VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A FINAL PROJECT REPORT

ON

"House Price Prediction using ML"

Submitted in partial fulfillment of the requirements for the for the award of

degree of Bachelor of Engineering in

COMPUTER SCIENCE & ENGINEERING

Submitted By

MILLEE SAMUKCHAM	**1MJ17CS731**
M BHAVANA	**1MJ17CS725**
YELURU SREEJA REDDY	**1MJ17CS751**
RIYA CHOUDHARY	**1MJ17CS744**

Under the Guidance of

Dr.S.Umamaheswaran,

Associate Professor,

Department of Computer Science & Engineering

An Autonomous Institute
Approved by AICTE, New Delhi
Affiliated to VTU, Belagavi
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA & NAAC

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MVJ COLLEGE OF ENGINEERING
Whitefield, Near ITPB
BANGALORE-67
Academic Year 2020-21

MVJ COLLEGE OF ENGINEERING
Since 1982
Engineering A Better Tomorrow

An Autonomous Institute

MVJ COLLEGE OF ENGINEERING

Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that phase II of the project work, entitled “**House Price Prediction Using ML”** is a bonafide work carried out by

MILLEE SAMUKCHAM	**1MJ17CS731**
M BHAVANA	**1MJ17CS725**
YELURU SREEJA REDDY	**1MJ17CS751**
RIYA CHOUDHARY	**1MJ17CS744**

in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2020-21. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements.

Signature of Internal Examiner	**Signature of HOD**	**Signature of Principal**
(Dr.Ramya.C)	**(Mrs. Tamilarasi R)**	**(Dr.P.Mahabaleswarappa)**

MVJ COLLEGE OF ENGINEERING
Since 1982
Engineering A Better Tomorrow

An Autonomous Institute

MVJ COLLEGE OF ENGINEERING

Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Millee Samukcham ,M Bhavana ,Yeluru Sreeja Reddy and Riya Choudhary** hereby declare that the entire work titled "**HOUSE PRICE PREDICTION USING ML"** embodied in this project report has been carried out by us during the 8th semester of **Bachelor of Engineering degree at MVJCE, Bangalore** under the esteemed guidance of Dr. Ramya.C, (AP, Dept. of CSE, MVJCE) affiliated to **Visvesvaraya Technological University, Belagavi**. The work embodied in this dissertation work is original and it has not been submitted in part of full for any other degree in any University.

1.MILLEE SAMUKCHAM	**1MJ17CS731**
2. M BHAVANA	**1MJ17CS725**
3. YELURU SREEJA REDDY	**1MJ17CS751**
4. RIYA CHOUDHARY	**1MJ17CS744**

Place: Bangalore

Date:25th June,2021

ACKNOWLEDGEMENT

It gives me a great pleasure to acknowledge with thanks the assistance and contribution of many individuals who have been actively involved at various stages of this project to make it a success.

Firstly, I am grateful to this esteemed institution MVJ College of Engineering for providing us an opportunity to pursue our degree course.

I thank our Principal, Dr. P.MAHABALESHWARAPPA and Vice Principal, M. BRINDHA for their constant support and guidance.

I wish to place on record my grateful thanks to our beloved Mrs. TAMILARASI.R, HOD/CSE for her constant encouragement and all the support provided during this project work.

I consider it a privilege and honour to express my sincere gratitude to my guide Dr. S. Umamaheswaran, Associate Professor, Department of Computer Science & Engineering for his valuable guidance throughout the tenure of this mini project work, and whose support and encouragement made this work possible.

I would also like to sincerely thank all our teaching and non-teaching staff, and my family for all their love and encourage.

Thanking you,

MILLEE SAMUKCHAM	**(1MJ17CS731)**
M BHAVANA	**(1MJ17CS725)**
YELURU SREEJA REDDY	**(1MJ17CS751)**
RIYA CHOUDHARY	**(1MJ17CS744)**

ABSTRACT

With the rise in industrialization ,people are also more careful nowadays when they are trying to buy a new house with their budgets and market strategies. Till date ,existing websites present only the house prices given by the owners and details of the house mostly infrastructure. Some websites even provide comparision between different houses with the same infrastructure. But, some people are not aware of how much a house with a certain infrastructure is supposed to cost and are not able to detect how much is good enough to be able to detect frauds. People also need other factors other than infrastructure to decide whether or not to buy a house Machine learning algorithm helps us in enhancing security alerts, ensuring public safety and improve medical enhancements.

We utilize logistic regression as our model because of its adaptable and probabilistic methodology on model selection. Our result exhibit that our approach of the issue need to be successful, and has the ability to process predictions that would be comparative with other house cost prediction models. More over on other hand housing value indices, the advancement of a housing cost prediction that tend to the advancement of real estate policies schemes. This study utilizes machine learning algorithms as a research method that develops housing price prediction models.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	Java Script
SQL	Structured Query Language
DFD	Data flow diagram

Chapter 1 INTRODUCTION

Data is at the heart of technical innovations, achieving any result is now possible using predictive models. Machine learning is extensively used in this approach. Machine learning means providing valid dataset and further on predictions are based on that, the machine itself learns how much importance a particular event may have on the entire system based on its pre-loaded data and accordingly predicts the result. Various modern applications of this technique include predicting stock prices, predicting the possibility of an earthquake, predicting company sales and the list has endless possibilities. For our research project, we have considered Mumbai as our primary location and are predicting real-time house prices for various localities in and around Mumbai. We have used parameters like 'square feet area', 'no. of Bedrooms', 'No of Bathrooms', 'Type of Flooring', 'Lift availability' ,'Parking availability' ,'Furnishing condition'. We have taken into account a verified dataset with diversity so as give accurate results for all conditions. We have used various algorithms explained below in various combinations and the weight for each algorithm is given based on the accuracy percentage. After evaluating for various test runs we conclude that instead of an individual algorithm a series of algorithm yields better results.

What is machine learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.1 Supervised Learning

Supervised machine learning algorithms can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Supervised Learning has input variables(x) and an output variable(y) and uses an algorithm to learn the mapping function from the input to the output. Supervised learning is the task of learning a function that maps an input to an output, based on example input output pairs. It infers a function from labeled training set consisting of a set of training examples. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

1.2 Unsupervised Learning

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Unsupervised Learning is the training of a model using information that is neither classified nor labeled. This model can be used to cluster the input data in classes on the basis of their statistical properties. Unsupervised machine learning is the task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations). Since the examples given to the learner are unlabeled, there is no evaluation of the accuracy of the structure that is output by the relevant algorithm—which is one way of distinguishing unsupervised learning from supervised learning and reinforcement learning.

A central case of unsupervised learning is the problem of density estimation in statistics, though unsupervised learning encompasses many other problems involving summarizing and explaining key features of the data. The classical example of unsupervised learning in the study of both natural and artificial neural networks is subsumed by Donald Hebb's principle, that is, neurons that fire together wire together.

Unsupervised learning occurs when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values as. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

1.3 Project Overview

This project contains two modules:

1) Admin Module
2) User Module

Admin Module:

- Admin can view his profile and edit his profile details.
- He can also view the users and delete any of them.
- He can upload the dataset and view it.
- He can change his password and logout.

User Module:

- A user can login and view his/her details. He/she can also edit the profile details.
- A user can check for the city, area, and type of house he/she is looking for and results will be displayed for the same.

- A user can also check for house prediction based on the city, area, and budget he has.
- Finally, he can check for house prediction based on entire city.
- He/she has the ability to change the password. And can logout from his profile.

1.4 System Development Environment

1.4.1 HTML

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML. Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags:

<! -- -->	specifies comments
<A>……….</A>	Creates hypertext links
<B>……….</B>	Formats text as bold
<BIG>……….</BIG>	Formats text in large font.
<BODY>…</BODY>	Contains all tags and text in the HTML document
<CENTER>...</CENTER>	Creates text
<DD>…</DD>	Definition of a term
<DL>...</DL>	Creates definition list
<FONT>…</FONT>	Formats text with a particular font
<FORM>...</FORM>	Encloses a fill-out form
<FRAME>...</FRAME>	Defines a particular frame in a set of frames
<H#>…</H#>	Creates heading of different levels(1 – 6)
<HEAD>...</HEAD>	Contains tags that specify information about a document
<HR>...</HR>	Creates a horizontal rule
<HTML>…</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>…</SCRIPT>	Contains client-side or server-side script
<TABLE>…</TABLE>	Creates a table
<TD>…</TD>	Indicates table data in a table
<TR>…</TR>	Designates a table row
<TH>…</TH>	Creates a heading in a table

Attributes

The attributes of an element are name-value pairs, separated by "=", and written within the start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML).Leaving attribute values unquoted is considered unsafe.Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: lang and dir.

The id attribute provides a document-wide unique identifier for an element. This can be used by style sheets to provide presentational properties, by browsers to focus attention on the specific element or by scripts to alter the contents or presentation of an element. The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document (or a set of documents) may use the designation class="notation" to indicate that all elements with this class value are all subordinate to the main text of the document (or documents). Such notation classes of elements might be gathered together and presented as footnotes on a page, rather than appearing in the place where they appear in the source HTML.

An author may use the style non-attributal codes presentational properties to a particular element. It is considered better practice to use an element’s son- id page and select the element with a style sheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties. The title is used to attach sub textual explanation to an element. In most browsers this title attribute is displayed as what is often referred to as a tool tip. The generic inline span element can be used to demonstrate these various non-attributes.The preceding displays as HTML (pointing the cursor at the abbreviation should display the title text in most browsers).

Advantages

- A HTML document is small and hence easy to send over the net.
- It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

1.4.2 Javascript

What is JAVASCRIPT? JavaScript, originally supported by Netscape Navigator, is the most popular Web scripting language today. JavaScript lets you embed programs right in your Web pages and run these programs using the Web browser. You place these programs in a <SCRIPT> element. If you want the script to write directly to the Web page, place it in the <BODY> element.

EX:

```
 <HTML>
     <HEAD>
           <TITLE></TITLE>
     </HEAD>
     <BODY>
        <SCRIPT LANGUAGE=”JavaScript”>
        </SCRIPT>
     </BODY>
 </HTML>
```

JAVASCRIPTS OBJECTS

JavaScript is an object-oriented language. JavaScript comes with a number of predefined objects.

Objects of the JavaScript

1. **Document**: Corresponds to the current Web page’s body. Using this object, you have access to the HTML of the page itself, including the all links, images and anchors in it.

2. **Form**: Holds information about HTML forms in the current page.

3. **Frame**: Refers to a frame in the browser’s window.

4. **History**: Holds the records of sites the Web browser has visited before reaching the current page.

5. **Location**: Holds information about the location of the current web page.

6. **Navigator**: Refers to the browser itself, letting you determine what browser the user has.

7. **Window**: Refers to the current browser window.

JAVASCRIPTS EVENTS

Some of the events of JavaScript

1. on Change: Occurs when data in a control, like a text field, changes.

2. on Click: Occurs when an element is clicked.

3. on Focus: Occurs when an element gets the focus.

4. on Mouse Down: Occurs when a mouse button goes down.

5. on Reset: Occurs when the user clicks the reset button.

JAVASCRIPTS FUNCTIONS

Declaration of function Syntax:

```
function function_name () { … … }
```

1.4.3 RDBMS CONCEPTS

1. DATA ABSTRACTION

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. However in order for the system to be usable, data must be retrieved efficiently. The efficiency lead to the design of complex data structure for the representation of data in the database. Certain complexity must be hidden from the database system users. This accomplished by defining several levels of abstraction at which the database may be viewed.

2. CLASSIFICATION OF DATABASE

There are 3 types of database approaches given below,

a. Hierarchical Database: In this type of model data is represented in simple tree structured. The record at the top of three is known as root, the root may have any number of dependents. Each of these may have any number of low level dependents and so on up to any number of levels. The disadvantages of the approach are that no independent record occurrence can exist without it's superior.

b. Network Database: In a Network database, data is represented by Network structure. In this approach record occurrence can have any number of superiors as well as any number of immediate dependents thus allow many to many correspondence directly than an hierarchical approach. The main disadvantage of the Network model is data representation is very complex resulting in complexity of the DML (Data Manipulation Language).

c. Relational Database: The Relational model represents data and relationships among data by a collection of tables each of which has a number of columns with unique names. 1.4.4 THE SQL LANGUAGE SQL is a language for relational database. SQL is a non-procedural i.e., when we use SQL we specify what we want to be done not how to do it.

1.4.4 THE SQL LANGUAGE

SQL is a language for relational database. SQL is a non-procedural i.e., when we use SQL we specify what we want to be done not how to do it.

Features of SQL

1. SQL is an interactive query language.

2. SQL is a database administration language.

3. SQL is a database programming language.

4. SQL is a client/server language.

5. SQL is a distributed database language.

6. SQL is a database gateway language.

Basic SQL Commands

1. Data Definition Language commands (DDL)
2. Data Manipulation Language commands (DML)
3. Transaction Control Language commands (TCL)
4. Data control Language commands (DCL)

1.4.5 JAVA

Initially the language was called as "*oak*" but it was renamed as "*Java*" in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer's language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.
- Finally, Java is to Internet programming where C was to system programming.

IMPORTANCE OF JAVA TO THE INTERNET

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

JAVA CAN BE USED TO CREATE TWO TYPES OF PROGRAMS

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java's ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java – compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

FEATURES OF JAVA SECURITY

Every time you that you download a "normal" program you are risking a viral infection. Prior to java, most users did not download executable programs frequently, and those who did scan them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a "firewall" between a network application and your computer.

When you use a java-compatible web browser, you can safely download java applets without fear of virus infection or malicious intent.

PORTABILITY

For programs to be dynamically downloaded to all the various types of platforms connected to the internet, some means of generating portable executable code is needed .as you will see, the same mechanism that helps ensure security also helps create portability. Indeed, java's solution to these two problems is both elegant and efficient.

THE BYTE CODE

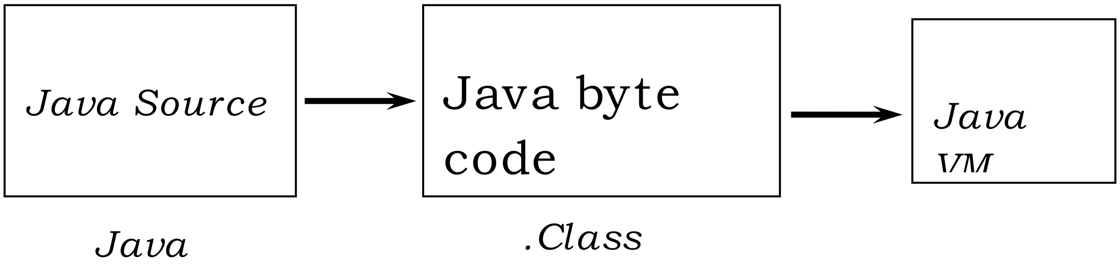
The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece – by – piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

Java Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

Overall Description



Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a. class file, which contains the byte code. The .Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of code

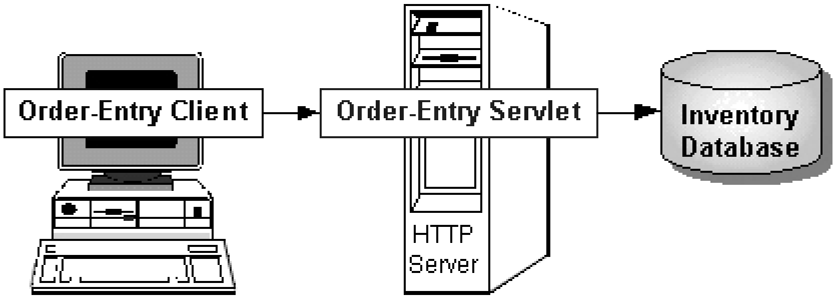
When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

1.4.6 Introduction to Servlets

Servlets provide a Java(TM)-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side -- object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform-independent, dynamically loadable, plug gable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

What is a Servlet?

Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.



Servlets are to servers what applets are to browsers. Unlike applets, however, Servlets have no graphical user interface. Servlets can be embedded in many different servers because the servlet API, which you use to write Servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support the Servlet API.

Use Servlets instead of CGI Scripts

- Servlets are an effective replacement for CGI scripts. They provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension.
- So use Servlets to handle HTTP client requests. For example, have Servlets process data posted over HTTPS using an HTML form, including purchase order or credit card data. A servlet like this could be

part of an order-entry and processing system, working with product and inventory databases, and perhaps an on-line payment system.

Other Uses for Servlets

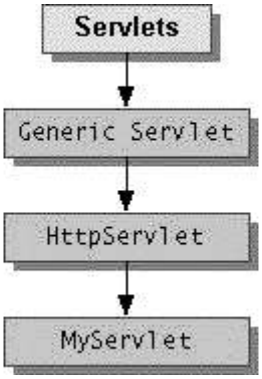
Here are a few more of the many applications for Servlets

- Allowing collaboration between people. A Servlet can handle multiple requests concurrently, and can synchronize requests. This allows Servlets to support systems such as on-line conferencing.

- Forwarding requests. Servlets can forward requests to other servers and Servlets. Thus Servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service over several servers, according to task type or organizational boundaries.
- Architecture of the Servlet Package
- The javax.servlet package provides interfaces and classes for writing Servlets. The architecture of the package is described below.

The Servlet Interface

- The central abstraction in the Servlet API is the Servlet interface. All Servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet.



- The Servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

Client Interaction

- When a servlet accepts a call from a client, it receives two objects:
- A ServletRequest , which encapsulates the communication from the client to the server.
- A ServletResponse , which encapsulates the communication from the servlet back to the client.
- ServletRequest and ServletResponse are interfaces defined by the javax.servlet package.

The ServletRequest Interface

- The ServletRequest interface allows the servlet access to: Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that receive the input stream, ServletInputStream. Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and PUT methods.

Interfaces that extend ServletRequest interface allow the servlet to retrieve more protocol-specific data. For example, the HttpServletRequest interface contains methods for accessing HTTP-specific header information.

THE SERVLETRESPONSE INTERFACE

The ServletResponse interface gives the servlet methods for replying to the client. It

- Allows the servlet to set the content length and MIME type of the reply.
- Provides an output stream, ServletOutputStream, and a Writer through which the servlet can send the reply data.

Interfaces that extend the ServletResponse interface give the servlet more protocol-specific capabilities. For example, the HttpServletResponse interface contains methods that allow the servlet to manipulate HTTP-specific header information.

Additional Capabilities of HTTP Servlets

The classes and interfaces described above make up a basic Servlet. HTTP Servlets have some additional objects that provide session-tracking capabilities. The servlet writer can use these APIs to maintain state between the servlet and the client that persists across multiple connections during some time period. HTTP Servlets also have objects that provide cookies. The servlet writer uses the cookie API to save data with the client and to retrieve this data.

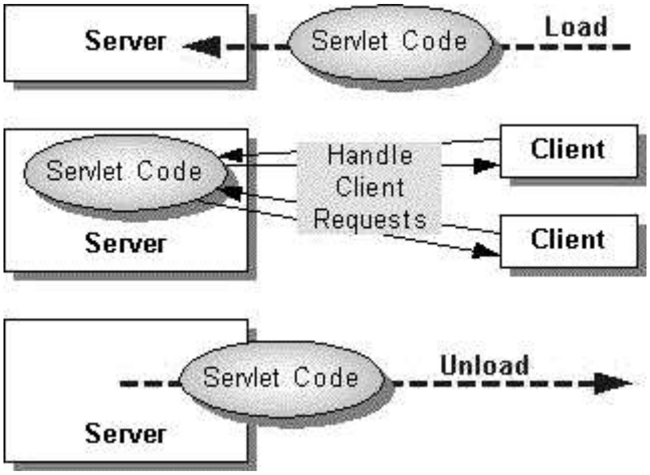
The classes mentioned in the Architecture of the Servlet Package section are shown in the example in bold:

- SimpleServlet extends the HttpServlet class, which implements the Servlet interface.
- SimpleServlet overrides the doGet method in the HttpServlet class. The doGet method is called when a client makes a GET request (the default HTTP request method) and results in the simple HTML page being returned to the client.
- Within the doGet method,
  A HttpServletRequest object represents the user's request.
  - An HttpServletResponse object represents the response to the user.
  - Because text data is returned to the client, the reply is sent using the Writer object obtained from the HttpServletResponse object.

Servlet Lifecycle

Each servlet has the same life cycle

- A server loads and initializes the servlet
- The servlet handles zero or more client requests
- The server removes the servlet



Initializing a Servlet

When a server loads a servlet, the server runs the servlet's init method. Initialization completes before client requests are handled and before the servlet is destroyed.

Even though most Servlets are run in multi-threaded servers, Servlets have no concurrency issues during servlet initialization. The server calls the init method once, when the server loads the servlet, and will not call the init method again unless the server is reloading the servlet. The server cannot reload a servlet until after the server has destroyed the servlet by calling the destroy method.

The init Method:

The init method provided by the HttpServlet class initializes the servlet and logs the initialization. To do initialization specific to your servlet, override the init () method following these rules:

If an initialization error occurs that renders the servlet incapable of handling client requests, throw an Unavailable Exception.

Initialization Parameters:

The second version of the init method calls the getInitParameter method. This method takes the parameter name as an argument and returns a String representation of the parameter's value.

The specification of initialization parameters is server-specific. In the Java Web Server, the parameters are specified with a servlet is added then configured in the Administration Tool. For an explanation of the

Administration screen where this setup is performed, see the Administration Tool: Adding Servlets online help document.In some cases, if we need to get the parameter names, we can use the getParameterNames method.

Destroying a Servlet:

Servlets run until the server is destroys them, for example at the request of a system administrator. When a server destroys a servlet, the server runs the servlet's destroy method. The method is run once; the server will not run that servlet again until after the server reloads and reinitializes the servlet.

When the destroy method runs, another thread might be running a service request. The Handling Service Threads at Servlet Termination section shows you how to provide a clean shutdown when there could be long-running threads still running service requests.

Using the Destroy Method:

The destroy method provided by the HttpServlet class destroys the servlet and logs the destruction. To destroy any resources specific to your servlet, override the destroy method. The destroy method should undo any initialization work and synchronize persistent state with the current in-memory state.

A server calls the destroy method after all service calls have been completed, or a server-specific number of seconds have passed, whichever comes first. If your servlet handles any long-running operations, service methods might still be running when the server calls the destroy method. You are responsible for making sure those threads complete. The next section shows you how.

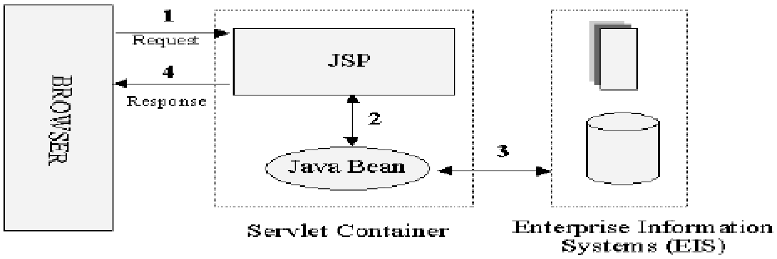
The destroy method shown above expects all client interactions to be completed when the destroy method is called, because the servlet has no long-running operations.

Java Server Page

Java Server Pages technology lets you put snippets of servlet code directly into a text-based document. A JSP page is a text-based document that contains two types of text: static template data, which can be expressed in any text-based format such as HTML, WML, and XML, and JSP elements, which determine how the page constructs dynamic content.

Java Server Page™ (JSP): An extensible Web technology that uses template data, custom elements, scripting languages, and server-side Java objects to return dynamic content to a client. Typically the template data is HTML or XML elements, and in many cases the client is a Web browser.

According to JSP model1 we can develop the application as,



According to above model the presentation logic has to be implemented in JSP page and the business logic has to be implemented as part of Java bean This model help us in separating the presentation and business logic. For large-scale projects instead of using model1 it is better to use model2 (MVC). Struts framework is based on model 2.

Java Server Pages (JSP) lets you separate the dynamic part of your pages from the static HTML. You simply write the regular HTML in the normal manner, using whatever Web-page-building tools you normally use. You then enclose the code for the dynamic parts in special tags, most of which start with "<%" and end with "%>". For example, here is a section of a JSP page that results in something like "Thanks for ordering Core Web Programming

For URL of

http://host/OrderConfirmation.jsp?title=Core+Web+Programming:

Thanks for ordering

<I><%= request.getParameter("title") %></I>

We normally give your file a .jsp extension, and typically install it in any place you could place a normal Web page. Although what you write often looks more like a regular HTML file than a servlet, behind the scenes, the JSP page just gets converted to a normal servlet, with the static HTML simply being printed to the output stream associated with the servlet's service method. This is normally done the first time the page is requested, and developers can simply request the page themselves when first installing it if they want to be sure that the first real user doesn't get a momentary delay when the JSP page is translated to a servlet and the servlet is compiled and loaded. Note also that many Web servers let you define define aliases that so that a URL that appears to reference an HTML file really points to a servlet or JSP page.

Aside from the regular HTML, there are three main types of JSP constructs that you embed in a page: scripting elements, directives, and actions. Scripting elements let you specify Java code that will become part of the resultant servlet, directives let you control the overall structure of the servlet, and actions let you specify existing components that should be used, and otherwise control the behavior of the JSP engine. To simplify the scripting elements, you have access to a number of predefined variables such as request in the snippet above.

Chapter 2 LITERATURE SURVEY

2.1 Literature Survey-1

TITLE	House Price Prediction Using Machine Learning BY - G. Naga Satish, Ch. V.Raghavendran, M.D.Sugnana Rao , Ch.Srinivasulu
YEAR	2018
DESCRIPTION	• Authors uses Linear Regression ,Multiple Regression analysis , lasso regression technique and gradient boosting regression each on the training dataset to train the model individually • Then print the predicted prices got from the best algorithm for a given test constraint. • They have managed to prepare a model that gives users for a novel best approach that will take a gander at future value predictions.
ISSUES	• A worry for their system is that preparation time is longer • Need to use various processors and parallel the computations sequentially to decrease the preparation time.

2.2 Literature survey-2

TITLE	PropTech for Proactive Pricing of Houses in Classified Advertisements in the Indian Real Estate Market BY - Sayan Putatunda
YEAR	Not specified
DESCRIPTION	• Author proposed a system for proactive pricing of houses in classified advertisements in the Indian real estate market using PropTech. • He aims to resolve the problems in regression technique. • They also used advanced algorithms such as Random Fore stand gradient boosting. • They also experimented with Artificial neural networks with different hyper-parameters. And their study shows that Random forest method is the best performer in terms of prediction accuracy
ISSUES	• Since this was a proof of concept (POC), so the models were implemented on a smaller dataset. However, the error margins can be reduced further if we use much larger datasets, which we aim to work on in the future.

2.3 Literature Survey-3

TITLE	Housing Prices Prediction With a Deep Learning and Random Forest Ensemble BY - Bruno Klaus de Aquino Afonso, Luckeciano Carvalho Melo,William Dihanster Gomes de Oliveira,Samuel Bruno da Silva Sousa,Lilian Berton
YEAR	2019
DESCRIPTION	• Their work uses the application of two different methods for insight into the data combined with powerful ML algorithms and then combined them into a final prediction. • They used two architectures (enriched RF and KISS) separately for data enrichment. • It showed that the Enriched RF works well with numeric features as it can derive rules not only depending on the value of an attribute but also on its presence or absence. • KISS, on the other hand can represent all kinds of data through the embedding layers. • They concluded that combining the two methods yielded the best result.
ISSUES	• Enriched RF used in this model couldn’t handle image or text data.

2.4 Literature Survey-4

TITLE	House Price Prediction Using Machine Learning and RPA BY - Prof.Pradnya Patil,Darshil Shah,Harshad Rajput,Jay Chheda
YEAR	2020
DESCRIPTION	• Their proposed system uses data scraping with the use of UIPath Studio Platform to develop RPA flowchart. • After the data is extracted, they perform data cleaning and then apply various algorithms. • They conclude that CatBoost is a clear winner in terms of accuracy for dataset that contains both categorical as well as numerical features.
ISSUES	• They claim that Random Forest quickly reaches some extent where more samples won’t improve the accuracy

Chapter 3 System Requirement

3.1 Hardware Requirement

- System : Pentium IV 2.4 Ghz
- Ram : 4GB
- Hard disk : 500GB

Any desktop/laptop system with above configuration or higher level

3.2 Software Requirements

- Operating system : Windows XP / 7
- Coding Language : Java (Jdk 1.7)
- Web Technology : Servlet, JSP
- Web Server : TomCAT 7.0
- IDE : Eclipse Galileo
- Database : My-SQL 5.0
- UGI for DB : SQLyog
- JDBC Connection : Type 4 - Native Driv**e**

Chapter 4 OBJECTIVE & METHODOLOGY

4.1 Objective

- To develop a model which predicts the property cost for a customer according to his\her interests.
- To predict the house price.
- To predict the selling prices of houses based on various factors other than infrastructures.

4.2 Existing System

- The real estate industry has become a competitive and nontransparent industry.
- The data mining process in such an industry provides an advantage to the developers by processing those data, forecasting future trends and thus assisting them to make favorable knowledge-driven decisions.
- The existing will build a predictive model to predict house price (price is a number from some defined range, so it will be regression task). For example, you want to sell a house and you don’t know the price which you can take—it can’t be too low or too high.
- To find house price you usually try to find similar properties in your neighborhood and based on gathered data you will try to assess your house price.

4.3 Disadvantages of The Existing System

- Prices of real estate properties are actually linked with our economy.
- But we are not provided with the tools or accurate measures of house prices even with the vast amount of data available.
- It can be extremely difficult to figure out the right set of attributes that are contributing to understanding the buyer’s behaviour. Hence, our proposed system aims to use machine learning to predict the selling prices of houses based on various factors other than infrastructures.

4.4 Proposed System

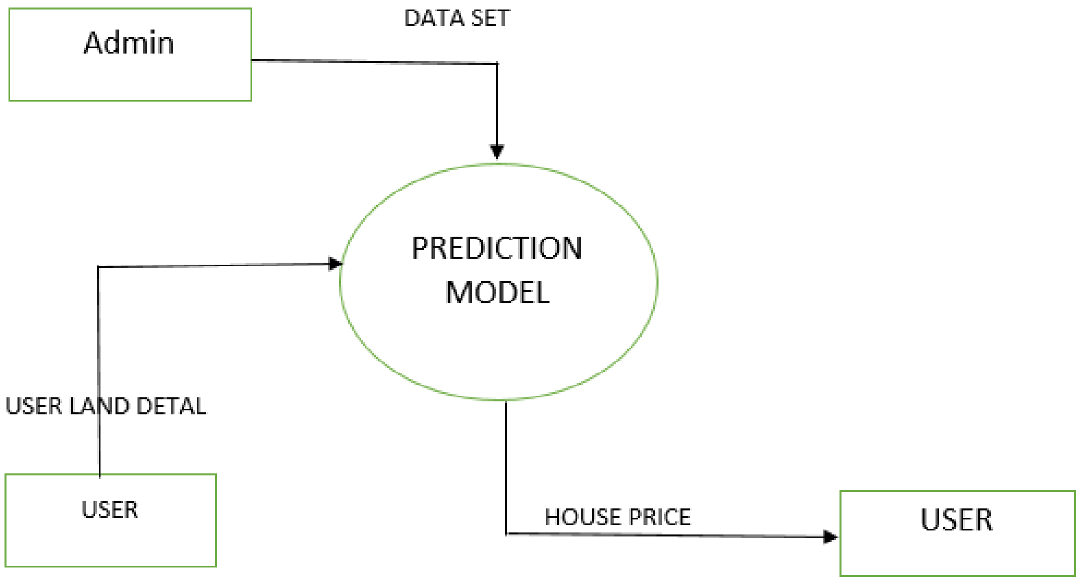
- Our main focus here is to develop a model which predicts the property cost for a customer according to his\her interests. Our model analyses a set of parameters selected by the customer so as to find an ideal price according to their requirements and interest.
- It uses a classical technique called logistic regression and collaborative filtering for prediction and tries to give an analysis of the results obtained.
- Our dataset comprises of various essential parameters and data mining has been at the root of our system. We initially cleaned up our entire dataset and also truncated the outlier values.
- Further, we weighed each parameter based on its importance in determining the pricing of the system and this led us to increase the value that each parameter withholds in the system.
- We shortlisted 3 different machine learning algorithms and tested our system with different combinations that can guarantee best possibly reliability of our results. Even after that, we followed a unique approach to increase the accuracy, our survey led to a conclusion that the actual real estate value also depends on nearby local amenities such as railway station, supermarket, school, hospital, temple, parks etc.
- And now we propose our unique approach that can counter this need. Now if we find any such public places in the circle, we increase the value of the property correspondingly. We carried this out with manual examples and this gave us tremendous results in terms of accuracy in prediction.

4.5 Advantages

- To provide an accurate prediction of housing.
- The system makes optimal use of Logistic Regression, Collaborative filtering.
- The efficiency of the algorithm has been further increased with use of Neural networks.

Chapter 5 SYSTEM DESIGN

5.1 Context Diagram

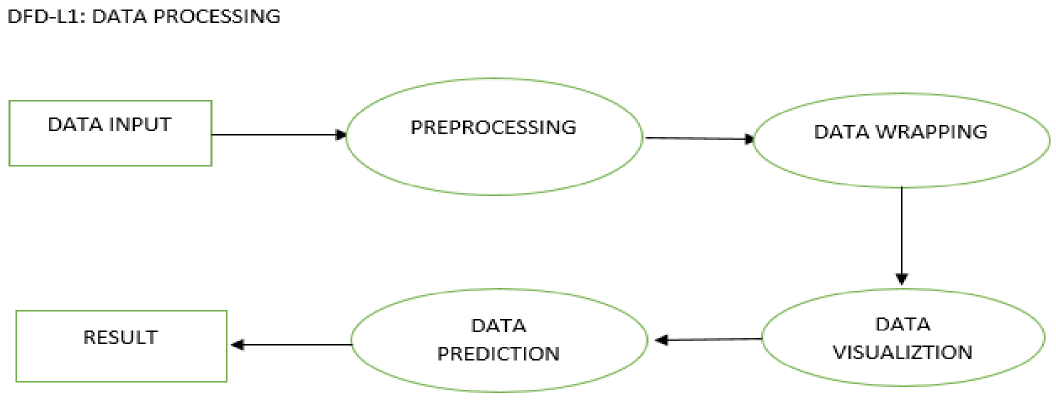


- Admin module provides the dataset to the prediction model.

- User module provides the required features/parameters as per his interest to the prediction model.

- The prediction model takes both as input, predicts and generates the price of the house, which will be provided to the user module.

5.3 DFD-Level 1: Data Processing



- **Input**: Raw data is provided as input to the Preprocessing Model

- **Preprocessing**: At this stage, the raw data is cleaned using various data cleaning techniques to remove null values, to fill missing values, etc., and the output is given for data wrapping.

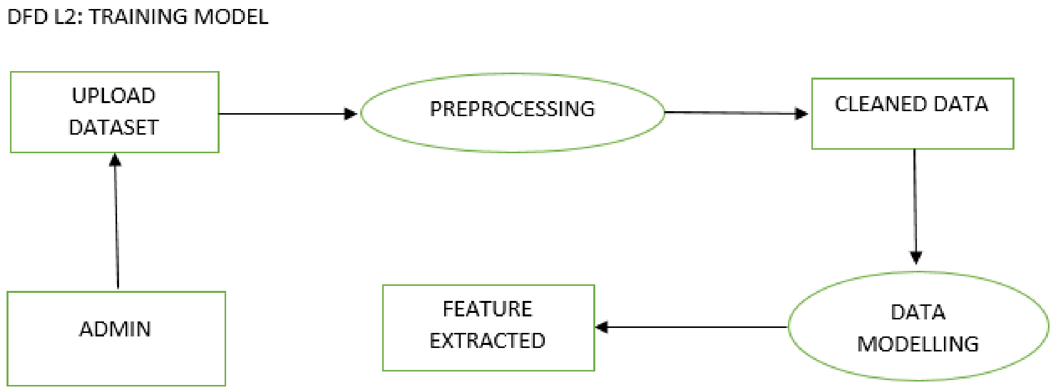
- **Data Wrapping**: At this stage, the system finds what are the attributes that contribute to the output, the higher the contribution, higher will be the weightage provided for the attribute during prediction.

- **Data Visualization**: The data is visualized using plotting techniques, to analyze the dataset.

- **Data prediction**: Prediction of output and analyzing the dataset.

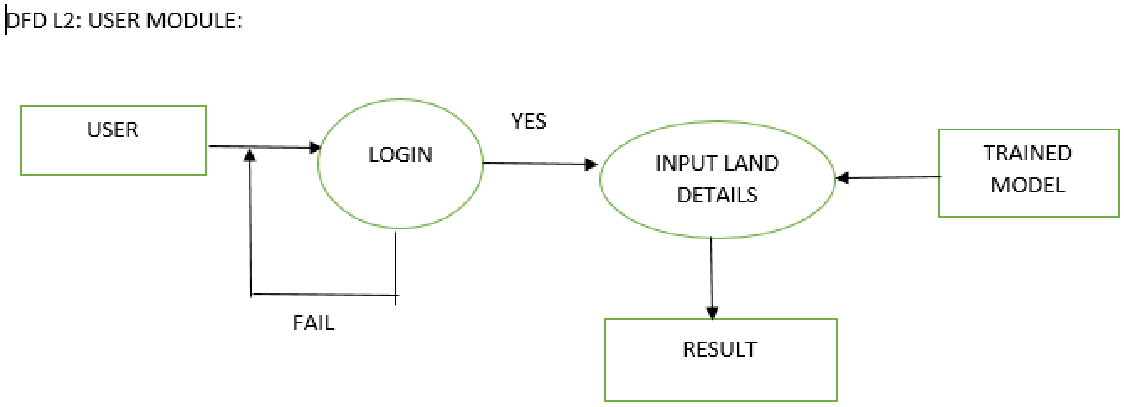
- **Result**: It is the cleaned and efficient data obtained after processing the raw data.

5.4 DFD-Level 2A: Training Model



- Admin uploads the dataset to the Preposeeing Model(DFD Level1)
- The obtained cleaned data is provided for Data Modelling,after which the required features are extracted, which actually effects the prediction value of the system.

5.5 DFD-Level 2B: User Module



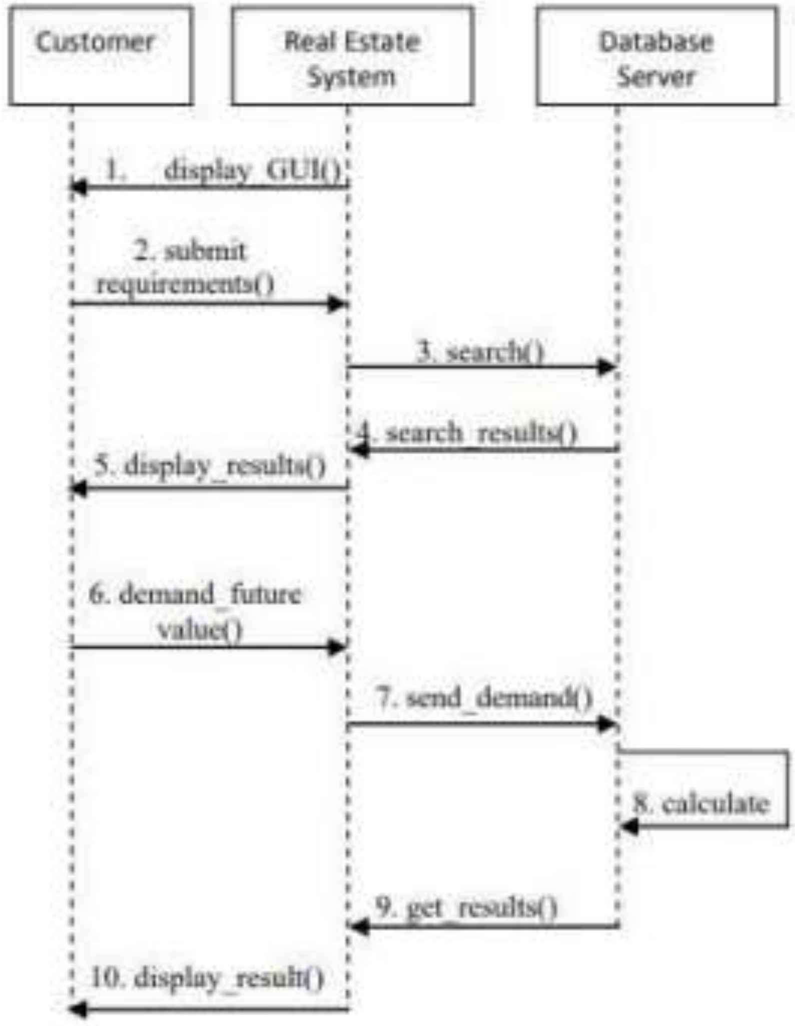
- User provides credentials to the system through the UI provided.

- If the login credentials match, then the login is successful and the user enters into the system.

- If not, user will be prompted with the same login page.

- The user provides the required features, and along with the trained model the result(house price) is predicted by the system and displayed to the user.

5.6 Sequence Diagram



1. The real estate system displays the GUI to the user for taking required inputs to authorize

2. The customer enters the required details for authentication and submits it to the system

3. The real estate system searches for the details entered if it is present in the database

4.The search result is given to the real estate system by the database

5. If the user exists the user is forwarded to the next appropriate page else displays an invalid message by the GUI to the user

6. Now the user once authenticated demands for future values to the system

7.The system then forwards this request to the database

8. The database does the required calculation for the request

9. The results is then forwarded to the system by the database

10. The result is displayed by the system to the user

Chapter 6 IMPLEMENTATION

6.1 Source Code

Adminhome.jsp:

```
<html>
<%
        String admin=request.getAttribute("admin").toString();
%>
<script language="JavaScript">
javascript:window.history.forward(-1);
</script>

<head>
        <link href="<%=request.getContextPath() %>/Files/CSS/style.css" rel="stylesheet" type="text/css" />
        <link rel="stylesheet" href="<%=request.getContextPath() %>/Files/CSS/button.css" type="text/css"/>
        <link href="<%=request.getContextPath()%>/Files/CSS/styles.css" rel="stylesheet" type="text/css" />
        <link href="<%=request.getContextPath()%>/Files/CSS/logins.css" rel="stylesheet" type="text/css" />
        <link href="<%=request.getContextPath()%>/Files/CSS/popup.css" rel="stylesheet" type="text/css" />
        <script src="<%=request.getContextPath()%>/Files/JS/jquery-1.6.4.min.js"
type="text/javascript"></script>

         <link href="//netdna.bootstrapcdn.com/font-awesome/3.2.1/css/font-awesome.css" rel="stylesheet">

         <link rel="stylesheet" href="css/style.css">

</head>
<body>
         <img src="<%=request.getContextPath() %>/Files/Images/1.png" width=1200px></img>
         <div style="position:absolute;top:140px;left:10px;">
         <!-- <font style="font-family: Monotype Corsiva; font-size: 30px; color:purple;'>Welcome <%=admin
%></font> -->
         <font style='font-family: Monotype Corsiva; font-size: 25px; color: purple;'>Welcome<%=admin
%></font> <br>

        <button id="expand-navigation">-</button>
       <section class="wrapper opened">
          <ul>
               <li><a href="<%=request.getContextPath() %>/AdminProfile?name=<%=admin %>"
target="afrm"><label style="font-size:12px" >Profile</label></a></li>
               <li><a href="<%=request.getContextPath() %>/UserList?submit=get" target="afrm"><label
style="font-size:12px" >User Details</label></a></li>
```

```
     <li><a href="<%=request.getContextPath() %>/Files/JSP/Admin/trans1.jsp" target="afrm"><label
style="font-size:12px">Upload Dataset</label></a></li>

    <li><a href="<%=request.getContextPath() %>/Files/JSP/Admin/viewdataset.jsp" target="afrm"><label
style="font-size:12px">View Dataset</label></a></li>

    <li><a href="<%=request.getContextPath() %>/index.jsp"><label style="font-size:12px">Sign
Out</label></a></li>
  </ul>
</section>

<div class="overlay on-overlay"></div>

        <script src="js/index.js"></script>



<%--    <font style="font-family: Monotype Corsiva; font-size: 30px; color:;">Welcome <%=admin %></font>
            <br><br>
            <a class="button_example" href="<%=request.getContextPath() %>/AdminProfile?name=<%=admin
%>" target="afrm">Show Profile&nbsp;&nbsp;</a><br></br>
            <!--  <a class="button_example" href="<%=request.getContextPath()%>/GroupList?submit=get"
target="afrm">Group List&nbsp;&nbsp;&nbsp;</a><br></br>-->
             <a class="button_example" href="<%=request.getContextPath() %>/UserList?submit=get"
target="afrm">User Details&nbsp;&nbsp;</a><br></br>
            <a class="button_example" href="<%=request.getContextPath() %>/CloudList?submit=get"
target="afrm">Cloud Details</a><br></br>
    <a class="button_example" href="<%=request.getContextPath() %>/HashDetails?submit=get"
target="afrm">Hash Details&nbsp;&nbsp;</a><br></br>
            <a class="button_example" href="<%=request.getContextPath() %>/Trans1?submit=get"
target="afrm">Transactions&nbsp;</a><br></br>
            <a class="button_example" href="<%=request.getContextPath() %>/index.jsp">&nbsp;&nbsp;Sign
Out&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</a> --%>
</div>
            <div style="position:absolute;top:140px;left:150px;">
                        <iframe frameborder="0" scrolling="auto" name="afrm" height="470" width="800" style;>
                        </iframe>
            </div>
</body>
            </html>
```

index.jsp:

```
<%@ page import="com.util.*"%>

<html>
<head>
<%
```

```
//int no=Utility.parse(request.getParameter("no"));
%>
<link href="<%=request.getContextPath()%>/Files/CSS/style.css"
        rel="stylesheet" type="text/css" />
<link href="<%=request.getContextPath()%>/Files/CSS/message.css"
        rel="stylesheet" type="text/css" />
<link rel="stylesheet"
        href="<%=request.getContextPath()%>/Files/CSS/login.css"
        type="text/css" />
<script type="text/javascript"
        src="<%=request.getContextPath()%>/Files/JS/style.js"></script>
<link href="<%=request.getContextPath()%>/Files/CSS/styles.css" rel="stylesheet" type="text/css" />
<link href="<%=request.getContextPath()%>/Files/CSS/logins.css" rel="stylesheet" type="text/css" />
<link href="<%=request.getContextPath()%>/Files/CSS/popup.css" rel="stylesheet" type="text/css" />
<script src="<%=request.getContextPath()%>/Files/JS/jquery-1.6.4.min.js" type="text/javascript"></script>
</head>

<body   onload="startTimer()" >
<div    style="position: centre;  left: -10px;" >
        <img src="<%=request.getContextPath()%>/Files/Images/1.png" width=1200px></img>
        <div class="TabMenu" id="label"     style="position: absolute;  left: 10px;"          >

                <a href="#overlay" id="button1"><span style="position: absolute; top: 110px; height:120;
width:120; background-color: BurlyWood ;">
                <br/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                <img src="<%=request.getContextPath() %>/Files/Images/admin5.jpg" align="center"
height="45" width="45"  />
                        &nbsp;&nbsp;<br/>&nbsp;&nbsp;&nbsp;&nbsp;
                        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font style="color:blue;"><b>Admin</b></font></a>
                </span>
                <a href="#overlay_signup" id="button1">
                   <span style="position: absolute; left: 1100px ;top:102px ; height:120; width:120;background-
color: BurlyWood ">
                   <br/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                  <img src="<%=request.getContextPath() %>/Files/Images/user2.jpg" height="45" width="45"
/><br/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                  <font style="color: blue;"><b>User</b></font>
                </a>
                </span>
                <!--  <a href="#overlay-tpa" id="button1">
               <span style="position: absolute; top: 310px; height:110; width:120;background-color: #B87070">
&nbsp;&nbsp;<br/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                 <img src="<%=request.getContextPath() %>/Files/Images/tpa.png" height="45" width="45" />
                        <br/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                         <font style="color: #62462B;"><b>Auditor</b></font></a>-->
                </span>
        </div>
```

```
	<!-- Admin Login Pop Up -->
	 <div id="overlay">

			<div id="popup" >
				<a href=""> <img class="close_button" src="<%=request.getContextPath()
%>/Files/Images/close1.jpg"height="50" width="50"  /></a>
<form class="login" action="<%=request.getContextPath()%>/AdminLogin" method="post">
<center>
<label><font style="font-family: Times New Roman; font-size: 30px; color: blue;">ADMIN
LOGIN</font></label></center>
<br> <label><font style="color: blue;">Admin Id</font></label>&nbsp;&nbsp;
<input type="text" tabindex="1" class="input"placeholder="Admin Id" name="name" required><br> <br>
<label><font style="color: blue;">Password</font></label>
<input type="password" class="input" tabindex="2" name="pass" placeholder="Password" required><br> <br>
					&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="image" id="" value="submit" src="<%=request.getContextPath()
%>/Files/Images/login6.jpg"alt="submit Button" class="gradientbuttons" width="80" height="80">
</form>
 </div>
 </div>
<!-- User Login Pop Up -->
<div id="overlay_signup">
  <div id="popup">
<a href=""><img class="close_button" src="<%=request.getContextPath()
%>/Files/Images/close1.jpg"height="50" width="50" /></a>
<form class="login" action="<%=request.getContextPath()%>/UserLogin" method="post">
<center><label><font style="font-family: Times New Roman; font-size: 30px; color: blue;">USER
LOGIN</font></label>
</center><br> <label><font style="color: blue;">User Id</font></label>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="text" tabindex="1" class="input" name="name" placeholder="User Id" required><br> <br>
<label><font style="color: blue;">Password</font></label><input type="password" class="input" tabindex="2"
name="pass" placeholder="Password" required><br> <br> <br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="image" id="" value="submit"
src="<%=request.getContextPath() %>/Files/Images/login6.jpg" alt="submit Button"class="gradientbuttons"
width="80" height="80">&nbsp;&nbsp;&nbsp;&nbsp;
</form>
</div>
</div>
<!-- Tpa Login Pop Up -->
<!--  <div id="overlay-tpa">
   <div id="popup">
	<a href=""> <img class="close_button" src="<%=request.getContextPath() %>/Files/Images/close.png"
height="50" width="50"/></a>
	<form class="login" action="<%=request.getContextPath()%>/AuditorLogin" method="post">
	<center><label><font style="font-family: Monotype Corsiva; font-size: 35px; color:
#DAA520;">Auditor's Login</font></label>
</center>
```

```
<br> <label><font style="color: #336633;">Auditor's Id</font></label>&nbsp;&nbsp;
<input type="text" tabindex="1" class="input" placeholder="Tpa Id" name="name" required><br> <br>
<label><font style="color: #336633;">Password</font></label> <input type="password" class="input"
tabindex="2" name="pass" placeholder="Password" required><br> <br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="image" id="" value="submit" src="<%=request.getContextPath() %>/Files/Images/login.png"
alt="submit Button" class="gradientbuttons" width="80" height="80">
</form>
</div></div>-->
<%int noo=Utility.parse(request.getParameter("no"));
if(noo==1)
{%>
<div class="error" id="message" style="height: 65; width: 250px; top:180" >
    <p>username and password mismatch</p>
</div>
<%}
%>
</body>
</html>
```

Adduser.jsp:

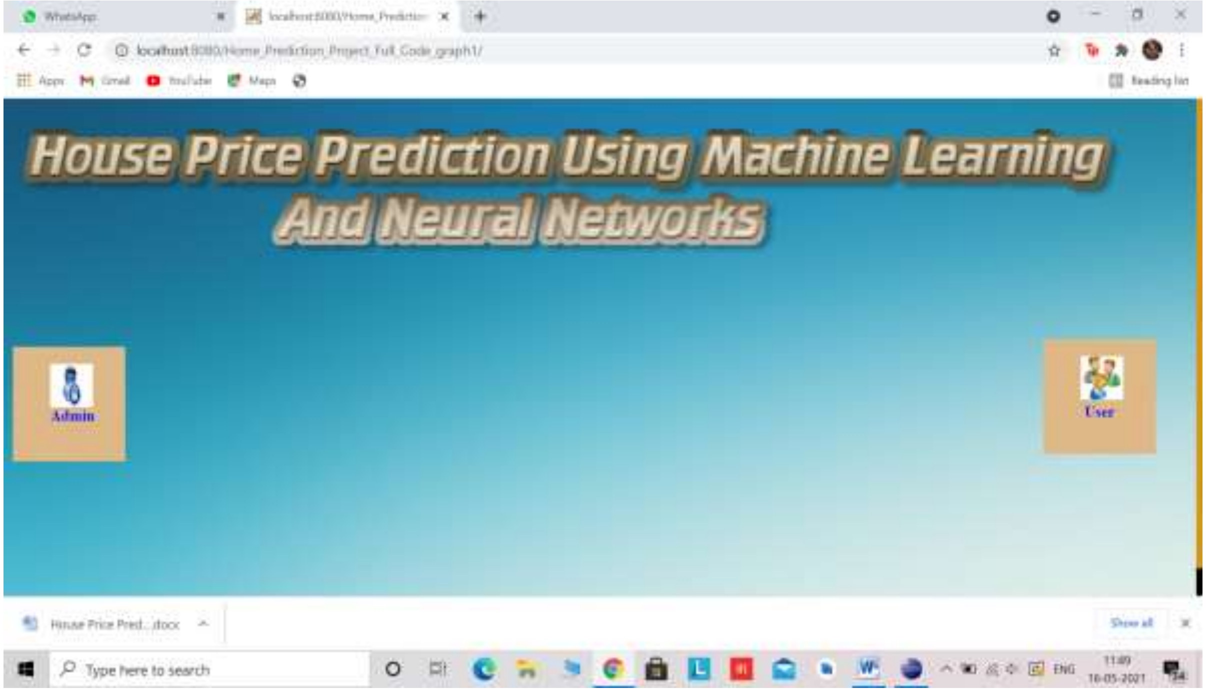
```
<%@ page import="com.util.*"%>
<%@ page import="java.sql.*"%>
<html>
<head>
<link href="<%=request.getContextPath() %>/Files/CSS/message.css"
        rel="stylesheet" type="text/css" />
<link rel="stylesheet"
        href="<%=request.getContextPath() %>/Files/CSS/login.css"
        type="text/css" />
<script type="text/javascript"
        src="<%=request.getContextPath() %>/Files/JS/style.js"></script>
<link href="<%=request.getContextPath()%>/Files/CSS/styles.css"
        rel="stylesheet" type="text/css" />
<link href="<%=request.getContextPath()%>/Files/CSS/logins.css"
        rel="stylesheet" type="text/css" />
<link href="<%=request.getContextPath()%>/Files/CSS/popup.css"
        rel="stylesheet" type="text/css" />
<script src="<%=request.getContextPath()%>/Files/JS/jquery-1.6.4.min.js"
        type="text/javascript"></script>
<style>
table, tr, td
th, td
{
padding:8px;
}
```

```
</style>
</head>
<%
   Class.forName("com.mysql.jdbc.Driver");
   Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sptheory","root", "admin");
   Statement st = con.createStatement();
   ResultSet rs=st.executeQuery("select * from userlogin");
   %>
<body onload="startTimer()">
   <!-- User Register Pop Up -->
      <div >
          <div id="popup" style="height: 400px; width: 600px; padding: 10px; background-color: #FFFFD1">
                <form action="<%=request.getContextPath() %>/UserList">
                       <input type="hidden" name="submit" value="Add"></input>
                        <input type="hidden" name="add1" value="YES"></input>
                        <div style="position: absolute; top: 30px; left: 50px;">
                       <center><label>
                          <font style="font-family: Monotype Corsiva; font-size: 35px; color:blue;">User
Registration</font>
</label></center><br></br>
<table style="width: 130">
  <tr>
          <td><label><font style="color: maroon;font-weight: bold;">UserId</font></label></td>
          <td><input type="text" tabindex="1" name="id" class="input" placeholder="User Id" required></td>
          <td> <label><font style="color: maroon;font-weight: bold;">Password</font></label></td>
           <td> <input type="password" class="input" name="pass" tabindex="2" placeholder="Password"
required></td>
  </tr>
  <tr>
     <td><label><font style="color: maroon;font-weight: bold;">User Name</font></label> </td>
     <td>
          <input type="text" tabindex="1" class="input" name="name" placeholder="User Name" required>
     </td>
     <td> <label><font style="color: maroon;font-weight: bold;">Gender</font></label></td>
     <td>
          <input name="gender" value="Male"  type="radio" checked="true" /> Male
          <input   name="gender" value="Female" type="radio" />Female
     </td>
  </tr>
  <tr>
         <td><label><font style="color: maroon;font-weight: bold;">Address</font></label></td>
        <td> <input type="text" tabindex="1" name="add" class="input" placeholder="Address" required></td>
        <td> <label><font style="color: maroon;font-weight: bold;">City</font></label>/td>
        <td> input type="text" class="input" tabindex="2" name="city" placeholder="City" required></td>
  </tr>
  <tr>
       <td><label><font style="color: maroon;font-weight: bold;">Email</font></label></td>
```

```
    <td><input type="text" tabindex="1" class="input" name="email"  placeholder="Email" pattern="\w+([-
+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" required></td>
    <td><label> <font style="color: maroon;font-weight: bold;">Phone</font></label></td>
    <td> <input type="text" class="input" tabindex="2" name="phone" placeholder="Phone" pattern="[7-
9]{1}[0-9]{9}" minlength="10" maxlength="10" required></td>
</tr>
</center>
</table>
<center>
     <input type="image" id="" value="register"src="<%=request.getContextPath()
%>/Files/Images/register.png" alt="submit Button" width="80" height="30">
</center>
</div>
</form>
</div>
</div>
<% int no = Utility.parse(request.getParameter("no"));
  if (no == 1) {
%>
<div class="error" id="message">
        <p>Oops ,Invalid userid / password..!</p>
</div>
<%}
if (no == 3) {
%>
<div class="error" id="message">
        <p>Oops ,Something went wrong ..!</p>
</div>
<%}
if (no == 4) {%>
 <div class="success" id="message">
        <p>You have registered successfully!</p>
</div>
<%}%>
</div>
<%if (Utility.parse(request.getParameter("no1")) == 5) {%>
        <div class="error" id="message" style="position: absolute">
                <p>Sorry this user id is already exists.....!</p>
        </div>
        <%}%>
</body>
</html>
```

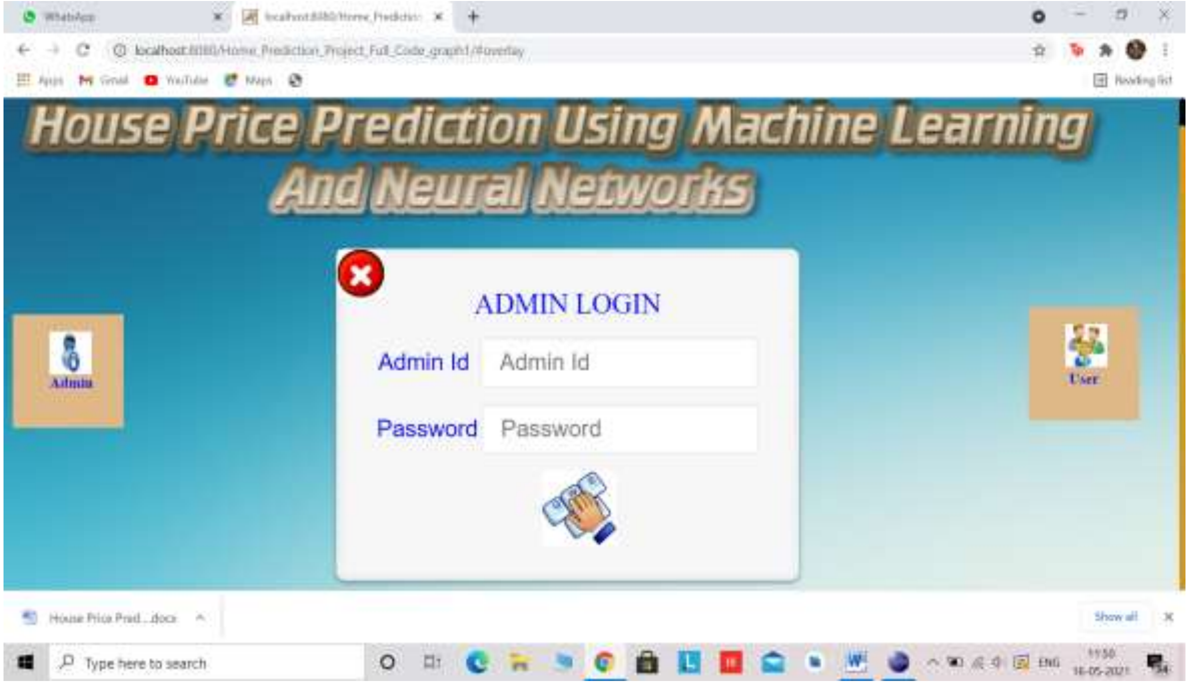
6.2 Snapshots

Home page



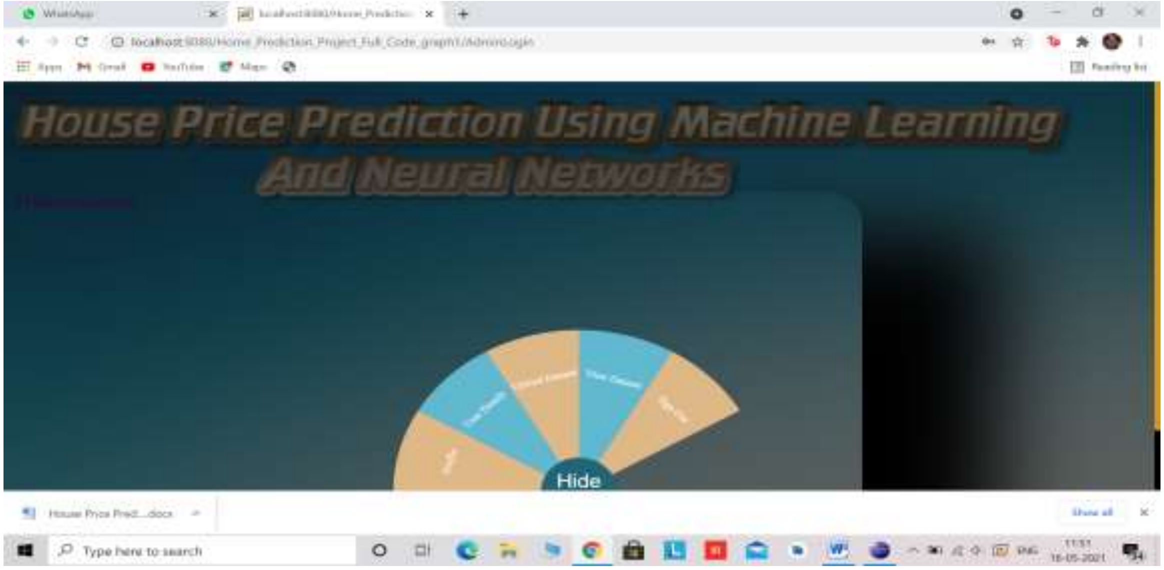
House Price Prediction Using Machine Learning And Neural Networks
Admin
User
Type here to search

Admin login

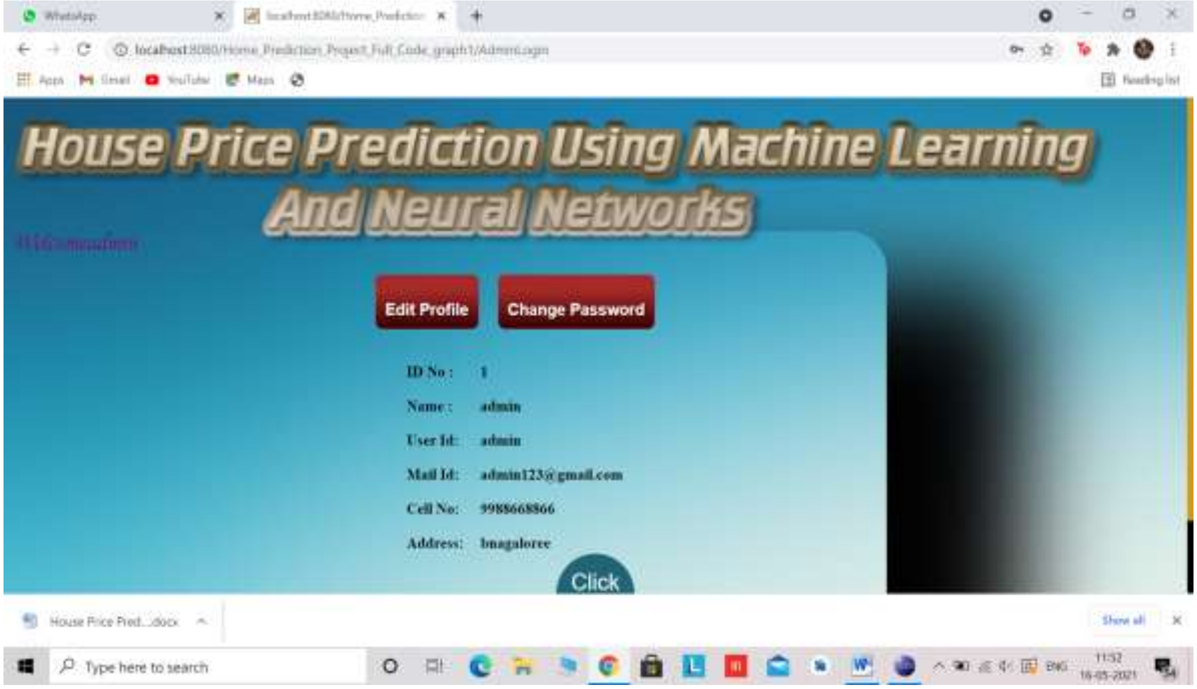


House Price Prediction Using Machine Learning And Neural Networks
ADMIN LOGIN
Admin Id
Password
Admin
User
Type here to search

Admin Homepage


House Price Prediction Using Machine Learning And Neural Networks
Hide
Type here to search

Admin can view and edit his details


House Price Prediction Using Machine Learning And Neural Networks
Edit Profile
Change Password
Click
Type here to search

ID No : 1

Name : admin

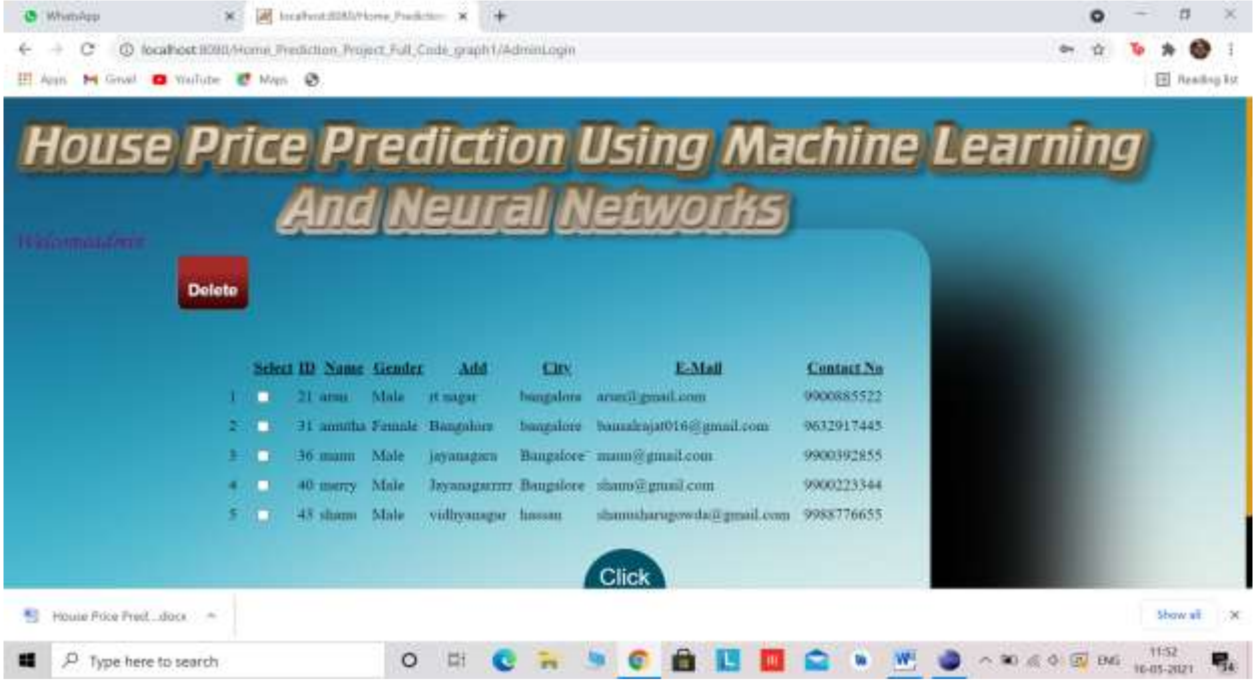
User Id: admin

Mail Id: admin123@gmail.com

Cell No: 9988668866

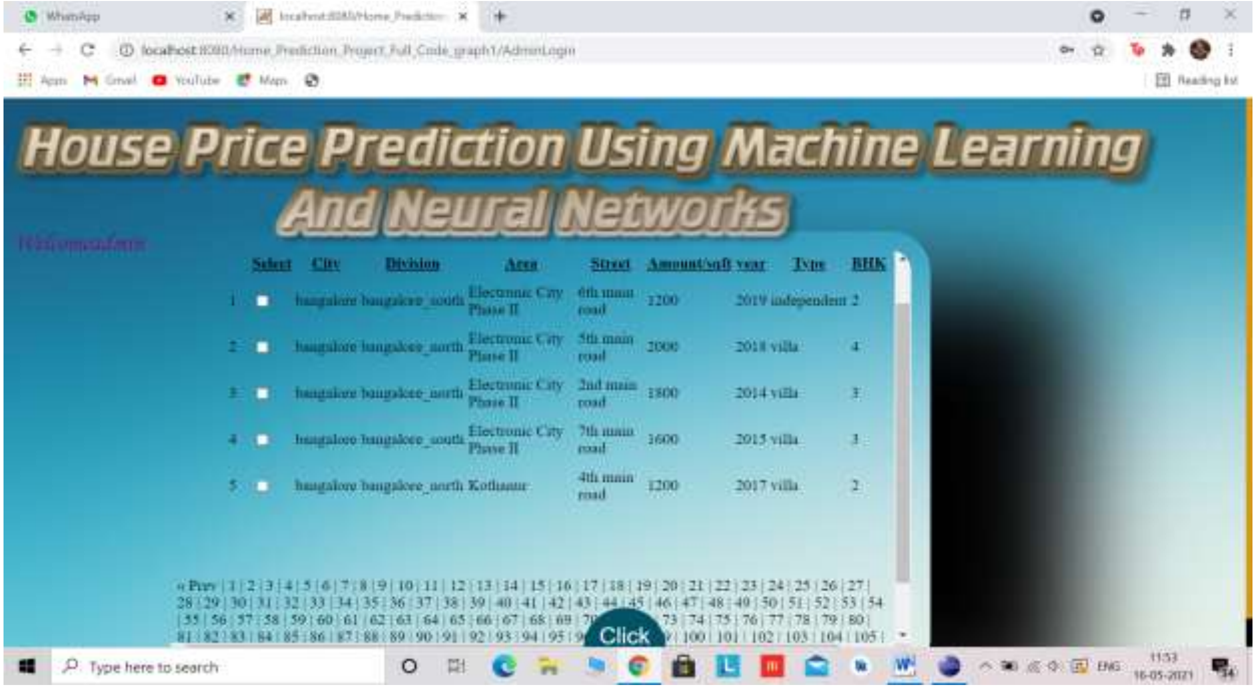
Address: bnagaloree

Viewing and deleting user details


House Price Prediction Using Machine Learning And Neural Networks
Welcomeadmin
Delete
Click
Type here to search

	Select	ID	Name	Gender	Add	City	E-Mail	Contact No
1	□	21	arun	Male	rt nagar	bangalore	arun@gmail.com	9900885522
2	□	31	amitha	Female	Bangalore	bangalore	banaalrajat016@gmail.com	9632917445
3	□	36	manu	Male	jayanagara	Bangalore	manu@gmail.com	9900392855
4	□	40	merry	Male	Jayanagarrrr	Bangalore	shanu@gmail.com	9900223344
5	□	43	shanu	Male	vidhyanagar	hassan	shanusharugowda@gmail.com	9988776655

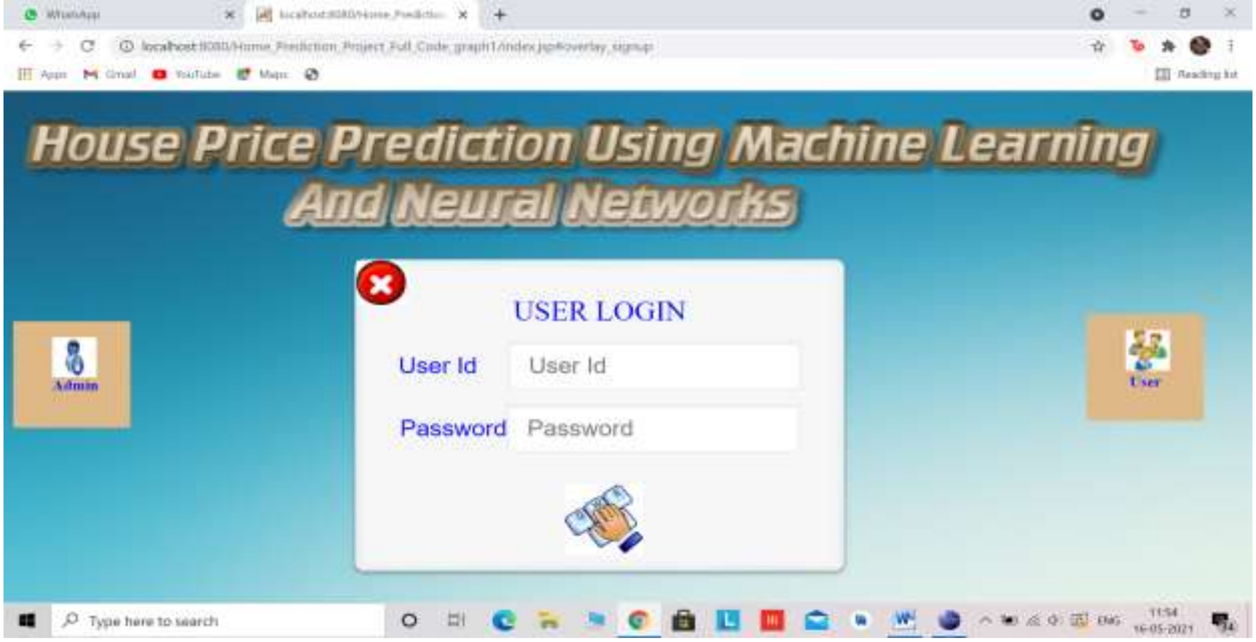
Upload and view dataset


House Price Prediction Using Machine Learning And Neural Networks
Welcomeadmin
Click
Type here to search

	Select	City	Division	Area	Street	Amount/sqft	year	Type	BHK
1	□	bangalore	bangalore_south	Electronic City Phase II	6th main road	1200	2019	independent	2
2	□	bangalore	bangalore_north	Electronic City Phase II	5th main road	2000	2018	villa	4
3	□	bangalore	bangalore_north	Electronic City Phase II	2nd main road	1800	2014	villa	3
4	□	bangalore	bangalore_south	Electronic City Phase II	7th main road	1600	2015	villa	3
5	□	bangalore	bangalore_north	Kothanur	4th main road	1200	2017	villa	2

« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 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 | 7 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 9 9 | 100 | 101 | 102 | 103 | 104 | 105 |

User login


House Price Prediction Using Machine Learning And Neural Networks

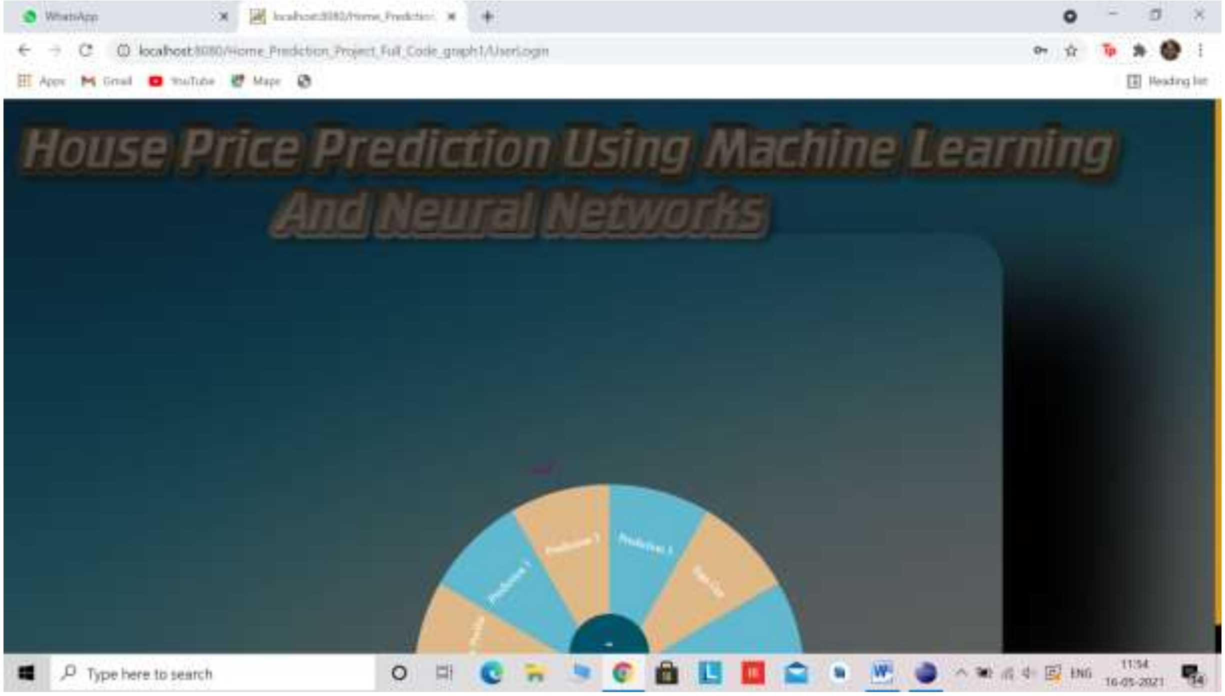
USER LOGIN

User Id: User Id

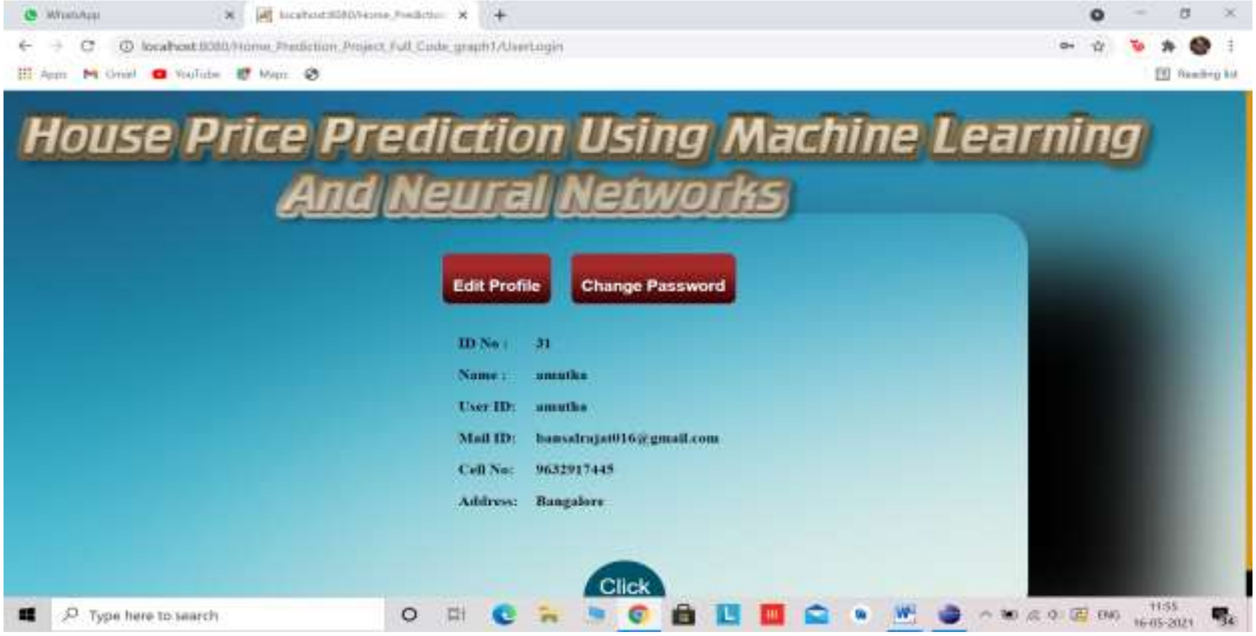
Password: Password

localhost:8080/Home_Prediction_Project_Full_Code_graph1/index.jsp#overlay_signup
Admin
User
Type here to search
11:54
16-05-2021

User Homepage


House Price Prediction Using Machine Learning And Neural Networks

User can view their profile and edit details


House Price Prediction Using Machine Learning And Neural Networks
Edit Profile
Change Password
Click
Type here to search

ID No : 31

Name : amutha

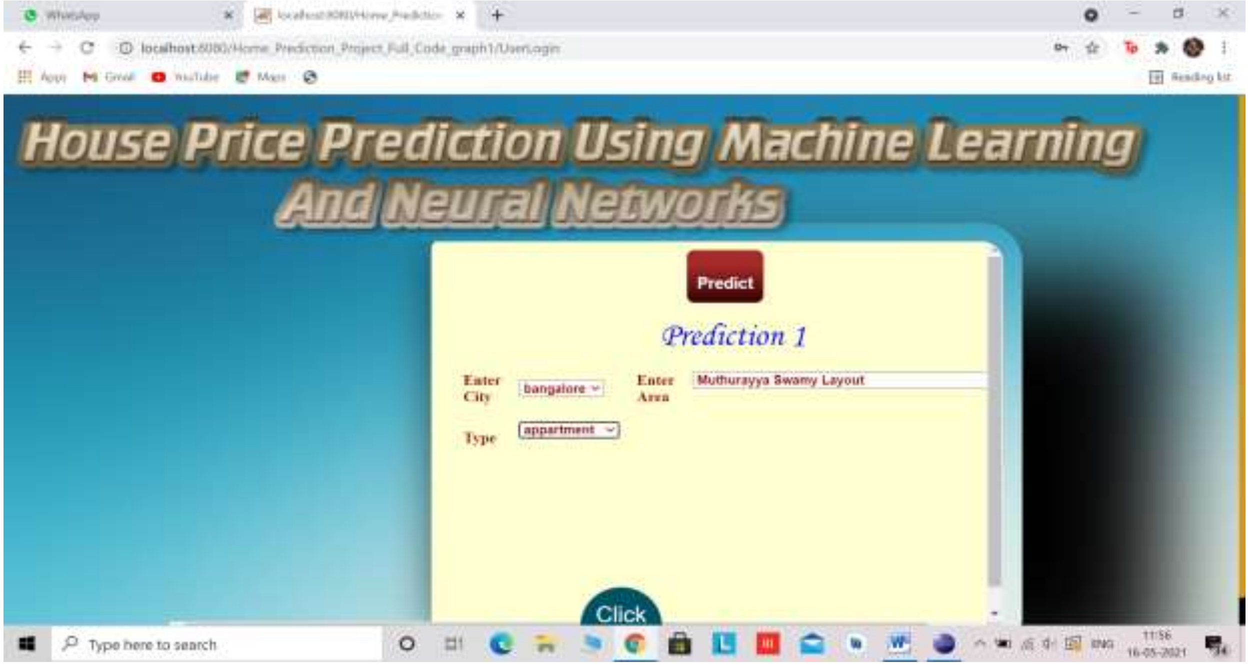
User ID: amutha

Mail ID: bansalrajat016@gmail.com

Cell No: 9632917445

Address: Bangalore

Prediction 1: based on city, area and type

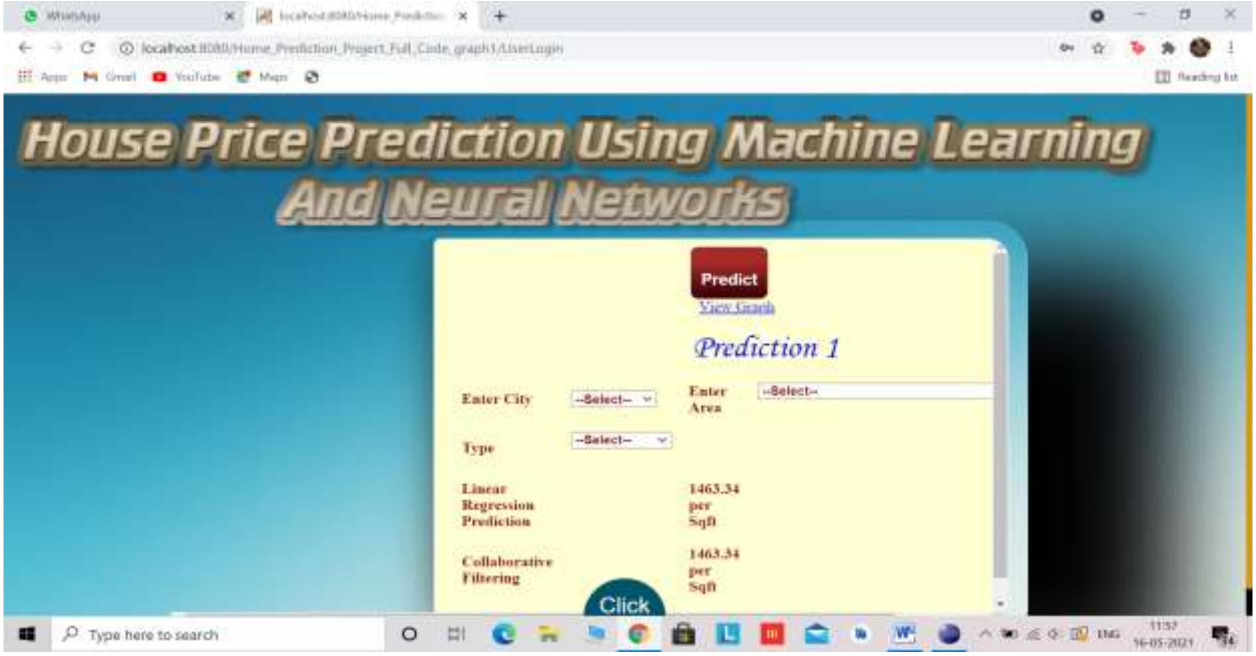

House Price Prediction Using Machine Learning And Neural Networks
Predict
Prediction 1
Click
Type here to search

Enter City: bangalore

Enter Area: Muthurayya Swamy Layout

Type: appartment

Prediction 1 result



House Price Prediction Using Machine Learning And Neural Networks

Predict

View Graph

Prediction 1

Enter City --Select--

Enter Area --Select--

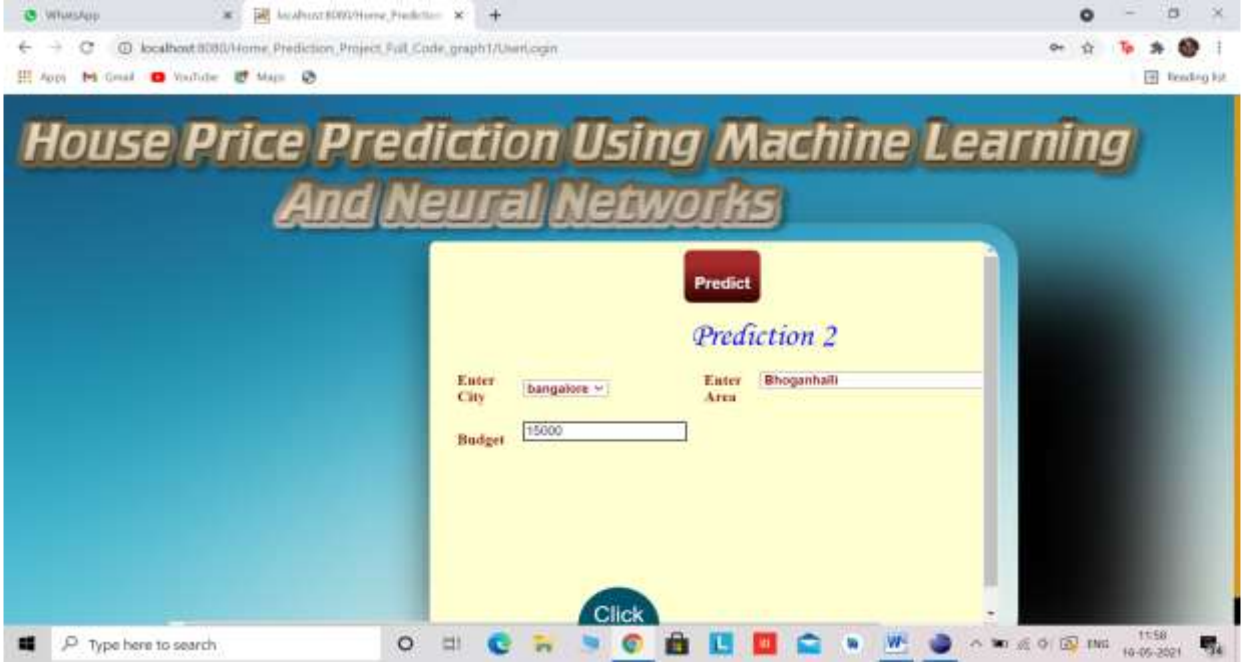
Type --Select--

Linear Regression Prediction 1463.34 per Sqft

Collaborative Filtering 1463.34 per Sqft

Click

Prediction 2: based on city, area and budget



House Price Prediction Using Machine Learning And Neural Networks

Predict

Prediction 2

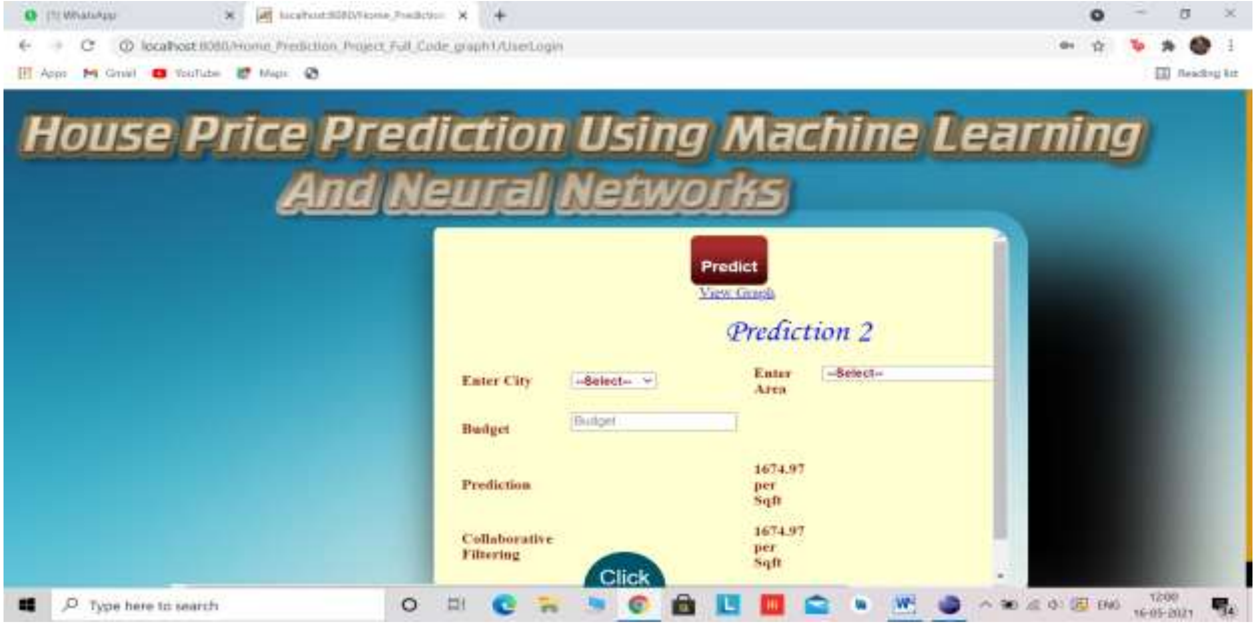
Enter City bangalore

Enter Area Bhoganhalli

Budget 15000

Click

Prediction 2 result


localhost:8080/Home_Prediction_Project_Full_Code_graph1/UserLogin
House Price Prediction Using Machine Learning And Neural Networks
Predict
View Graph
Click
Type here to search

Prediction 2

Enter City --Select--

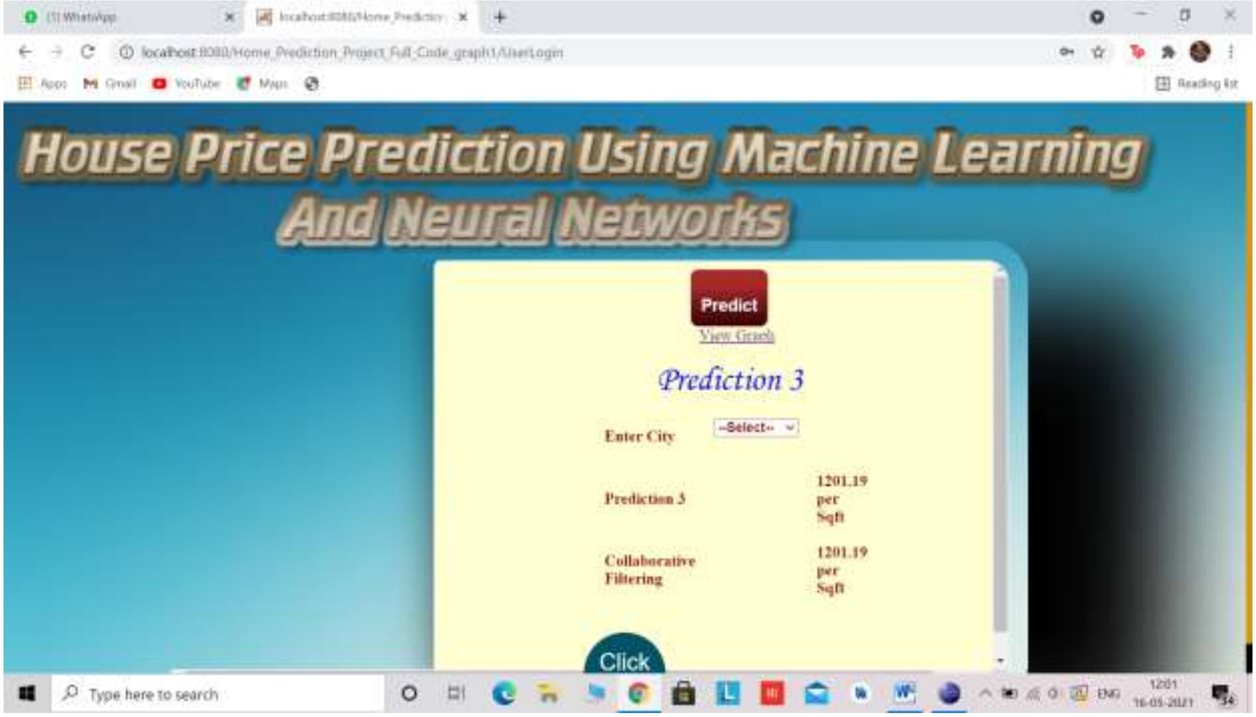
Enter Area --Select--

Budget

Prediction 1674.97 per Sqft

Collaborative Filtering 1674.97 per Sqft

Results for Prediction 3: based on total city


localhost:8080/Home_Prediction_Project_Full_Code_graph1/UserLogin
House Price Prediction Using Machine Learning And Neural Networks
Predict
View Graph
Click
Type here to search

Prediction 3

Enter City --Select--

Prediction 3 1201.19 per Sqft

Collaborative Filtering 1201.19 per Sqft

Chapter 7 CONCLUSION

A system that aims to provide an accurate prediction of housing prices has been developed. The system makes optimal use of Linear Regression and Collaborative Filtering. The efficiency of the algorithm ha been further increased with use of Neural networks. The system will satisfy customers by providing accurate output and preventing the risk of investing in the wrong house. Additional features for the customer's benefit can also be added to the system without disturbing it's core functionality.

Chapter 8 FUTURE SCOPE

In spite of hosting generated all the attempting provision that met our introductory requirements, there are different upgrades that could be produced later on. These incorporate upgrades we didn’t settle on because of constrained duration of the time. A real worry for the prediction framework may be the stacking period. Moreover, our data set takes more than one day should prepare. As opposed performing the computations sequentially, we might utilize various processors and parallel the computations involved, which might possibly decrease the preparation time furthermore prediction period. Include all the more functionalities under the model, we can give choices for client with select a district alternately locale should produce those high temperature maps, as opposed to entering in the list.

Chapter 9 REFERENCES

[1] A. Adair, J. Berry, W. McGreal, Hedonic modeling, housing submarkets and residential valuation, Journal of Property Research, 13 (1996) 67-83.

[2] O. Bin, A prediction comparison of housing sales prices by parametric versus semi-parametric regressions, Journal of Housing Economics, 13 (2004) 68-84.

[3] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, “How many trees in a random forest?” In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7376 LNAI, 2012, pp. 154–168, ISBN: 9783642315367. DOI: 10 . 1007 / 978 - 3 - 642 - 31537-4\ 13

[4] J. Schmidhuber, “Multi-column deep neural networks for image classification,” in Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), ser. CVPR ’12, Washington, DC, USA: IEEE Computer Society, 2012, pp. 3642–3649, ISBN: 978-1-4673-1226-4. [Online].

[5] T. Kauko, P. Hooimeijer, J. Hakfoort, Capturing housing market segmentation: An alternative approach based on neural network modeling, Housing Studies, 17 (2002) 875-894