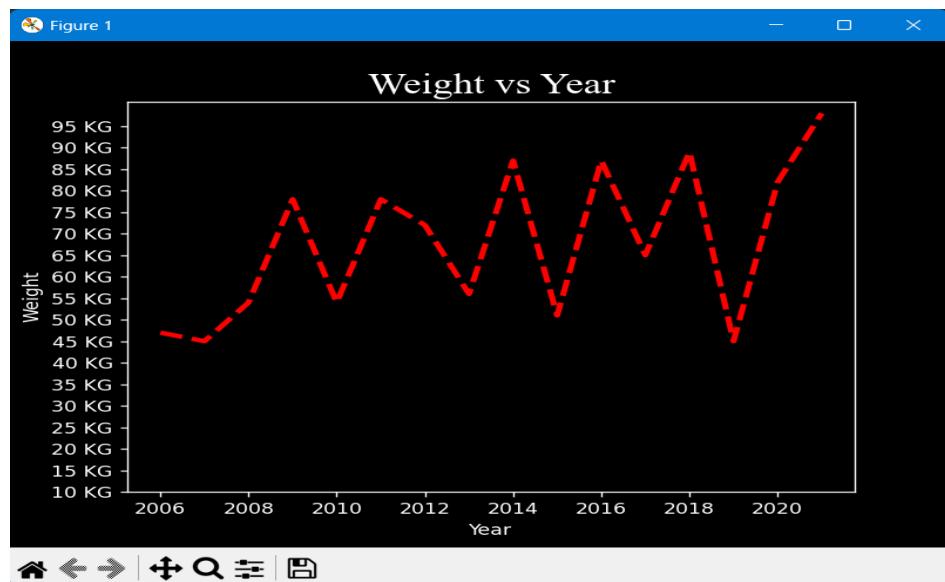


Figure 2.2.3.1

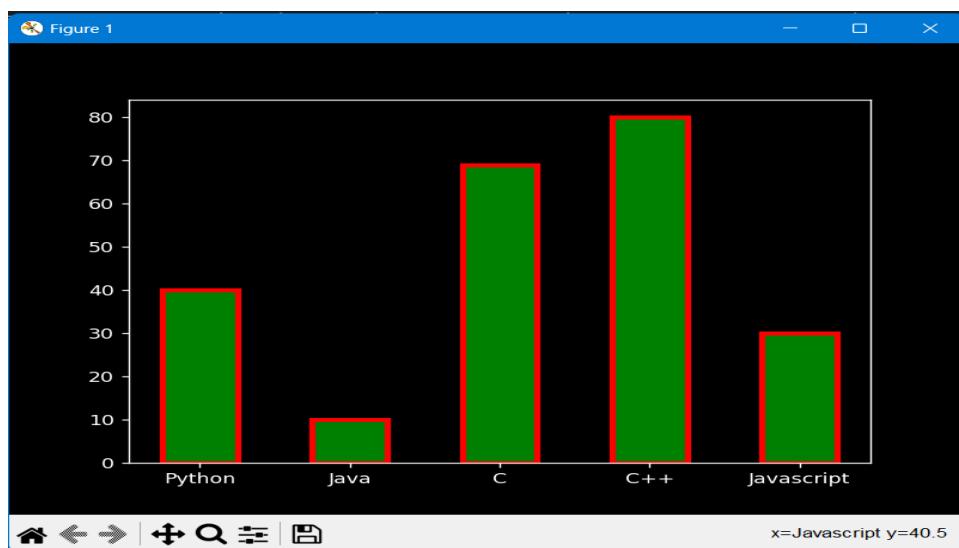


Figure 2.2.3.2

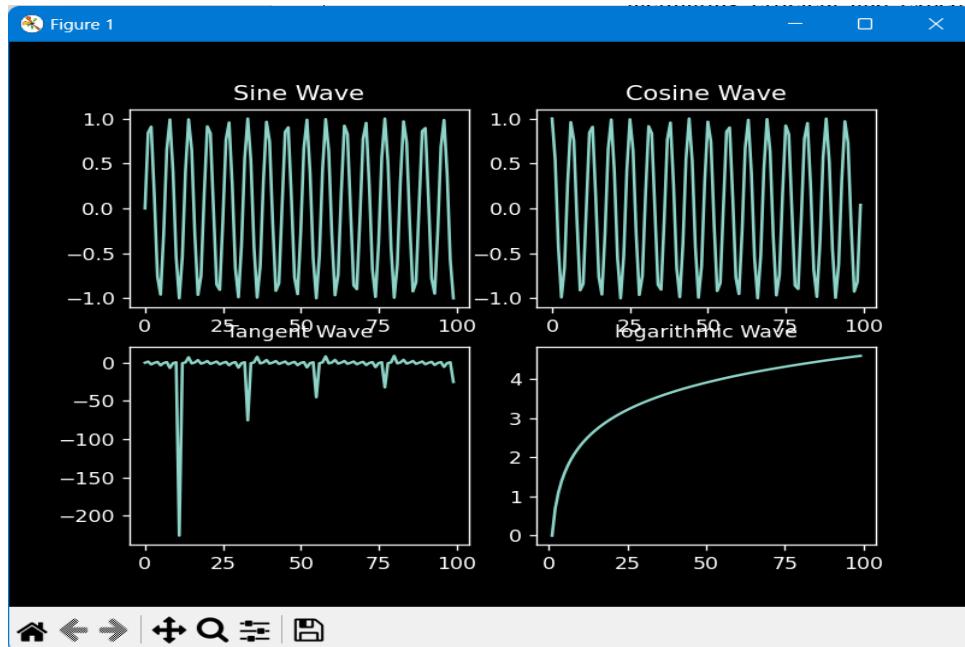


Figure 2.2.3.3

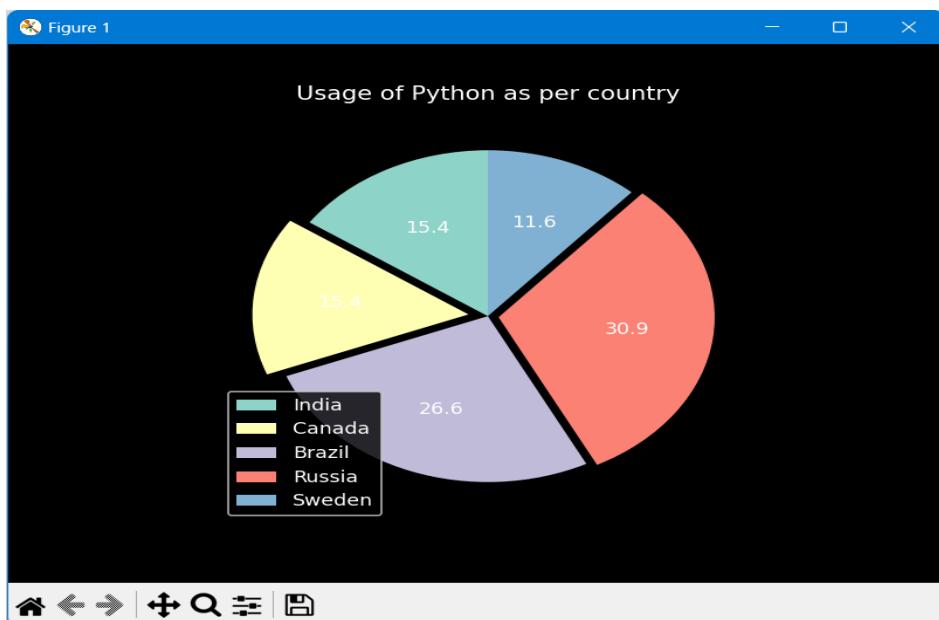


Figure 2.2.3.4

#### 2.2.4 Seaborn

Seaborn is a widely recognized Python library for data visualization that is constructed upon Matplotlib. It provides a user-friendly interface for generating visually captivating and useful statistical images. The software is specifically engineered to seamlessly integrate with Pandas dataframes, facilitating efficient and expedient visualization and exploration of data.

Seaborn provides a diverse range of robust tools for information visualization, encompassing scatter plots, line plots, bar plots, heat maps, and numerous more options. Additionally, it offers assistance for computationally sophisticated statistical analysis techniques, including regression analysis, distribution plots, and categorical plots.

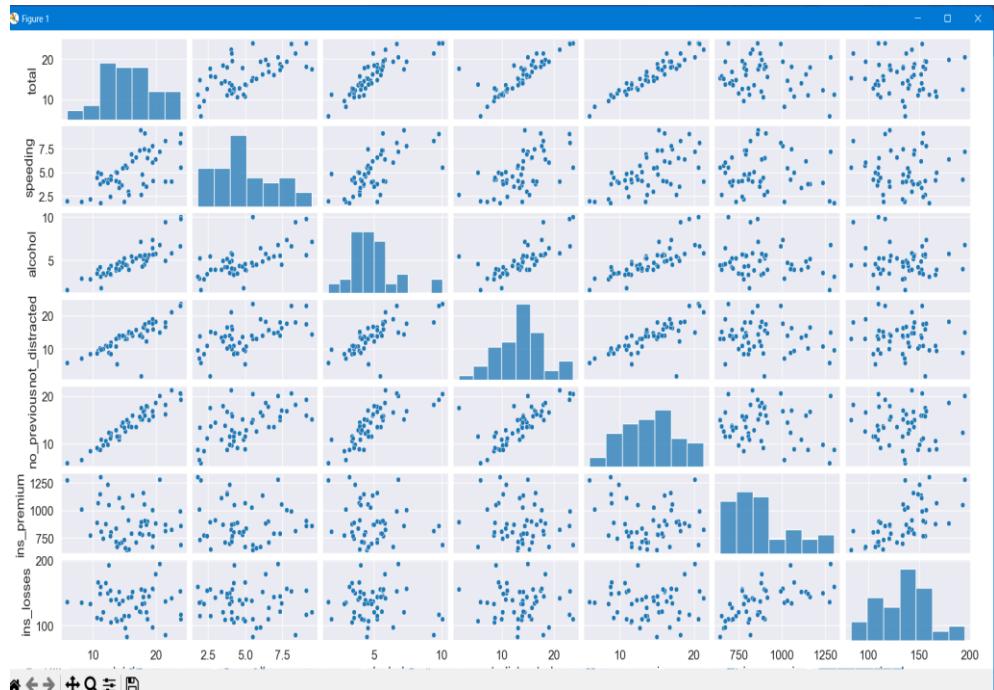


Figure 2.2.4.1

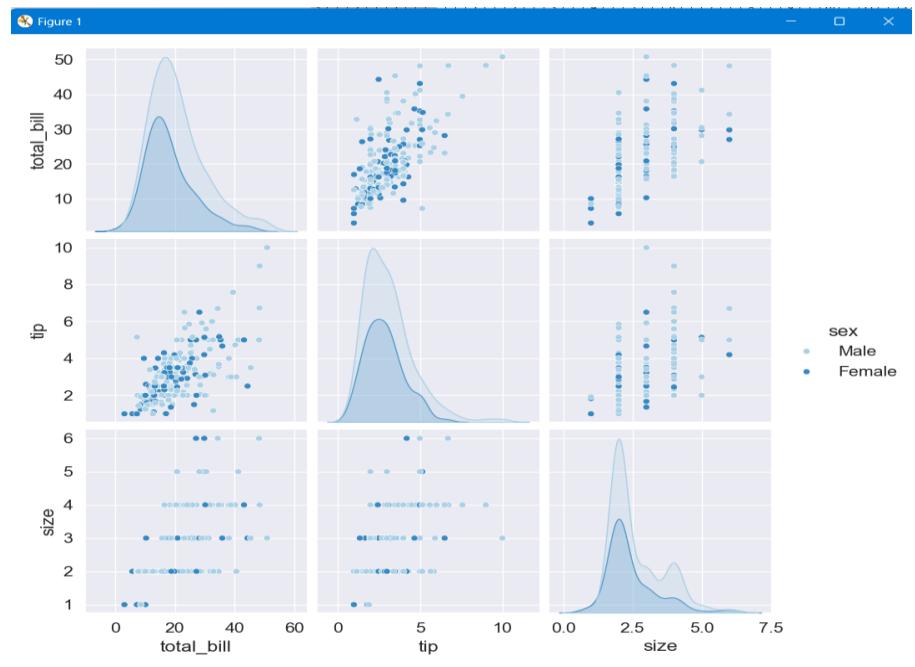


Figure 2.2.4.2

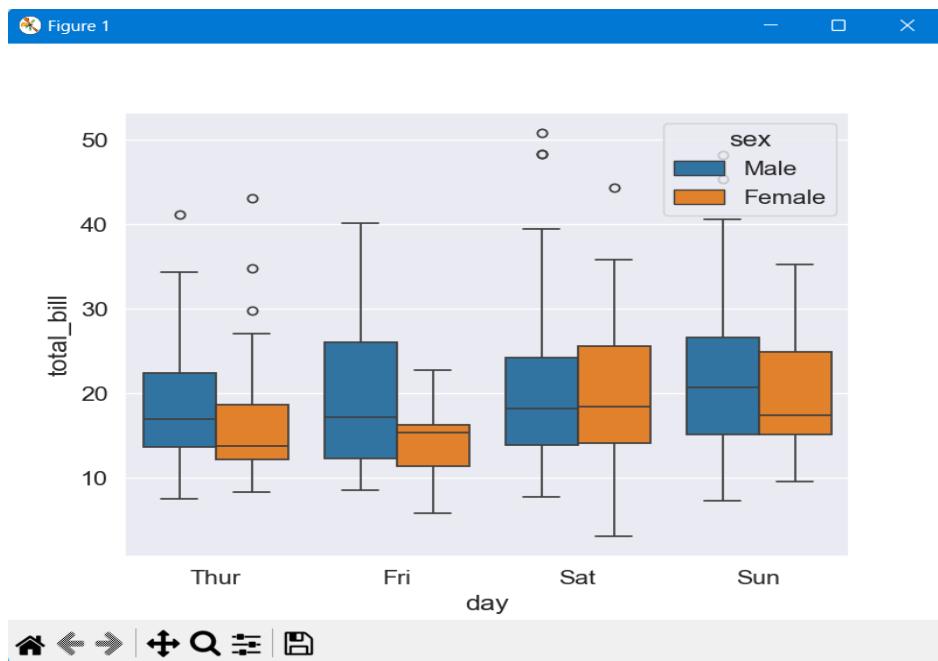


Figure 2.2.4.3

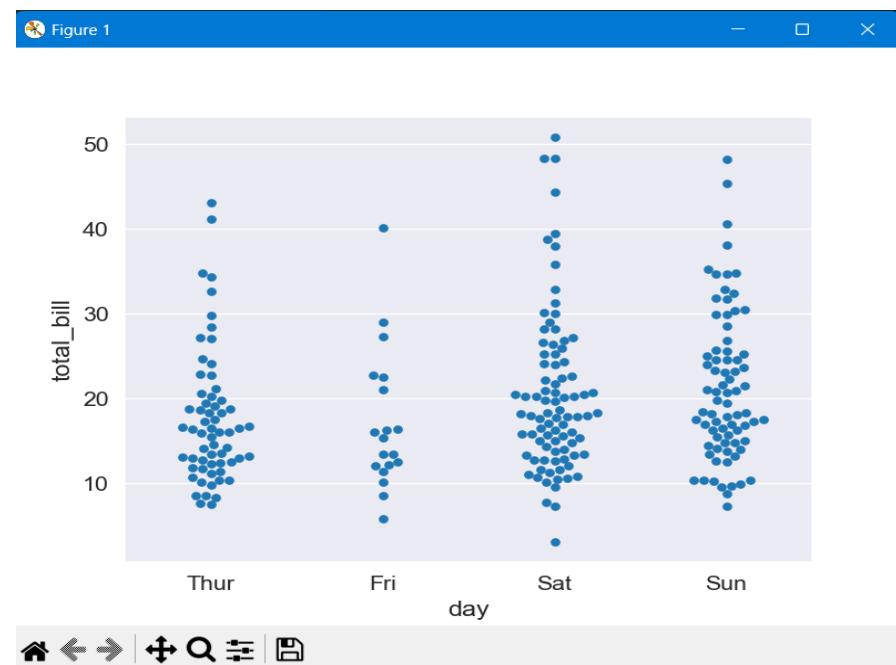


Figure 2.2.4.4

## 2.2.5 SciPy

SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands. As mentioned earlier, SciPy builds on NumPy and therefore if you import SciPy, there is no need to import NumPy.

```

5
6
new *
7  def f(x):
8      return (x - 3) ** 2
9
10
11 res = minimize(f, x0: 2)
12 print(res)
13
14 # Constant Sub Packages
15
16 # c = speed of vacuum
17 # h = Plank's const.
18 # G = Newton's gravitational const.
19 # e = Elementary charge
20 # R = Molar gaws const.
21 # Avogadro = Avogradro const.
22 # k = Boltzman const
23 # m_e = Mass of electron
24 # m_p = Mass of proton
25 # m_n = Mass of neutron
26 # etc....
27
28 # Mathematical Constants
29 from scipy.constants import pi, golden_ratio
30
31 print(f"Pi = {pi}")
32 print(f"Golden Ratio = {golden_ratio}")
33 radius = 5
34 print(f"Area = {pi * radius * radius}")
35 from scipy.constants import m_n, Avogadro

```

Figure 2.2.5.1

```

37 print(f"Mass of Neutron: {m_n} Kg")
38 print(f"Avogadro's Number: {Avogadro}")
39
40 # Values of Physical Constants
41 from scipy.constants import physical_constants
42
43 print(physical_constants["alpha particle mass"])
44 for key, value in physical_constants.items():
45     print(f"{key}: {value}")
46
47 # Conversion of Unit
48 from scipy.constants import kilo, yotta, deci
49
50 print(f"Kilo = {kilo} gram")
51 meters = 800
52 print(f" In Kilometers: {meters / yotta}")
53 print(f" In Kilometers: {meters / deci}")
54
55 # Conversation of temprature
56 from scipy.constants import convert_temperature
57
58 temprature_in_fahrenheit = 90
59 temprature_in_celsius = convert_temperature(temprature_in_fahrenheit, old_scale: "F", new_scale: "C")
60 print(f"Temprature in Celsius: {temprature_in_celsius}")
61
62 temprature_in_celsius = 0
63 temprature_in_kelvin = convert_temperature(temprature_in_fahrenheit, old_scale: "C", new_scale: "K")
64 print(f"Temprature in Kelvin: {temprature_in_kelvin}")

```

Figure 2.2.5.2

```

66 import numpy as np
67 import matplotlib.pyplot as plt
68 from scipy.cluster.vq import kmeans2
69
70 a = np.random.multivariate_normal(mean=[0, 1], cov=[[2, 1], [1, 1.5]], size=200)
71 b = np.random.multivariate_normal(mean=[0, 8], cov=[[1, -1], [-1, 3]], size=400)
72 c = np.random.multivariate_normal(mean=[6, 4], cov=[[5, 0], [0, 1.2]], size=250)
73 z = np.concatenate((a, b, c))
74 np.random.shuffle(z)
75 # plt.scatter(z[:,0], z[:,1])
76 # plt.show()
77
78 K = 3
79
80 centroids, labels = kmeans2(z, K, minit="points")
81 print(f"Centroids: {centroids}")
82 print(f"Z: {z}")
83 print(labels)
84
85 cluster_1 = z[labels == 0]
86 cluster_2 = z[labels == 1]
87 cluster_3 = z[labels == 2]
88 plt.scatter(cluster_1[:, 0], cluster_1[:, 1], label="cluster_1")
89 plt.scatter(cluster_2[:, 0], cluster_2[:, 1], label="cluster_2")
90 plt.scatter(cluster_3[:, 0], cluster_3[:, 1], label="cluster_3")
91 plt.scatter(centroids[:, 0], centroids[:, 1], label="Centroids", color="black")
92 plt.show()

```

Figure 2.2.5.3

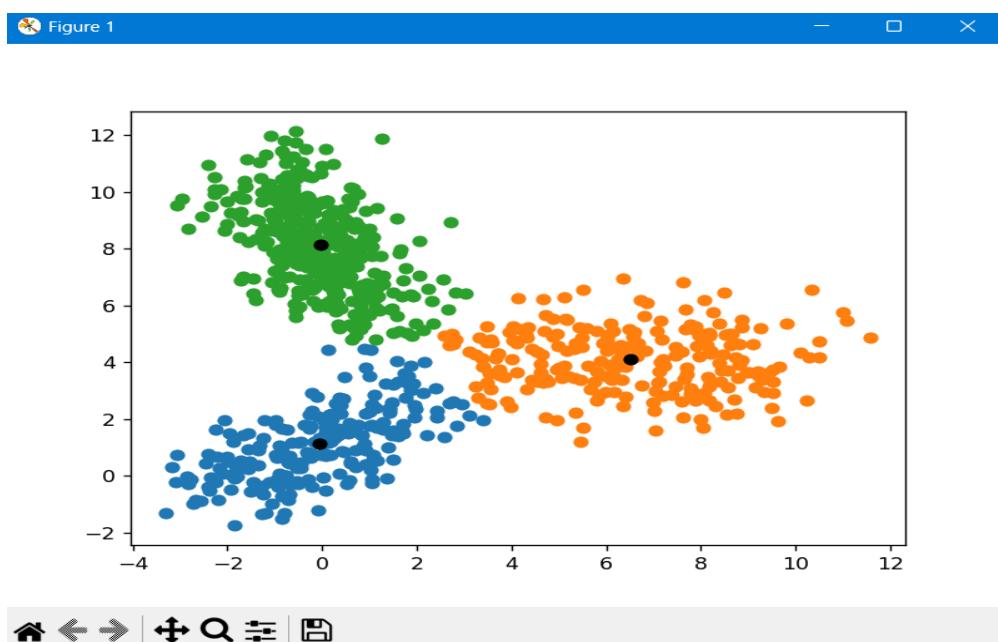


Figure 2.2.5.4

### 2.2.6 Scikit learn

Scikit-learn, also known as sklearn, is an open-source, machine learning and data modelling library for Python. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python libraries, NumPy and SciPy. It supports a standardized and concise model interface across models. For example, Scikit-learn makes use of a simple fit/predict workflow model for its classification algorithms.

### 2.2.7 OpenCV

OpenCV (Open-Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

```

1 import cv2
2 import numpy as np
3
4 img = cv2.imread(r"E:\data science Internship\pythonProject1\archive\Vegetable Images\test\Tomato\1031.jpg")
5 print(img.shape)
6
7 #How to convert RGB Image to a Gray scale Image
8
9 img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
10 print(img_gray.shape)
11 cv2.imshow( winname: "Vegetable_Gray", img_gray)
12 cv2.waitKey(0)
13
14 # Manipulate RGB
15 img_blue = img[:, :, 0]
16 img_green = img[:, :, 1]
17 img_red = img[:, :, 2]
18
19 new_img = np.hstack((img_blue, img_green, img_red)) # To show multiple images in same tab = hstack -- NumPy
20
21 # Image Resize
22 img_resize = cv2.resize(img, dsize: (256, 256))
23 print(img_resize.shape)
24
25 cv2.imshow( winname: "Vegetable", img_resize)
26 cv2.waitKey(0)
27

```

Figure 2.2.7.1

```

28 # Flip Image
29 img_flip = cv2.flip(img, flipCode: 0) # flipCode : 0 -- vertical flip, flipCode : 1 -- horizontal
30
31 cv2.imshow( winname: "Vegetable", img_flip)
32 cv2.waitKey(0)
33
34 # Cropping
35 img_crop = img[0:224, 0:200]
36 cv2.imshow( winname: "Vegetable", img_crop)
37 cv2.waitKey(0)
38
39 # Save Image
40 cv2.imwrite( filename: 'cropped_tomato.png', img_crop)
41
42 #Draw Shape ( which is used to scan the shape like square, circle, etc.
43 raw_img = np.zeros((512,512,3)) # to create a black image in which we will create shapes and te
44 #Rectangle
45 cv2.rectangle(raw_img, pt1=(100,100), pt2=(300,300), color=(255,0,0), thickness=3) #If you want t
46 #Circle
47 cv2.circle(raw_img, center=(100,400), radius=50, color=(0,0,255), thickness=3)
48 #Line
49 cv2.line(raw_img, pt1=(0,0), pt2=(512,512), thickness=2, color=(0,255,0))
50 #Text
51 cv2.putText(raw_img, org=(350,100), fontScale=4, color=(0,255,255), thickness=2,
52             lineType=cv2.LINE_AA, text="Hello", fontFace=cv2.FONT_HERSHEY_PLAIN)

```

Figure 2.2.7.2

```

55 cv2.imshow( winname: "Vegetable", raw_img)
56 cv2.waitKey(0)
57
58 # Live Direct drawing
59 new *
60 def draw(event, x, y, flags, params):
61     pass
62 cv2.namedWindow(winname = "Window")
63 img_raw = np.zeros((512,512,3))
64
65 while True:
66     cv2.imshow( winname: "Image", img_raw)
67
68     if cv2.waitKey(1) & 0xFF == ord('x'):
69         break
70 cv2.destroyAllWindows()

```

Figure 2.2.7.3

## CHAPTER 3: LEARNING PHASE - II

### 3.1 EXPLORING DATABASE

After going through the basics, I decided to add an authentication factor to the project. For that purpose, I explored and learned about one of the database systems – MySQL. It is based on relational database management systems (RDBMS).

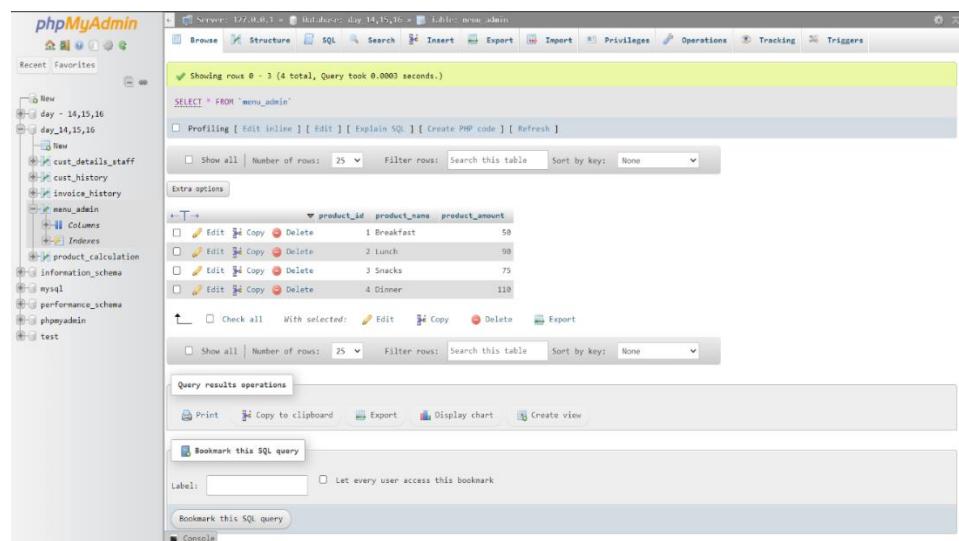


Figure 3.1.1

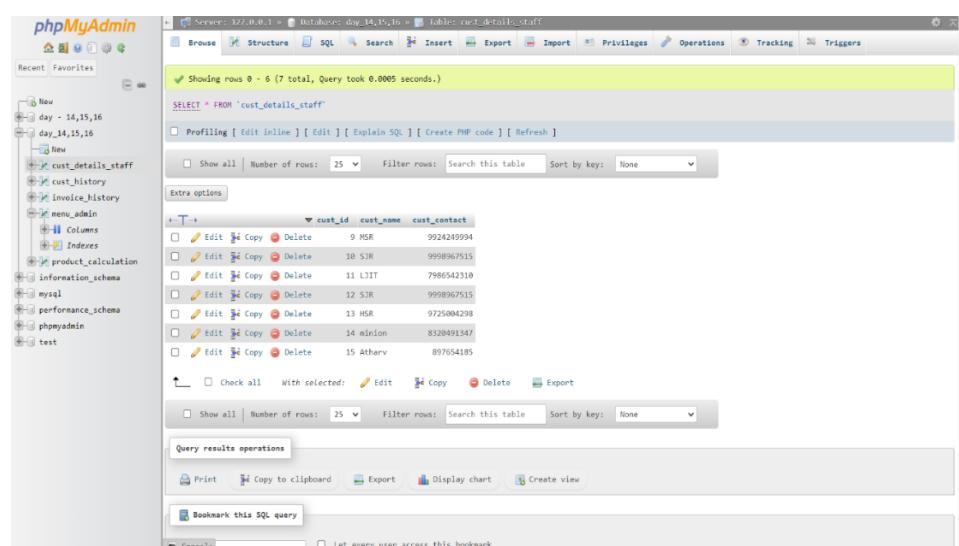


Figure 3.1.2

```
"C:\data science Internship\venv\Scripts\python.exe" "C:\data science Internship\pythonProject1\day_14,15,16\main.py"
Connection successful
Welcome to the Hotel Management System
Press 'A' to login as an Admin
Press 'S' to login as Staff
A
Enter your Admin Password: 0410
You have Logged in Successfully...
Welcome Admin...

Press 1 to Edit Menu Items.
Press 2 to check history:
1
press 1 to add food item to database.
press 2 to Update food item to database.
press 3 to Delete food item to database.
press 4 to View details of all food items to database.
press 5 to Search details of food item to database.
press 6 to exit.

Please select an option from the menu: 4
+-----+-----+-----+
| product_id | Product Name | Price |
+-----+-----+-----+
| 1 | Breakfast | 50.0 |
| 2 | Lunch | 90.0 |
| 3 | Snacks | 75.0 |
| 4 | Dinner | 110.0 |
+-----+-----+-----+

Press 1 to Edit Menu Items.
Press 2 to check history:
ence Internship > pythonProject1 > day_14,15,16 > main.py
1:1 (58 chars, 1 line)
```

Figure 3.1.3

```
Press 1 to Edit Menu Items.
Press 2 to check history:
2
+-----+-----+-----+-----+-----+
| Invoice ID | Customer ID | Customer Contact | Price | Date |
+-----+-----+-----+-----+-----+
| 8791 | 9 | 2147483647 | 445.0 | 2024-02-05 12:32:55 |
| 5294 | 9 | 2147483647 | 330.0 | 2024-02-05 12:32:55 |
| 5446 | 9 | 9924249994 | 710.0 | 2024-02-05 12:32:55 |
| 6743 | 9 | 9924249994 | 1155.0 | 2024-02-05 12:32:55 |
| 4023 | 9 | 9924249994 | 360.0 | 2024-02-05 12:32:55 |
| 6315 | 9 | 9924249994 | 370.0 | 2024-02-05 12:39:37 |
| 4169 | 13 | 9725004298 | 640.0 | 2024-02-05 12:41:40 |
| 6104 | 9 | 9924249994 | 295.0 | 2024-02-05 21:58:57 |
+-----+-----+-----+-----+-----+
Enter your Invoice ID for which you would like to see the details: 5294
+-----+-----+-----+-----+
| Invoice ID | Product Name | Product Quantity | Total Amount |
+-----+-----+-----+-----+
| 5294 | Breakfast | 3 | 150.0 |
| 5294 | Lunch | 2 | 180.0 |
+-----+-----+-----+-----+

Press 1 to Edit Menu Items.
Press 2 to check history:
|
```

Figure 3.1.3

```

Logging out as an Admin...
Press 'A' to login as an Admin
Press 'S' to login as Staff
S
Enter your Password to login: 0410
Welcome to Hotel Management System

Press 1 if you are going to add new user
Press 2 if you want to access old user

Enter your choice: 1

....ADD NEW CUSTOMER DETAILS.....

Enter Customer Name: Atharv
Enter Customer Contact Number: 897654185

Customer Details Added Successfully...

Press 1 to Buy Products...
Press 2 to Log out...
1

```

Figure 3.1.4

```

Enter your contact number : 897654185
+-----+-----+
| Product ID | Product Name | Price |
+-----+-----+-----+
| 1 | Breakfast | 50.0 |
| 2 | Lunch | 90.0 |
| 3 | Snacks | 75.0 |
| 4 | Dinner | 110.0 |
+-----+-----+
What would you like to order?
Please type the Product ID from the menu: 2
Enter the quantity of the product: 3
Order is taken. Thank you!
Do you want to add more products to your order? (yes/no): yes
+-----+-----+-----+
| Product ID | Product Name | Price |
+-----+-----+-----+
| 1 | Breakfast | 50.0 |
| 2 | Lunch | 90.0 |
| 3 | Snacks | 75.0 |
| 4 | Dinner | 110.0 |
+-----+-----+
What would you like to order?
Please type the Product ID from the menu: 4
Enter the quantity of the product: 1
Order is taken. Thank you!
Do you want to add more products to your order? (yes/no): no
Order processing completed. Final Amount: $380.0
Here's your Invoice ID for the payment: 3465
Press 'A' to login as an Admin
Press 'S' to login as Staff

```

Figure 3.1.5

```

Press 'S' to login as Staff
S
Enter your Password to login: 0410
Welcome to Hotel Management System

Press 1 if you are going to add new user
Press 2 if you want to access old user

Enter your choice: 2
hi...

....PLEASE PROVIDE SOME DETAILS TO LOGIN.....


Enter your Contact Number: 897654185
Press 1 to buy product
Press 2 to view your previous history.2
+-----+
| Contact Number | Customer ID | Invoice ID | Final Amount | Date Time |
+-----+
| 897654185     |      15    |    3465    |     380.0    | 2024-03-18 12:42:56 |
+-----+
Enter your Invoice ID for which you would like to see the details: 3465
+-----+
| Invoice ID | Product Name | Product Quantity | Total Amount |
+-----+
| 3465       |   Lunch        |         3          |    270.0    |
| 3465       |   Dinner        |         1          |    110.0    |
+-----+
Press 'A' to login as an Admin
Press 'S' to login as Staff
|

```

Figure 3.1.6

## 3.2 EXPLORING MACHINE LEARNING ALGORITHMS

Without the need for explicit programming, machine learning algorithms are models that enable computers to recognize patterns, forecast, or form opinions based on data. To lift towards final project, I have explored various machine learning algorithms Such as HOG, R-CNN, SSD, YOLO including their pros and cons.

### 1.1.1 Histogram of Oriented Gradients (HOG)

A feature extractor is what HOG uses to find things in a picture. In HOG, a feature description is a picture of a part of an image from which we only take out the most important details and ignore the rest. This is what the feature descriptor does: it turns the image's total size into an array or feature vector. The gradient orientation method is used in HOG to find the most important parts of an image.

There are some issues with the Histogram of Oriented Gradients (HOG): Even though it was very new when it first came out, this method had a lot of issues. It takes a long time to do complicated calculations on pixels in pictures, and it doesn't always work well when you need to find things in small spaces.

What is the best time to use HOG? – HOG is often the best way to try other object detection algorithms and see how well they work. No matter what, HOG is useful for finding most objects and recognizing face landmarks with a fair amount of accuracy.

HOG is often used to find people going down the street because it has smooth edges. Object recognition of certain things is another general use.

### 3.2.2 Region-based Convolutional Neural Networks (R-CNN)

We use selective features in the R-CNN models to try to get the most important features out, which are usually around 2000 features. A selective search algorithm can help you figure out how to choose the most important extractions. This can lead to these more important regional ideas.

There are some drawbacks of this algorithm: this algorithm takes long time to get all the region ideas and then the best areas, even though the CNN models that have already been trained do a good job of getting features. Not only does the R-CNN model learn slowly, but it also takes a long time to make predictions. The answer needs a lot of computer power, which makes it more likely that the process will work. The building as a whole can be thought of as quite expensive because of this.

When Should You Use R-CNN? – An R-CNN model that works like the HOG object detection method should be used as a starting point to check how well the object detection models work. Most of the time, the more recent versions of R-CNN are better because the time it takes to guess images and objects is longer than expected.

R-CNN can be used in a number of different ways to solve different kinds of object recognition problems. Tracking things from a camera on a drone, finding text in an image, and turning on object detection in Google Lens are some examples.

### **3.2.3 Fast R- CNN**

In Fast R- CNN, all of the picture is sent through the pre-trained Convolutional Neural Network in the fast R-CNN method, rather than each sub-segment. The region of interest pooling method is unique because it uses two inputs—a pre-trained model and a selective search algorithm—to create an output layer that is fully linked. We will learn more about the Faster R-CNN network in this part. This network is better than the fast R-CNN mode.

**Issues:** One of the main problems with the Faster R-CNN method is that it takes too long for different items to be given. The type of system used can sometimes change the speed. Other CNN methods take longer to use. This one takes less time. R-CNN needs about 40 to 50 seconds to figure out what things are in a picture most of the time. Fast R-CNN only takes two seconds, but it finds the best answer in 0.2 seconds.

### **3.2.4 Single Shot Detector (SSD)**

The single-shot detector for multi-box predictions is one of the fastest ways to do object detection jobs in real time. The Faster R-CNN methods can make accurate predictions, but the whole process takes a long time and needs the real-time job to run at about 7 frames per second, which is not good.

**Limitations of SSD:** The single-shot detector for multi-box predictions is one of the fastest ways to do object detection jobs in real time. The Faster R-CNN methods can make accurate predictions, but the whole process takes a long

time and needs the real-time job to run at about 7 frames per second, which is not good.

The single-shot detector for multi-box predictions is one of the fastest ways to do object detection jobs in real time. The Faster R-CNN methods can make accurate predictions, but the whole process takes a long time and needs the real-time job to run at about 7 frames per second, which is not good.

Different datasets, like the PASCAL VOC, COCO, and ILSVRC datasets, can be used to train and test the Single-shot detection. Larger objects, like people, tables, chairs, and other similar things, can be found quickly and accurately by them.

### **3.2.5 YOLO (You Only Look Once)**

Many people know that YOLO is one of the best model designs and ways to find things. When you search Google for algorithms on object recognition, the YOLO design is often the first thing that comes up. Next, we'll talk about the different types of YOLO. With one of the best neural network designs, the YOLO model is able to be very accurate and work quickly. A lot of people like it because it works quickly and correctly.

One of the best things about YOLO is that it can do a lot of work very quickly compared to other object recognition and training methods. The YOLO algorithm not only has a fast-computing speed, but it also has a high level of accuracy because it reduces the background mistakes that other methods make. YOLO's design makes it easier for the model to learn about and understand many things at once.

Nevertheless, there are some limitations of YOLO such as, The memory rate is low, so you can't see small details in a picture or movie. You can't find two things that are very close to each other because of how bounding boxes work

### 3.2.6 Random Forest

Random Forest is a powerful ensemble learning algorithm in machine learning that operates by constructing a multitude of decision trees during training and outputting the mode (classification) or mean prediction (regression) of the individual trees. Each tree in the forest is trained on a random subset of the training data and a random subset of the features.

This randomness helps to decorrelate the individual trees, reducing the risk of overfitting and improving the model's generalization ability. During prediction, the algorithm aggregates the predictions of all the trees to produce a final prediction, making Random Forest robust and accurate in various tasks. It is widely used for classification and regression problems, as it can handle high-dimensional data, large datasets, and noisy data effectively.

Additionally, Random Forest provides valuable insights into feature importance, making it a popular choice for data analysis and predictive modeling tasks.

### 3.2.7 Linear Regression

Linear regression stands as a cornerstone within supervised learning, serving to anticipate continuous outcomes. Its essence lies in depicting the interconnection between a reliant variable, or target, and one or more autonomous variables, or features, through the assembly of a linear equation upon observed data. At its core, the algorithm posits that this relationship can be aptly captured by a straight line, imbuing it with simplicity and interpretability.

Through the process of training, the algorithm discerns coefficients, encompassing slope and intercept, which are honed to minimize the dissonance between predicted and actual values within the training set,

typically employing the method of least squares. Following this phase, the model stands primed to extrapolate predictions onto novel data by way of multiplying input features by their respective coefficients and aggregating them, yielding the anticipated outcome.

Linear regression's pervasive utility spans across disciplines such as economics, finance, and social sciences, underpinning essential functions like sales prognostication, trend delineation, and risk evaluation.

### **3.2.8 Light Gradient Boosting Model**

LightGBM (Light Gradient Boosting Machine) is a powerful gradient boosting framework that has gained popularity in the machine learning community due to its efficiency and scalability. It is based on the gradient boosting decision tree (GBDT) algorithm but introduces novel techniques to improve training speed and accuracy.

LightGBM employs a leaf-wise growth strategy, where it grows the tree node-wise, prioritizing nodes with the maximum loss reduction, resulting in a deeper and potentially unbalanced tree. Additionally, it uses histogram-based algorithms to discretize continuous feature values into discrete bins, reducing memory usage and speeding up the training process.

LightGBM also implements techniques such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to further enhance efficiency while maintaining high predictive performance. These features make LightGBM well-suited for large-scale datasets and real-world applications where speed and accuracy are crucial.

### **3.2.9 Decision Tree**

The decision tree algorithm is a highly adaptable supervised learning method utilized across machine learning tasks encompassing classification and regression. It functions by iteratively segmenting datasets into subsets

based on attributes deemed most informative.

At each node, the algorithm identifies the attribute offering optimal data segregation, with the objective of maximizing homogeneity within each subset. This iterative process persists until predefined termination conditions, such as reaching a specified maximum tree depth, are met. Decision trees present notable simplicity and interpretability, rendering them conducive to visualization and comprehension even among non-specialists.

Nevertheless, they are susceptible to overfitting, particularly evident in instances of deep trees or data containing substantial noise. Despite these limitations, decision trees maintain relevance across diverse domains, including finance, healthcare, and marketing, owing to their capacity to handle diverse data types and intrinsic feature selection capabilities.

### **3.2.10 Naïve Base**

The Naive Bayes algorithm serves as a fundamental probabilistic classifier extensively utilized in machine learning applications such as text classification, spam detection, and recommendation systems. It relies on Bayes' theorem to estimate the probability of a hypothesis given observed evidence. Despite its straightforward nature, Naive Bayes demonstrates considerable efficacy, particularly in scenarios involving high-dimensional datasets.

The "naive" descriptor pertains to its assumption of feature independence, positing that features are conditionally independent given the class label. This assumption simplifies probability calculations, rendering the algorithm computationally efficient, especially for large datasets.

Although the independence assumption may not always hold in real-world contexts, Naive Bayes frequently yields satisfactory results and serves as a benchmark model for evaluating more sophisticated algorithms.

### **3.3 EXPLORING FLASK FRAMEWORK**

Flask stands out as a lightweight and adaptable web framework tailored for Python, engineered to streamline the process of constructing web applications swiftly and effortlessly. Recognized for its straightforwardness, adaptability, and minimalist design ethos, Flask empowers developers to craft web applications with a notably reduced burden of repetitive code.

Fundamental to Flask are its provisions for crucial web development functionalities including URL routing, template rendering, and request management. By adhering to the WSGI (Web Server Gateway Interface) specification, Flask ensures compatibility across a spectrum of web servers and deployment configurations.

Moreover, Flask boasts extensive extensibility, boasting a rich ecosystem of extensions that enable the integration of additional features such as authentication mechanisms, database functionality, and RESTful API support. The inherent simplicity and modular architecture of Flask render it a preferred choice for both novices and seasoned developers seeking to construct web applications efficiently.

## CHAPTER 4: SYSTEM ANALYSIS

After learning enough about my training modules, I was introduced to a task. I had to make views and templates for admin interface. I had to look after whether all the implemented concepts of Data Science were working well or not.

### 4.1 INTRODUCTION

The benefits of this website were to give a user a new feeling of using and learning data science and machine learning. Where he can explore more about this domain and can look after a few amazing projects that works on those concepts.

### 4.2 GOALS

The primary objective of the project is to empower users with the capability to perform analysis on any dataset through interactive interfaces, eliminating the need for them to develop algorithms from the ground up.

### 4.4 FEASIBILITY STUDY

Feasibility study is a process that identifies, describes and evaluates proposed system and selects the best system. During the study, the problem definition is solved and all aspects of the problem to be included in the system are determined. Size of project, cost of benefits is also estimated with greater accuracy. A good feasibility study will show the strengths and defects before the project is planned or budgeted.

#### 4.4.1 Technical Study

The feasibility study evaluates the availability and integration of necessary technology for the proposed system, considering its stakeholders, particularly users. The technical assessment examines whether users possess the requisite expertise to comprehend and utilize the new system effectively.

This evaluation relies on preliminary system requirements design to ascertain the technical and legal viability of the proposal. It involves scrutinizing the hardware and software components to ensure they align with the needs of the proposed system. Furthermore, it examines whether the technology can be developed using existing equipment and possesses the technical capability to accommodate the data requirements of the current system.

#### **4.4.2 Economic Study**

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification. It identifies the financial benefits and cost associated with the development of the system. Economic feasibility is often known as the cost benefit analysis. To carry out an economic feasibility study it is necessary to estimate actual money value against activities needed for implementing the system.

While implementing our system we can ensure that the cost of the prospective new venture will ultimately be profitable to the people. So, we can say it is financially feasible.

#### **4.4.3 Operational Study**

Operational feasibility evaluates the extent to which a proposed system effectively addresses identified problems and leverages opportunities outlined during scope definition. It assesses how well the system meets the requirements identified in the requirement analysis phase of system development. This evaluation centers on the alignment of the proposed project with the existing environment and objectives, particularly concerning the development schedule. Operational feasibility scrutinizes aspects related to human resources, organizational structure, and political considerations.

## CHAPTER 5: SYSTEM DESIGN

The purpose of this document is to give a detailed description of the prerequisite steps for the project. It will illustrate the complete declaration for the development of the system.

### 5.1 EXPLORING DATASET

Understanding the dataset holds paramount importance in constructing a precise predictive model for housing prices. The primary objective during data analysis is to identify patterns, anomalies, and other pertinent factors that can contribute to predicting housing prices accurately. A comprehensive exploration of all variables influencing housing prices, encompassing features like balcony presence, bedroom and bathroom count, is integral to this process. Additionally, categorical variables such as location and availability might also be included in the dataset. This systematic approach facilitates the extraction of invaluable insights from the dataset, paving the way for the development of robust predictive models for house price estimation.

A	B	C	D	E	F	G	H	I	J
Name	Property_Type	Bedrooms	Baths	Balcony	City	Location	Total_Area	Price_per_SQFT	Price
1 DAC Praphi	Flat	3	3 No	Chennai	Kasturbai Nagar, West Tambaram,Chennai	1320	7580	10000000	
2 DRA 90 Degrees	Flat	3	3 No	Chennai	Rose Avenue, Pallikaranai,Chennai	1444	9000	13000000	
4 CasaGrand Vistaaz	Villa	3	3 No	Chennai	Kolapakkam - Vandalur,Chennai	1566	6900	10800000	
5 Vasanthan	Flat	3	3 Yes	Chennai	Maruthi Nagar, Sembakkam,Chennai	1712	6720	11500000	
6 Golden Jubilee Flats	Flat	3	3 Yes	Chennai	Thirumangalam, Anna Nagar West,Chennai	1100	9090	10000000	
7 Prestige Bella Vista	Flat	3	3 Yes	Chennai	Iyyappanthangal,Chennai	1791	7260	13000000	
8 BSR Nerukundam	Independent House	4	3 No	Chennai	Tirumala Nagar, Koyambedu,Chennai	1400	8210	11500000	
9 Pacifica Happiness Towers	Flat	4	4 Yes	Chennai	Padur,Chennai	2232	5600	12500000	
10 Pacifica Arun Villas	Villa	4	4 Yes	Chennai	Padur,Chennai	2117	6610	14000000	
11 Prestige Courtyards by Prestige Estates Projects Ltd.	Flat	3	3 Yes	Chennai	Sholinganallur,Chennai	1689	7820	13200000	
12 Casagrand Arena	Villa	3	3 Yes	Chennai	Oragadam Ambattur,Chennai	1545	8090	12500000	
13 Pacifica Arun Villas by Pacifica Companies	Flat	4	4 Yes	Chennai	Padur,Chennai	2200	6820	15000000	
14 Navina Kushik	Flat	3	3 Yes	Chennai	Ramakrishna Puram, West Mambalam,Chennai	1269	12610	16000000	
15 Sudharshan Homes	Flat	2	3 No	Chennai	Periyar Nagar, Velachery,Chennai	1152	9000	10400000	
16 Krishna Flats	Flat	2	3 No	Chennai	Natesa Nagar, Virungambakkam,Chennai	1365	8420	11500000	
17 Vijay Raja Classic	Villa	3	3 Yes	Chennai	Uniamcheri, Kolapakkam, Kolapakkam - Vandalur,Chennai	1734	6340	11000000	
18 VGN Coasta	Flat	3	3 No	Chennai	Muttukadu, Chennai	1540	8440	13000000	
19 Tulip Apartment	Flat	2	3 No	Chennai	Kamdar Nagar, Nungambakkam,Chennai	1150	11740	13500000	
20 Man Pearl Reside	Flat	3	3 Yes	Chennai	Thiruvengatapuram, Choolaimedu,Chennai	1260	8730	11000000	
21 Jain West Minster	Flat	3	3 Yes	Chennai	Pushpa Colony, Saligramam,Chennai	1385	10830	15000000	
22 Jadhav Jain Atisaya	Flat	2	3 No	Chennai	Trustpuram, Kodambakkam,Chennai	1120	10710	12000000	
23 VGN LA PARISIENNE	Flat	3	3 Yes	Chennai	Nolambur, Ambattur Industrial Estate,Chennai	1309	8400	11000000	
24 Mahalakshmi Apartments	Flat	3	3 No	Chennai	Rajaji Nagar Extension, Madipakkam,Chennai	1149	10880	12500000	
25 Sri Rama Nivas	Flat	3	3 Yes	Chennai	Reddi Koppar, West Mambalam,Chennai	1225	10610	13000000	
26 Appaswamy Arcot Terrace	Flat	2	3 No	Chennai	Somasundara Bharathi Nagar, Vadapalan,Chennai	1231	13000	16000000	
27 Radiance Icon	Flat	2	3 No	Chennai	Nerkundram, Koyambedu,Chennai	1002	11980	12000000	
28 Punya Windermere	Flat	3	4 Yes	Chennai	Pallikaranai, Chennai	2160	6480	14000000	
29 Nakar Navkar Residency by Nakar Homes	Flat	3	3 Yes	Chennai	Korukkupet, Washermanpet,Chennai	1750	7260	12700000	

Figure 5.1.1

A	B	C	D	E	F	G	H	I	J
11174 Nabajgarh Extension, Nabajgarh, New Delhi	Independent House	1	1 No	New Delhi Nabajgarh Extension, Nabajgarh, New Delhi	360	5000	1800000		
11175 Rohini east sector 7, Sector 7 Rohini, New Delhi	Flat	1	1 No	New Delhi Rohini east sector 7, Sector 7 Rohini, New Delhi	323	11140	3600000		
11176 Rani Garden, Geeta Colony, New Delhi	Independent House	3	3 Yes	New Delhi Rani Garden, Geeta Colony, New Delhi	1260	5950	7500000		
11177 New Krishna Nagar, New Krishna Nagar, Krishna Nagar, New De Flat		3	3 No	New Delhi New Krishna Nagar, New Krishna Nagar, Krishna Nagar, New De Flat	1000	7500	7500000		
11178 Mahavir Enclave Part 1, Mahavir Enclave, New Delhi	Independent House	1	3 No	New Delhi Mahavir Enclave Part 1, Mahavir Enclave, New Delhi	900	1000	900000		
11179 Mehrauli, New Delhi	Independent House	1	1 Yes	New Delhi Mehrauli, New Delhi	225	8000	1800000		
11180 Duggal Colony, Khanpur, New Delhi	Independent House	3	3 Yes	New Delhi Duggal Colony, Khanpur, New Delhi	900	11000	9900000		
11181 Mittal properties, Pocket B, Sidhartha Nagar, New Delhi	Flat	2	2 Yes	New Delhi Mittal properties, Pocket B, Sidhartha Nagar, New Delhi	720	8050	5800000		
11182 Capital green, Moti Nagar, New Delhi	Flat	1	1 Yes	New Delhi Capital green, Moti Nagar, New Delhi	375	10670	4000000		
11183 DDA Janta Flats Sector 16 Rohini, Sector 17, Sector 17 Rohini, N Independent House		2	2 No	New Delhi DDA Janta Flats Sector 16 Rohini, Sector 17, Sector 17 Rohir	550	5090	2800000		
11184 rohini sector 15 g2 block, Sector 15C, Sector 15 Rohini, New De Flat		1	2 No	New Delhi rohini sector 15 g2 block, Sector 15C, Sector 15 Rohini, New De Flat	500	10400	5200000		
11185 Sri Jagannath Apartments, Deoli Gaon Nai Basti, Khanpur, New Flat		4	3 Yes	New Delhi Sri Jagannath Apartments, Deoli Gaon Nai Basti, Khanpur, N	1000	8000	8000000		
11186 Sharda Puri, Ramesh Nagar, New Delhi	Flat	1	2 No	New Delhi Sharda Puri, Ramesh Nagar, New Delhi	446	8970	4000000		
11187 Narela Mandi, Narela, New Delhi	Independent House	2	2 No	New Delhi Narela Mandi, Narela, New Delhi	700	3430	2400000		
11188 Block 22, Tilak Nagar, New Delhi	Independent House	5	3 Yes	New Delhi Block 22, Tilak Nagar, New Delhi	1530	3590	5500000		
11189 Sharda Puri, Ramesh Nagar, New Delhi	Flat	2	2 No	New Delhi Sharda Puri, Ramesh Nagar, New Delhi	446	8970	4000000		
11190 Bharat Vihar, New Delhi	Independent House	3	1 No	New Delhi Bharat Vihar, New Delhi	333	6010	2000000		
11191 C Block, Preet Vihar, New Delhi	Flat	3	2 Yes	New Delhi C Block, Preet Vihar, New Delhi	600	5000	3000000		
11192 Block D, Preet Vihar, New Delhi	Independent House	4	3 Yes	New Delhi Block D, Preet Vihar, New Delhi	1000	4000	4000000		
11193 Hari vihar, Sector 16 Dwarka, New Delhi	Independent House	2	2 No	New Delhi Hari vihar, Sector 16 Dwarka, New Delhi	450	10000	4500000		
11194 Sector 31 Rohini, New Delhi	Independent House	3	3 No	New Delhi Sector 31 Rohini, New Delhi	963	3630	3500000		
11195 Prem Nagar III, Kirari Suleman Nagar, New Delhi	Independent House	2	2 No	New Delhi Prem Nagar III, Kirari Suleman Nagar, New Delhi	465	6020	2800000		
11196 Bulder Floor, Jaipur, New Delhi	Flat	1	1 No	New Delhi Bulder floor, Jaipur, New Delhi	200	4500	900000		
11197 Raghu Nagar, Dabri, New Delhi	Independent House	7	2 Yes	New Delhi Raghu Nagar, Dabri, New Delhi	405	13580	5500000		
11198 Krishna Park Extension, Tilak Nagar, New Delhi	Flat	2	3 Yes	New Delhi Krishna Park Extension, Tilak Nagar, New Delhi	1053	3800	4000000		
11199 Rawta, Jaffarpur Kalan, New Delhi	Independent House	1	2 Yes	New Delhi Rawta, Jaffarpur Kalan, New Delhi	472	2970	1400000		
11200 Rani Garden, Geeta Colony, New Delhi	Flat	1	1 Yes	New Delhi Rani Garden, Geeta Colony, New Delhi	378	7940	3000000		
11201 Lig flat rohini, Sector 16E, Sector 16 Rohini, New Delhi	Flat	3	2 Yes	New Delhi Lig flat rohini, Sector 16E, Sector 16 Rohini, New Delhi	700	8570	6000000		
11202 Sector 3B, Sector 3 Rohini, New Delhi	Independent House	1	1 Yes	New Delhi Sector 3B, Sector 3 Rohini, New Delhi	344	15990	5500000		

Figure 5.1.2

## 5.2 DATA CLEANING

Data cleaning serves as the fundamental cornerstone of any data-centric undertaking, pivotal in ensuring precision, reliability, and credibility in subsequent analyses. This process is instrumental in upholding dataset consistency and completeness through meticulous handling of missing values, standardization of formats, and error detection. By addressing discrepancies and rectifying inconsistencies, data quality is significantly enhanced, rendering it more conducive to insightful modeling and analysis. Moreover, the elimination of outliers and redundant data not only enhances the efficacy and performance of predictive models but also streamlines the analytical workflow. Ultimately, the rigorous practice of data cleaning facilitates informed decision-making and facilitates the derivation of actionable insights by averting erroneous conclusions and instilling confidence in the accuracy of outcomes.

```

1 import numpy as np
2 import pandas as pd
3
4 import warnings
5 from Tools.scripts.dutree import display
6
7 # This code is to see full detailed output in pycharm
8 pd.set_option('display.max_rows', None)
9 pd.set_option('display.max_columns', None)
10 pd.set_option('display.width', None)
11 pd.set_option('display.max_colwidth', 0) # This ensures that contents of columns are not truncated
12
13
14 warnings.filterwarnings('ignore')
15
16 df = pd.read_csv(r'C:\data science Internship\pythonProject1\Real Estate Data V21.csv')
17 print(df.head(10))
18 df2 = df
19 print(df2.shape)
20
21 # capturing BHK values in different columns
22 BHK = []
23 for i in range(0, 14520):
24     BHK.append(df2['Property Title'][i][:-2])
25
26 # Assigning a list to a new column called "Bedrooms"
27 df2['Bedrooms'] = BHK
28
29 # Replacing the errors, misspelt, or unwanted symbols in the "Bedrooms" column
30 df2['Bedrooms'] = df2['Bedrooms'].str.replace(".", "")
31 df2['Bedrooms'] = df2['Bedrooms'].str.replace(":", "")
32 df2['Bedrooms'] = df2['Bedrooms'].str.replace("St", "")
33 df2['Bedrooms'] = df2['Bedrooms'].str.replace("", "")

```

Figure 5.2.1

```

34 df3 = df2
35 print(df3['Bedrooms'].value_counts())
36
37 # Finding the index number of the column which has empty values,
38 # print(df3[df3['Bedrooms'] == ""].index)
39
40 # removing those columns which are too small
41 bad_df = df3.index.isin([1186, 3497, 4003, 4005, 5889, 7527, 7594, 7993, 9148, 12355, 14487])
42 df3 = df3[~bad_df]
43
44 # We check again
45 df3['Bedrooms'].value_counts()
46
47 # We can see repeated digits because of the space it has in front of it, we now remove that space
48
49 df3['Bedrooms'] = df3['Bedrooms'].str.replace("Sh", "1")
50 df3['Bedrooms'] = df3['Bedrooms'].str.replace("1 ", "1")
51 df3['Bedrooms'] = df3['Bedrooms'].str.replace("2 ", "2")
52 df3['Bedrooms'] = df3['Bedrooms'].str.replace("3 ", "3")
53 df3['Bedrooms'] = df3['Bedrooms'].str.replace("4 ", "4")
54 df3['Bedrooms'] = df3['Bedrooms'].str.replace("5 ", "5")
55 df3['Bedrooms'] = df3['Bedrooms'].str.replace("6 ", "6")
56 df3['Bedrooms'] = df3['Bedrooms'].str.replace("7 ", "7")
57 df3['Bedrooms'] = df3['Bedrooms'].str.replace("8 ", "8")
58 df3['Bedrooms'] = df3['Bedrooms'].str.replace("9 ", "9")
59 df3['Bedrooms'] = df3['Bedrooms'].str.replace("0 ", "0")
60 df3['Bedrooms'] = df3['Bedrooms'].str.replace("Sh", "3")
61
62 # Now, the values are good and unique
63 df3['Bedrooms'].value_counts()
64
65 # Converting the column to INT type, so that we can use it for a conditional formatting.
66 df3['Bedrooms'] = df3['Bedrooms'].astype(int)

```

Figure 5.2.2

```

67 # Using conditional Formatting to Assign BHK and RK for houses with 1 it is RK rest is BHK
68 df3['Bedrooms1'] = np.where(df3['Bedrooms'] > 1, df3['Bedrooms'].astype(str) + 'BHk',
69                             df3['Bedrooms'].astype(str) + 'RK')
70
71 print(df3.head(4))
72
73 # finding how many rows are same in both columns
74 print(df3["Name"].isin(df3["Location"]).value_counts())
75
76 # Dividing the dataset based on the criteria above
77 df_split1 = df3[-df3["Name"].isin(df3["Location"])]
78
79 df_split2 = df3[df3["Name"].isin(df3["Location"])]
80
81 # Checking which column to include in the new and modified dataset with column in order.
82 print(df_split2.columns)
83
84 df_split_Edit = df_split2[['Name', 'Bedrooms1', 'Baths', 'Balcony', 'Price', 'Location', 'Total_Area',
85                           'Price_per_SQFT', 'Property Title', 'Description']]
86
87 # Renaming the column to match with the 2nd splitted column
88 # df_split_Edit = df_split.Edit.rename(columns={'Name': 'Name1'})
89
90 # Modifying 2nd split dataframe with columns we require
91 df_split1_edit = df_split1[['Name', 'Bedrooms1', 'Baths', 'Balcony', 'Price', 'Location', 'Total_Area',
92                           'Price_per_SQFT', 'Property Title', 'Description']]
93
94 print(df_split1_edit.head(4))
95
96 # Concatenating the above splitted dataset
97 again_new_data = pd.concat([df_split1_edit, df_split_Edit], ignore_index=True)
98
99 print(again_new_data.shape)

```

Figure 5.2.3

```

100 print(again_new_data.head(4))
101
102 # Let's find how many properties which has Villa in their title
103 print("No of Villa Properties are: {}".format(
104     len(again_new_data[again_new_data['Property Title'].str.contains('Villa')]))
105 print(again_new_data[again_new_data['Property Title'].str.contains('Villa')][:4])
106
107 # Now, we use np.where to replace values based on condition where Property is Villa
108 again_new_data['Property_Type'] = np.where(again_new_data['Property Title'].str.contains('Villa') == True, 'Villa',
109                                         'To Add')
110
111 # This is the same condition as above but for Independent House, and this time we return the
112 # values which is already present in the "Property_Type", so previous values don't get replaced.
113 again_new_data['Property_Type'] = np.where(again_new_data['Property Title'].str.contains('Independent House') == True,
114                                         'Independent House',
115                                         again_new_data['Property_Type'])
116
117 # This is the same condition but for Flats
118 again_new_data['Property_Type'] = np.where(again_new_data['Property Title'].str.contains('Flat') == True, 'Flat',
119                                         again_new_data['Property_Type'])
120
121 # Finding whether data selected is accurate
122 print(again_new_data['Price'][1].replace(',', '').replace(' Cr', '')) # Output: 10000000
123
124 # Making New split
125 new_data_split = again_new_data
126
127 # Assigning a new column containing only crores
128 new_data_split['Price'] = new_data_split['Price'].map(lambda x: x.lstrip('₹').rstrip(' Cr'))
129
130 # Splitting that dataset into Crores and Lakhs to solve them two different conditions
131 crores = new_data_split[new_data_split['Price'].str.contains(' L')]


```

Figure 5.2.4

```

134     lakhs = new_data_split[new_data_split['Price'].str.contains(' L')]
135     croress['Price'] = croress['Price'].map(lambda x: x.rstrip('k'))
136
137     croress['Price'] = croress['Price'].astype(float)
138
139     croress['Price'] = croress['Price'] * 10000000
140
141     print(croress.head(2))
142
143     lakhs['Price'] = lakhs['Price'].map(lambda x: x.rstrip(' L'))
144
145     lakhs['Price'] = lakhs['Price'].map(lambda x: x.rstrip(' Lcs'))
146
147     lakhs['Price'] = lakhs['Price'].astype(float)
148
149     lakhs['Price'] = lakhs['Price'] * 100000
150
151     print(lakhs.head())
152
153     again_new_data_mega = pd.concat([crosess, lakhs], ignore_index=True)
154     print(again_new_data_mega.head())
155
156     # As before, assigning an empty list to capture the City names, along with replaces values with spaces
157     lococity = []
158     for i in range(0, 14517):
159         lococity.append(
160             again_new_data_mega['Location'][i].split('.')[1].replace(' Bangalore', 'Bangalore').replace(' Pune',
161             'Pune').replace(
162                 ' Chennai', 'Chennai').replace(' New Delhi', 'New Delhi').replace(' Hyderabad', 'Hyderabad').replace(
163                 ' Mumbai', 'Mumbai').replace(' Kolkata', 'Kolkata'))
164
165     # Assigning the list to the new column City
166     again_new_data_mega['City'] = lococity

```

Figure 5.2.5

```

167     # Checking the value counts for presence of any unwanted data
168     print(again_new_data_mega['City'].value_counts())
169
170     # Checking columns to select required columns
171     print(again_new_data_mega.columns)
172
173     # Modifying Dataset with required Columns
174     again_new_data_mega1 = again_new_data_mega[
175         ['Name', 'Property_Type', 'Bedrooms', 'Baths', 'Balcony', 'City', 'Location',
176         'Total_Area', 'Price_per_SQFT', 'Price', 'Property Title', 'Description']]
177     print(again_new_data_mega1.head())
178
179     # Everytime before starting, we check value for iteration, which can also be used as range(0, len())
180     print("No of rows to iterate:{}".format(len(again_new_data_mega)))
181
182     # Assigning empty column to capture the sentences as a list
183     descrip = []
184     for i in range(0, 14517):
185         descrip.append(again_new_data_mega1['Description'][i].split('.')[1:-4])
186
187     descrip1 = []
188     for i in range(0, 14517):
189         descrip1.append('. '.join(descrip[i]))
190
191     print("Difference can be seen below")
192     print("Before")
193     print(descrip[0])
194     print(' ' * 50)
195     print(' ' * 50)
196     print(' ' * 50)
197     print("After")
198     print(descrip[0])

```

Figure 5.2.6

```

201 # Assigning list to the column
202 again_new_data_mega1['Description1'] = descriptiv
203
204 # Dropping not needed columns
205 again_new_data_mega2 = again_new_data_mega1.drop('Description', axis=1)
206
207 # Finalized and clean Dataset
208 print(again_new_data_mega2.head(4))
209
210 again_new_data_mega2.to_csv('again_new_data_mega2.csv', index=False)
211 print(again_new_data_mega2.head())
212
213 # to check Null values
214 print(again_new_data_mega2.isna().sum())

```

Figure 5.2.7

The screenshot shows the PyCharm IDE interface. On the left, there's a project tree with a 'Run' configuration named 'Data Cleaning'. The main area shows the code for 'Data Cleaning.py'. Below the code, the terminal window displays the output of the script's execution. The output shows a list of real estate properties with columns: Name, Property Title, Price, Location, and Total.

Name	Property Title	Price	Location	Total
0 Casagrand ECR 14	4 BHK Flat for sale in Kanathur Reddikuppam, Chennai	₹1.99 Cr	Kanathur Reddikuppam, Chennai	2583
1 Ramanathan Nagar, Pozhichalur,Chennai	10 BHK Independent House for sale in Pozhichalur, Chennai	₹2.25 Cr	Ramanathan Nagar, Pozhichalur,Chennai	7000
2 DAC Prarthi	3 BHK Flat for sale in West Tambaram, Chennai	₹1.0 Cr	KasthuriBai Nagar, West Tambaram,Chennai	1320
3 Naveenilaya,Chepauk, Triplicane,Chennai	7 BHK Independent House for sale in Triplicane, Chennai	₹3.35 Cr	Naveenilaya,Chepauk, Triplicane,Chennai	4250
4 VBA Spring Field Phase 1	2 BHK Flat for sale in Avadi, Chennai	₹46.0 L	Avadi, Chennai	968
5 KG Earth Homes	2 BHK Flat for sale in Siruseri, Chennai	₹40.0 L	Siruseri, Chennai	940
6 THIRAN FLATS ,Gowrivakkam, Sembakkam,Chennai	2 BHK Flat for sale in Sembakkam, Chennai	₹60.0 L	THIRAN FLATS ,Gowrivakkam, Sembakkam,Chennai	880
7 TK Jasmine Grove	3 BHK Independent House for sale in Mahindra World City, Chennai	₹72.35 L	Mahindra World City, Chennai	1700
8 Avenue	2 BHK Flat For sale in West Tambaram, Chennai	₹62.0 L	Brindavan Colony, West Tambaram,Chennai	840
9 Guru Kothai Apartments	1 BHK Flat for sale in Chromepet, Chennai	₹30.0 L	New Colony, Chromepet,Chennai	535
(14528, 9)				
Bedrooms				
2 5691				
1 3278				
3 3132				
4 990				
5 438				
6 296				
10 200				
7 143				
8 132				
9 78				
5 44				
2 38				
1 30				
5 24				
4 11				
4 10				
9 4				

Figure 5.2.8

```

0 Casagrand ECR 14          4 BHK Flat for sale in Kanathur Reddikuppam, Chennai    >1.99 Cr  Kanathur Reddikuppam, Chennai      2583    7700.0
1 Ramanathan Nagar, Pozhichalur,Chennai  10 BHK Independent House for sale in Pozhichalur, Chennai >2.25 Cr  Ramanathan Nagar, Pozhichalur,Chennai      7000    3210.0
2 DAC Prethi                 3 BHK Flat for sale in West Tambaram, Chennai     >1.0 Cr   Kasthuribai Nagar, West Tambaram,Chennai      1320    7580.0
3 Naveenlaya,Chepauk, Triplicane,Chennai  7 BHK Independent House for sale in Triplicane, Chennai >3.33 Cr  Naveenlaya,Chepauk, Triplicane,Chennai      4250    7840.0
Name: count, dtype: int64
Index(['Name', 'Property Title', 'Price', 'Location', 'Total_Area',
       'Price_per_SQFT', 'Description', 'Baths', 'Balcony',
       'Bedrooms1'],
      dtype='object')
   Name Bedrooms1  Baths  Balcony  Price           Location  Total_Area  Price_per_SQFT
0  Casagrand ECR 14        4BHK    4    Yes  >1.99 Cr  Kanathur Reddikuppam, Chennai      2583    7700.0
1  DAC Prethi              3BHK    3    No   >1.0 Cr   Kasthuribai Nagar, West Tambaram,Chennai      1320    7580.0
2  VGN Spring Field Phase 1 2BHK    3    Yes  >48.0 L   Avadi, Chennai      960    5000.0
3  KG Earth Homes          2BHK    3    No  >40.0 L  Siruseri, Chennai      940    4250.0
No of Villa Properties are: 737
   Name Bedrooms1  Baths  Balcony  Price           Location  Total_Area  Price_per_SQFT
12 CasaGrand Vistaaz        3BHK    3    No  >1.08 Cr  Kolapakkam - Vandulur, Chennai      1566    6900.0
20 Jones Dawn Villas by Jones Foundations 3BHK    3    Yes  >87.0 L  Ponnai, Chennai      1782    4680.0
61 Adityaram Nagar Phase 5 4BHK    5    Yes  >2.6 Cr  Adityaram Nagar, Panchayat,Chennai      3500    7430.0
64 Pacifica Aurum Villas   4BHK    4    Yes  >1.4 Cr   Padur, Chennai      2117    6610.0
Prop
   Name Bedrooms1  Baths  Balcony  Price           Location  Total_Area  Price_per_SQFT
0  Casagrand ECR 14        4BHK    4    Yes  >1.99 Cr  Kanathur Reddikuppam, Chennai      2583    7700.0
1  DAC Prethi              3BHK    3    No   >1.0 Cr   Kasthuribai Nagar, West Tambaram,Chennai      1320    7580.0
2  VGN Spring Field Phase 1 2BHK    3    Yes  >48.0 L   Avadi, Chennai      960    5000.0
3  KG Earth Homes          2BHK    3    No  >40.0 L  Siruseri, Chennai      940    4250.0
2 BHK Flat for sale in Avadi, Chennai
3 BHK Flat for sale in West Tambaram, Chen
4 BHK Flat for sale in Kanathur Reddikuppam
4 BHK Flat for sale in Siruseri, Chennai
3 BHK Villa for sale in Kolapakkam
3 BHK Villa for sale in Ponnai, Ch
4 BHK Villa for sale in Panayur, Ch
4 BHK Villa for sale in Padur, Ch

```

Figure 5.2.9

```

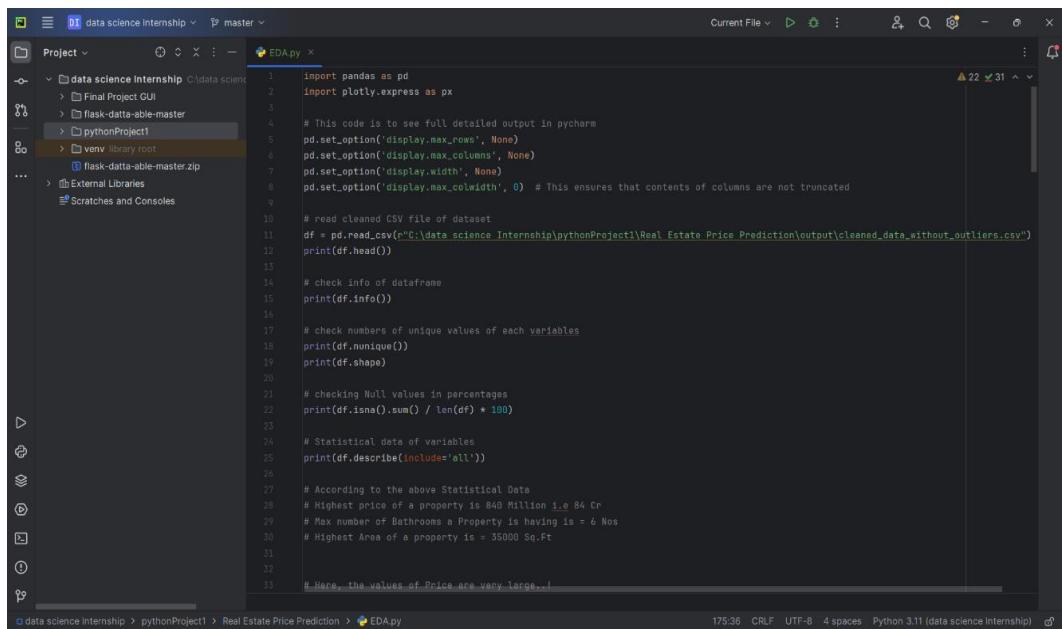
['Best 4 BHK Apartment for modern-day lifestyle is now available for sale', ' No brokerage involved, Posted by Owner', ' Grab this 4 BHK property for sale in one of Chennai's t
-----
After
Best 4 BHK Apartment for modern-day lifestyle is now available for sale. No brokerage involved, Posted by Owner. Grab this 4 BHK property for sale in one of Chennai's top locat
   Name Property_Type Bedrooms1  Baths  Balcony  City           Location  Total_Area  Price_per_SQFT  Price
0  Casagrand ECR 14  Flat        4BHK    4    Yes  Chennai  Kanathur Reddikuppam, Chennai      2583    7700.0  19900000.0  4 BHK Flat for sale in Kai
1  DAC Prethi        Flat        3BHK    3    No   Chennai  Kasthuribai Nagar, West Tambaram,Chennai      1320    7580.0  10800000.0  3 BHK Flat for sale in We
2  DRA 90 Degrees    Flat        3BHK    3    No   Chennai  Rose Avenue, Pallikaranai,Chennai      1444    9000.0  13000000.0  3 BHK Flat for sale in Pa
3  Ramcons Dominion Flat        3BHK    3    Yes  Chennai  Tirumurthy Nagar, Nungambakkam,Chennai      1658    15680.0  26000000.0  3 BHK Flat for sale in Nu
   Name          0
Property_Type  0
Bedrooms1      0
Baths          0
Balcony         0
City            0
Location        0
Total_Area      0
Price_per_SQFT  0
Price          0
Property Title  0
Description1    0
dtype: int64
   Name          object

```

Figure 5.2.10

## 5.3 EXPLORATORY DATA ANALYSIS (EDA)

A vital first step in the data analysis process is exploratory data analysis (EDA), which seeks to comprehend the underlying patterns, trends, and correlations within a dataset. It entails analyzing the data both statistically and visually in order to draw conclusions and develop theories for additional research. EDA is crucial because it gives analysts a thorough picture of the data, enabling them to spot outliers, missing numbers, and other problems with data quality early on. Analysts can decide on data cleaning, feature engineering, and modeling methodologies by developing a deeper grasp of the dataset's properties. Typically, exploratory statistical analyses like correlation matrices and clustering algorithms are used in conjunction with descriptive statistics and data visualization tools like histograms, scatter plots, and box plots to undertake exploratory data analysis (EDA).



```

import pandas as pd
import plotly.express as px

# This code is to see full detailed output in pycharm
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth', 0) # This ensures that contents of columns are not truncated

# read cleaned CSV file of dataset
df = pd.read_csv("C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")
print(df.head())

# check info of dataframe
print(df.info())

# check numbers of unique values of each variables
print(df.unique())
print(df.shape)

# checking Null values in percentages
print(df.isna().sum() / len(df) * 100)

# Statistical data of variables
print(df.describe(include='all'))

# According to the above Statistical Data
# Highest price of a property is 840 Million i.e 84 Cr
# Max number of Bathrooms a Property is having is = 6 Nos
# Highest Area of a property is = 35000 Sq.Ft
# Here, the values of Price are very large...

```

Figure 5.3.1

```

# Reducing Scientific Values of price by creating new Variables like
# Calculate 'Price in Million' column
df['Price in Million'] = df['Price'] / 1000000

# Define the position where you want to insert the 'Price in Million' column
insert_index = df.columns.get_loc('Price') + 1

# Insert the 'Price in Million' column at the desired position
df.insert(insert_index, column='Price in Million', df.pop('Price in Million'))

# Display the DataFrame to verify the column order
print(df.head())

# Lets Check out the Records for each Max Values
# Checking Property with Highest Pricing Records
print("\n\nProperty with Highest Pricing Records")
print(df[df['Price in Million'] == df['Price in Million'].max()])

# Highest Rated Property as per Dataset Price
# Property Name : Yesvantpura
# Property Type : Independent House
# Property State : Bangalore
# Property Location : Kolar
# BHK : 1.5
# Property Selling Price : 84 Crore
# Property Price as per Sq.ft : 84 lakh (Not Correct)
# Balcony : No

# Property with Highest Total Area
print("\n\nProperty with Highest Total Area")
print(df[df['Total_Area'] == df['Total_Area'].max()])

# Highest Rated Property as per Dataset Total Area

```

Figure 5.3.2

```

# Property Name : Veer Sandra
# Property Type : Independent House
# Total Area : 35000 Sq.ft
# Property State : Bangalore
# Property Location : Electronic City Phase II
# BHK : 5+
# Property Selling Price : 90 Lakh
# Property Price as per Sq.ft : 899 Crore (Not Correct may be the Sq.ft Price is not correct)
# Balcony : No

# Property with Highest Nos of Bathrooms
a = df['Baths'] == df['Baths'].max()
print("\nProperties with highest Nos of Bath: ", a.shape)
print("\n Name of all those Properties which have highest numbers of bathrooms:\n ", a['Name'].unique())

# 161 Property Has the Highest nos of bathrooms i.e 6 Nos
# Nos of Properties With Highest Nos of BHK (Bedroom-Hall-Kitchen)
# We will need to Separate numbers of BHK From BHK column

# df['Bedrooms1'].unique()
# sp = df['Bedrooms1'].apply(lambda x: x.split(' '))
# df['No of BHK'] = df['Bedrooms1'].apply(lambda x: x.split('')[0])
# print(df['No of BHK'].unique())

import re
# Define a function to extract the numeric part from strings
# usage new
def extract_numeric(s):
    match = re.search(pattern=r'\d+', s) # Find the first sequence of digits in the string
    if match:
        return int(match.group()) # Convert the matched digits to integer

```

Figure 5.3.3

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** data science Internship > master
- Project Tree:** data science Internship (C:\data science) contains Final Project GUI, flask-database-master, pythonProject1 (selected), venv\library host, flask-database-master.zip, External Libraries, and Scratches and Consoles.
- Code Cell:** The current cell, EDA.py, contains Python code for data extraction and analysis. It includes functions for extracting numeric values from strings, handling 'No of BHK' columns, and creating a bar chart for city-wise data counts.

```
# Define a function to extract the numeric part from strings
# Usage new
def extract_numeric(s):
    match = re.search(pattern=r'\d+', s) # Find the first sequence of digits in the string
    if match:
        return int(match.group()) # Convert the matched digits to integer
    else:
        return None # Return None if no digits are found

# Apply the function to extract numeric values from 'Bedrooms1' column
df['No of BHK'] = df['Bedrooms1'].apply(extract_numeric)

# Extract numeric part from 'BHk' column using regular expressions
df['No of BHK'] = df['Bedrooms1'].apply(lambda x: re.findall(pattern=r'\d+', x)[0] if isinstance(x, str) else x)

# printing Unique Numbers of bedrooms
print(df['No of BHK'].unique())

df['No of BHK'] = df['No of BHK'].str.replace('+', '')
df['No of BHK'] = df['No of BHK'].astype('Float')
print(df['No of BHK'].dtype)

s = df[df['No of BHK'] == df['No of BHK'].max()]
g = s
print(g)

n = df.groupby('City')['No of BHK'].agg(['max'])
nn = df.merge(n, how='inner', left_on=['No of BHK', 'City'], right_on=['max', 'City'])
dd = nn[['City', 'No of BHK']].value_counts()
rr = dd.unstack()
fig = px.bar(rr)
fig.show()
```

- Bottom Status Bar:** data science Internship > pythonProject1 > Real Estate Price Prediction > EDA.py, 93:1 CRLF UTF-8 4 spaces Python 3.11 (data science Internship)

Figure 5.3.4

```
data science Internship > pythonProject1
Project ▾ EDA.py x
127 # Cheapest Property with highest no of BHK as per Dataset
128 print(g['Price in Million'].min())
129 # Cheapest Property with highest no of BHK as per Dataset
130 # Property Name : Venus county,Haragadde, Bangalore
131 # Property Type : Independent House
132 # Total Area : 1988 Sq.ft
133 # Property State : Bangalore
134 # Property Location : Venus county,Haragadde, Bangalore
135 # BHK : 10
136 # Property Selling Price : 5 Lakh
137 # Property Price as per Sq.ft : 252000
138 # Balcony : Yes
139 # Bathrooms : 3 Nos
140
141
142
143 import plotly.graph_objects as go
144
145 # Define the labels and values
146 labels = ['Yes', 'No']
147 values = [8574, 5943]
148
149 # Create the pie chart figure
150 fig = go.Figure(data=[go.Pie(labels=labels, values=values)])
151
152 # Add a title to the pie chart
153 fig.update_layout(title_text='Balcony - Yes/No Distribution')
154
155 # Show the pie chart
156 fig.show()
157
158 # df.rename(columns = {'State':'City'},inplace = True)
159 u = df.groupby('City')['Price in Million'].agg(['mean']).reset_index()
```

Figure 5.3.5

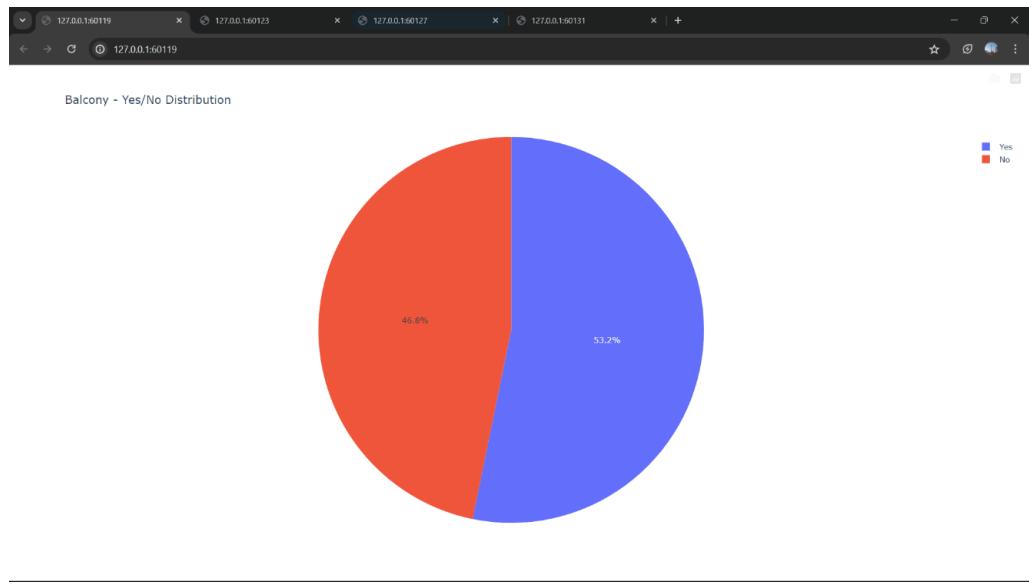


Figure 5.3.6

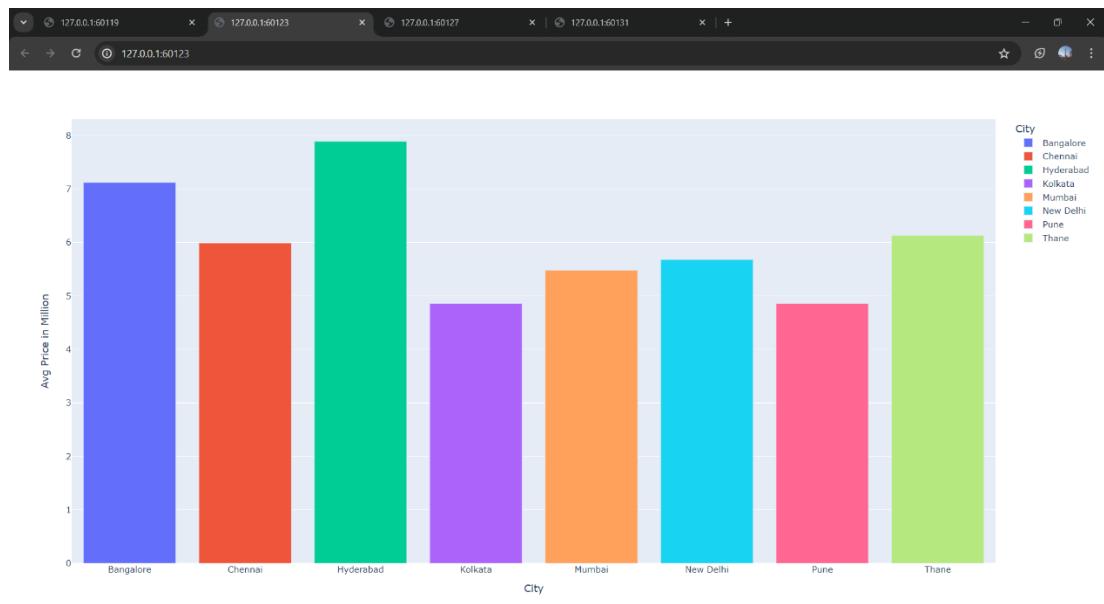


Figure 5.3.7

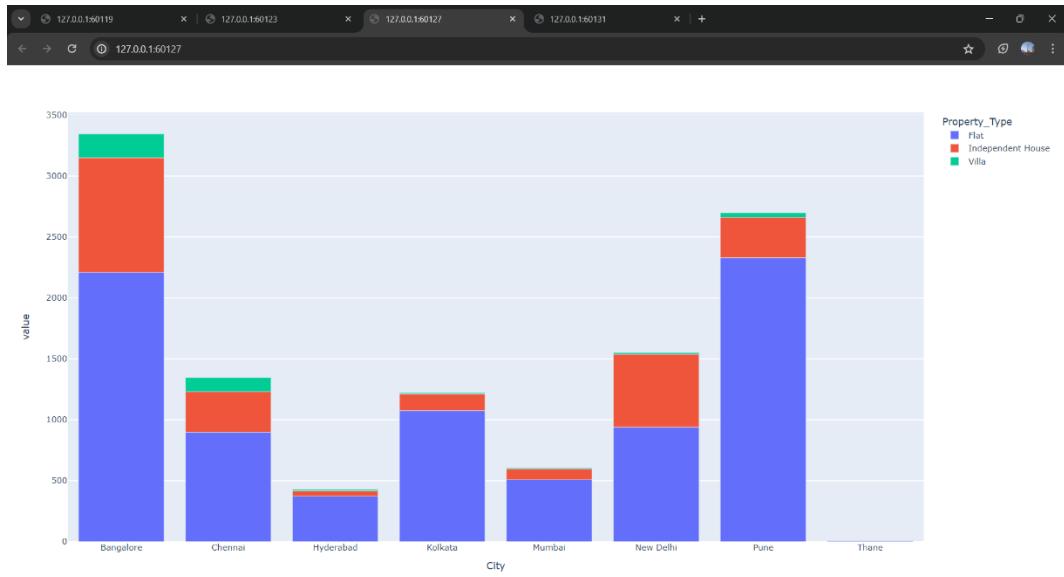


Figure 5.3.8

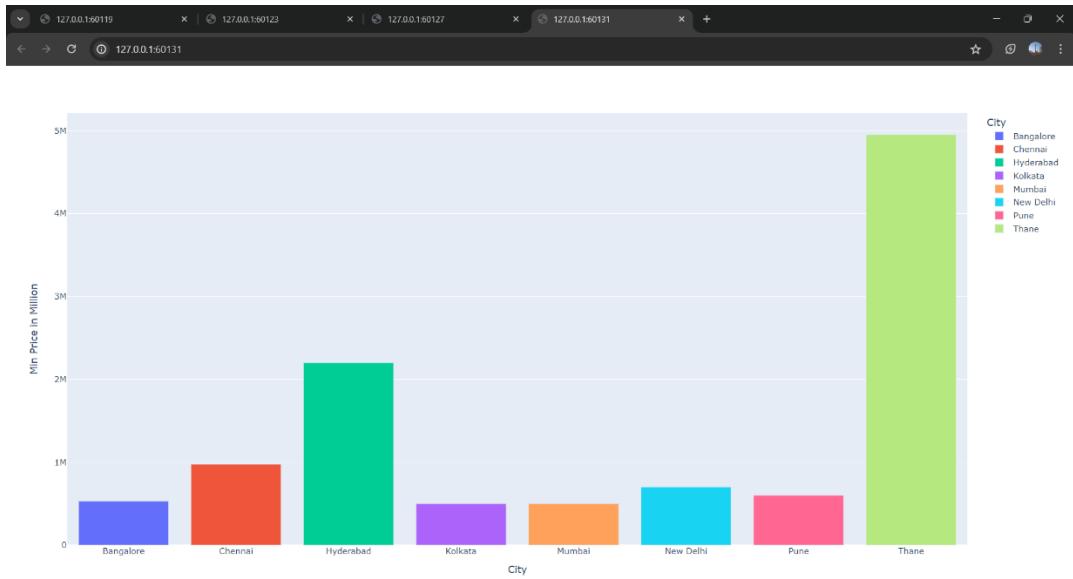


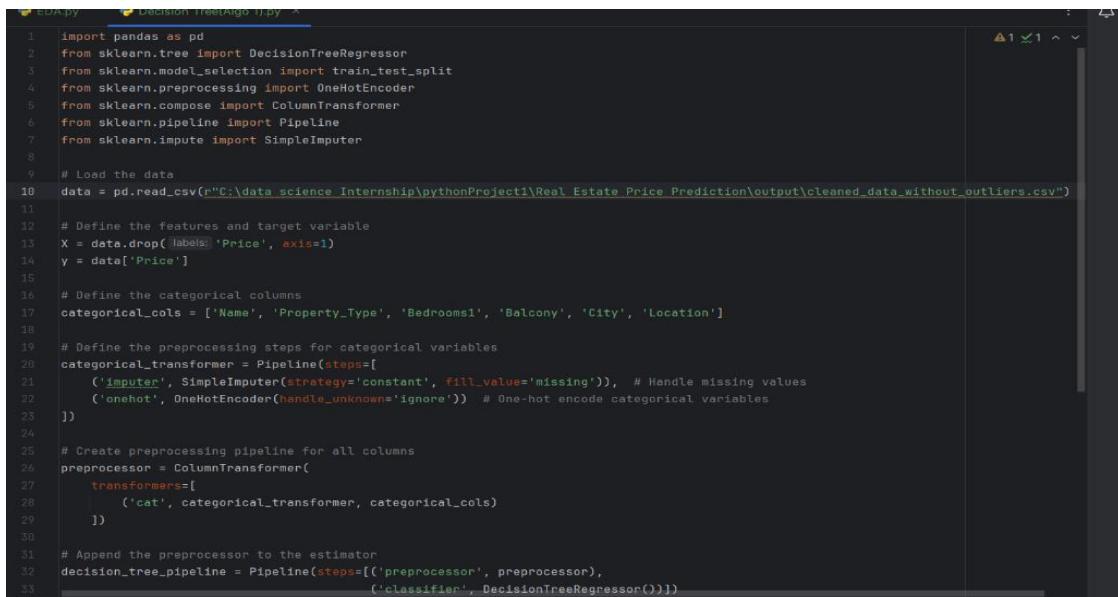
Figure 5.3.9

In any data analysis or machine learning endeavor, the selection of algorithms assumes paramount importance as it fundamentally shapes the success trajectory and modeling approach of the project. Leveraging my dataset, I conducted an array of algorithms to forecast pricing and subsequently assessed the accuracy post model training.

## 5.4 ALGORITHMS

An approach that typically uses recursive operations to solve a mathematical problem in a limited number of steps

### 5.4.1 Decision Tree

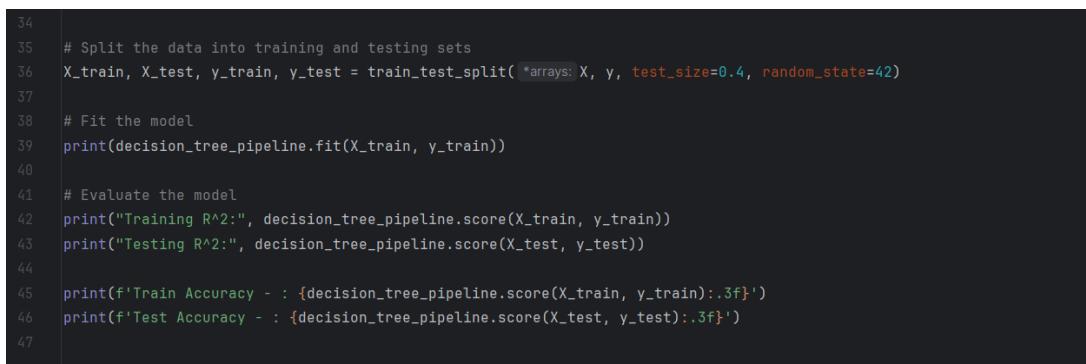


```

1 import pandas as pd
2 from sklearn.tree import DecisionTreeRegressor
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import OneHotEncoder
5 from sklearn.compose import ColumnTransformer
6 from sklearn.pipeline import Pipeline
7 from sklearn.impute import SimpleImputer
8
9 # Load the data
10 data = pd.read_csv(r"C:\data_science_Internship\pythonProject1\Real_Estate_Price_Prediction\output\cleaned_data_without_outliers.csv")
11
12 # Define the features and target variable
13 X = data.drop(['Price'], axis=1)
14 y = data['Price']
15
16 # Define the categorical columns
17 categorical_cols = ['Name', 'Property_Type', 'Bedrooms1', 'Balcony', 'City', 'Location']
18
19 # Define the preprocessing steps for categorical variables
20 categorical_transformer = Pipeline(steps=[
21     ('imputer', SimpleImputer(strategy='constant', fill_value='missing')), # Handle missing values
22     ('onehot', OneHotEncoder(handle_unknown='ignore')) # One-hot encode categorical variables
23 ])
24
25 # Create preprocessing pipeline for all columns
26 preprocessor = ColumnTransformer(
27     transformers=[
28         ('cat', categorical_transformer, categorical_cols)
29     ])
30
31 # Append the preprocessor to the estimator
32 decision_tree_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
33                                         ('classifier', DecisionTreeRegressor())])

```

Figure 5.4.1.1



```

34
35 # Split the data into training and testing sets
36 X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.4, random_state=42)
37
38 # Fit the model
39 print(decision_tree_pipeline.fit(X_train, y_train))
40
41 # Evaluate the model
42 print("Training R^2:", decision_tree_pipeline.score(X_train, y_train))
43 print("Testing R^2:", decision_tree_pipeline.score(X_test, y_test))
44
45 print(f'Train Accuracy - : {decision_tree_pipeline.score(X_train, y_train):.3f}')
46 print(f'Test Accuracy - : {decision_tree_pipeline.score(X_test, y_test):.3f}')
47

```

Figure 5.4.1.2

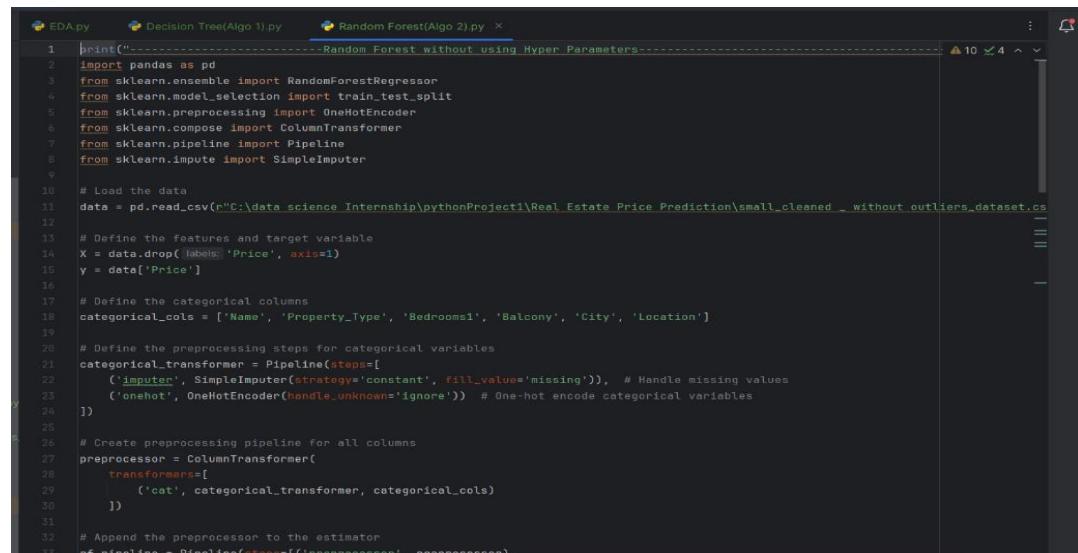
```

Training R^2: 0.9904563145340908
Testing R^2: 0.22782508762903408
Train Accuracy - : 0.990
Test Accuracy - : 0.228

```

Figure 5.4.1.3

## 5.4.2 Random Forest

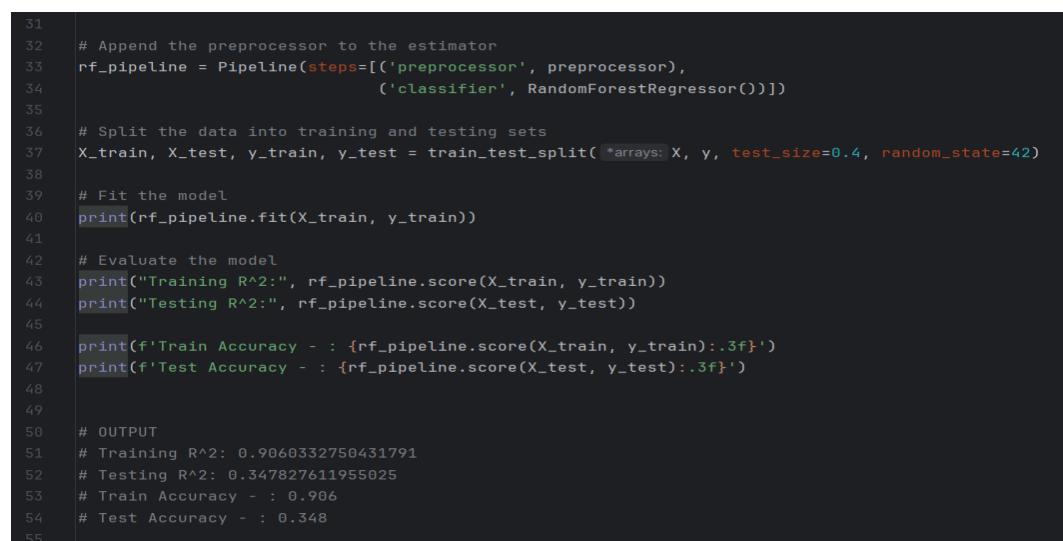


```

1 print("-----Random Forest without using Hyper Parameters-----")
2 import pandas as pd
3 from sklearn.ensemble import RandomForestRegressor
4 from sklearn.model_selection import train_test_split
5 from sklearn.preprocessing import OneHotEncoder
6 from sklearn.compose import ColumnTransformer
7 from sklearn.pipeline import Pipeline
8 from sklearn.impute import SimpleImputer
9
10 # Load the data
11 data = pd.read_csv(r"C:\data science Internship\pythonProject\Real Estate Price Prediction\small_cleaned _without_outliers_dataset.csv")
12
13 # Define the features and target variable
14 X = data.drop(['Price'], axis=1)
15 y = data['Price']
16
17 # Define the categorical columns
18 categorical_cols = ['Name', 'Property_Type', 'Bedrooms1', 'Balcony', 'City', 'Location']
19
20 # Define the preprocessing steps for categorical variables
21 categorical_transformer = Pipeline(steps=[
22     ('imputer', SimpleImputer(strategy='constant', fill_value='missing')), # Handle missing values
23     ('onehot', OneHotEncoder(handle_unknown='ignore')) # One-hot encode categorical variables
24 ])
25
26 # Create preprocessing pipeline for all columns
27 preprocessor = ColumnTransformer(
28     transformers=[
29         ('cat', categorical_transformer, categorical_cols)
30     ]
31 )
32 # Append the preprocessor to the estimator
33 rf_pipeline = Pipeline(steps=[('preprocessor', preprocessor),

```

Figure 5.4.2.1



```

31 # Append the preprocessor to the estimator
32 rf_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
33                             ('classifier', RandomForestRegressor())])
34
35 # Split the data into training and testing sets
36 X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.4, random_state=42)
37
38 # Fit the model
39 rf_pipeline.fit(X_train, y_train)
40
41 # Evaluate the model
42 print("Training R^2:", rf_pipeline.score(X_train, y_train))
43 print("Testing R^2:", rf_pipeline.score(X_test, y_test))
44
45 print(f'Train Accuracy - : {rf_pipeline.score(X_train, y_train):.3f}')
46 print(f'Test Accuracy - : {rf_pipeline.score(X_test, y_test):.3f}')
47
48
49 # OUTPUT
50 # Training R^2: 0.9060332750431791
51 # Testing R^2: 0.347827611955025
52 # Train Accuracy - : 0.906
53 # Test Accuracy - : 0.348
54
55

```

Figure 5.4.2.2

```

Training R^2: 0.9441759123453746
Testing R^2: 0.6030151291599787
Train Accuracy - : 0.944
Test Accuracy - : 0.603

```

Figure 5.4.2.3

### 5.4.3 Light Gradient Boosting Model

```

import pandas as pd
import lightgbm
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

# Load the data
df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")

# Specify the feature columns to use for prediction
feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'Total_Area', 'Price_per_SQFT']

# Convert 'Yes'/ 'No' in 'Balcony' column to 1/0
df_train['Balcony'] = df_train['Balcony'].map({'Yes': 1, 'No': 0})

# Drop rows with missing values in the selected feature columns
df_train.dropna(subset=feature_columns, inplace=True)

# Extract features and target variable
X = df_train[feature_columns]
y = df_train['Price']

# Split the data into training and validation sets
X_train, X_valid, y_train, y_valid = train_test_split(*arrays X, y, test_size=0.2, random_state=42)

# Define LightGBM parameters
parameters = {
    'boosting_type': 'gbdt',
    'objective': 'regression',
    'metric': 'l2',
    'num_leaves': 31,
    'learning_rate': 0.05,
    'feature_fraction': 0.9
}

```

Figure 5.4.3.1

```

    'bagging_fraction': 0.8,
    'bagging_freq': 5,
    'verbose': 0,
    'early_stopping_rounds': 50
}

# Create LightGBM datasets
train_data = lightgbm.Dataset(X_train, label=y_train)
valid_data = lightgbm.Dataset(X_valid, label=y_valid)

# Train the LightGBM model
model_lgbm = lightgbm.train(params=parameters,
                            train_set=train_data,
                            valid_sets=[train_data, valid_data],
                            num_boost_round=5000)

# Predictions on training and validation data
y_train_pred = model_lgbm.predict(X_train)
y_valid_pred = model_lgbm.predict(X_valid)

# Evaluate the model using R^2 score
r2_train = r2_score(y_train, y_train_pred)
r2_valid = r2_score(y_valid, y_valid_pred)
print("R^2 Train: {:.4f}".format(r2_train))
print("R^2 Valid: {:.4f}".format(r2_valid))

```

Figure 5.4.3.2

```
"C:\data science Internship\output\Linear_Regression\output\linear_regressor.pkl
R^2 Train: 0.9992
R^2 Valid: 0.9990
```

Figure 5.4.3.3

#### 5.4.4 Linear Regression

```
master ~
Random Forest(Algo 2).py LightGBM Algo(3).py Accuracy Comparison.py Linear Regression(Algo 4).py
1 import pandas as pd
2 from sklearn.linear_model import LinearRegression
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import OneHotEncoder
5 from sklearn.compose import ColumnTransformer
6 from sklearn.pipeline import Pipeline
7 from sklearn.impute import SimpleImputer
8
9 # Load the data
10 data = pd.read_csv(r"C:\data science Internship\pythonProject\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")
11
12 # Define the features and target variable
13 X = data.drop(['Price'], axis=1)
14 y = data['Price']
15
16 # Define the categorical columns
17 categorical_cols = ['Name', 'Property_Type', 'Bedrooms1', 'Balcony', 'City', 'Location']
18
19 # Define the preprocessing steps for categorical variables
20 categorical_transformer = Pipeline(steps=[
21     ('imputer', SimpleImputer(strategy='constant', fill_value='missing')), # Handle missing values
22     ('onehot', OneHotEncoder(handle_unknown='ignore')) # One-hot encode categorical variables
23 ])
24
25 # Create preprocessing pipeline for all columns
26 preprocessor = ColumnTransformer(
27     transformers=[
28         ('cat', categorical_transformer, categorical_cols)
29     ])
30
```

Figure 5.4.4.1

```
# Create preprocessing pipeline for all columns
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', categorical_transformer, categorical_cols)
    ])

# Append the preprocessor to the estimator
linear_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                  ('classifier', LinearRegression())])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.4, random_state=42)

# Fit the model
print(linear_pipeline.fit(X_train, y_train))

# Evaluate the model
print("Training R^2:", linear_pipeline.score(X_train, y_train))
print("Testing R^2:", linear_pipeline.score(X_test, y_test))

print(f'Train Accuracy - : {linear_pipeline.score(X_train, y_train):.3f}')
print(f'Test Accuracy - : {linear_pipeline.score(X_test, y_test):.3f}')
```

Figure 5.4.4.2

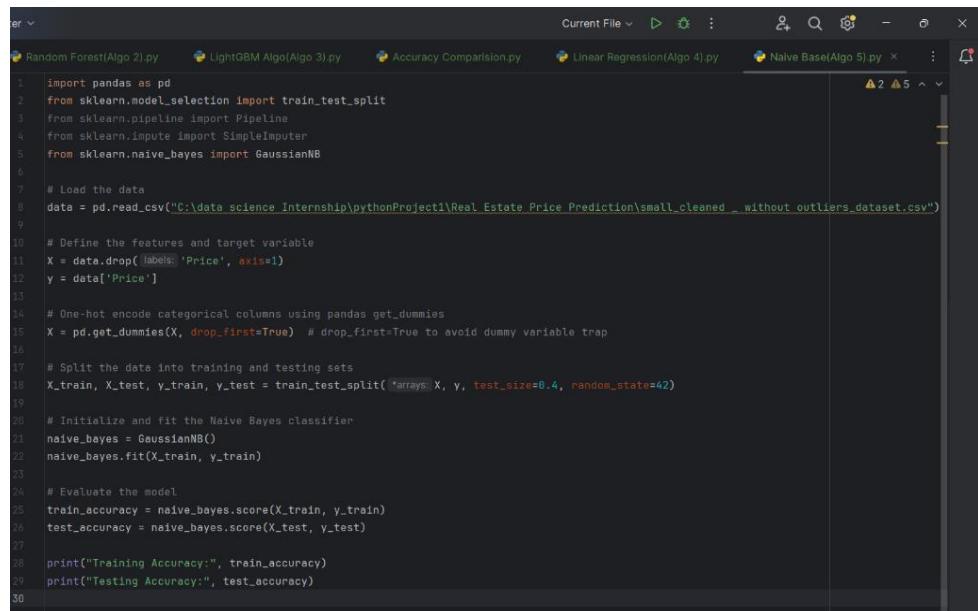
```

Training R^2: 0.9338286255782781
Testing R^2: 0.38710968837148185
Train Accuracy - : 0.934
Test Accuracy - : 0.387

```

Figure 5.4.4.3

#### 5.4.5 Naïve Base



```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.pipeline import Pipeline
4 from sklearn.impute import SimpleImputer
5 from sklearn.naive_bayes import GaussianNB
6
7 # Load the data
8 data = pd.read_csv("C:/data_science_internship/pythonProject1/Real_Estate_Price_Prediction/small_cleaned .. without outliers_dataset.csv")
9
10 # Define the features and target variable
11 X = data.drop( labels=['Price'], axis=1)
12 y = data['Price']
13
14 # One-hot encode categorical columns using pandas get_dummies
15 X = pd.get_dummies(X, drop_first=True) # drop_first=True to avoid dummy variable trap
16
17 # Split the data into training and testing sets
18 X_train, X_test, y_train, y_test = train_test_split(*arrays=X, y, test_size=0.4, random_state=42)
19
20 # Initialize and fit the Naive Bayes classifier
21 naive_bayes = GaussianNB()
22 naive_bayes.fit(X_train, y_train)
23
24 # Evaluate the model
25 train_accuracy = naive_bayes.score(X_train, y_train)
26 test_accuracy = naive_bayes.score(X_test, y_test)
27
28 print("Training Accuracy:", train_accuracy)
29 print("Testing Accuracy:", test_accuracy)
30

```

Figure 5.4.5.1

```

Training Accuracy: 0.8511582060128142
Testing Accuracy: 0.05617147080561715

Process finished with exit code 0

```

Figure 5.4.5.2

## 5.5 ACCURACY COMPARISON

A key component of this evaluation process is accuracy comparison, which makes it possible to compare the predictive capacities of various models. I utilized matplotlib to create a bar chart, enabling a comparative analysis of the accuracy across multiple algorithms. Through this examination, it was discerned that the decision tree algorithm yielded the most favorable outcomes..

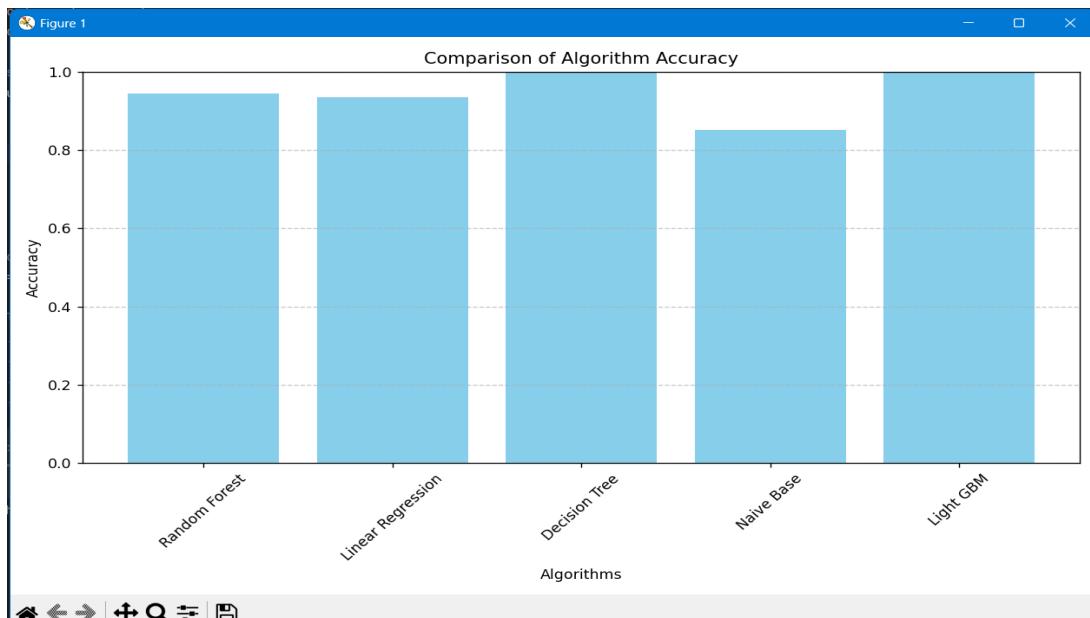


Figure 5.5.1

## 5.6 CREATING REAL ESTATE PRICE PREDICTION MODEL

In the real estate sector, a price prediction model is a potent instrument that provides insights into the dynamic swings in property values. Since a model is required to accomplish the aim, I chose one that matches my dataset flawlessly. I was able to develop a price prediction model with it and get an accurate estimation of the chosen model.

```

1 import pandas as pd
2 import lightgbm
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import r2_score
5
6 # Load the data
7 df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")
8
9 # Specify the columns to use for prediction
10 feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'Total_Area', 'Price_per_SQFT']
11
12 # Preprocess the 'Balcony' column
13 df_train['Balcony'] = df_train['Balcony'].replace({'Yes': 1, 'No': 0}) # Convert 'Yes'/'No' to numeric (1/0)
14
15 # Drop rows with missing values in the selected feature columns
16 df_train.dropna(subset=feature_columns, inplace=True)
17
18 # Extract features and target variable
19 X = df_train[feature_columns]
20 y = df_train['Price']
21
22 # Split the data into training and validation sets
23 X_train, X_valid, y_train, y_valid = train_test_split(*X, y, test_size=0.2, random_state=42)
24
25 # Create LightGBM datasets
26 train_data = lightgbm.Dataset(X_train, label=y_train)
27 valid_data = lightgbm.Dataset(X_valid, label=y_valid)
28
29 # Define LightGBM parameters
30 parameters = {
31     'boosting_type': 'gbdt',
32     'objective': 'regression',
33     'metric': 'l2',
34     'num_leaves': 31,
35     'learning_rate': 0.05,
36     'feature_fraction': 0.9,
37     'bagging_fraction': 0.8,
38     'bagging_freq': 5,
39     'verbose': 0,
40     'early_stopping_rounds': 50
41 }
42
43 # Train the LightGBM model
44 model_lgbm = lightgbm.train(params=parameters,
45                             train_set=train_data,
46                             valid_sets=[train_data, valid_data],
47                             num_boost_rounds=5000)
48
49 # Predictions
50 y_train_pred = model_lgbm.predict(X_train)
51 y_valid_pred = model_lgbm.predict(X_valid)
52
53 # Evaluate the model using R^2 score
54 r2_train = r2_score(y_train, y_train_pred)
55 r2_valid = r2_score(y_valid, y_valid_pred)
56 print("R^2 Train: {:.4f}".format(r2_train))
57 print("R^2 Valid: {:.4f}".format(r2_valid))
58
59 # Take user inputs for prediction
60 user_inputs = {}
61 for feature in feature_columns:
62     user_input = input(f"Enter value for {feature}: ")
63     if feature == 'Balcony':
64         user_inputs[feature] = [1 if user_input.lower() == 'yes' else 0] # Convert 'Yes'/'No' to 1/0
65     else:
66         user_inputs[feature] = [float(user_input)] # Convert input to float for other features

```

Figure 5.6.1

```

1 import pandas as pd
2 import lightgbm
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import r2_score
5
6 # Load the data
7 df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")
8
9 # Specify the columns to use for prediction
10 feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'Total_Area', 'Price_per_SQFT']
11
12 # Preprocess the 'Balcony' column
13 df_train['Balcony'] = df_train['Balcony'].replace({'Yes': 1, 'No': 0}) # Convert 'Yes'/'No' to numeric (1/0)
14
15 # Drop rows with missing values in the selected feature columns
16 df_train.dropna(subset=feature_columns, inplace=True)
17
18 # Extract features and target variable
19 X = df_train[feature_columns]
20 y = df_train['Price']
21
22 # Split the data into training and validation sets
23 X_train, X_valid, y_train, y_valid = train_test_split(*X, y, test_size=0.2, random_state=42)
24
25 # Create LightGBM datasets
26 train_data = lightgbm.Dataset(X_train, label=y_train)
27 valid_data = lightgbm.Dataset(X_valid, label=y_valid)
28
29 # Define LightGBM parameters
30 parameters = {
31     'boosting_type': 'gbdt',
32     'objective': 'regression',
33     'metric': 'l2',
34     'num_leaves': 31,
35     'learning_rate': 0.05,
36     'feature_fraction': 0.9,
37     'bagging_fraction': 0.8,
38     'bagging_freq': 5,
39     'verbose': 0,
40     'early_stopping_rounds': 50
41 }
42
43 # Train the LightGBM model
44 model_lgbm = lightgbm.train(params=parameters,
45                             train_set=train_data,
46                             valid_sets=[train_data, valid_data],
47                             num_boost_rounds=5000)
48
49 # Predictions
50 y_train_pred = model_lgbm.predict(X_train)
51 y_valid_pred = model_lgbm.predict(X_valid)
52
53 # Evaluate the model using R^2 score
54 r2_train = r2_score(y_train, y_train_pred)
55 r2_valid = r2_score(y_valid, y_valid_pred)
56 print("R^2 Train: {:.4f}".format(r2_train))
57 print("R^2 Valid: {:.4f}".format(r2_valid))
58
59 # Take user inputs for prediction
60 user_inputs = {}
61 for feature in feature_columns:
62     user_input = input(f"Enter value for {feature}: ")
63     if feature == 'Balcony':
64         user_inputs[feature] = [1 if user_input.lower() == 'yes' else 0] # Convert 'Yes'/'No' to 1/0
65     else:
66         user_inputs[feature] = [float(user_input)] # Convert input to float for other features

```

Figure 5.6.2

```
# Create DataFrame from user inputs
user_df = pd.DataFrame(user_inputs)

# Make predictions on user input data
predicted_price = model_lgbm.predict(user_df)

# Print the predicted price
print("Predicted Price:", predicted_price[0])
```

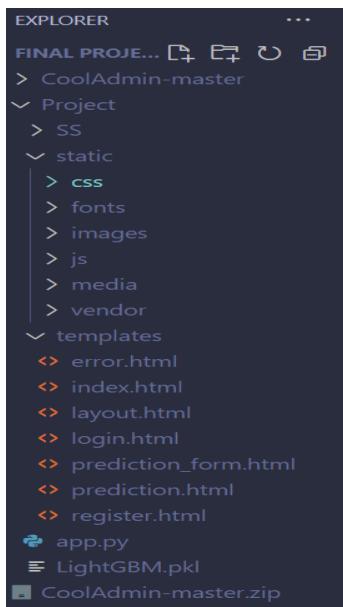
Figure 5.6.3

## CHAPTER 6: IMPLEMENTATION

### 6.1 SETTING-UP FLASK

For my application, I have developed a graphical user interface using the Flask framework. Setting up the file structure was the first step, and I utilized it to make a fantastic file structure for my entire application. I created using Flask Framework.

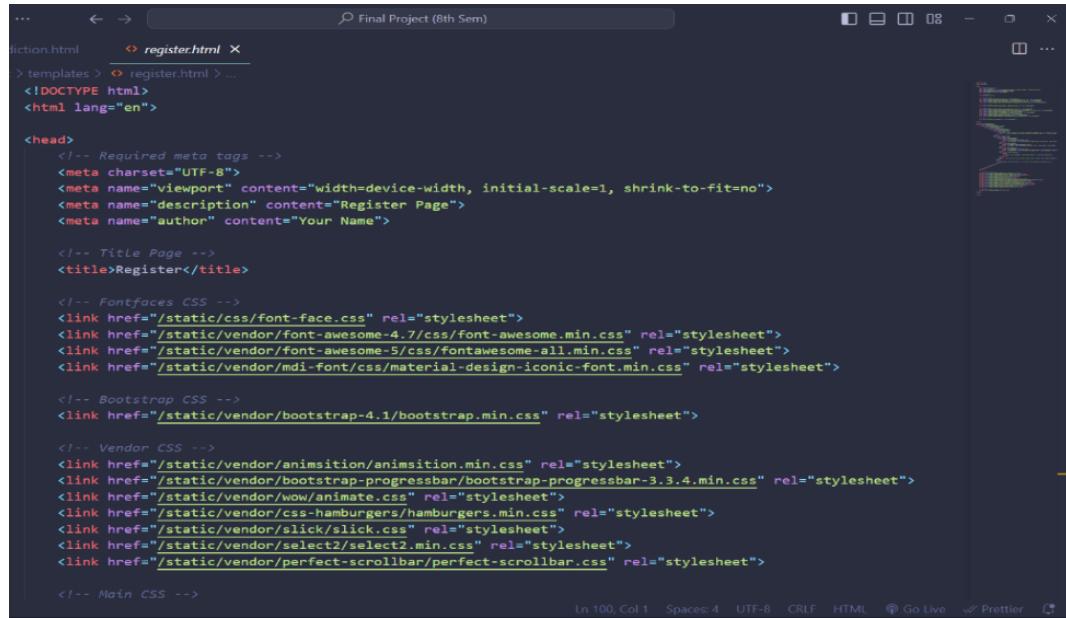
I made two distinct directories in the file structure. one is called template, and inside the template folder, I've made a ton of HTML files to make a beautiful user interface. In the static folder, I've added a variety of javascript and bootstrap files to enhance the online program.



**Figure 6.1.1**

### 6.2 REGISTRATION

To register any user on the local server, I have made a registration form. To do this, I have added registration.html to the template dictionary and appended this HTML file with the app.py application file.



```

<!-- Required meta tags -->
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="Register Page">
<meta name="author" content="Your Name">

<!-- Title Page -->
<title>Register</title>

<!-- Fontfaces CSS -->
<link href="/static/css/font-face.css" rel="stylesheet">
<link href="/static/vendor/font-awesome-4.7/css/font-awesome.min.css" rel="stylesheet">
<link href="/static/vendor/font-awesome-5/css/fontawesome-all.min.css" rel="stylesheet">
<link href="/static/vendor mdi-font/css/material-design-iconic-font.min.css" rel="stylesheet">

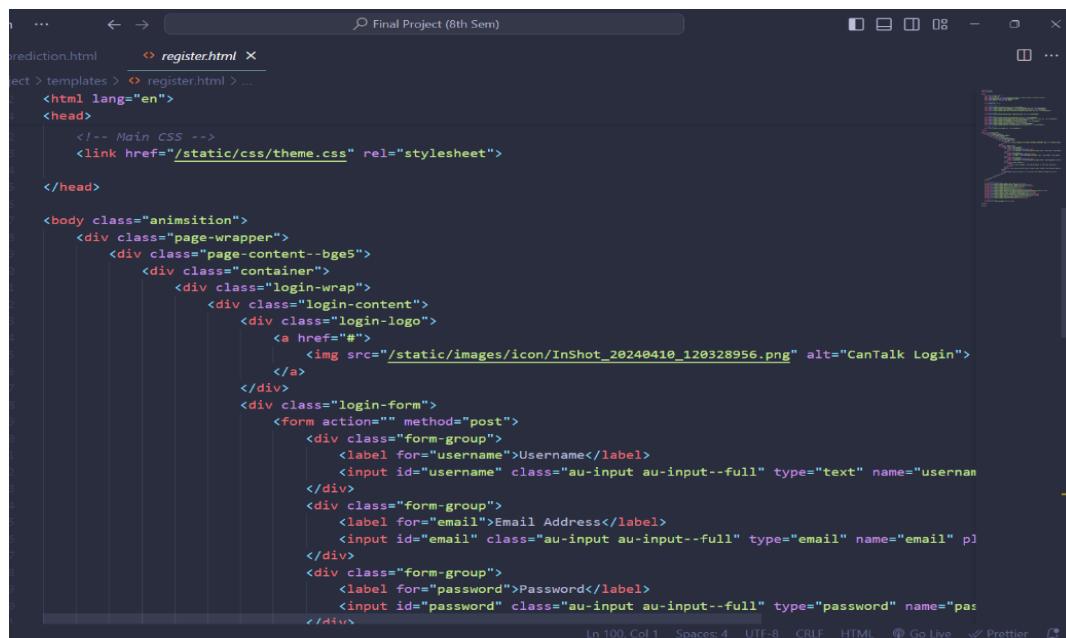
<!-- Bootstrap CSS -->
<link href="/static/vendor/bootstrap-4.1/bootstrap.min.css" rel="stylesheet">

<!-- Vendor CSS -->
<link href="/static/vendor/animotion/animotion.min.css" rel="stylesheet">
<link href="/static/vendor/bootstrap-progressbar/bootstrap-progressbar-3.3.4.min.css" rel="stylesheet">
<link href="/static/vendor/wow/animate.css" rel="stylesheet">
<link href="/static/vendor/css-hamburgers/hamburgers.min.css" rel="stylesheet">
<link href="/static/vendor/slick/slick.css" rel="stylesheet">
<link href="/static/vendor/select2/select2.min.css" rel="stylesheet">
<link href="/static/vendor/perfect-scrollbar/perfect-scrollbar.css" rel="stylesheet">

<!-- Main CSS -->

```

Figure 6.2.1



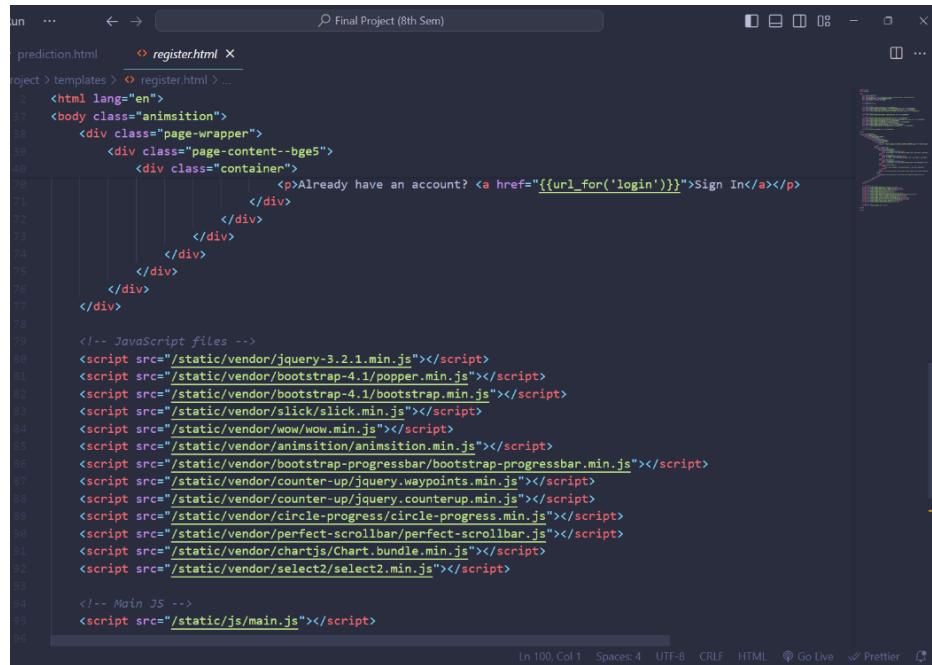
```

<!-- Main CSS -->
<link href="/static/css/theme.css" rel="stylesheet">

<body class="animotion">
    <div class="page-wrapper">
        <div class="page-content--bgc5">
            <div class="container">
                <div class="login-wrap">
                    <div class="login-content">
                        <div class="login-logo">
                            <a href="#">
                                
                            </a>
                        </div>
                        <div class="login-form">
                            <form action="" method="post">
                                <div class="form-group">
                                    <label for="username">Username</label>
                                    <input id="username" class="au-input au-input--full" type="text" name="username">
                                </div>
                                <div class="form-group">
                                    <label for="email">Email Address</label>
                                    <input id="email" class="au-input au-input--full" type="email" name="email" p1>
                                </div>
                                <div class="form-group">
                                    <label for="password">Password</label>
                                    <input id="password" class="au-input au-input--full" type="password" name="password">
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Figure 6.2.2



```

1  un ... ← → | Final Project (8th Sem)
2  prediction.html register.html ✘
3  Project > templates > register.html > ...
4  1 <html lang="en">
5  2   <body class="animsition">
6  3     <div class="page-wrapper">
7  4       <div class="page-content--bgc5">
8  5         <div class="container">
9  6           <p>Already have an account? <a href="{{url_for('login')}}">Sign In</a></p>
10 7         </div>
11 8       </div>
12 9     </div>
13 10   </div>
14 11 </div>
15
16  <!-- JavaScript files -->
17  <script src="/static/vendor/jquery-3.2.1.min.js"></script>
18  <script src="/static/vendor/bootstrap-4.1/popper.min.js"></script>
19  <script src="/static/vendor/bootstrap-4.1/bootstrap.min.js"></script>
20  <script src="/static/vendor/slick/slick.min.js"></script>
21  <script src="/static/vendor/wow/wow.min.js"></script>
22  <script src="/static/vendor/animations/animations.min.js"></script>
23  <script src="/static/vendor/bootstrap-progressbar/bootstrap-progressbar.min.js"></script>
24  <script src="/static/vendor/counter-up/jquery.waypoints.min.js"></script>
25  <script src="/static/vendor/counter-up/jquery.counterup.min.js"></script>
26  <script src="/static/vendor/circle-progress/circle-progress.min.js"></script>
27  <script src="/static/vendor/perfect-scrollbar/perfect-scrollbar.js"></script>
28  <script src="/static/vendor/chartjs/Chart.bundle.min.js"></script>
29  <script src="/static/vendor/select2/select2.min.js"></script>
30
31  <!-- Main JS -->
32  <script src="/static/js/main.js"></script>
33
34
35
36
37
38
39
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

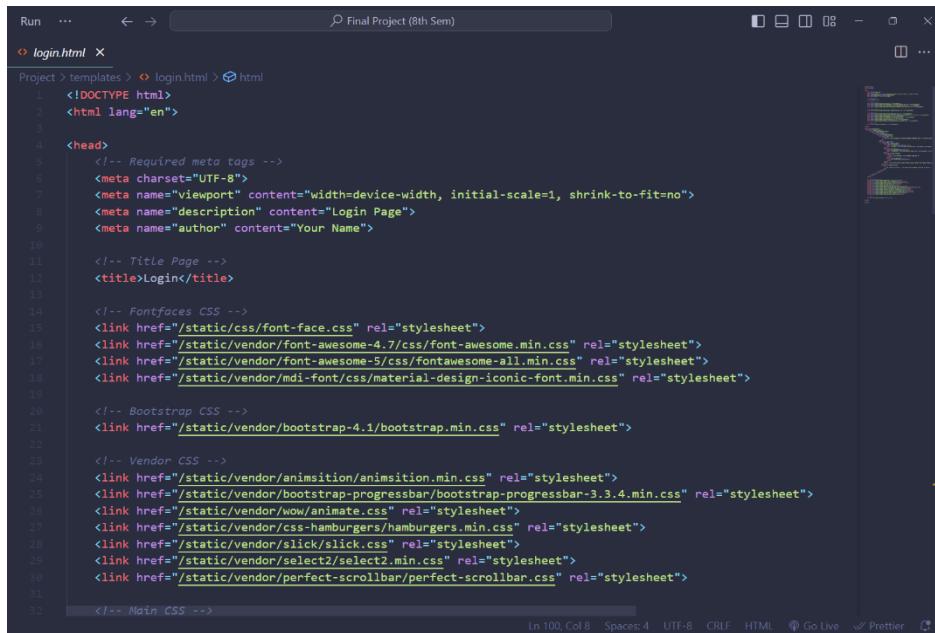
```

In 100 Col 1 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ✨ Prettier

Figure 6.2.3

## 6.3 LOGIN

The purpose of the login form is to allow any existing user to log in to the program. To that end, I have connected the login form to the local server, allowing it to retrieve and validate user data.



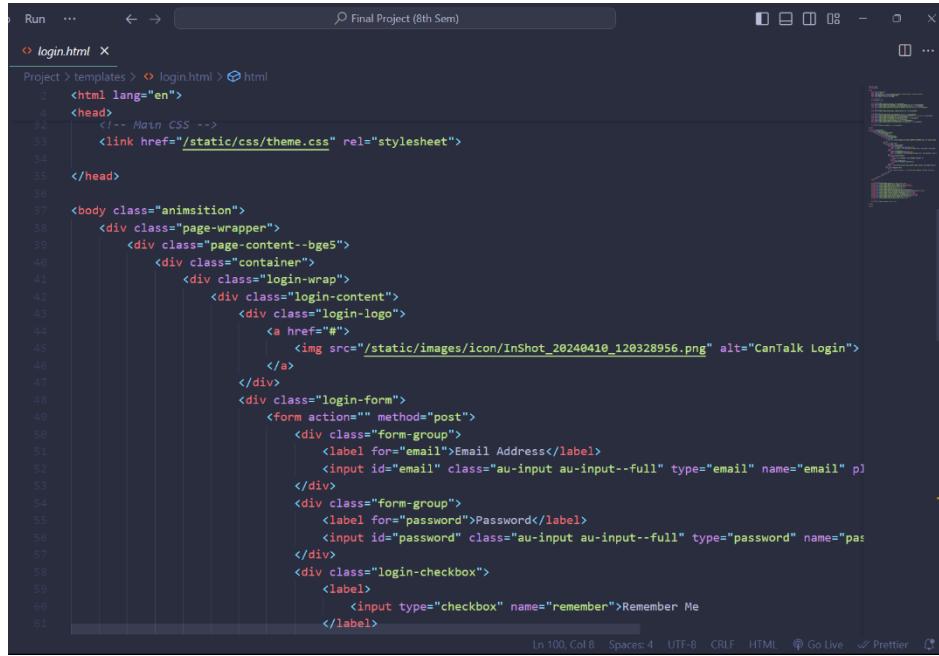
```

Run ... ← → | Final Project (8th Sem)
1  login.html ✘
2  Project > templates > login.html > html
3
4  1 <!DOCTYPE html>
5  2 <html lang="en">
6
7  3   <head>
8  4     <!-- Required meta tags -->
9  5     <meta charset="UTF-8">
10 6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
11 7     <meta name="description" content="Login Page">
12 8     <meta name="author" content="Your Name">
13
14 9   <!-- Title Page -->
15 10  <title>Login</title>
16
17 11  <!-- Fontfaces CSS -->
18 12  <link href="/static/css/font-face.css" rel="stylesheet">
19 13  <link href="/static/vendor/font-awesome-4.7/css/font-awesome.min.css" rel="stylesheet">
20 14  <link href="/static/vendor/font-awesome-5/css/fontawesome-all.min.css" rel="stylesheet">
21 15  <link href="/static/vendor/mdi-font/css/material-design-iconic-font.min.css" rel="stylesheet">
22
23 16  <!-- Bootstrap CSS -->
24 17  <link href="/static/vendor/bootstrap-4.1/bootstrap.min.css" rel="stylesheet">
25
26 18  <!-- Vendor CSS -->
27 19  <link href="/static/vendor/animations/animations.min.css" rel="stylesheet">
28 20  <link href="/static/vendor/bootstrap-progressbar/bootstrap-progressbar-3.3.4.min.css" rel="stylesheet">
29 21  <link href="/static/vendor/wow/animate.css" rel="stylesheet">
30 22  <link href="/static/vendor/css-hamburgers/hamburgers.min.css" rel="stylesheet">
31 23  <link href="/static/vendor/slick/slick.css" rel="stylesheet">
32 24  <link href="/static/vendor/select2/select2.min.css" rel="stylesheet">
33 25  <link href="/static/vendor/perfect-scrollbar/perfect-scrollbar.css" rel="stylesheet">
34
35  <!-- Main CSS -->
36
37
38
39
39
40
41
42
43
44
45
46
47
48
49
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96

```

In 100 Col 8 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ✨ Prettier

Figure 6.3.1

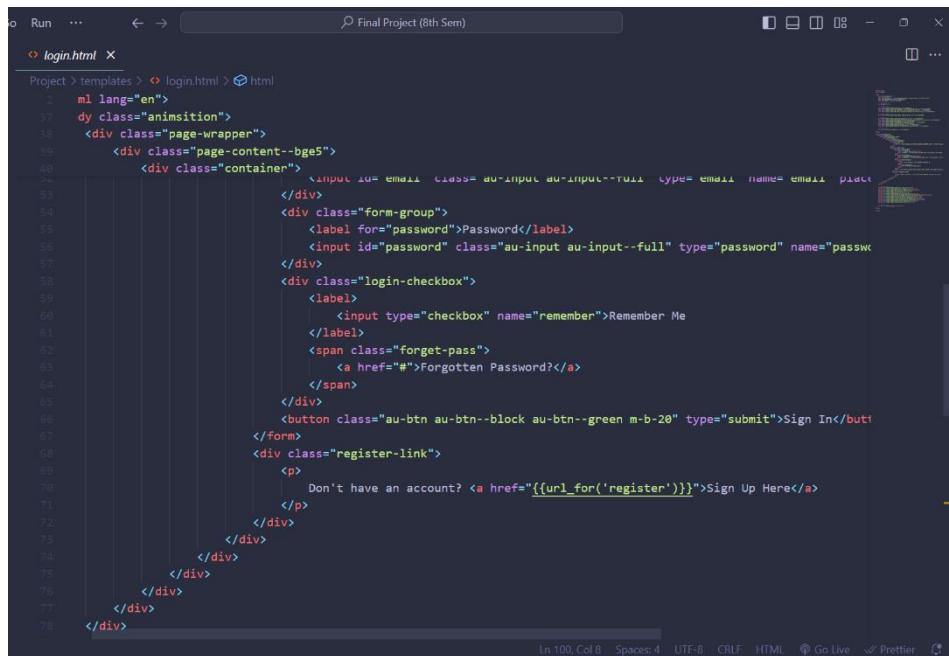


```

<html lang="en">
<head>
    <!-- Main CSS -->
    <link href="/static/css/theme.css" rel="stylesheet">
</head>
<body class="animsition">
    <div class="page-wrapper">
        <div class="page-content--bge5">
            <div class="container">
                <div class="login-wrap">
                    <div class="login-content">
                        <div class="login-logo">
                            <a href="#">
                                
                            </a>
                        </div>
                        <div class="login-form">
                            <form action="" method="post">
                                <div class="form-group">
                                    <label for="email">Email Address</label>
                                    <input id="email" class="au-input au-input--full" type="email" name="email" placeholder="Email Address">
                                </div>
                                <div class="form-group">
                                    <label for="password">Password</label>
                                    <input id="password" class="au-input au-input--full" type="password" name="password" placeholder="Password">
                                </div>
                                <div class="login-checkbox">
                                    <label>
                                        <input type="checkbox" name="remember">Remember Me
                                    </label>
                                </div>
                            </form>
                            <div class="forget-pass">
                                <a href="#">Forgotten Password?</a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>

```

Figure 6.3.2



```

<html lang="en">
<head>
    <!-- Main CSS -->
    <link href="/static/css/theme.css" rel="stylesheet">
</head>
<body class="animsition">
    <div class="page-wrapper">
        <div class="page-content--bge5">
            <div class="container">
                <div class="login-wrap">
                    <div class="login-content">
                        <div class="login-logo">
                            <a href="#">
                                
                            </a>
                        </div>
                        <div class="login-form">
                            <form action="" method="post">
                                <div class="form-group">
                                    <label for="email">Email Address</label>
                                    <input id="email" class="au-input au-input--full" type="email" name="email" placeholder="Email Address">
                                </div>
                                <div class="form-group">
                                    <label for="password">Password</label>
                                    <input id="password" class="au-input au-input--full" type="password" name="password" placeholder="Password">
                                </div>
                                <div class="login-checkbox">
                                    <label>
                                        <input type="checkbox" name="remember">Remember Me
                                    </label>
                                    <span class="forget-pass">
                                        <a href="#">Forgotten Password?</a>
                                    </span>
                                </div>
                            </form>
                            <div class="register-link">
                                <p>Don't have an account? <a href="{{url_for('register')}}">Sign Up Here</a></p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>

```

Figure 6.3.3

```

 1 <!-- JavaScript files -->
 2 <script src="/static/vendor/jquery-3.2.1.min.js"></script>
 3 <script src="/static/vendor/bootstrap-4.1/popper.min.js"></script>
 4 <script src="/static/vendor/bootstrap-4.1/bootstrap.min.js"></script>
 5 <script src="/static/vendor/slick/slick.min.js"></script>
 6 <script src="/static/vendor/wow/wow.min.js"></script>
 7 <script src="/static/vendor/animationsition.animationsition.min.js"></script>
 8 <script src="/static/vendor/bootstrap-progressbar/bootstrap-progressbar.min.js"></script>
 9 <script src="/static/vendor/counter-up/jquery.waypoints.min.js"></script>
10 <script src="/static/vendor/counter-up/jquery.counterup.min.js"></script>
11 <script src="/static/vendor/circle-progress/circle-progress.min.js"></script>
12 <script src="/static/vendor/perfect-scrollbar/perfect-scrollbar.js"></script>
13 <script src="/static/vendor/chartjs/Chart.bundle.min.js"></script>
14 <script src="/static/vendor/select2/select2.min.js"></script>
15
16 <!-- Main JS -->
17 <script src="/static/js/main.js"></script>
18
19 </body>
20
21 </html>
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

Ln 100, Col 8 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live

Figure 6.3.4

## 6.4 PIKAL FILE

The pickle file, which contains all of the program's logic and allows it to run the prediction model, is the heart and soul of this application.

```

 1 import pandas as pd
 2 import lightgbm
 3 from sklearn.model_selection import train_test_split
 4 from sklearn.metrics import r2_score
 5
 6 # Load the data
 7 df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_dat
 8
 9 # Specify the feature columns to use for prediction
10 feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'total_sqft', 'Price_per_SQFT']
11
12 # Convert 'Yes'/'No' in 'Balcony' column to 1/0
13 df_train['Balcony'] = df_train['Balcony'].map({'Yes': 1, 'No': 0})
14
15 # Drop rows with missing values in the selected feature columns
16 df_train.dropna(subset=feature_columns, inplace=True)
17
18 # Extract features and target variable
19 X = df_train[feature_columns]
20 y = df_train['Price']
21
22 # Split the data into training and validation sets
23 X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2, random_state=42)
24
25 # Define LightGBM parameters
26 parameters = {
27     'boosting_type': 'gbdt',
28     'objective': 'regression',
29     'metric': 'l2',
30     'num_leaves': 31,
31     'learning_rate': 0.05,
32     'feature_fraction': 0.9,
33 }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

Auto run command is working. Check out the README to make it do s...  
Ln 54, Col 1 Spaces: 4 UTF-8 CRLF Plain Text ⚡ Go Live ⚡ Prettier ⚡

Figure 6.4.1

```

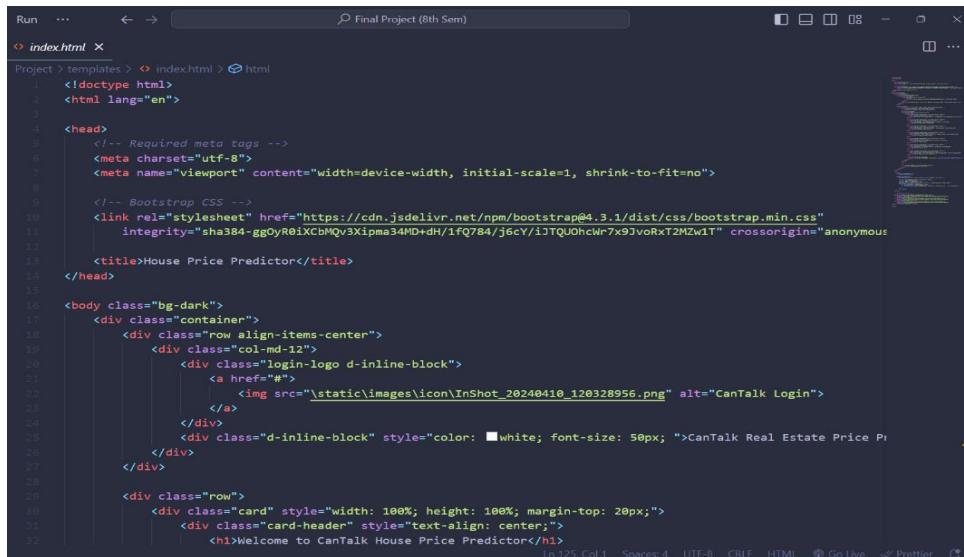
26     parameters = {
27         'learning_rate': 0.05,
28         'feature_fraction': 0.9,
29         'bagging_fraction': 0.8,
30         'bagging_freq': 5,
31         'verbose': 0,
32         'early_stopping_rounds': 50
33     }
34
35
36
37
38
39 # Create LightGBM datasets
40 train_data = lightgbm.Dataset(X_train, label=y_train)
41 valid_data = lightgbm.Dataset(X_valid, label=y_valid)
42
43 # Train the LightGBM model
44 model_lgbm = lightgbm.train(params=parameters,
45                             train_set=train_data,
46                             valid_sets=[train_data, valid_data],
47                             num_boost_round=5000)
48
49 # Predictions on training and validation data
50 y_train_pred = model_lgbm.predict(X_train)
51 y_valid_pred = model_lgbm.predict(X_valid)
52

```

Figure 6.4.2

## 6.5 INDEX FILE

The index file is used to retrieve user-provided values. It has a form where the user can enter various user values for the type of property, such as price per square foot, number of rooms, number of bathrooms, and total square feet.



```

<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
        integrity="sha384-ggOyR0iXcbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhW7x9JvoRxT2MZw1T" crossorigin="anonymous"/>
    <title>House Price Predictor</title>
</head>
<body class="bg-dark">
    <div class="container">
        <div class="row align-items-center">
            <div class="col-md-12">
                <div class="login-logo d-inline-block">
                    <a href="#">
                        
                    </a>
                </div>
                <div class="d-inline-block" style="color: white; font-size: 50px; ">CanTalk Real Estate Price P</div>
            </div>
        </div>
        <div class="row">
            <div class="card" style="width: 100%; height: 100%; margin-top: 20px;">
                <div class="card-header" style="text-align: center;">
                    <h1>Welcome to CanTalk House Price Predictor</h1>
                </div>
            </div>
        </div>
    </div>
</body>

```

Figure 6.5.1

```

<html lang="en">
  <body class="bg-dark">
    <div class="container">
      <div class="card" style="width: 100%; height: 100%; margin-top: 20px;">
        <h1>Welcome to CanTalk House Price Predictor</h1>
        <br class="card-body">
        <form method="post" accept-charset="utf-8">
          <div class="row">
            <div class="col-md-6 form-group" style="text-align: center;">
              <label><b>Enter Bedrooms</b></label>
              <input type="text" class="form-control" id="Bedrooms1" name="Bedrooms1" placeholder="Enter No of Bedrooms" style="width: 100%; height: 100%; margin-top: 20px;">
            </div>
            <div class="col-md-6 form-group" style="text-align: center;">
              <label><b>Enter No of Bathrooms</b></label>
              <input type="number" class="form-control" id="Baths" name="Baths" placeholder="Enter No of Bathrooms" style="width: 100%; height: 100%; margin-top: 20px;">
            </div>
            <div class="col-md-6 form-group" style="text-align: center;">
              <label><b>Enter Total Area:</b></label>
              <input type="number" class="form-control" id="total_sqft" name="total_sqft" placeholder="Enter Total Area" style="width: 100%; height: 100%; margin-top: 20px;">
            </div>
            <div class="col-md-6 form-group" style="text-align: center;">
              <label><b>Enter Property Type</b></label>
              <input type="text" class="form-control" id="Property_Type" name="Property_Type" placeholder="Enter Property Type" style="width: 100%; height: 100%; margin-top: 20px;">
            </div>
          </div>
        </form>
      </div>
    </div>
  </body>
</html>

```

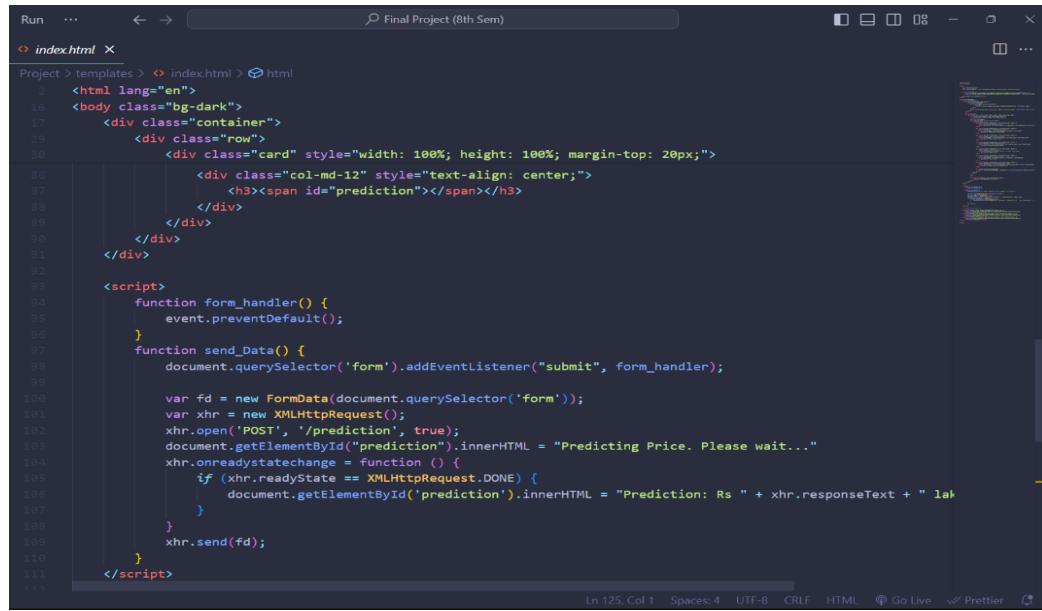
Figure 6.5.2

```

<html lang="en">
  <body class="bg-dark">
    <div class="container">
      <div class="card" style="width: 100%; height: 100%; margin-top: 20px;">
        <div class="row">
          <div class="col-md-6 form-group" style="text-align: center;">
            <label><b>Enter City</b></label>
            <input type="text" class="form-control" id="City" name="City" placeholder="Enter City" style="width: 100%; height: 100%; margin-top: 20px;">
          </div>
          <div class="col-md-6 form-group" style="text-align: center;">
            <label><b>Having Balcony?</b></label>
            <input type="text" class="form-control" id="Balcony" name="Balcony" placeholder="Yes or No" style="width: 100%; height: 100%; margin-top: 20px;">
          </div>
          <div class="col-md-6 form-group" style="text-align: center;">
            <label><b>Enter Price per squarefit</b></label>
            <input type="number" class="form-control" id="Price_per_SQFT" name="Price_per_SQFT" placeholder="Enter Price per Squarefit" style="width: 100%; height: 100%; margin-top: 20px;">
          </div>
          <div class="col-md-12 form-group">
            <button class="btn btn-primary form-control" onclick="send_Data()">Predict Price</button>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

Figure 6.5.3



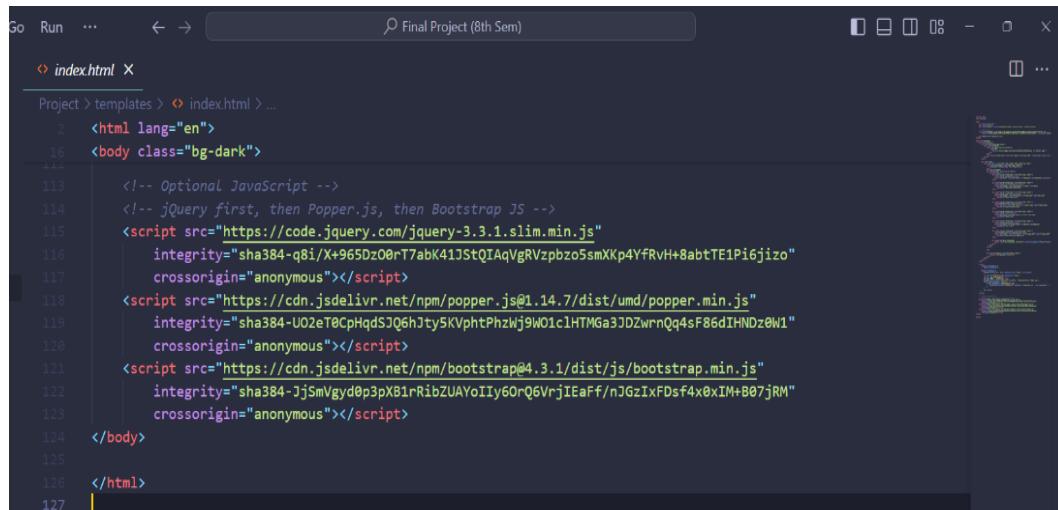
```

<html lang="en">
<body class="bg-dark">
<div class="container">
<div class="row">
<div class="card" style="width: 100%; height: 100%; margin-top: 20px;">
<div class="col-md-12" style="text-align: center;">
<h3><span id="prediction"></span></h3>
</div>
</div>
</div>
</div>
<script>
    function form_handler() {
        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', '/prediction', true);
        document.getElementById('prediction').innerHTML = "Predicting Price. Please wait...";
        xhr.onreadystatechange = function () {
            if (xhr.readyState == XMLHttpRequest.DONE) {
                document.getElementById('prediction').innerHTML = "Prediction: Rs " + xhr.responseText + " lakhs";
            }
        }
        xhr.send(fd);
    }
</script>

```

Figure 6.5.4



```

<html lang="en">
<body class="bg-dark">


<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
       integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVpbzo5smXKp4YfRvH+8abTE1Pi6jizo"
       crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
       integrity="sha384-UO2eT0CpHqdSJQ6hJty5KvphPhzwj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1"
       crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
       integrity="sha384-JJSmVgyd0p3pX81rRibZUAYoIIy6OrQ6VrjIEaFF/nJGzIxFDsf4x0xIM+B07jRM"
       crossorigin="anonymous"></script>
</body>
</html>

```

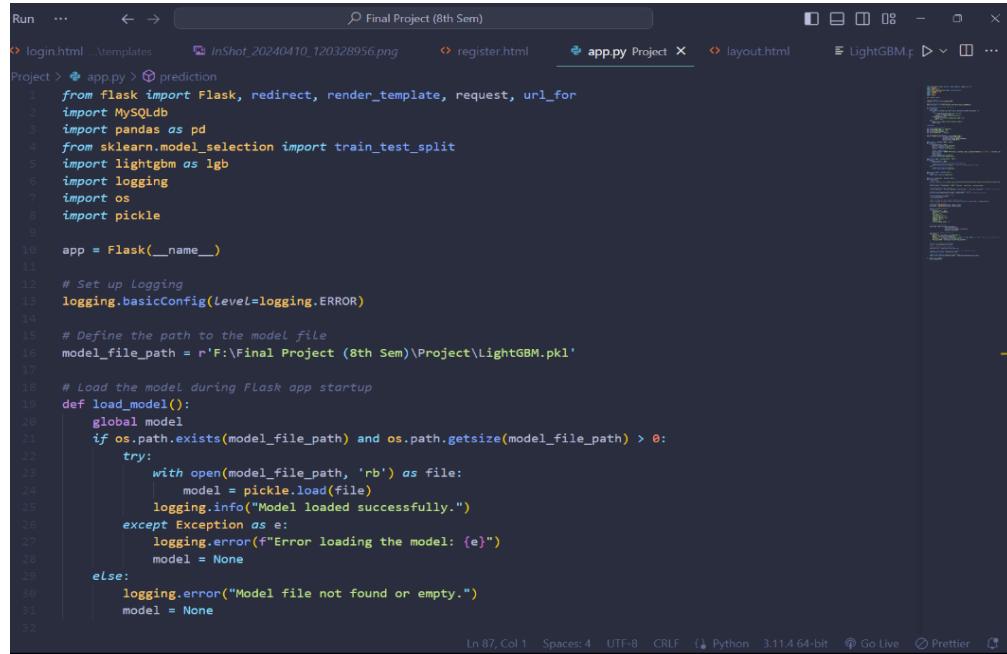
Figure 6.5.5

## 6.6 APP.PY

The main file that links all of the application's connections in the project is called app.py.

It was made with the help of several libraries, including flask, mysqldb, pandas, scikit-

learn, light gbm, logging, and OS, among others. It loads the model, pickle file, and other HTML files that are contained in the template directory.

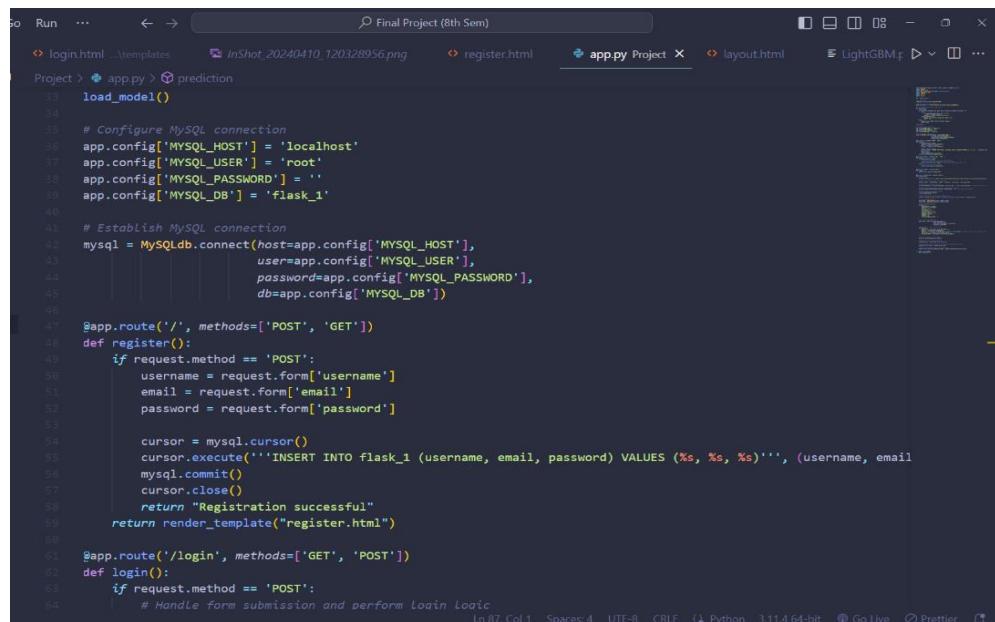


```

Run ... ← → ⌂ Final Project (8th Sem)
login.html templates InShot_20240410_120328956.png register.html app.py Project layout.html LightGBM.p
Project > app.py > prediction
1  from flask import Flask, redirect, render_template, request, url_for
2  import MySQLdb
3  import pandas as pd
4  from sklearn.model_selection import train_test_split
5  import lightgbm as lgb
6  import logging
7  import os
8  import pickle
9
10 app = Flask(__name__)
11
12 # Set up Logging
13 logging.basicConfig(level=logging.ERROR)
14
15 # Define the path to the model file
16 model_file_path = r'F:\Final Project (8th Sem)\Project\LightGBM.pkl'
17
18 # Load the model during Flask app startup
19 def load_model():
20     global model
21     if os.path.exists(model_file_path) and os.path.getsize(model_file_path) > 0:
22         try:
23             with open(model_file_path, 'rb') as file:
24                 model = pickle.load(file)
25                 logging.info("Model loaded successfully.")
26         except Exception as e:
27             logging.error(f"Error loading the model: {e}")
28             model = None
29     else:
30         logging.error("Model file not found or empty.")
31     model = None
32
33 Ln 87, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier

```

Figure 6.6.1

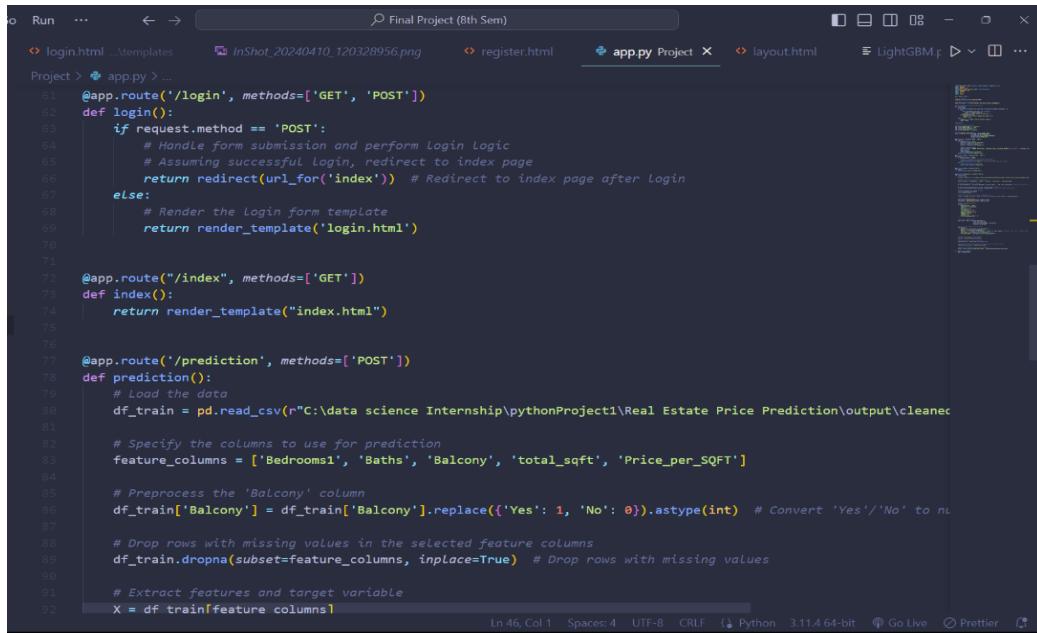


```

Run ... ← → ⌂ Final Project (8th Sem)
login.html templates InShot_20240410_120328956.png register.html app.py Project layout.html LightGBM.p
Project > app.py > prediction
33 load_model()
34
35 # Configure MySQL connection
36 app.config['MYSQL_HOST'] = 'localhost'
37 app.config['MYSQL_USER'] = 'root'
38 app.config['MYSQL_PASSWORD'] = ''
39 app.config['MYSQL_DB'] = 'flask_1'
40
41 # Establish MySQL connection
42 mysql = MySQLdb.connect(host=app.config['MYSQL_HOST'],
43                         user=app.config['MYSQL_USER'],
44                         password=app.config['MYSQL_PASSWORD'],
45                         db=app.config['MYSQL_DB'])
46
47 @app.route('/', methods=['POST', 'GET'])
48 def register():
49     if request.method == 'POST':
50         username = request.form['username']
51         email = request.form['email']
52         password = request.form['password']
53
54         cursor = mysql.cursor()
55         cursor.execute("INSERT INTO flask_1 (username, email, password) VALUES (%s, %s, %s)", (username, email, password))
56         mysql.commit()
57         cursor.close()
58         return "Registration successful"
59         return render_template("register.html")
60
61 @app.route('/login', methods=['GET', 'POST'])
62 def login():
63     if request.method == 'POST':
64         # Handle form submission and perform login logic
65
66 Ln 87, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live Prettier

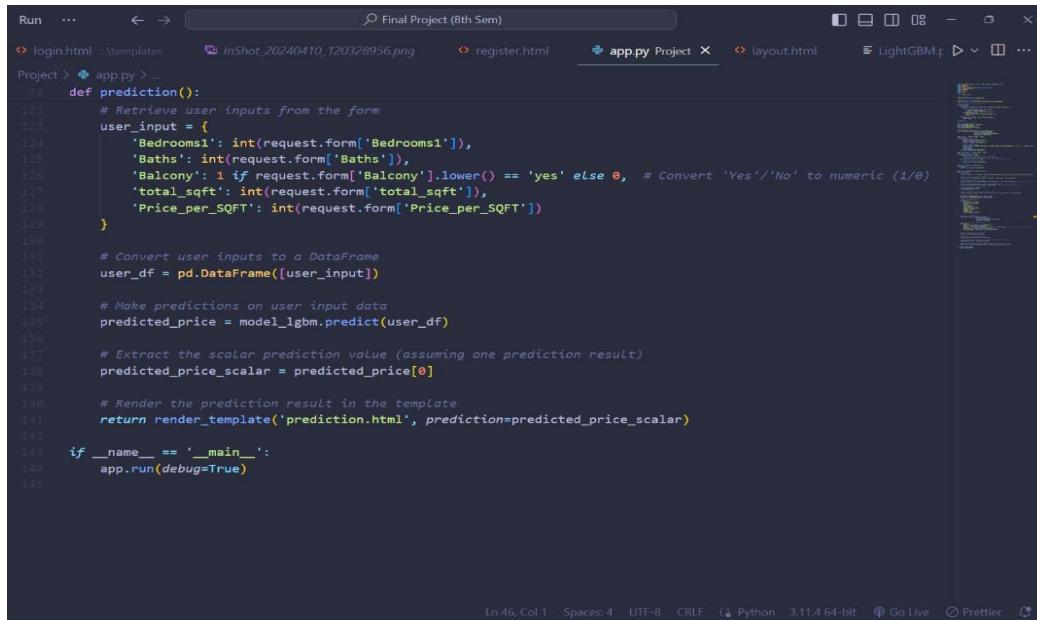
```

Figure 6.6.2



```

Run ... ← → Final Project (8th Sem)
login.html ...templates InShot_20240410_120328956.png register.html app.py Project layout.html LightGBM.f ...
Project > app.py > ...
61     @app.route('/login', methods=['GET', 'POST'])
62     def login():
63         if request.method == 'POST':
64             # Handle form submission and perform Login Logic
65             # Assuming successful Login, redirect to index page
66             return redirect(url_for('index')) # Redirect to index page after Login
67         else:
68             # Render the Login form template
69             return render_template('login.html')
70
71
72     @app.route('/index', methods=['GET'])
73     def index():
74         return render_template("index.html")
75
76
77     @app.route('/prediction', methods=['POST'])
78     def prediction():
79         # Load the data
80         df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned.csv")
81
82         # Specify the columns to use for prediction
83         feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'total_sqft', 'Price_per_SQFT']
84
85         # Preprocess the 'Balcony' column
86         df_train['Balcony'] = df_train['Balcony'].replace({'Yes': 1, 'No': 0}).astype(int) # Convert 'Yes'/ 'No' to numbers
87
88         # Drop rows with missing values in the selected feature columns
89         df_train.dropna(subset=feature_columns, inplace=True) # Drop rows with missing values
90
91         # Extract features and target variable
92         X = df_train[feature_columns]
93
94
95         # Split the data into training and validation sets
96         X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2, random_state=42)
97
98         # Create LightGBM datasets
99         train_data = lgb.Dataset(X_train, label=y_train)
100        valid_data = lgb.Dataset(X_valid, label=y_valid)
101
102        # Define LightGBM parameters
103        parameters = {
104            'boosting_type': 'gbdt',
105            'objective': 'regression',
106            'metric': 'l2',
107            'num_leaves': 31,
108            'learning_rate': 0.05,
109            'feature_fraction': 0.9,
110            'bagging_fraction': 0.8,
111            'bagging_freq': 5,
112            'verbose': 0,
113            'early_stopping_rounds': 50
114        }
115
116        # Train the LightGBM model
117        model_lgbm = lgb.train(params=parameters,
118                               train_set=train_data,
119                               valid_sets=[train_data, valid_data],
120                               num_boost_round=5000)
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
757
758
759
759
760
761
762
763
764
765
766
767
767
768
769
769
770
771
772
773
774
775
776
777
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
807
808
809
809
810
811
812
813
814
815
816
816
817
817
818
819
819
820
821
822
823
824
825
826
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
867
867
868
869
869
870
871
872
873
874
875
876
877
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
907
908
909
909
910
911
912
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
16
```



The screenshot shows a code editor with a dark theme. The file being edited is `app.py`. The code is a Python script for a web application using Flask and LightGBM. It defines a `prediction()` function to handle user inputs from a form, convert them into a DataFrame, make predictions using a LightGBM model, and return the result. It also includes a `if __name__ == '__main__': app.run(debug=True)` block at the bottom. The code editor has tabs for `login.html`, `InShot_20240410_120328956.png`, `register.html`, `app.py`, `Project`, `layout.html`, and `LightGBM`. The status bar at the bottom shows file statistics and toolbars for Python 3.11.4 64-bit, Go Live, and Prettier.

Figure 6.6.5

## 6.7 GUI

The graphical user interface (GUI) is what the user can see and utilize. When utilizing this form, the user can first view the registration form as shown here. If the user is already registered, he can proceed straight to the login form by clicking the sign in button at the bottom of the registration form. Otherwise, the user can register himself to the local server.

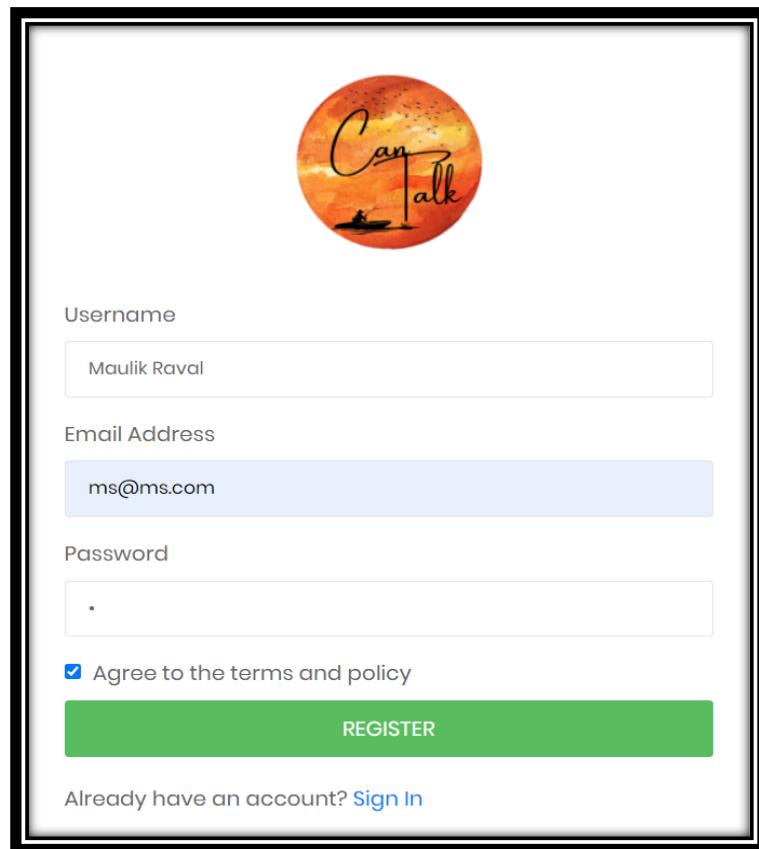


Figure 6.7.1

As soon as the user enters their login, email address, and password, the information is saved in the localhost database, as seen below.

	ID	username	email	password
<input type="checkbox"/>	1	msr	ms@ms.com	1
<input type="checkbox"/>	2	cantor	cantor@cantalk.ca	123
<input type="checkbox"/>	3	maulik.raval	maulik.raval@mail.ca	121
<input type="checkbox"/>	4	maulik.raval	maulik.raval@mail.ca	121
<input type="checkbox"/>	5	bits	campus@bits.in	098

Figure 6.7.2

The next time they visit, the existing user will be taken straight to the login page, where they will need to enter their email address and password to log in. The application will compare the user's information entered by the user with that of the local host database.

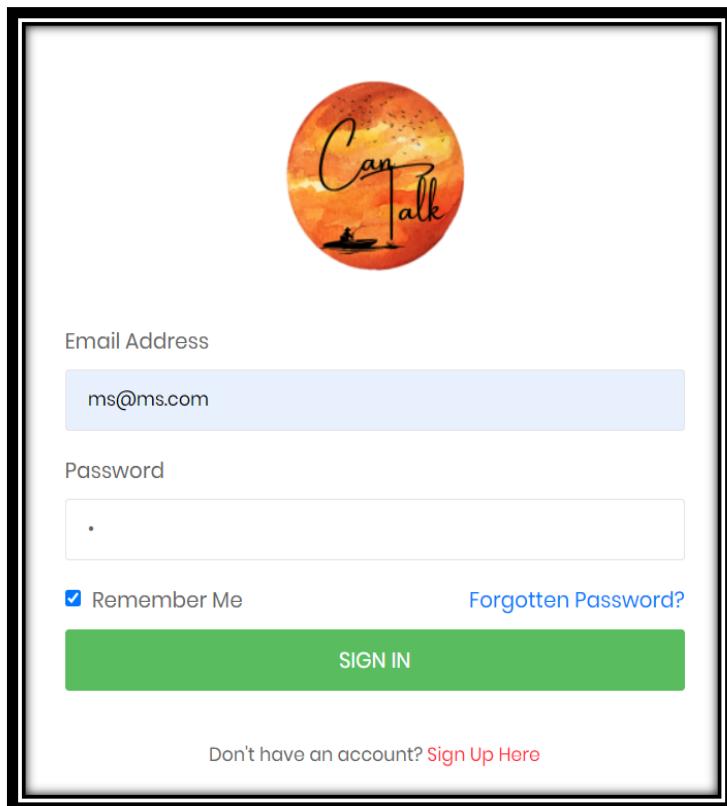


Figure 6.7.3

The user will engage with the Cantalk real estate price prediction system after their login credentials have been verified. Here, they will be able to submit various details, including the number of bedrooms, baths, and total space of the property, as well as whether or not it has a balcony and how much it will cost per square foot.

The screenshot shows a web browser window with the URL 127.0.0.1:5000/index. The title bar reads "CanTalk Real Estate Price Prediction System". The main content area is titled "Welcome to CanTalk House Price Predictor". It contains the following input fields:

- Enter Bedrooms: 3
- Enter No of Bathrooms: 3
- Enter Total Area: 1320
- Enter Property Type: Flat
- Enter City: Chennai
- Having Balcony?: Yes
- Enter Price per squarefit: 7580

A blue "Predict Price" button is at the bottom.

Figure 6.7.4

The screenshot shows a web browser window with the URL 127.0.0.1:5000/index. The title bar reads "CanTalk Real Estate Price Prediction System". The main content area is titled "Welcome to CanTalk House Price Predictor". It contains the following input fields:

- Enter Bedrooms: Enter BHK
- Enter No of Bathrooms: Enter No of Bathrooms
- Enter Total Area: Enter Total Area
- Enter Property Type: Enter Property Type
- Enter City: Enter City
- Having Balcony?: Yes or No
- Enter Price per squarefit: Enter Price per Squarefit

A blue "Predict Price" button is at the bottom.

Figure 6.7.5

### Welcome to CanTalk House Price Predictor

Enter Bedrooms <input type="text" value="3"/>	Enter No of Bathrooms <input type="text" value="3"/>
Enter Total Area: <input type="text" value="1320"/>	Enter Property Type <input type="text" value="Flat"/>
Enter City <input type="text" value="Chennai"/>	Having Balcony? <input type="text" value="Yes"/>
Enter Price per squarefit <input type="text" value="7580"/>	
<input type="button" value="Predict Price"/>	

Figure 6.7.6

Predicted Price:  
Rs 9934878.84 Rs.

[Back to Home](#) lakhs

Figure 6.7.7

## CHEPTE 7: CONCLUSION

In wrapping up my price prediction project, it's evident that it holds considerable promise for future forecasting endeavours. By harnessing the power of advanced algorithms and meticulously scrutinizing historical data, I've successfully engineered a robust model adept at accurately delineating price trends. The culmination of my research not only yields actionable insights but also furnishes valuable resources for companies and investors navigating the complexities of market behaviour.

This venture represents more than just a static snapshot; it's a dynamic journey marked by ongoing refinement and evolution. As I continue to fine-tune my methodologies and incorporate fresh streams of data, the efficacy of my predictive capabilities will undoubtedly ascend. This iterative process not only fortifies the reliability of my model but also augments the discernment it provides to stakeholders seeking to make informed decisions in the financial landscape.

Moreover, the implications of this endeavour extend far beyond the confines of my immediate research. The insights garnered from this project hold the potential to catalyse paradigm shifts in how we approach market analysis and decision-making. By leveraging the symbiotic relationship between cutting-edge technology and historical market data, we can forge a path toward a more nuanced understanding of market dynamics.

In essence, my price prediction project encapsulates the fusion of data-driven ingenuity and strategic foresight. It serves as a beacon illuminating the possibilities that lie ahead in the realm of financial forecasting. As I embark on the next phase of this journey, I remain steadfast in my commitment to pushing the boundaries of predictive analytics and empowering stakeholders with the knowledge they need to navigate the ever-changing tides of the market.

## CHAPTER 8: REFERENCES

<https://www.kaggle.com>

[www.github.com](https://www.github.com)

<https://seaborn.pydata.org>

<https://www.python.org>

<https://www.w3schools.com>

<https://stackoverflow.com>