D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY

BALUSUMUDI, BHIMAVARAM, W.G. Dist., A.P., PIN-534202

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

(Accredited with A++ Grade by NAAC & Accredited by NBA)

Ph:08816-221238, Email: dnrcet@gmail.com, Website: https://dnrcet.org

Vision of the Institution:

To evolve as a Quality Institution in Teaching, Innovative Research, Entrepreneurship and Consultation in Engineering & Technology, empower rural youth globally competent and self-disciplined technocrats.

Mission of the Institution:

IM1: Inculcate technical knowledge, soft skills through student centric teaching & learning.

IM2: Strengthen industry institute interaction, provide solutions to the ever- Changing requirements.

IM3: Implant entrepreneurial attitude and ethical values.

IM4: Create work culture towards learning, Research & Development.

IM5: Develop a unique practice that instills responsibility and accountability among the stakeholders

D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY
BALUSUMUDI, BHIMAVARAM, W.G. Dist., A.P., PIN-534202
(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)
(Accredited with A++Grade by NAAC & Accredited by NBA)
Ph:08816-221238, Email: dnrcet@gmail.com, Website: https://dnrcet.org

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION:

To become an identified development center for high quality professionals in the area of Computer Science & Engineering serving the societal needs.

MISSION:

DM1: Train the stakeholders in the area of Computer Science & Engineering.

DM2: Organize innovative technical training and leadership activities to groom professionals.

DM3: Provide quality resources towards research and development on Artificial Intelligence.

A Project Report on

ENHANCING HOUSE PRICE PREDICTION BASED MACHINE LEARNING USING MODIFIED EXTREME BOOSTING

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

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By

M. LAKSHMI PRASANNA (219P1A0565)
G. LIKHITHA (219P1A0548)
D. JANAYATRI (219P1A0533)
J. YUVA RATNA (219P1A0554)

Under the Esteemed Guidance of

Ms. K. SIVA SYAMALA, M. Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

D.N.R COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

(Accredited by NAAC with A++ Grade & NBA)

Balusumudi, Bhimavaram, W. G. Dist., A.P. -534202

2021–2025

D.N.R COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)
(Accredited by NAAC with A++ Grade & NBA)
Balusumudi, Bhimavaram -534202 W. G. Dist., A.P., India.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project report titled **"ENHANCING HOUSE PRICE PREDICTION BASED MACHINE LEARNING USING MODIFIED EXTREME BOOSTING"** is a Bonafide record of project work carried out by**, M. LAKSHMI PRASANNA, G. LIKHITHA, D. JANAYATRI, J. YUVA RATNA** bearing with the **Regd. No's: 219P1A0565, 219P1A0548, 219P1A0533, 219P1A0554** students of B. tech final year in the Department of Computer Science & Engineering, in D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY, Balusumudi, Bhimavaram. The project work is carried out during the academic year 2024-2025 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering.

Guide& Internal Examiner ***Project Co-Ordinator***

Head of the Department ***External Examiner***

ACKNOWLEDGEMENT

We would like to take the privilege of the opportunity to express our gratitude for the project work of **"ENHANCING HOUSE PRICE PREDICTION BASED MACHINE LEARNING USING MODIFIED EXTREME BOOSTING"** enabled us to express our special thanks to our guide **K. SIVA SYAMALA**, Assistant Professor, Department of Computer Science & Engineering, **D. N. R COLLEGE OF ENGINEERING & TECHNOLOGY**, for his valuable guidance since the project started.

We owe our gratitude to our beloved Professor & Head of the Department of Computer Science & Engineering, **Dr. G Satyanarayana**, for being a source of inspiration and encouraging us in successful completion of the project work.

We are thankful to our honorable principal **Dr. M Anjan Kumar**, who has shown keen interest and encouraged us by providing all the facilities to complete our project successfully.

We wish to express our sincere thanks to our Management by providing all the facilities and infrastructure to complete our project successfully and we wish to express our sincere thanks to all teaching and non-teaching staff of Computer Science and Engineering.

We are very thankful to all our friends and family members who had given us good co-operation and suggestions throughout this project and helped us in successful completion.

M. LAKSHMI PRASANNA (219P1A0565)
G. LIKHITHA (219P1A0548)
D. JANAYATRI (219P5A0533)
J. YUVA RATNA (219P1A0554)

DECLARATION BY THE CANDIDATES

We, M. LAKSHMI PRASANNA, G. LIKHITHA, D. JANAYATRI, J. YUVA RATNA bearing Reg no: 219P1A0565, 219P1A0548, 219P1A0533, 219P1A0554, here by declare that the project report titled **"ENHANCING HOUSE PRICE PREDICTION BASED MACHINE LEARNING USING MODIFIED EXTREME BOOSTING"** under the guidance of **K. SIVA SYAMALA,** M. tech., is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering.

This is a record of bona fide work carried out by us and the results embodied in this project have not been reproduced/copy from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree.

M. LAKSHMI PRASANNA (219P1A0565)
G. LIKHITHA (219P1A0548)
D. JANAYATRI (219P5A0533)
J. YUVA RATNA (219P1A0554)

ABSTRACT

In recent years, machine learning has become increasingly important in making everyday predictions, such as improving safety in autonomous vehicles and enhancing customer service. With its growing application, machine learning is becoming more integral across various industries. When it comes to predicting house prices, factors like location, region, and population play a significant role in determining property values. While traditional machine learning models have been used to forecast house prices, many studies often focus on basic approaches and neglect the potential of more advanced, lesser-known models. In this study, we introduce Modified Extreme Gradient Boosting (MXGB) as our approach, chosen for its ability to adapt to different data patterns and efficiently select the best models. Our model achieves an accuracy of over 80%, making it a highly reliable tool for forecasting housing market trends. In today's rapidly changing society, the ability to predict housing market trends is becoming increasingly important. As housing costs continue to fluctuate, reliable predictions can offer valuable insights, helping individuals, investors, and policymakers make more informed decisions in a volatile market.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OFABBREVATIONS

S. No.	Abbreviations
1	XG BOOST-Modified Extreme Gradient Boosting
2	L1-Lasso Regression
3	L2-Ridge Regression
4	DTR-Decision Tree Regression
5	ABR-Ada Boost Regressor
6	RFR-Random Forest Regression
7	GBR-Gradient Boosting Regression
8	LR-Linear Regression
9	UML-Unified Modeling Languages
10	IDLE– Integrated Development and Learning Environment

1. INTRODUCTION

1.1 BRIEF INFORMATION

The housing market is a crucial part of the real estate sector, significantly impacting the overall health of any economy. For many people, owning a home is a personal achievement and a symbol of status, driving young professionals to aspire to homeownership, while investors view real estate as a profitable opportunity. A strong housing market is essential for a growing economy, as homeownership represents both a milestone for individuals and an investment for investors. The relationship between home prices and homeownership rates is often inversely correlated, with rising prices making homeownership less accessible. Most research has focused on countries with high homeownership rates, especially those experiencing economic growth. Affordable housing is vital for long-term market stability, and housing costs greatly affect individuals' ability to secure stable investments. Compared to more volatile markets like stocks, real estate has been more stable, with fluctuations in home prices influencing both homebuyers and investors. Recently, the real estate market has become one of the most attractive fields for investment, with a growing need for effective models to predict property values. As urban populations grow, housing shortages have become a significant issue, and better predictive models are needed to understand the factors influencing home prices. In this study, we propose Modified Extreme Gradient Boosting (MXGB) as a predictive model for home prices. Unlike traditional models, MXGB adapts more effectively to data patterns, offering a more accurate and probabilistic approach, which can help homebuyers, investors, and policymakers make more informed decisions in a fluctuating market.

1.2 PURPOSE

This project aims to create an accurate model for predicting home prices using Modified Extreme Gradient Boosting (MXGB). The goal is to improve predictions by considering factors such as location, size, and economic conditions. This model will help investors, homebuyers, and policymakers make informed decisions in the housing market. By utilizing advanced machine learning techniques, the project seeks to provide more reliable forecasts compared to traditional models. Ultimately, it aims to support better investment strategies and decision-making while minimizing risks related to fluctuating home prices

1.3 MOTIVATION

This project is motivated by the challenges people face in today's housing market, where property prices are constantly changing. Many individuals and investors struggle to make informed decisions due to the unpredictable market. Traditional methods don't always provide accurate predictions. With growing urban populations and increased housing demand, having a reliable way to forecast home prices is crucial. This project aims to create a more accurate model using machine learning techniques. By improving predictions, it can help people make better decisions when buying or investing in homes. The model will reduce risks.

1.4 PROBLEMSTATEMENT

Predicting house prices is difficult because many factors affect the value, such as location, size, and market trends. Traditional methods are often not accurate enough, making it hard for buyers, sellers, and investors to make the right decisions. Some machine learning algorithms like combining Lasso and Gradient boosting regressions and SVM &Random Forest can predict prices, but they do not always handle with complex patterns well or adjust to changing market conditions. This can lead to the incorrect price estimates**.**

2. LITERATURE SURVEY

This literature survey aims to explore previous research and methodologies used in predicting house prices through machine learning. Our project, titled "Machine Learning-Based House Price Prediction Using Modified Extreme Boosting," focuses on enhancing prediction accuracy by utilizing optimized boosting algorithms. By reviewing related works, we aim to identify effective models, techniques, and feature engineering practices that can improve the performance of our system. This survey helps us understand the strengths and limitations of existing approaches, guiding the development of our proposed model.

1)Artificial intelligence algorithms to predict Italian real estate market prices

AUTHORS :Luca Rampini, Fulvio Re Cecconi

The assessment of the Real Estate (RE) prices depends on multiple factors that traditional evaluation methods often struggle to fully understand. Housing prices, in particular, are the foundations for a better knowledge of the Built Environment and its characteristics. Recently, Machine Learning (ML) techniques, which are a subset of Artificial Intelligence, are gaining momentum in solving complex, non-linear problems like house price forecasting. Hence, this study deployed three popular ML techniques to predict dwelling prices in two cities in Italy.

2)Artificial Intelligence Approach for Modeling House Price Prediction

AUTHORS: Melihsah Cekic; Kubra Nur Korkmaz; Habib Mukus; Alaa Ali Hameed; Akhtar Jamil;

Real estate has a vast market volume across the globe. This domain has been growing significantly in the past few decades. An accurate prediction can help buyers, and other decision-makers make better decisions. However, developing a model that can effectively predict house prices in complex environments is still a challenging task. This paper proposes machine learning models for the accurate prediction of real estate house prices. Furthermore, we investigated the feature importance and various data analysis methods to improve the prediction accuracy. Linear Regression, Decision Tree, XGBoost, Extra Trees, and Random Forest were used in this study. For all models, hyperparameters were first calculated using k-fold cross-validation, and then they were trained to apply to test data. The models were tested on the Boston housing dataset.

3)Predicting house prices with machine learning methods

AUTHORS: ALAN IHRE ISAK ENGSTROM.

In this study, the machine learning algorithms K-Nearest-Neighbours regression (K-NN) and Random Forest (RF) regression were used to predict house prices from a set of features in the Ames housing data set. The algorithms were selected from an assessment of previous research and the intent was to compare their relative performance at this task. Software implementations for the experiment were selected from the scikit learn Python library and executed to calculate the error between the actual and predicted sales price using four different metrics. Hyperparameters for the algorithms used were optimally selected and the cleaned data set was split using five-fold cross-validation to reduce the risk of bias. An optimal subset of hyperparameters for the two algorithms was selected through the grid search algorithm for the best prediction. The Random Forest was found to consistently perform better than the KNN algorithm in terms of smaller errors and be better suited as a prediction model for the house price problem.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In this existing system house prices are predicted using simple algorithms like Hybrid regression model combining Lasso & Gradient boosting regressions to predict house prices Use location and size but don't track market changes, Work well but struggle with big data. Additionally, other algorithms like SVM &Random Forest are mentioned as possible alternatives but are not the primary focus of the study these models used but don't adjust to changes. By using these algoritms they get best accuracy for house price prediction was with a hybrid model combining 65% Lasso and 35% Gradient Boosting.

DISADVANTAGES OF EXISTING SYSTEM

1. Not Accurate: Many models miss important details
2. Can't Adjust to Market Changes: Prices change, but these models don't update well.
3. Overfitting: Some models work on old data but fail for new prices.
4. Misses Key Factors: Things like new roads, schools, crime rates, and demand are ignored
5. Slow: Some methods take too long to process data.

3.2 PROPOSED SYSTEM

To overcome the limitations of existing house price prediction based on some factors like location, size, situation, market trends. This project introduces a Modified Extreme Gradient Boosting (XG Boost) model. This advanced machine learning approach improves accuracy by learning from past data and adjusting to market changes. By using this algorithm we expect accuracy range is above 80%.

ADVANTAGES OF PROPOSED SYSTEM

1. More Accurate – Predicts house prices better than old methods.
2. Uses Important Factors – Considers crime rates, transport, and economy for better results.
3. Adapts to Market Changes – Learns and updates when house prices change.
4. Understands Complex Data – Works well with different price factors.
5. Less Manual Work – Adjusts settings automatically for better performance.

3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.3.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system

and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.4 SYSTEM REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) is a complete description of the behaviour of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

3.4.1 Functional Requirements

- Data Collection
- Data Pre-processing
- Training and Testing
- Modelling
- Predicting

3.4.2 No Functional Requirements

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement

- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

3.5 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Architecture – All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.

Processing power – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category.

Memory – All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Secondary storage – Hard-disk requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display adapter – Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements.

Peripherals – Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc.

1. Operating System: Windows
2. Processor: i3 and above
3.Ram: 8gb minimum
4.HardDisk: At least 1TB

3.6 SOFTWAREREQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform – In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.

Operating system is one of the first requirements mentioned when defining system requirements(software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

APIs and drivers – Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms.

Web browser – Most web applications and software depending heavily on Internet technologies make use of the default browser installed on system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, which makes use of ActiveX controls, despite their vulnerabilities.

1) Software: Python IDLE V (3.8.2)

2) PrimaryLanguage: Python

3) Frontend Framework: Django

4) Back-end Framework: Visual Studio Code (VS Code)

5) Database: SQLite

6)Front-End Technologies: HTML, CSS, JavaScript and Bootstrap4

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

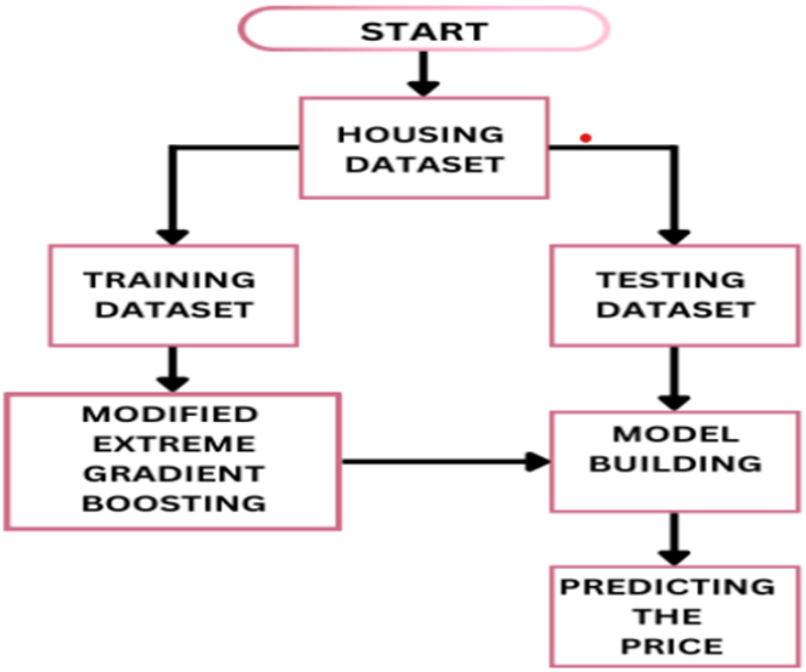


Fig:4.1SystemArchitecture

4.2 MODULES

To implement this project we used the following modules given below. They are:

- User
- Admin
- Data Preprocessing
- Machine Learning Results

1. User Module

- **Purpose:** This module focuses on the user experience and interaction with the system.
  - **Register:** Users must register with a valid email and mobile number for future communication.
  - **Activation:** After registration, users can only log in once the admin activates

their account.

- **Upload Dataset**: Users can upload datasets that match the expected column format. The data must be float format for processing.
- **Data Extension**: Users may add new entries to the existing dataset using the Django interface.
- **Execute Processing**: Users can initiate processing and view the results based on the algorithms used.

2. Admin Module

- **Purpose**: This module is responsible for system oversight and user management.
- **Login:** Admins can log in using their credentials to access administrative functions.
- **Activate Users:** Admins approve user registrations, enabling them to access the system.
- **View Data:** Admins can view uploaded data through the web interface.
- **Access Results:** Admins can access and view data processing results generated by the system.
- **Accuracy Overview:** Once all algorithm executions are complete, admins can see overall performance.

3. Data Preprocessing Module

- **Purpose**: This module prepares raw data for machine learning algorithms.
- **Data Objects:** The dataset is treated as a collection of data objects, described by various features.
- **Feature Definition:** Features may include physical properties, event timings, or other relevant characteristics.
- **Cleaning Techniques**: This includes noise removal, handling missing values, modifying defaults where applicable, and attribute grouping.
- **Enhanced Prediction:** Proper preprocessing improves prediction accuracy at various analysis levels.

4. Machine Learning Results Module

- **Purpose:** This module presents the results from the applied machine learning algorithms.

- **Data Splitting:** The dataset is divided into training and testing sets.
- **Algorithms Used:** Algorithms such as Random Forest (RF), K-Nearest Neighbour (KNN), and Linear Regressor (LR) are applied.
- **Result Display:** The system displays outcomes based on the results generated by each algorithm.
- **Best Model Identification**: The model that produces the most accurate results is identified as the best performing regressor.

4.3 UMLDiagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Beindependentofparticularprogramminglanguagesanddevelopmentprocess.
4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

There are nine types of UMLDiagrams

- Class Diagram
- Use Case Diagram
- Object Diagram
- Sequence Diagram
- Collaboration Diagram
- State chart Diagram
- Activity Diagram
- Component Diagram

4.3.1 Use case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

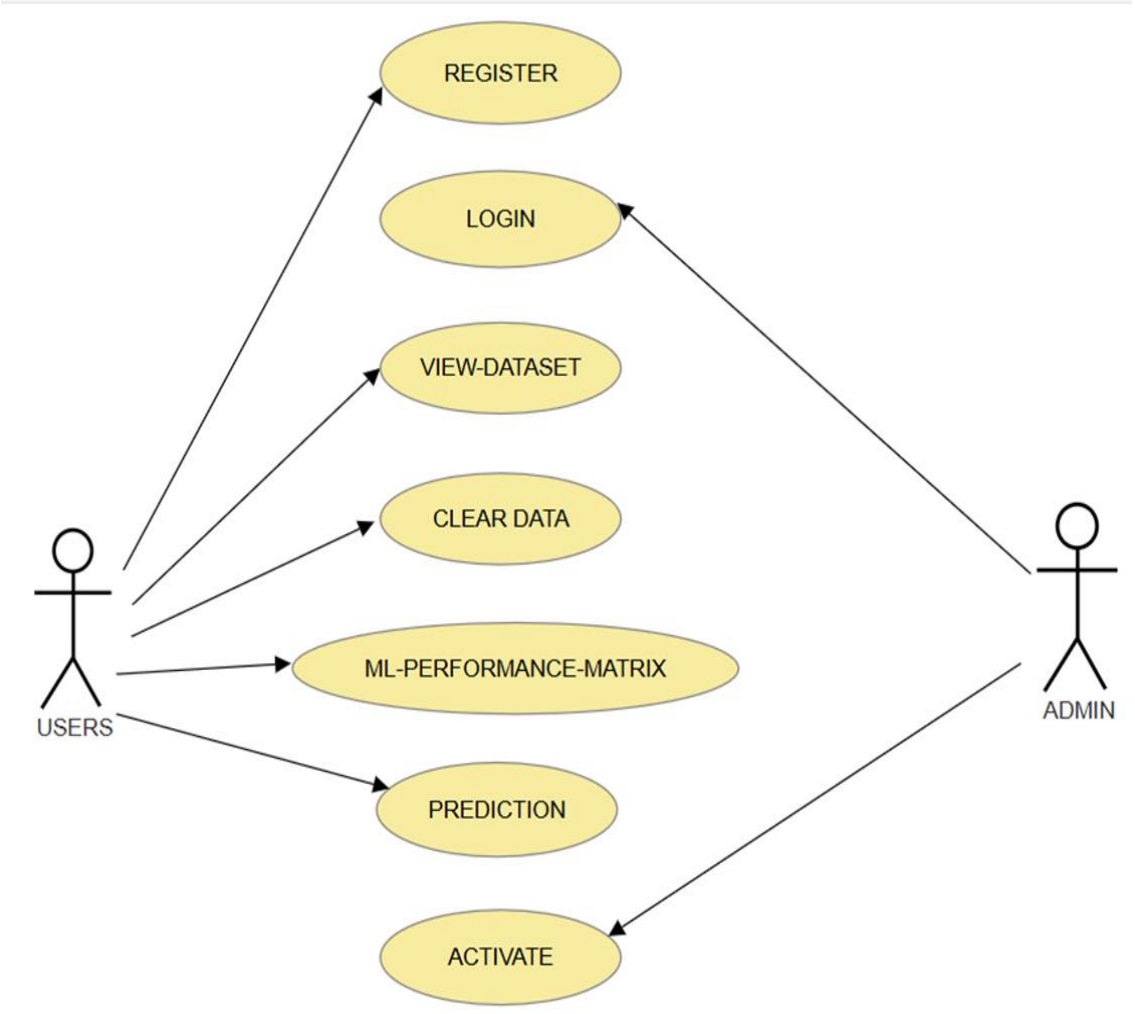


Fig:4.1Usecase Diagram

4.3.2 Class Diagram

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

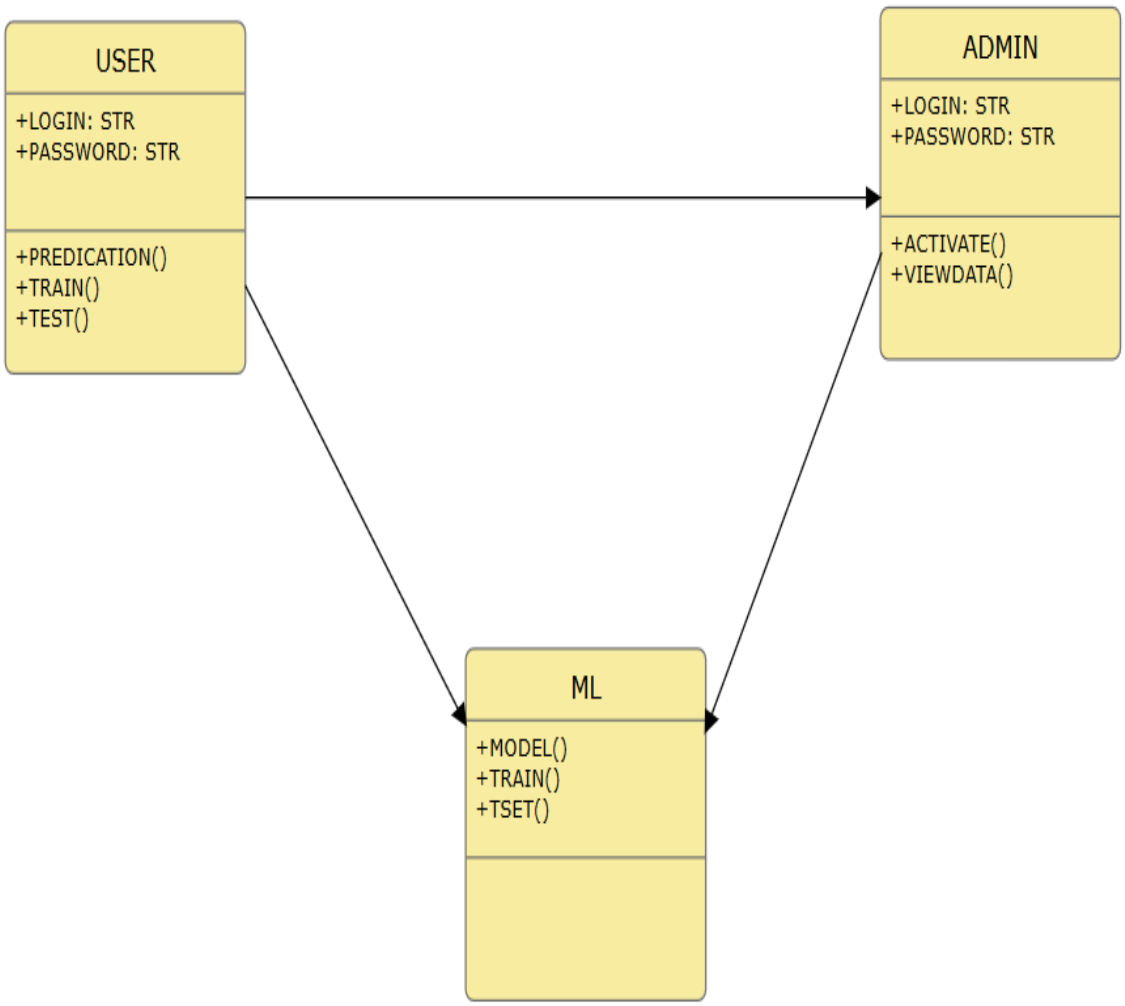


Fig:4.2Class Diagram

4.3.3 Sequence Diagram

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

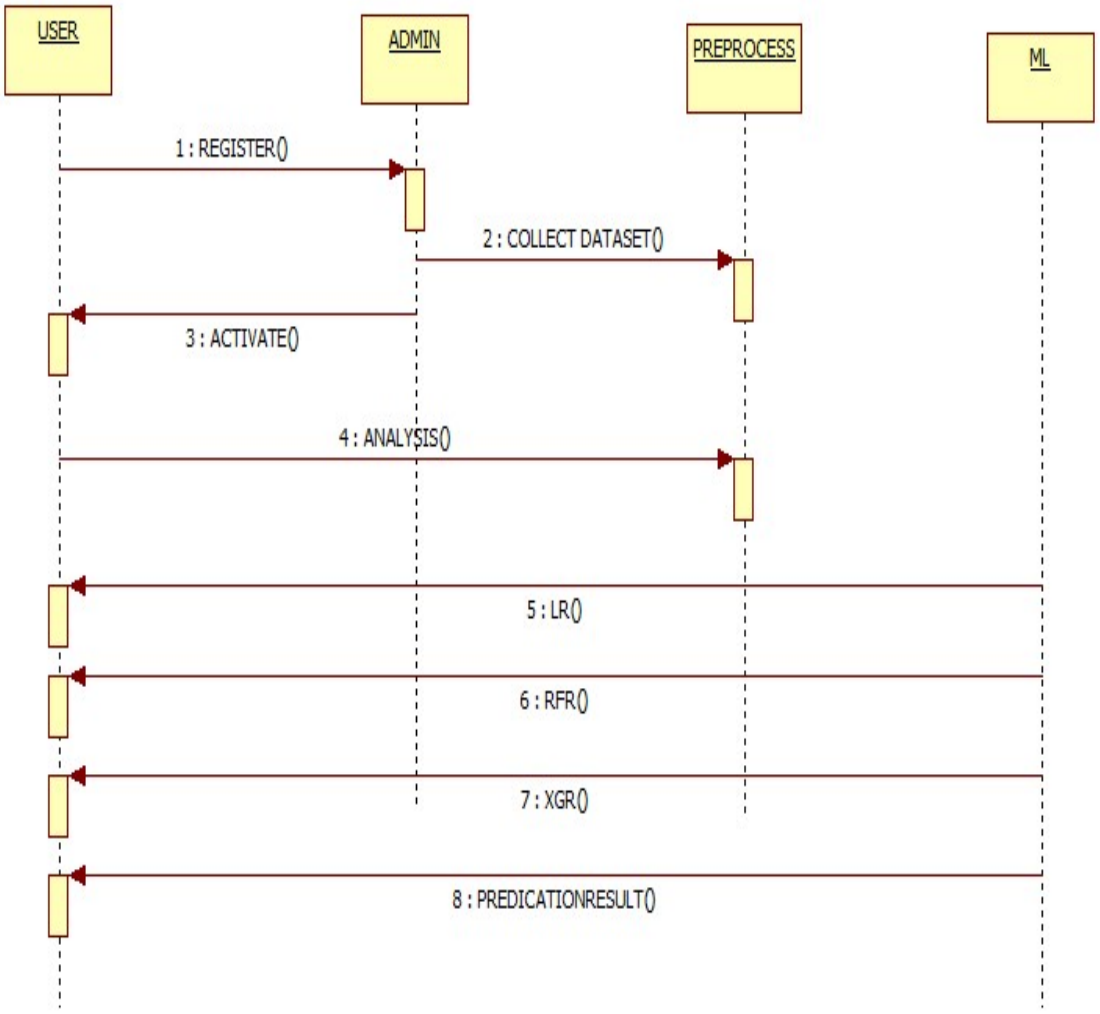


Fig:4.3Sequence Diagram

4.3.4 Activity diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds off lows, the activity diagram has come up with a fork, join, etc. It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

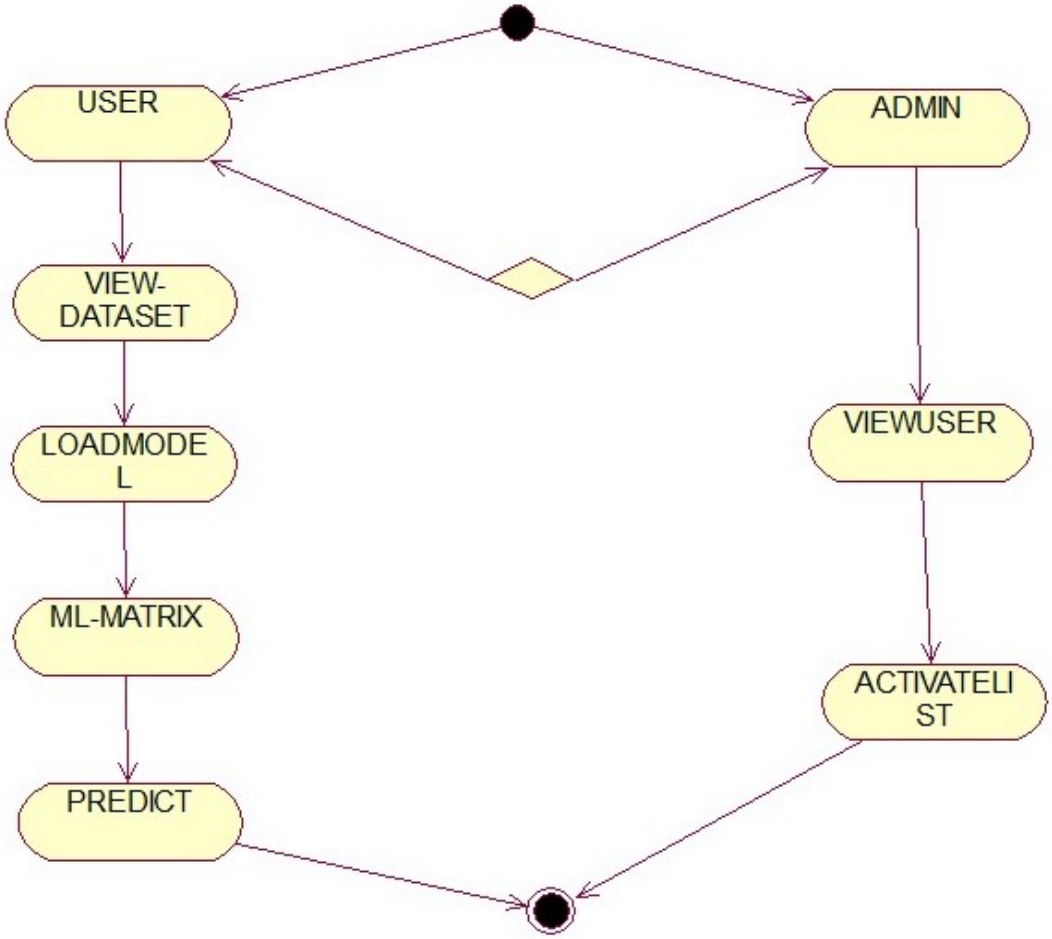


Fig:4.4Activity Diagram

5. SYSTEM IMPLEMENTATION

5.1 FRONTEND IMPLEMENTATION

PYTHON LANGUAGE

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is a dynamic, high-level, free open source, and interpreted programming language. It supports object-oriented programming as well as procedural-oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamically typed language. For example, x = 10 Here, x can be anything such as String, int, etc.

Features in Python:

There are many features in Python, some of which are discussed below as follows:

1. Free and Open Source

Python language is freely available at the official website and you can download it from

the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to Code

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Easy to Read

As you will see, learning Python is quite simple. As was already established, Python's syntax is really straight forward. The code block is defined by the indentations rather than by semicolons or brackets.

4. Object-Oriented Language

One of the key features of Python is Object-Oriented programming. Python supports object- oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support

Graphical User interfaces can be made using a module such as PyQt5,PyQt4,wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

6. High-Level Language

Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. Extensible feature

Python is an Extensible language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.

8. Easy to Debug

Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.

9. Python is a Portable language

Python language is also a portable language. For example, if we have Python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

10. Python is an Integrated language

Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.

11. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. Like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to debug our code. The source code of Python is converted into an immediate form called bytecode.

12. Large Standard Library

Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

13. Dynamically Typed Language

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

14. Frontend and backend development

With a new project Py script, you can run and write Python codes in HTML with the help of some simple tags <Py-script>, <Py-env>, etc. This will help you do frontend development work in Python like javascript. Backend is the strong for teof Python it's extensively used for this work cause of its frameworks like Django and Flask.

15. Allocating Memory Dynamically

In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value. Developers do not need to write int y = 18if the integer value 15 is set to y. You may just type y=18.

Install Python Step-by-Step in Windows and Mac:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its not able use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac:

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.8.2 or in other words, It is Python3.

Note: The python version3.8.2 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python3. Download the Python Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Fig:5.1 Download Python IDLE

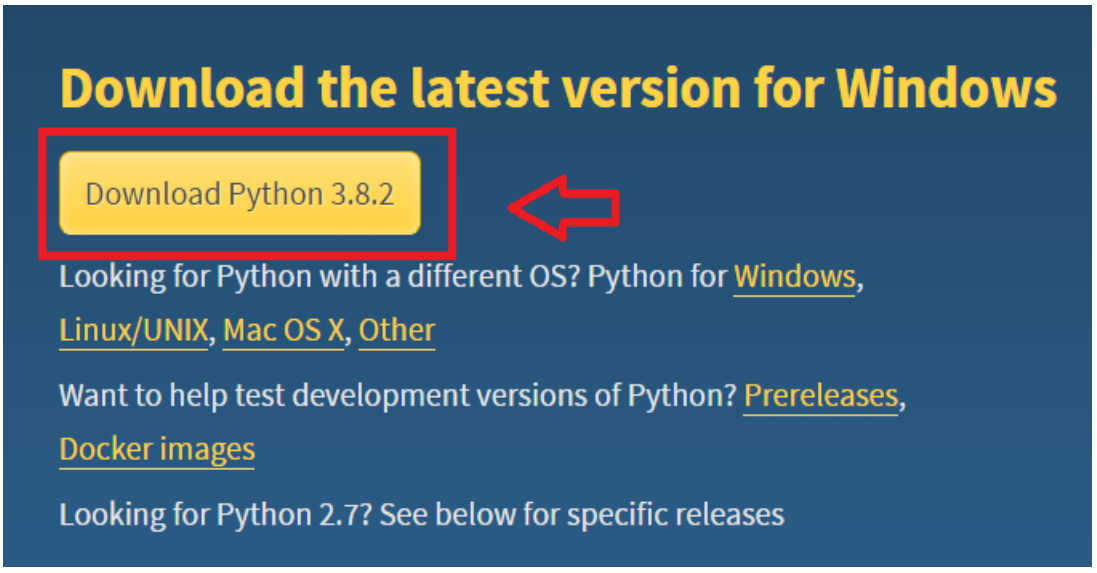


Fig:5.2Click on latest version

Step 3: You can either select the Download Python for windows 3.8.2 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here,

we are downloading the most recent python version for windows 3.8.2

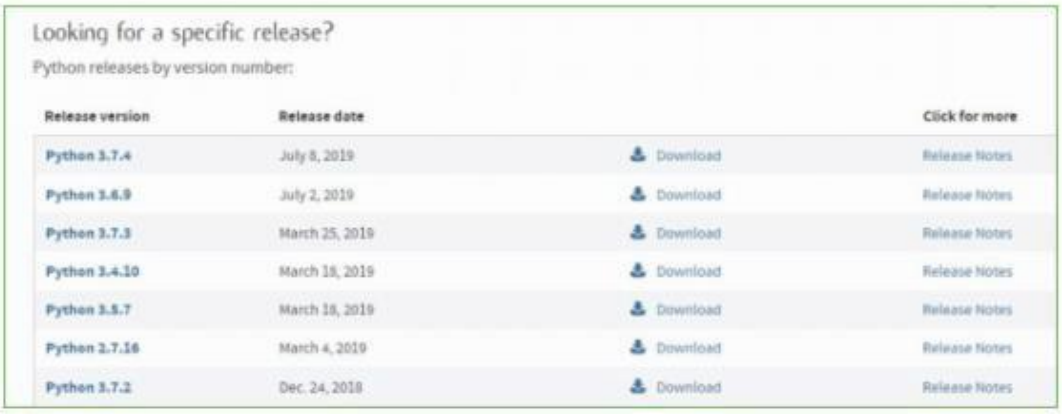


Fig:5.3 Select files

Step4: Scroll down the page until you find the Files option.

Step5: Here you see a different version of python along with the operating system.

Fig:5.4 Install all users

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windowsx86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which

version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

Fig:5.5 Select Path

Step2: Before you click on Install Now, Make sure to put a tick on Add Python3.8 to PATH.

6

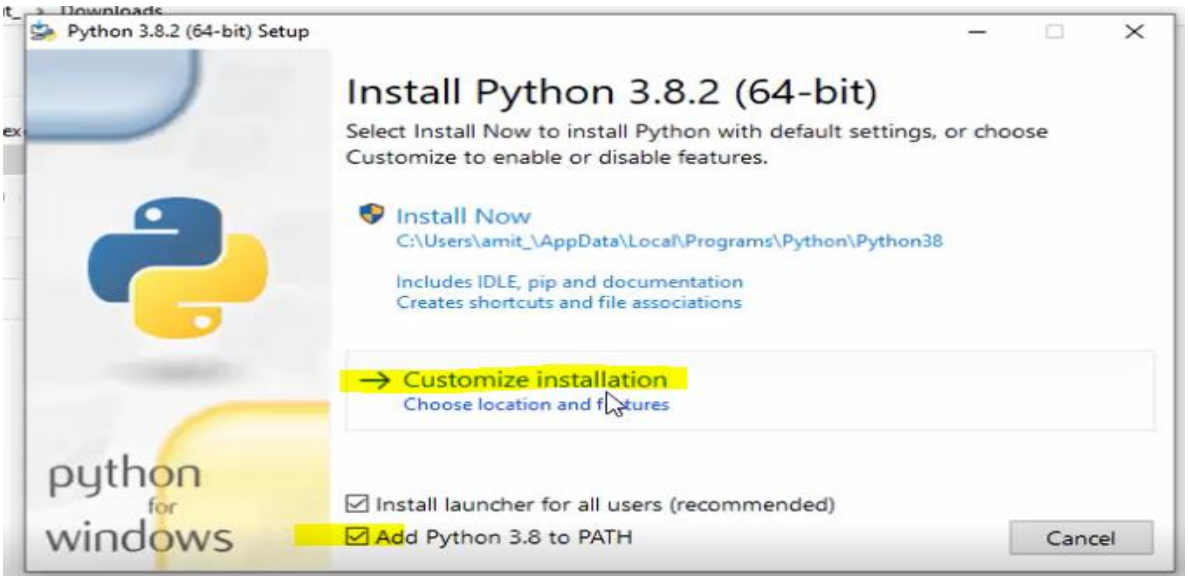


Fig:5.6 Click on Install

Step3: Click on Install NOW After the installation is successful. Click on Close.

Fig:5.7 Python IDLE Setup

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step2: In the Windows Run Command, type “cmd”.

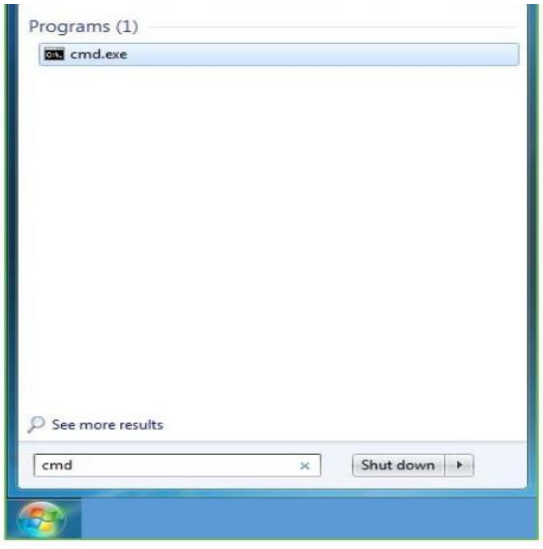


Fig:5.8 Search for command prompt

Step 3: Open the Command prompt option.

Step4: Let us test whether the python is correctly installed. Type python –V and press Enter.

Fig:5.9 Screen shows pycharm

Step 5: You will get the answer as 3.8.2

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step1: Click on Start

Step2: In the Windows Run command, type “python idle”.

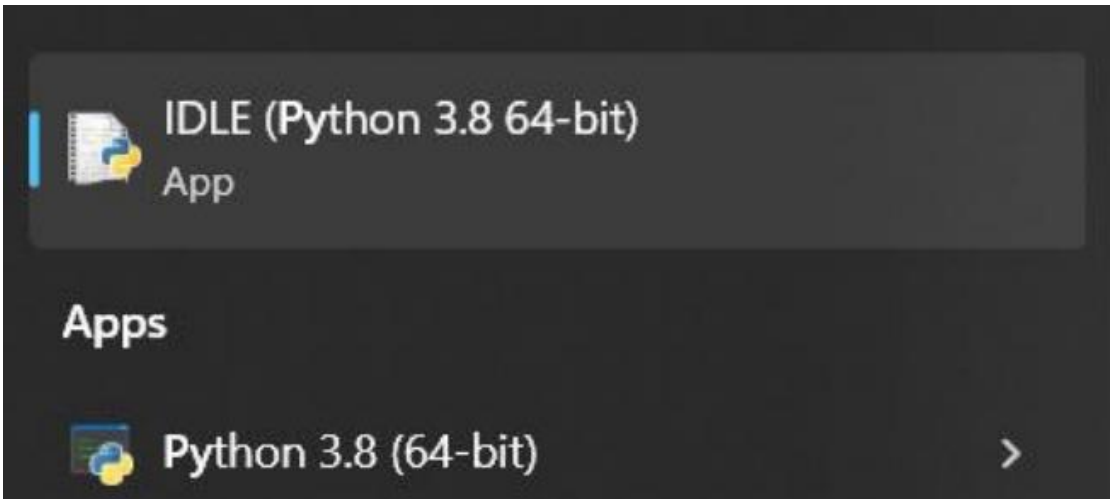


Fig:5.10 Search for python IDLE

Step3: Click on IDLE(Python3.864-bit) and launch the program

Step4: To go a head with working in IDLE you must first save the file. Click on File>Click on Save

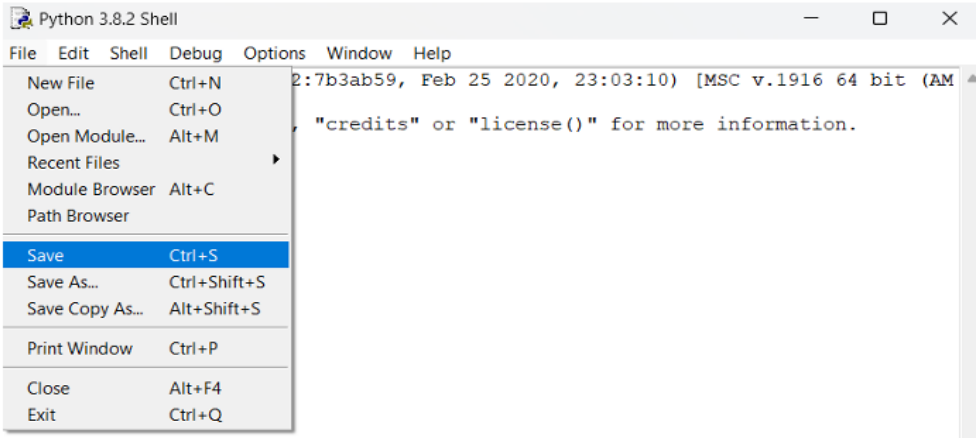


Fig:5.11 Python files

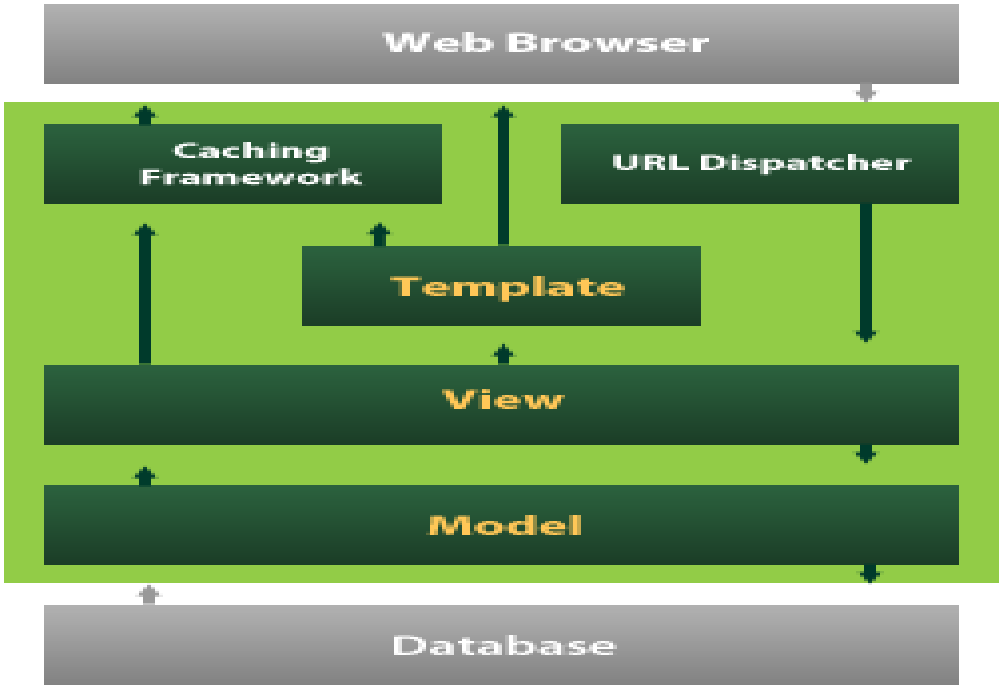
Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step6: Now for e.g. enter print.

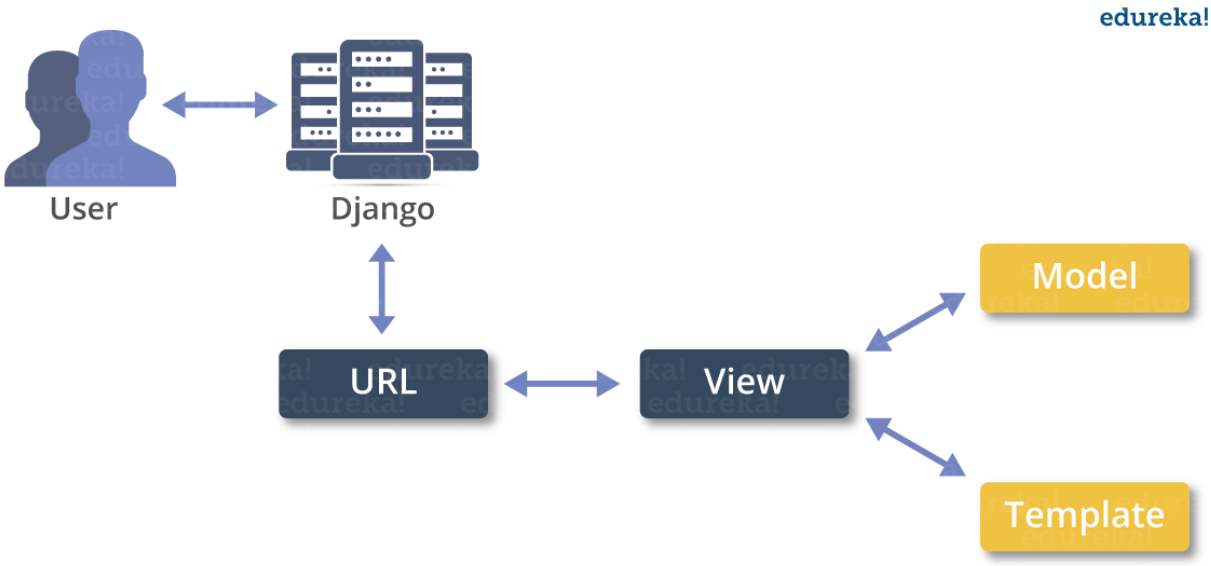
5.2 BACKEND IMPLEMENTATION

DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don’t repeat_yourself. Python is used throughout, even for settings files and data models.



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.



Create a Project

Whether you are on Windows or Linux, just get a terminal or a cmd prompt and navigate to the place you want your project to be created, then use this code −

$ Django -admin start project my project

This will create a "my project" folder with the following structure −

My project/

manage.py

my project/

__init__.py

settings.py

urls.py

wsgi.py

The Project Structure

The “my project” folder is just your project container, it actually contains two elements −

manage.py − This file is kind of your project local Django-admin for interacting with your project via command line (start the development server, sync db...). To get a full list of command accessible via manage.py you can use the code −

$ python manage.py help

The “my project” subfolder − This folder is the actual python package of your project. It contains four files −

__init__.py − Just for python, treat this folder as package.

settings.py − As the name indicates, your project settings.

urls.py − All links of your project and the function to call. A kind of toc of your project.

wsgi.py − If you need to deploy your project over WSGI.

Setting Up Your Project

Your project is set up in the subfolder my project/settings.py. Following are some important options you might need to set −

DEBUG = True

This option lets you set if your project is in debug mode or not. Debug mode lets you get more information about your project's error. Never set it to ‘True’ for a live project. However, this has to be set to ‘True’ if you want the Django light server to serve static files. Do it only in the development mode.

```
DATABASES = {
  'default': {
    'ENGINE': 'django.db.backends.sqlite3',
    'NAME': 'database.sql',
    'USER': '',
    'PASSWORD': '',
    'HOST': '',
    'PORT': '',
  }
}
```

Database is set in the ‘Database’ dictionary. The example above is for SQLite engine. As stated earlier, Django also supports −

MySQL (django.db.backends.my sql)

Post GreSQL (django.db.backends.postgresql_psycopg2)
Oracle (django.db.backends.oracle) and NoSQL DB
MongoDB (django_mongodb_engine)
Before setting any new engine, make sure you have the correct db driver installed.

You can also set others options like: TIME_ZONE, LANGUAGE_CODE, TEMPLATE…

Now that your project is created and configured make sure it's working −

$ python manage.py run server
You will get something like the following on running the above code − Validating models...

0 errors found
September 03, 2015 - 11:41:50
Django version 1.6.11, using settings 'my project .settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

A project is a sum of many applications. Every application has an objective and can be reused into another project, like the contact form on a website can be an application, and can be reused for others. See it as a module of your project.

Create an Application

We assume you are in your project folder. In our main “my project” folder, the same folder then manage.py −

$ python manage.py start app my app
You just created my app application and like project, Django create a “my app” folder with the application structure −

My app/
  __init__.py
  admin.py

models.py

tests.py

views.py

__init__.py − Just to make sure python handles this folder as a package.

admin.py − This file helps you make the app modifiable in the admin interface.

models.py − This is where all the application models are stored.

tests.py − This is where your unit tests are.

views.py − This is where your application views are.

Get the Project to Know About Your Application

At this stage we have our "my app" application, now we need to register it with our Django project "my project". To do so, update INSTALLED_APPS tuple in the settings.py file of your project (add your app name) −

```
INSTALLED_APPS = (
  'django.contrib. admin',
  'django.contrib.auth',
  'django.contrib.contenttypes',
  'django.contrib.sessions',
  'django.contrib.messages',
  'django.contrib.staticfiles',
  'myapp',
)
```

Creating forms in Django, is really similar to creating a model. Here again, we just need to inherit from Django class and the class attributes will be the form fields. Let's add a forms.py file in my app folder to contain our app forms. We will create a login form.

myapp/forms.py

```
#-*- coding: utf-8 -*-
from django import forms

class Login Form(forms. Form):
   user = forms. Char Field(max_length = 100)
   password = forms. Char Field(widget = forms. Password Input())
```

As seen above, the field type can take "widget" argument for html rendering; in our case, we want the password to be hidden, not displayed. Many others widget are present in Django: Date Input for dates, Check box Input for checkboxes, etc.

Using Form in a View

There are two kinds of HTTP requests, GET and POST. In Django, the request object passed as parameter to your view has an attribute called "method" where the type of the request is set, and all data passed via POST can be accessed via the request. POST dictionary.

Let's create a login view in our myapp/views.py −

```
#-*- coding: utf-8 -*-
from my app. forms import Login Form

def login(request):
   username = "not logged in"

   if request. method == "POST":
      #Get the posted form
      My Login Form = Login Form(request.POST)

      if My Login Form. is_valid():
         username = My Login Form. cleaned_data['username']
   else:
      My Login Form = Login form()

   return render(request, 'loggedin.html', {"username" : username})
```

The view will display the result of the login form posted through the loggedin.html. To test it, we will first need the login form template. Let's call it login.html.

```
<html>
  <body>
<form name = "form" action = "{% url "myapp.views.login" %}"
        method = "POST" >{% csrf_token %}

        <div style = "max-width:470px;">
          <center>
            <input type = "text" style = "margin-left:20%;"
                    placeholder = "Identifiant" name = "username" />
          </center>
        </div>

         <br>
            <div style = "max-width:470px;">
              <center>
                <input type = "password" style = "margin-left:20%;"
                  placeholder = "password" name = "password" />
              </center>
            </div>

            <br>

        <div style = "max-width:470px;">
              <center>

            <button style = "border:0px; background-color:#4285F4; margin-top:8%;
                  height:35px; width:80%;margin-left:19%;" type = "submit"
                  value = "Login" >
                  <strong>Login</strong>
                 </button>
```

```
            </center>
        </div>
          </form>

       </body>
</html>
```

The template will display a login form and post the result to our login view above. You have probably noticed the tag in the template, which is just to prevent Cross-site Request Forgery (CSRF) attack on your site.

{% csrf_token %}

Once we have the login template, we need the loggedin.html template that will be rendered after form treatment.

```
<html>

  <body>
    You are : <strong>{{username}}</strong>
  </body>

</html>
```

Now, we just need our pair of URLs to get started: myapp/urls.py

```
from django.conf.urls import patterns, url
from django.views.generic import TemplateView

urlpatterns = patterns('myapp.views',
url(r'^connection/',TemplateView.as_view(template_name = 'login.html')),
url(r'^login/', 'login', name = 'login'))
```

When accessing "/myapp/connection", we will get the following login.html template rendered −

Setting Up Sessions

In Django, enabling session is done in your project settings.py, by adding some lines to the

MIDDLEWARE_CLASSES and the INSTALLED_APPS options. This should be done while creating the project, but it's always good to know, so MIDDLEWARE_CLASSES should have –

'django.contrib.sessions.middleware.SessionMiddleware'
And INSTALLED_APPS should have −

'django.contrib.sessions'
By default, Django saves session information in database (django_session table or collection), but you can configure the engine to store information using other ways like: in file or in cache.

When session is enabled, every request (first argument of any view in Django) has a session (dict) attribute.

Let's create a simple sample to see how to create and save sessions. We have built a simple login system before (see Django form processing chapter and Django Cookies Handling chapter). Let us save the username in a cookie so, if not signed out, when accessing our login page you won't see the login form. Basically, let's make our login system we used in Django Cookies handling more secure, by saving cookies server side.

For this, first lets change our login view to save our username cookie server side –

```
def login(request):
  username = 'not logged in'

  if request.method == 'POST':
    MyLoginForm = LoginForm(request.POST)

    if MyLoginForm.is_valid():
      username = MyLoginForm.cleaned_data['username']
      request.session['username'] = username
    else:
      MyLoginForm = LoginForm()
```

```
  return render(request, 'loggedin.html', {"username" : username}
```

Then let us create formView view for the login form, where we won't display the form if cookie is set −

```
def formView(request):
  if request.session.has_key('username'):
    username = request.session['username']
    return render(request, 'loggedin.html', {"username" : username})
  else:
    return render(request, 'login.html', {})
```

Now let us change the url.py file to change the url so it pairs with our new view −

```
from django.conf.urls import patterns, url
from django.views.generic import TemplateView
urlpatterns = patterns('myapp.views',
  url(r'^connection/','formView', name = 'loginform'),
  url(r'^login/', 'login', name = 'login'))
```

When accessing /myapp/connection, you will get to see the following page

5.2.1Algorithms

Modified Extreme Gradient Boosting (XG BOOST):

This is the primary algorithm used for house price prediction in our project. It is an advanced ensemble learning technique based on gradient boosting. The "modified" version improves traditional XG Boost by optimizing parameters such as learning rate, tree depth, and regularization, resulting in higher accuracy and better generalization. It uses a probabilistic model selection approach, which adapts well to complex, non-linear relationships present in real estate data. Due to its robustness against overfitting and superior performance in handling missing data and outliers, it is ideal for predicting house prices. The model achieved the highest prediction accuracy of **0.82912**, making it the most effective among all models used in this study.

Linear Regression:

A simple model that assumes a linear relationship between input features and the target variable. It is easy to interpret and fast to train, but performs poorly with non-linear patterns.

Ridge Regression:

Enhances linear regression with L2 regularization to reduce model complexity. It helps in reducing overfitting, especially when multicollinearity exists in the data.

Lasso Regression:

Uses L1 regularization, which allows automatic feature selection by shrinking some coefficients to zero. It's useful when we want to simplify the model by keeping only the most important variables.

Decision Tree Regression:

A tree-structured model that splits data based on feature values for prediction. It is easy to understand but can easily overfit if not properly pruned.

Ada Boost Regressor:

Ensemble method that combines weak learners sequentially, each focusing on correcting previous errors for better accuracy.

Random Forest Regression:

An ensemble of multiple decision trees that improves accuracy by averaging predictions. It reduces variance and is generally more stable and robust than a single decision tree.

Gradient Boosting Regression:

Builds an additive model in a forward stage-wise fashion by minimizing the loss function. Each new tree focuses on correcting the errors made by the previous ones for better accuracy.

5.3 SOURCECODE

```
from django.shortcuts import render, HttpResponse
from django.contrib import messages
from sklearn.tree import DecisionTreeClassifier
from .forms import UserRegistrationForm
from .models import UserRegistrationModel
# Create your views here.
def UserRegisterActions(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            print('Data is Valid')
                    form.save()
            messages.success(request, 'You have been successfully registered')
            form = UserRegistrationForm()
            return render(request, 'UserRegistrations.html', {'form': form})
        else:
            messages.success(request, 'Email or Mobile Already Existed')
            print("Invalid form")
    else:
        form = UserRegistrationForm()
    return render(request, 'UserRegistrations.html', {'form': form})
def UserLoginCheck(request):
```

```
    if request.method == "POST":
        loginid = request.POST.get('loginid')
        pswd = request.POST.get('pswd')
        print("Login ID = ", loginid, ' Password = ', pswd)
        try:
            check=UserRegistrationModel.objects.get(loginid=loginid,
password=pswd)
            status = check.status
            print('Status is = ', status)
            if status == "activated":
                request.session['id'] = check.id
                request.session['loggeduser'] = check.name
                request.session['loginid'] = loginid
                request.session['email'] = check.email
                print("User id At", check.id, status)
                return render(request, 'users/UserHomePage.html', {})
            else:
                messages.success(request, 'Your Account Not at activated')
                return render(request, 'UserLogin.html')
        except Exception as e:
            print('Exception is ', str(e))
            pass
        messages.success(request, 'Invalid Login id and password')
    return render(request, 'UserLogin.html', {})
```

```
def UserHome(request):
    return render(request, 'users/UserHomePage.html', {})
            def Viewdata(request):
    import os
    import pandas as pd
    from django.conf import settings
    path =os.path.join(settings.MEDIA_ROOT,'Bengaluru_House_Data.csv')
    #path = os.path.join(settings.MEDIA_ROOT,
"World_Happiness_2015_2017_.csv")
    data = pd.read_csv(path, nrows=500)
    print(data)
    data = data.to_html()
    return render (request, "users/Viewdata.html", {"data":data})
def processdata(request):
    import os
    import pandas as pd
    from django.conf import settings
    path = os.path.join(settings.MEDIA_ROOT,'danaiahprocessdataset.csv')
    data = pd.read_csv(path,nrows=100)
    print(data)
    data = data.to_html
    return render(request,'users/process.html',{'data':data})
def prediction(request):
    if request.method == 'POST':
```

```
    # location,sqft,bath,bhk
    location = request.POST.get('location')
    sqft = request.POST.get('sqft')
    bath= request.POST.get('bath')
    bhk = request.POST.get('bhk')
    userinput = [[location,sqft,bath,bhk]]
    print('*****************************')
    print(userinput)
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    import os
    from django.conf import settings
       path = os.path.join(settings.MEDIA_ROOT + "\\" +
"Bengaluru_House_Data.csv")
    df = pd.read_csv(path)
    # print(df.head())
    df.drop(['society'], axis=1, inplace=True)
    df2 = df.dropna()
    df2['bhk'] = df2['size'].apply(lambda x: int(x.split(' ')[0]))
    def sqft_value(x):
       try:
          float(x)
       except:
```

```
        return False
    return True
df2[~df2.total_sqft.apply(sqft_value)].head(10)
def convert_sqft_to_num(x):
    tokens = x.split('-')
    if len(tokens) == 2:
        return (float(tokens[0])+float(tokens[1]))/2
    try:
        return float(x)
    except:
        return None

df3 = df2.copy()
df3.total_sqft = df3.total_sqft.apply(convert_sqft_to_num)
print('---------convert_sqft_to_num---------------')
df4 = df3.copy()
df4['price_per_sqft'] = (df4['price']*100000) / df4['total_sqft']
df4.location = df4.location.apply(lambda x: x.strip())
location_stats = df4['location'].value_counts(ascending=False)
# location_stats
location_stats_less_than_10 = location_stats[location_stats<=10]
# location_stats_less_than_10
print('-----------location_stats_less_than_10------------')
df4.location = df4.location.apply(lambda x: 'other' if x in
location_stats_less_than_10       else x)
```

```
        # len(df4.location.unique())
        df4[df4.total_sqft/df4.bhk<300].head()
        df5 = df4[~(df4.total_sqft/df4.bhk<300)]
        print(df5['location'].value_counts())
        def remove_pps_outliers(df):
            df_out = pd.DataFrame()
            for key, subdf in df.groupby('location'):
                m = np.mean(subdf.price_per_sqft)
                st = np.std(subdf.price_per_sqft)
                reduced_df = subdf[(subdf.price_per_sqft>(m-st)) &
(subdf.price_per_sqft<=(m+st))]
                df_out = pd.concat([df_out,reduced_df],ignore_index=True)
            return df_out
        df6 = remove_pps_outliers(df5)
        print('------outlier romved------')
        df9 = df6.drop(['area_type', 'availability', 'balcony',
'size','price_per_sqft'],axis='columns')
                dummies = pd.get_dummies(df9.location)
                df10 =
       pd.concat([df9,dummies.drop('other',axis='columns')],axis='columns')
      df11 = df10.drop('location',axis='columns')
      X = df11.drop(['price'],axis='columns')
      Y = df11.price
      #split
      print('-----splitting------')
```

```
        from sklearn.model_selection import train_test_split
        X_train, X_test, Y_train, Y_test =
train_test_split(X,Y,test_size=0.2,random_state=10)
        from sklearn.model_selection import GridSearchCV
        # We are going to compare 3 regression models
        from sklearn.linear_model import LinearRegression
        from sklearn.linear_model import Lasso
        from sklearn.tree import DecisionTreeRegressor
        lr_clf = LinearRegression()
        lr_clf.fit(X_train,Y_train)
        #lr_clf.score(X_test,y_test)
        # Predict test set
        y_pred = lr_clf.predict(X_test)
        # Check the accuracy of the model
        score = lr_clf.score(X_test, Y_test)
        print('Score ====> ',score)
        def predict_price(location,sqft,bath,bhk):
            loc_index = np.where(X.columns==location)[0][0]
            x = np.zeros(len(X.columns))
            x[0] = sqft
            x[1] = bath
            x[2] = bhk
            if loc_index >= 0:
                x[loc_index] = 1
            return lr_clf.predict([x])[0]
```

```
        pred_res = predict_price(location,int(sqft), int(bath), int(bhk))
        pred_res1 = round(pred_res,2)
        print('===== > ',pred_res1)
        return render(request, "users/predictions_form.html",
{"test_pred":pred_res1})
    else:
        return render(request, 'users/predictions_form.html', {})
    return render(request, 'users/predictions_form.html')
def ML(request):
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    import os
    from django.conf import settings
   #  df =
pd.read_csv('C:\\Users\\MMC\\Downloads\\Bengaluru_House_Data.csv')
    path = os.path.join(settings.MEDIA_ROOT + "\\" +
"Bengaluru_House_Data.csv")
    df = pd.read_csv(path)
    df2 = df.dropna()
    df2['bhk'] = df2['size'].apply(lambda x: int(x.split(' ')[0]))
    def sqft_value(x):
        try:
            float(x)
```

```
    except:
        return False
    return True
df2[~df2.total_sqft.apply(sqft_value)].head(10)
def convert_sqft_to_num(x):
        tokens = x.split('-')
        if len(tokens) == 2:
            return (float(tokens[0])+float(tokens[1]))/2
        try:
            return float(x)
        except:
            return None
             df3 = df2.copy()
df3.total_sqft = df3.total_sqft.apply(convert_sqft_to_num)
print('---------convert_sqft_to_num---------------')
df4 = df3.copy()
df4['price_per_sqft'] = (df4['price']*100000) / df4['total_sqft']
df4.location = df4.location.apply(lambda x: x.strip())
location_stats = df4['location'].value_counts(ascending=False)
# location_stats
location_stats_less_than_10 = location_stats[location_stats<=10]
# location_stats_less_than_10
#  print('-----------location_stats_less_than_10------------')
```

```
    df4.location = df4.location.apply(lambda x: 'other' if x in
location_stats_less_than_10 else x)
   #  # len(df4.location.unique())
    df4[df4.total_sqft/df4.bhk<300].head()
    df5 = df4[~(df4.total_sqft/df4.bhk<300)]
   #  print(df5['location'].value_counts())
     def remove_pps_outliers(df):
         df_out = pd.DataFrame()
         for key, subdf in df.groupby('location'):
             m = np.mean(subdf.price_per_sqft)
             st = np.std(subdf.price_per_sqft)
              reduced_df = subdf[(subdf.price_per_sqft>(m-st)) &
 (subdf.price_per_sqft<=(m+st))]
             df_out = pd.concat([df_out,reduced_df],ignore_index=True)
         return df_out
    df6 = remove_pps_outliers(df5)
          #  print('------outlier romved------')
    df9 = df6.drop(['area_type', 'availability', 'balcony',
 'size','price_per_sqft','society'],axis='columns')
    dummies = pd.get_dummies(df9.location)
    df10 =
 pd.concat([df9,dummies.drop('other',axis='columns')],axis='columns')
    df11 = df10.drop('location',axis='columns')
    X = df11.drop(['price'],axis='columns')
    Y = df11.price
```

```
    #split
    print('-----splitting------')
    from sklearn.model_selection import train_test_split
     X_train, X_test, Y_train, Y_test =
train_test_split(X,Y,test_size=0.2,random_state=10)
    from sklearn.model_selection import GridSearchCV
    # We are going to compare 3 regression models
       #algrothem-1
    from sklearn.linear_model import LinearRegression
    lr_clf = LinearRegression()
    lr_clf.fit(X_train,Y_train)
   #  accuracy = lr_clf.score(X_test,Y_test)*100
    # Predict test set
    y_pred = lr_clf.predict(X_test)
     from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score
   #  accuracy = accuracy_score(Y_test, y_pred) * 100
    accuracy = lr_clf.score(X_test,Y_test)*100
   #  clf.score(X_test, y_test)
   #  accuracy = accuracy_score(Y_test, Y_test)
    print('Accuracy:', accuracy)
   #  from sklearn.metrics import precision_score
   #  precision = precision_score(Y_test, y_pred) * 100
   #  print('Precision Score:', precision)
   # #  from sklearn.metrics import recall_score
```

```
  #  recall = recall_score(Y_test, y_pred) * 100
  #  print('recall_score:',recall)
  # #  from sklearn.metrics import f1_score
  #  f1score = f1_score(Y_test, y_pred) * 100
  #  print('f1score:',f1score)
 #second algrothem
   from sklearn.linear_model import Lasso
   lasso = Lasso()
   lasso.fit(X_train,Y_train)
   y_pred1 = lasso.predict(X_test)
   from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score
  #  accuracy1 = accuracy_score(Y_test, y_pred1) * 100
   accuracy1 = lasso.score(X_test,Y_test)*100
   print('Accuracy:', accuracy1)
  # #  from sklearn.metrics import precision_score
  #  precision1 = precision_score(Y_test, y_pred1) * 100
  #  print('Precision Score:', precision1)
  # #  from sklearn.metrics import recall_score
  #  recall1 = recall_score(Y_test, y_pred1) * 100
  #  print('recall_score:',recall1)
  # #  from sklearn.metrics import f1_score
  #  f1score1 = f1_score(Y_test, y_pred1) * 100
  #  print('f1score:',f1score1)
```

```
#third algrothem
   from sklearn.tree import DecisionTreeRegressor
   DTR = DecisionTreeRegressor()
   DTR.fit(X_train,Y_train)
   y_pred2 = DTR.predict(X_test)
   from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score
  #  accuracy2 = accuracy_score(Y_test, y_pred2) * 100
   accuracy2 = lasso.score(X_test,Y_test)*100
   print('Accuracy:', accuracy2)
  # #  from sklearn.metrics import precision_score
  #  precision2 = precision_score(Y_test, y_pred2) * 100
  #  print('Precision Score:', precision2)
  # #  from sklearn.metrics import recall_score
  #  recall2 = recall_score(Y_test, y_pred2) * 100
  #  print('recall_score:',recall2)
  # #  from sklearn.metrics import f1_score
  #  f1score2 = f1_score(Y_test, y_pred2) * 100
  #  print('f1score:',f1score2)
#fourth algrothem
   from sklearn.ensemble import  RandomForestRegressor
   RFR  = RandomForestRegressor()
   RFR.fit(X_train,Y_train)
   y_pred3 = RFR.predict(X_test)
   from sklearn.metrics import
```

```
accuracy_score,precision_score,recall_score,f1_score
  #  accuracy3 = accuracy_score(Y_test, y_pred3) * 100
   accuracy3 = RFR.score(X_test,Y_test)*100
   print('Accuracy:', accuracy3)
  # #  from sklearn.metrics import precision_score
  #  precision3 = precision_score(Y_test, y_pred3) * 100
  #  print('Precision Score:', precision3)
  # #  from sklearn.metrics import recall_score
  #  recall3 = recall_score(Y_test, y_pred3) * 100
  #  print('recall_score:',recall3)
  # #  from sklearn.metrics import f1_score
  #  f1score3 = f1_score(Y_test, y_pred3) * 100
  #  print('f1score:',f1score3)
#fifith algrothem
   from sklearn.ensemble import AdaBoostRegressor
   ABR = AdaBoostRegressor()
   ABR.fit(X_train,Y_train)
   y_pred4 = ABR.predict(X_test)
   from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score
    #  accuracy4 = accuracy_score(Y_test, y_pred4) * 100
   accuracy4 = ABR.score(X_test,Y_test)*100
    print('Accuracy:', accuracy4)
    # #  from sklearn.metrics import precision_score
    #  precision4 = precision_score(Y_test, y_pred) * 100
```

```
    #  print('Precision Score:', precision4)
    # #  from sklearn.metrics import recall_score
    #  recall4 = recall_score(Y_test, y_pred4) * 100
    #  print('recall_score:',recall4)
    # #  from sklearn.metrics import f1_score
    #  f1score4 = f1_score(Y_test, y_pred4) * 100
    #  print('f1score:',f1score4)
#sixth algrothem
    from sklearn.ensemble import GradientBoostingRegressor
    GBR = GradientBoostingRegressor()
    GBR.fit(X_train,Y_train)
    y_pred5 = GBR.predict(X_test)
     from sklearn.metrics import
 accuracy_score,precision_score,recall_score,f1_score
   #  accuracy5 = accuracy_score(Y_test, y_pred5) * 100
    accuracy5 = GBR.score(X_test,Y_test)*100
    print('Accuracy:', accuracy5)
   # #  from sklearn.metrics import precision_score
   #  precision5 = precision_score(Y_test, y_pred5) * 100
   #  print('Precision Score:', precision5)
   # #  from sklearn.metrics import recall_score
   #  recall5 = recall_score(Y_test, y_pred5) * 100
   #  print('recall_score:',recall5)
   # #  from sklearn.metrics import f1_score
   #  f1score5 = f1_score(Y_test, y_pred5) * 100
```

```
   #  print('f1score:',f1score5)
#seven algrothem
    from xgboost import XGBRegressor
    XGB = XGBRegressor()
    XGB.fit(X_train,Y_train)
    y_pred6 = XGB.predict(X_test)
     from sklearn.metrics import
 accuracy_score,precision_score,recall_score,f1_score
   #  accuracy6 = accuracy_score(Y_test, y_pred6) * 100
    accuracy6 = lasso.score(X_test,Y_test)*100
    print('Accuracy:', accuracy6)
   # #  from sklearn.metrics import precision_score
   #  precision6 = precision_score(Y_test, y_pred6) * 100
   #  print('Precision Score:', precision6)
   # #  from sklearn.metrics import recall_score
   #  recall6 = recall_score(Y_test, y_pred6) * 100
    #  print('recall_score:',recall6)
    # #  from sklearn.metrics import f1_score
    #  f1score6 = f1_score(Y_test, y_pred6) * 100
    #  print('f1score:',f1score6)
       accuracy = {'LR':accuracy,  'LASSO': accuracy1,  'DTR': accuracy2,
'RFR':accuracy3,  'ABR':accuracy4,  'GBR':accuracy5,  'XBR':accuracy6}
       #  accuracy = {'LR':accuracy,  'LASSO': accuracy1,  'DTR':
 accuracy2,  'RFR':accuracy3,  'ABR':accuracy4,  'GBR':accuracy5,
 'XBR':accuracy6}
```

```
  #  precision ={'LR':precision, 'LASSO': precision1, 'DTR': precision2,
 'RFR':precision3, 'ABR':precision4,  'GBR':precision5,  'XBR':precision6}
  #  recall = {'LR': recall,    'LASSO': recall1,    'DTR': recall2,
 'RFR':recall3,    'ABR':recall4,    'GBR':recall5,     'XBR':recall6}
  #  f1score = {'LR':f1score,   'LASSO':f1score1,   'DTR':f1score2,
 'RFR':f1score3,    'ABR':f1score4,    'GBR':f1score5,    'XBR':f1score6}
  # # confusionmatrix = {'DT':confusionmatrix, 'KNN':confusionmatrix2,
 'RF':confusionmatrix3}
  #'confusionmatrix:',confusionmatrix
  # roc = {'RF': roc1, 'SVM': roc2, 'LogisticRegression': roc,  'MLP': roc5}
   return render(request, 'users/ML.html',
              {"accuracy": accuracy})

    #   {"accuracy": accuracy, "precision": precision, "recall":recall,
 'f1score':f1score})
```

base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>house_price_prediction:</title>
  <meta content="" name="description">
  <meta content="" name="keywords">
```

```
<!-- Favicons -->
<link href="{%static 'img/favicon.png'%}" rel="icon">
<link href="{%static 'img/apple-touch-icon.png'%}" rel="apple-touch-icon">
<!-- Google Fonts -->
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,
400i,700,700i|Roboto:100,300,400,500,700|Philosopher:400,400i,700,70
0i" rel="stylesheet">
<!-- Vendor CSS Files -->
<link href="{%static 'vendor/aos/aos.css'%}" rel="stylesheet">
<link href="{%static 'vendor/bootstrap/css/bootstrap.min.css'%}"
rel="stylesheet">
<link href="{%static 'vendor/bootstrap-icons/bootstrap-icons.css'%}"
rel="stylesheet">
<link href="{%static 'vendor/glightbox/css/glightbox.min.css'%}"
rel="stylesheet">
<link href="{%static 'vendor/swiper/swiper-bundle.min.css'%}"
rel="stylesheet">
<!-- Template Main CSS File -->
<link href="{%static 'css/style.css'%}" rel="stylesheet">
<!--
==========================================================
* Template Name: eStartup
* Updated: Jul 27 2023 with Bootstrap v5.3.1
* Template URL: https://bootstrapmade.com/estartup-bootstrap-landing-page-
```

```
 template/
 * Author: BootstrapMade.com
 * License: https://bootstrapmade.com/license/
 ======================================================== --
 >
</head>
<body>
 <!-- ======= Header ======= -->
 <header id="header" class="header fixed-top d-flex align-items-center">
  <div class="container d-flex align-items-center justify-content-between">
   <div id="logo">
    <h1><a href="{% url 'index' %}"><span>HOUSE_</span>
 House_Price_Prediction</a></h1>
   </div>
   <nav id="navbar" class="navbar">
    <ul>
     <li><a class="nav-link scrollto" href="{% url 'index'
 %}">HOME</a></li>
     <li><a class="nav-link scrollto" href="{% url 'AdminLogin'
 %}">ADMIN-LOGIN</a></li>
     <li><a class="nav-link scrollto" href="{% url 'UserLogin' %}">USER-
 LOGIN</a></li>
     <li><a class="nav-link scrollto" href="{% url 'UserRegister'
 %}">USER-REGISTER</a></li>
```

```
    </nav><!-- .navbar -->
  </div>
</header><!-- End Header -->
<!-- ======= Hero Section ======= -->
<br><br>
{%block contents%}
{%endblock%}
<section id="hero">
  <div class="hero-container" data-aos="fade-in">
    <h1>house_price_prediction:</h1>
    <h2> developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house</h2>
    <!-- <h1>What is house_price_prediction?</h1>
    <h2>house_price_prediction is an open-source deep learning library developed by Intel that makes it easy to benchmark different anomaly detection algorithms on both public and custom datasets, all by simply modifying a config file.</h2>- -->
    <img src="../static/img/dani.jpg" alt="Hero Imgs" data-aos="zoom-out" data-aos-delay="100">
  </div>
</section><!-- End Hero Section -->
<main id="main">
```

```
    <!-- ======= Newsletter Section ======= -->
    <section id="newsletter" class="newsletter text-center">
      <div class="overlay padd-section">
        <div class="container" data-aos="zoom-in">


          <div class="row justify-content-center">
            <div class="col-md-9 col-lg-6">
            </div>
          </div>
      </div>
    </div>
  </section><!-- End Newsletter Section -->/


 </main><!-- End #main –
<!-- ======= Footer ======= -->
 <footer class="footer">
  <div class="container">
    <div class="row">
  <div class="copyrights">
    <div class="container">
      <p>&copy; Copyrights <b>house_price_prediction.</b> All rights
 reserved.</p>
      <div class="credits">
        Designed by <a href="#"><b>house_price_prediction.</b></a>
```

```
      </div>
    </div>
  </div>
 </footer><!-- End  Footer -->
 <a href="#" class="back-to-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>
 <!-- Vendor JS Files -->
 <script src="{% static 'vendor/aos/aos.js'%}"></script>
 <script src="{% static
'vendor/bootstrap/js/bootstrap.bundle.min.js'%}"></script>
 <script src="{% static 'vendor/glightbox/js/glightbox.min.js'%}"></script>
 <script src="{% static 'vendor/swiper/swiper-bundle.min.js'%}"></script>
 <script src="{% static 'vendor/php-email-form/validate.js'%}"></script>
 <!-- Template Main JS File -->
 <script src="{% static 'js/main.js'%}"></script>
</body>
</html>
```

urls.py

```
"""house_price_prediction URL Configuration
The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
```

Class-based views

1. Add an import: from other_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

```
from django.contrib import admin
from django.urls import path
from house_price_prediction import views as mainView
from admins import views as admins
from users import views as usr
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", mainView.index, name="index"),
    path("AdminLogin/", mainView.AdminLogin, name="AdminLogin"),
    path("UserLogin/", mainView.UserLogin, name="UserLogin"),
    path("UserRegister/", mainView.UserRegister, name="UserRegister"),
    # Adminviews
    path("AdminLoginCheck/", admins.AdminLoginCheck,
name="AdminLoginCheck"),
    path("AdminHome/", admins.AdminHome, name="AdminHome"),
    path('RegisterUsersView/', admins.RegisterUsersView,
name='RegisterUsersView'),
    path('ActivaUsers/', admins.ActivaUsers, name='ActivaUsers'),
    path('DeleteUsers/', admins.DeleteUsers, name='DeleteUsers'),
```

```
        # User Views
        path("UserRegisterActions/", usr.UserRegisterActions,
name="UserRegisterActions"),
        path("UserLoginCheck/", usr.UserLoginCheck,
name="UserLoginCheck"),
        path("UserHome/", usr.UserHome, name="UserHome"),
        path('Viewdata',usr.Viewdata,name='Viewdata'),
        path("processdata/", usr.processdata, name="processdata"),
        path("ML/", usr.ML, name="ML"),
        path("prediction",usr.prediction,name="prediction")
    ]
    admin/base.html
    {% load static %}
    <!DOCTYPE html>
    <html lang="en">
    <head>
      <meta charset="utf-8">
      <meta content="width=device-width, initial-scale=1.0"
    name="viewport">
      <title>house_price_prediction: A Deep Learning Library for
    Anomaly Detection</title>
      <meta content="" name="description">
      <meta content="" name="keywords">
      <!-- Favicons -->
      <link href="{%static 'img/favicon.png'%}" rel="icon">
      <link href="{%static 'img/apple-touch-icon.png'%}" rel="apple-touch-
    icon">
      <!-- Google Fonts -->
      <link
    href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i
    ,400,400i,700,700i|Roboto:100,300,400,500,700|Philosopher:400,40
    0i,700,700i" rel="stylesheet">
```

```
  <!-- Vendor CSS Files -->
  <link href="{%static 'vendor/aos/aos.css'%}" rel="stylesheet">
  <link href="{%static 'vendor/bootstrap/css/bootstrap.min.css'%}"
rel="stylesheet">
  <link href="{%static 'vendor/bootstrap-icons/bootstrap-icons.css'%}"
rel="stylesheet">
  <link href="{%static 'vendor/glightbox/css/glightbox.min.css'%}"
rel="stylesheet">
  <link href="{%static 'vendor/swiper/swiper-bundle.min.css'%}"
rel="stylesheet">
  <!-- Template Main CSS File -->
  <link href="{%static 'css/style.css'%}" rel="stylesheet">
  <!--
=======================================================
=
  * Template Name: eStartup
  * Updated: Jul 27 2023 with Bootstrap v5.3.1
  * Template URL: https://bootstrapmade.com/estartup-bootstrap-
landing-page-template/
  * Author: BootstrapMade.com
  * License: https://bootstrapmade.com/license/

=======================================================
== -->
</head>

<body>

  <!-- ======= Header ======= -->
  <header id="header" class="header fixed-top d-flex align-items-
center">
    <div class="container d-flex align-items-center justify-content-
between">
```

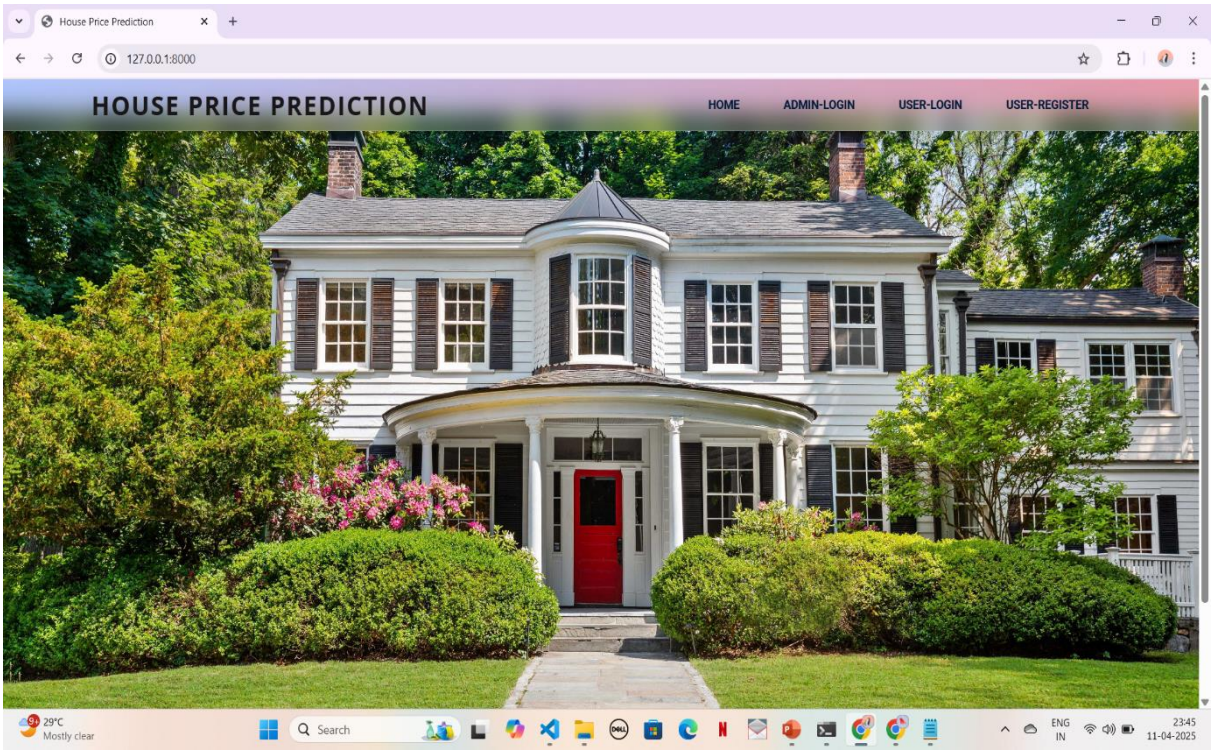
```
    <div id="logo">
      <h1><a href="index.html"><span>House_</span>
Price_Prediction</a></h1>
    </div>
    <nav id="navbar" class="navbar">
      <ul>
        <li><a class="nav-link scrollto " href="{% url 'AdminHome'
%}">Home</a></li>
        <li><a class="nav-link scrollto" href="{% url
'RegisterUsersView' %}">User Details</a></li>
        <li><a class="nav-link scrollto" href="{% url 'index'
%}">Logout</a></li>
          </nav><!-- .navbar -->
   </div>
  </header><!-- End Header -->
  <!-- ======= Hero Section ======= -->
  <br><br> <br><br>
  {%block contents%}
  {%endblock%}
  <section id="hero">
    <div class="hero-container" data-aos="fade-in">
     <!-- <h1>house_price_prediction: A Deep Learning Library for
Anomaly Detection</h1>
     <h2>This paper introduces house_price_prediction, a novel
library for unsupervised anomaly detection and localization.</h2>
     -->
     <img src="../static/img/dani.jpg" alt="Hero Imgs" data-aos="zoom-
out" data-aos-delay="100">
    </div>
  </section><!-- End Hero Section -->
  <main id="main">
```

```
    <!-- ======= Newsletter Section ======= -->
    <section id="newsletter" class="newsletter text-center">
      <div class="overlay padd-section">
        <div class="container" data-aos="zoom-in">
          <div class="row justify-content-center">
            <div class="col-md-9 col-lg-6">
            </div>
          </div>
        </div>
      </div>
    </section><!-- End Newsletter Section -->/
  </main><!-- End #main -->
  <!-- ======= Footer ======= -->
  <footer class="footer">
    <div class="container">
      <div class="row">
    <div class="copyrights">
      <div class="container">
        <p>&copy; Copyrights <b>house_price_prediction.</b> All rights
reserved.</p>
        <div class="credits">
          Designed by <a href="#"><b>house_price_prediction.</b></a>
        </div>
      </div>
    </div>
  </footer><!-- End  Footer -->
  <a href="#" class="back-to-top d-flex align-items-center justify-
content-center"><i class="bi bi-arrow-up-short"></i></a>
  <!-- Vendor JS Files -->
  <script src="{% static 'vendor/aos/aos.js'%}"></script>
  <script src="{% static
'vendor/bootstrap/js/bootstrap.bundle.min.js'%}"></script>
```

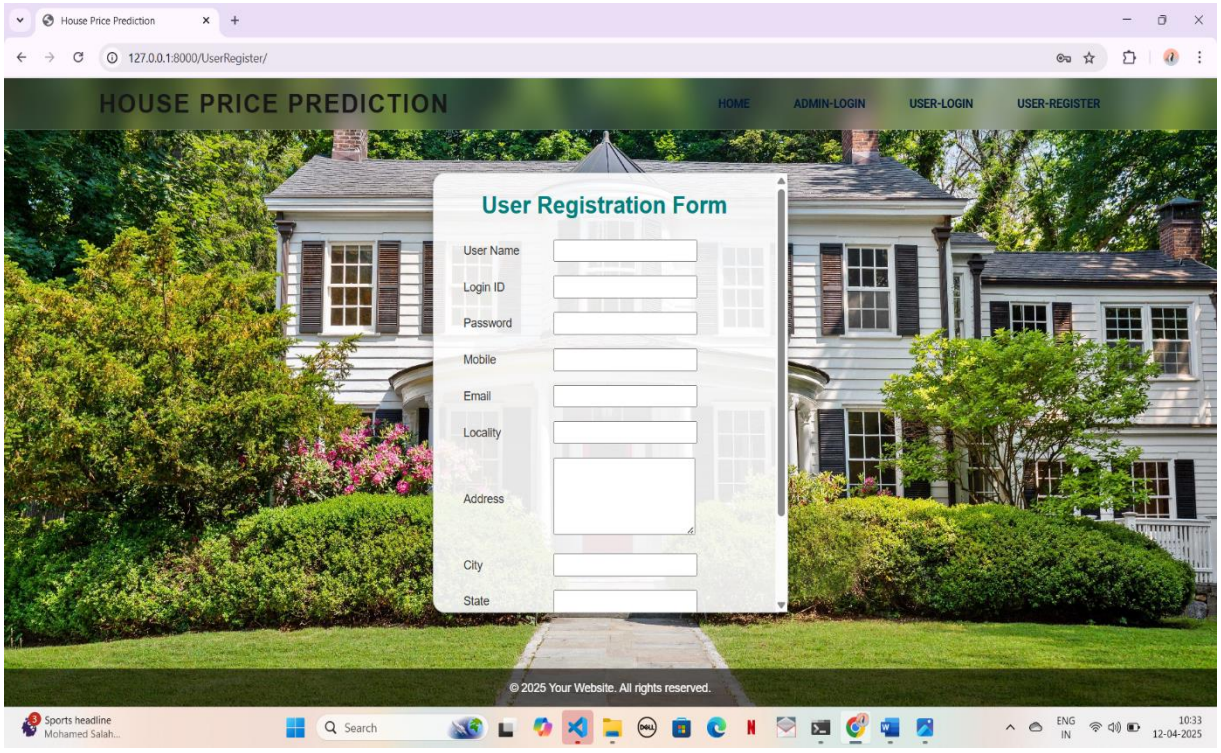
```
  <script src="{% static
'vendor/glightbox/js/glightbox.min.js'%}"></script>
  <script src="{% static 'vendor/swiper/swiper-
bundle.min.js'%}"></script>
  <script src="{% static 'vendor/php-email-
form/validate.js'%}"></script>
  <!-- Template Main JS File -->
  <script src="{% static 'js/main.js'%}"></script>
</body>
</html>
```

5. 4 OUTPUT SCREENS

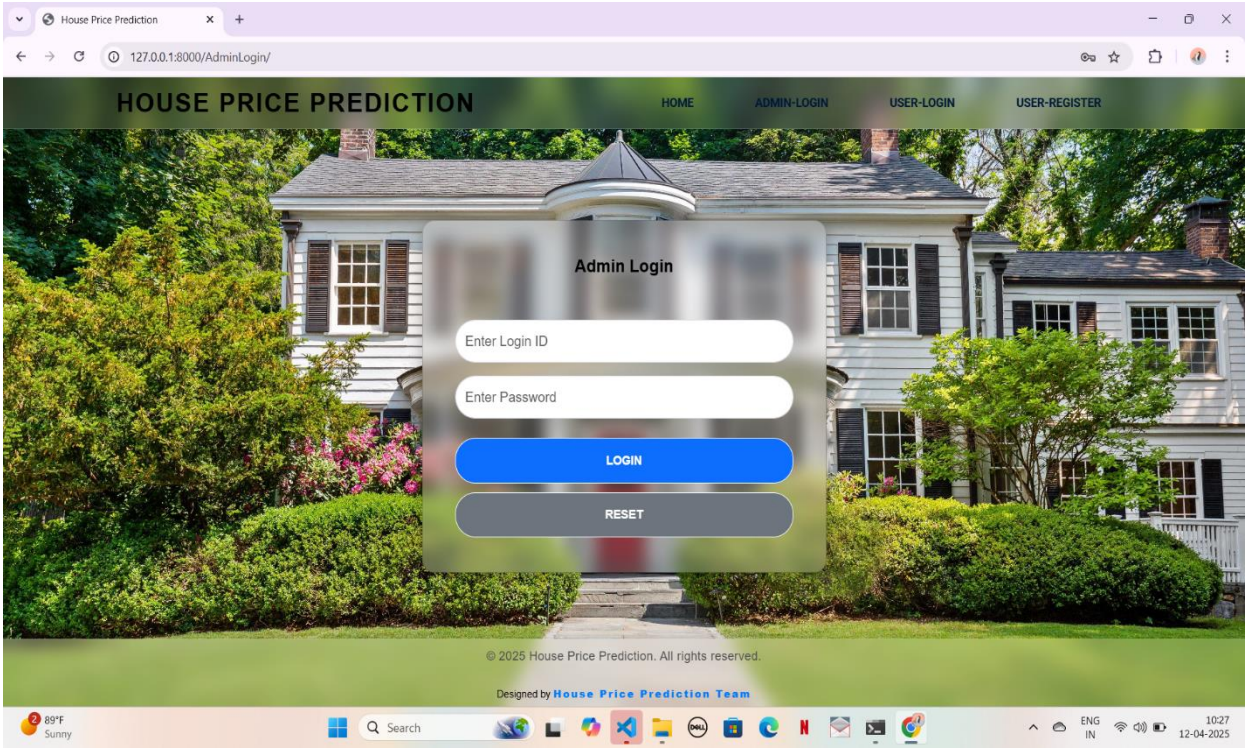
HOME PAGE



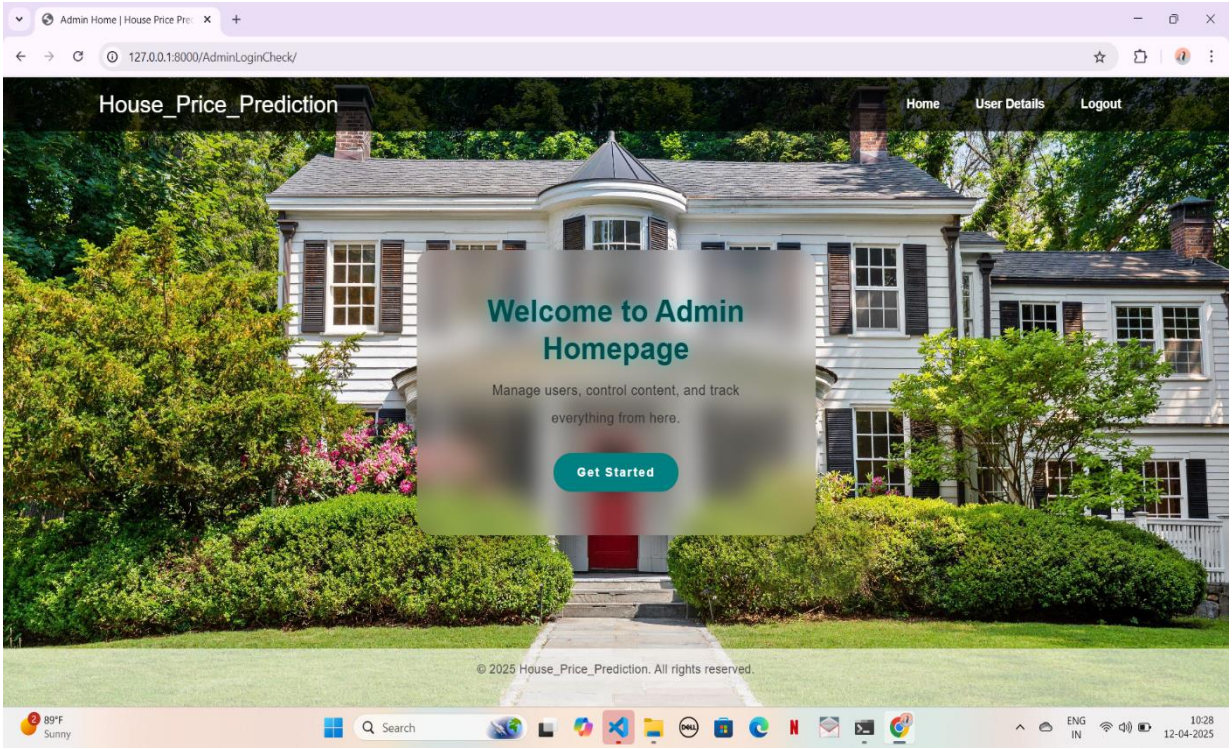
REGISTRATION PAGE



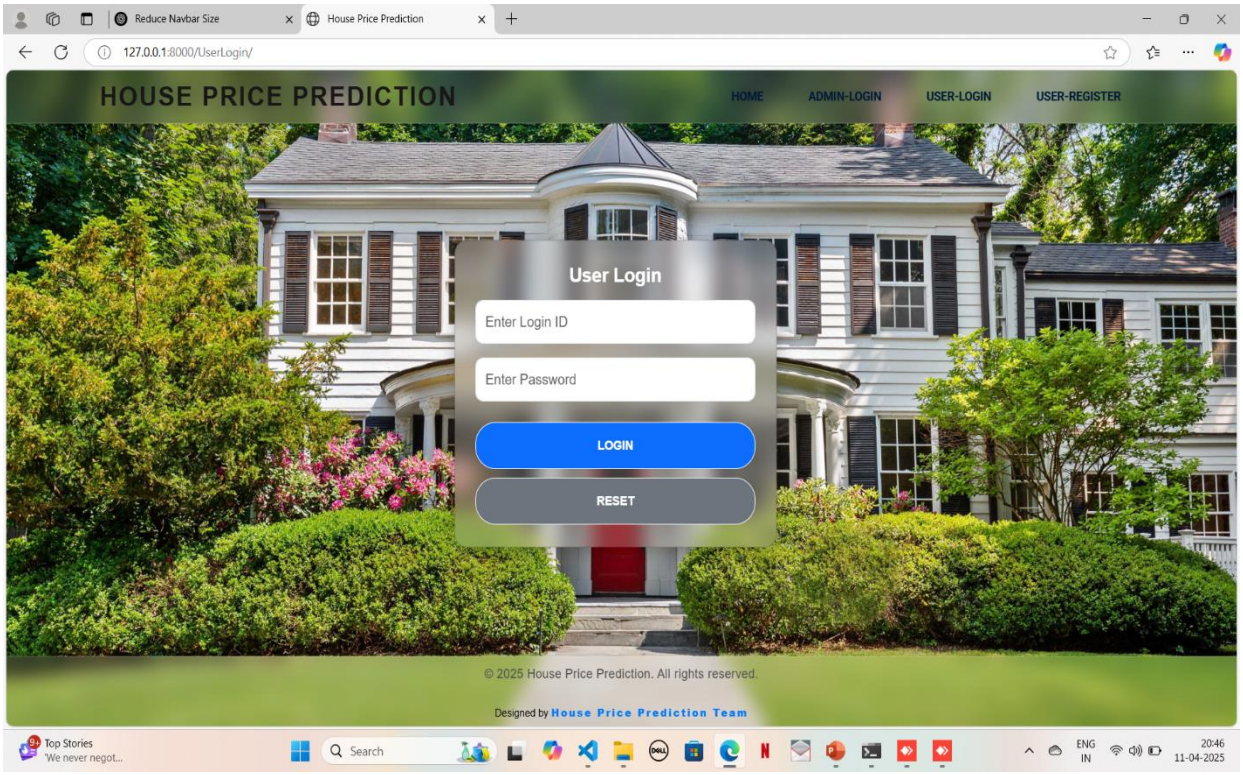
ADMIN LOGIN PAGE



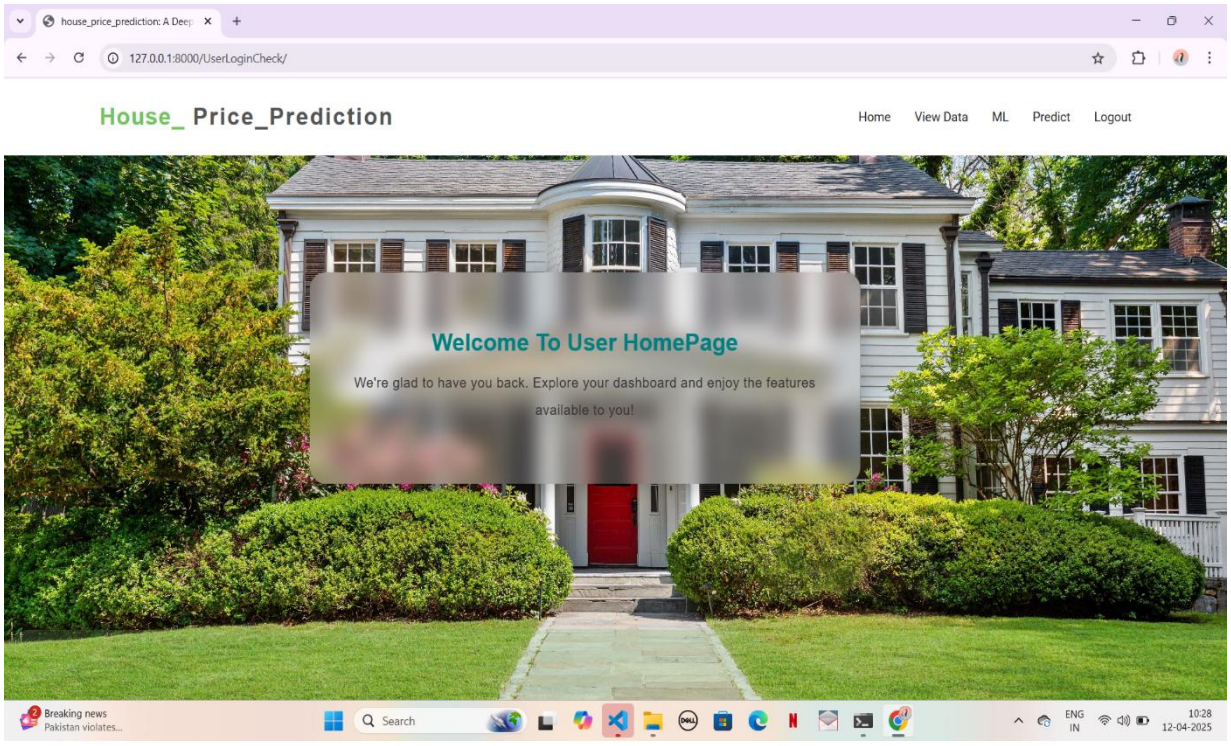
ADMIN HOME PAGE



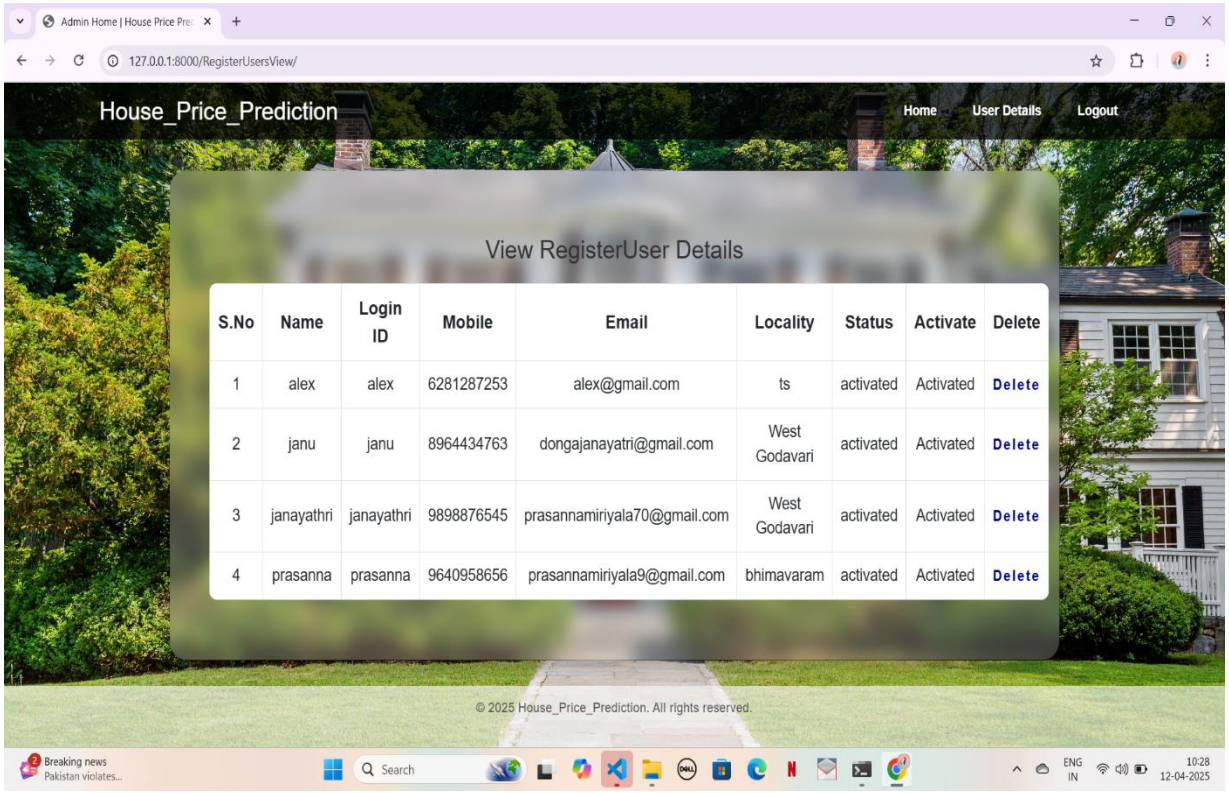
USER LOGIN PAGE



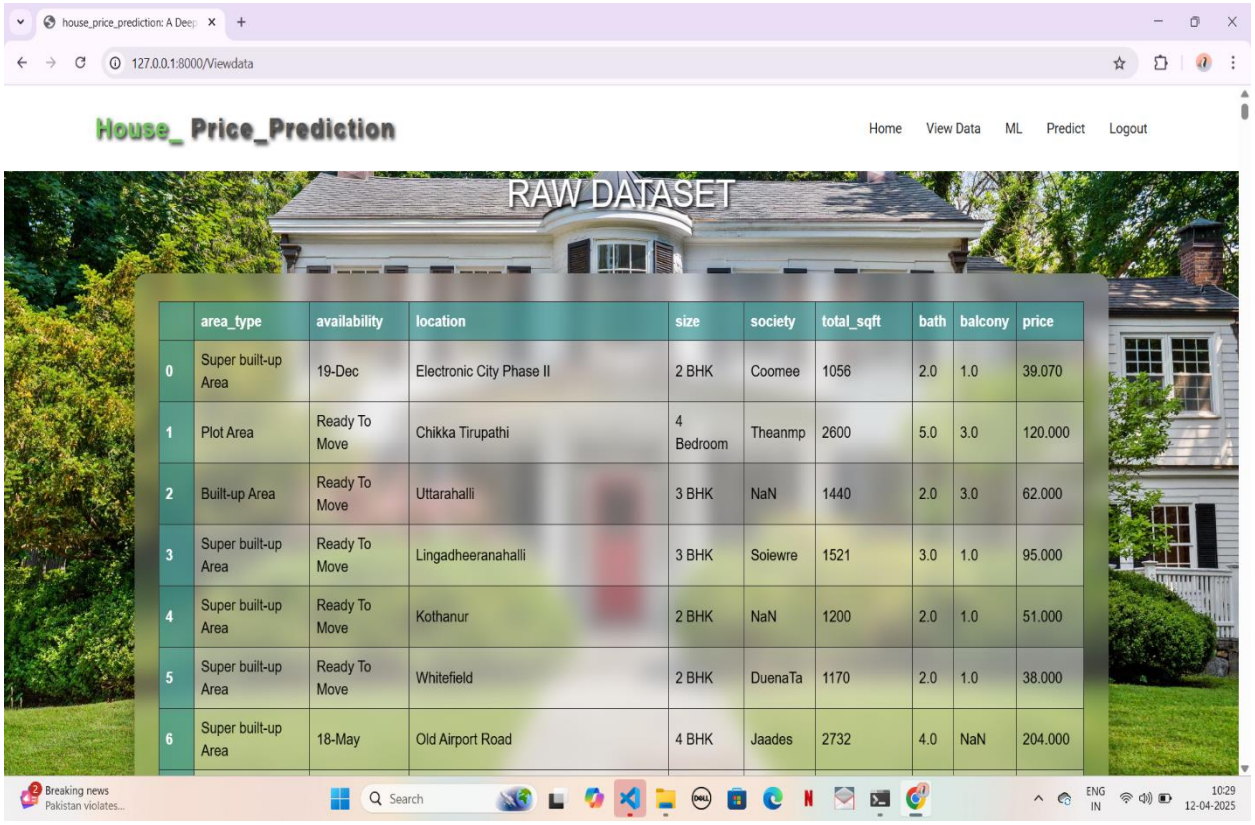
USER HOME PAGE



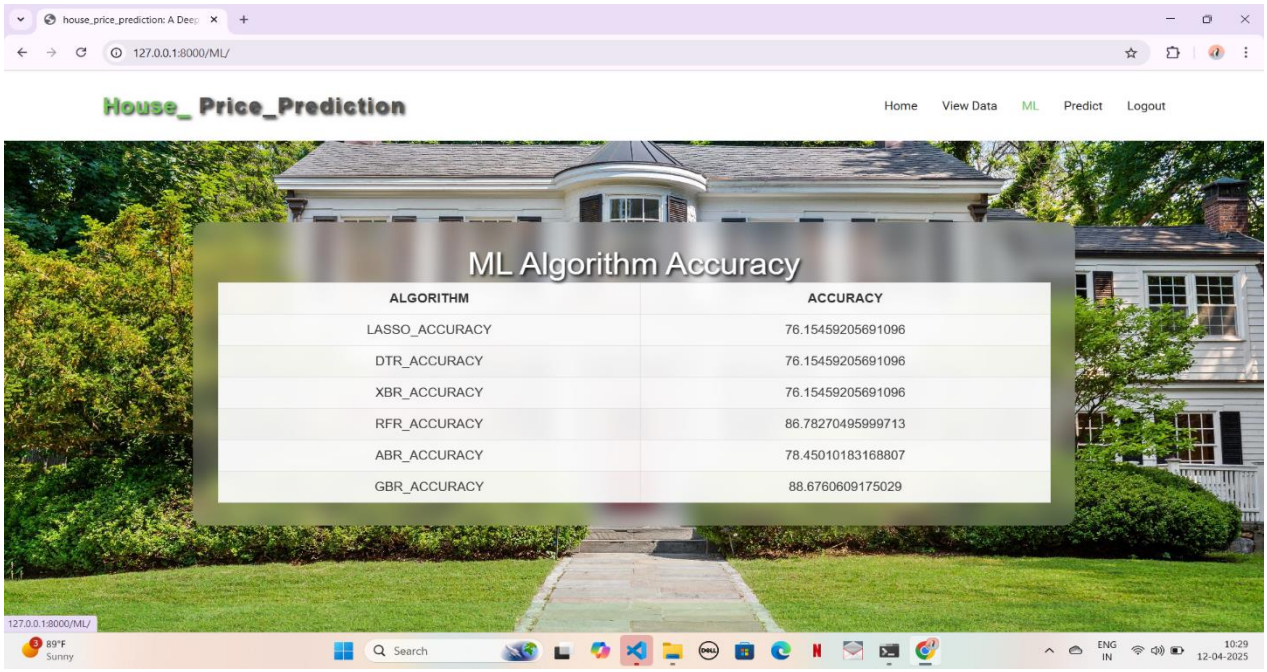
ACTIVATE USER



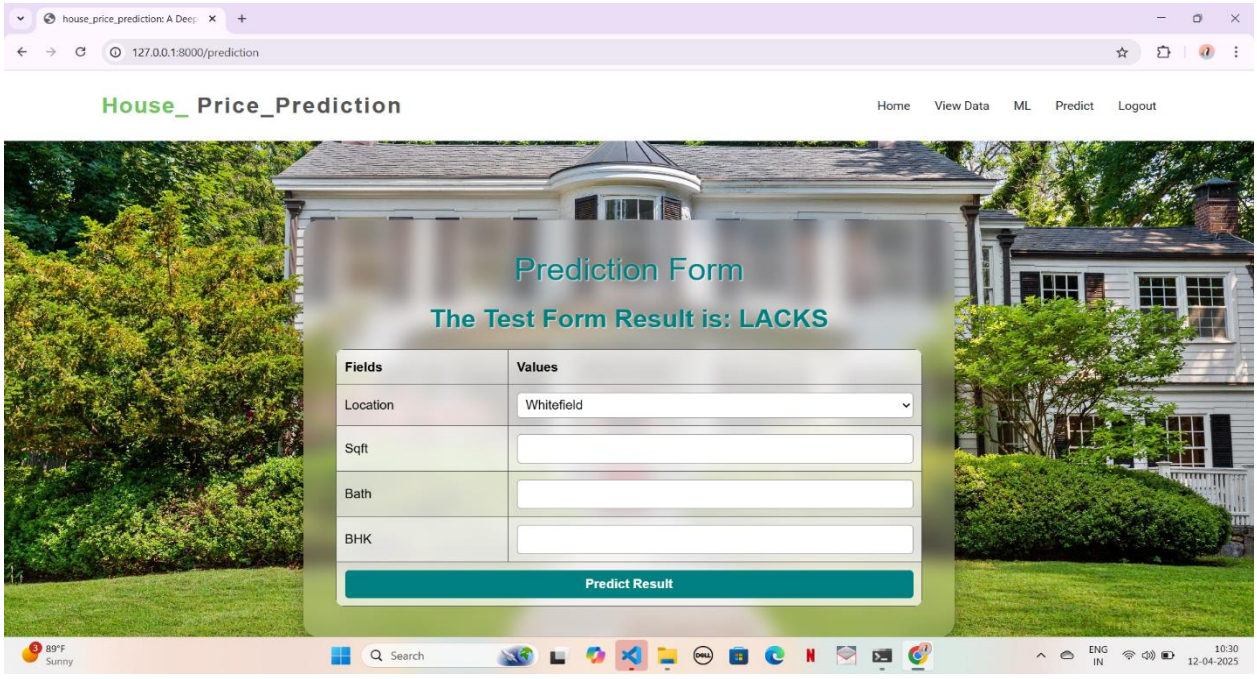
VIEW DATASET



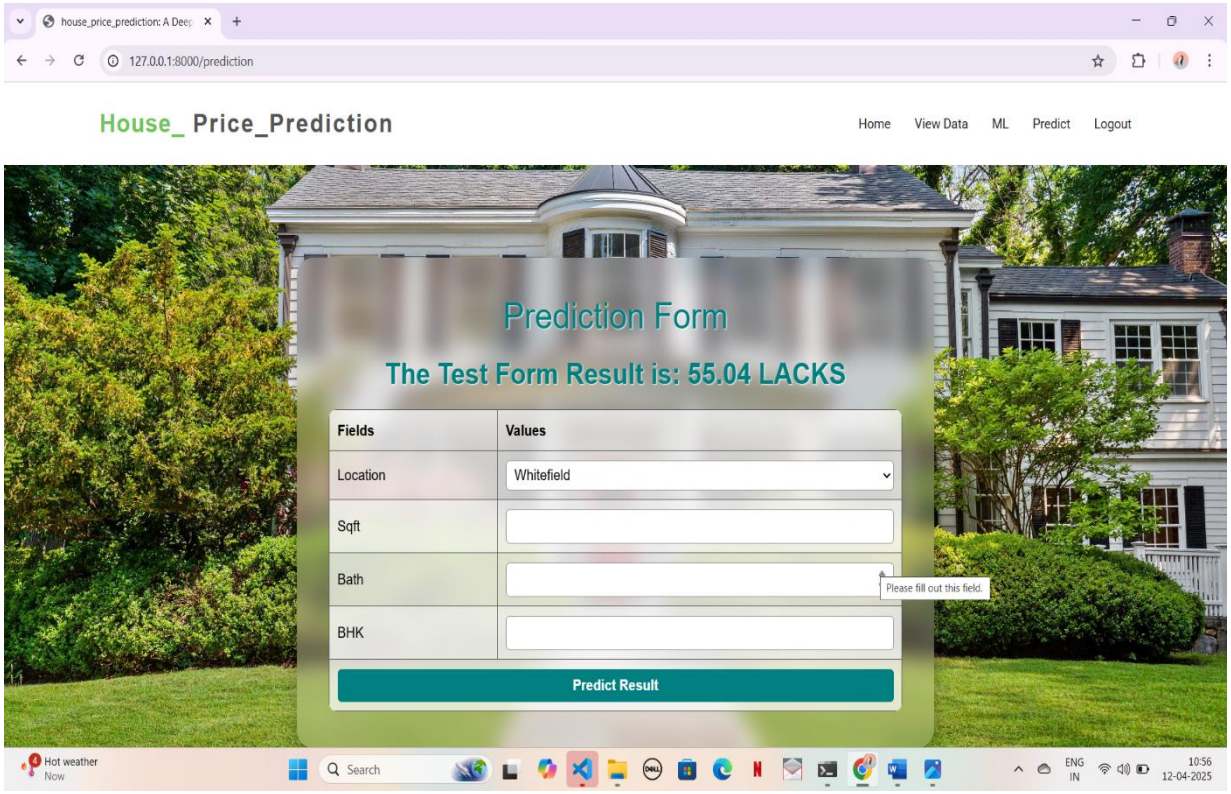
MACHINE LEARNING



PREDICTION PAGE



OUTPUT PAGE


house_price_prediction: A Deep
127.0.0.1:8000/prediction
House_ Price_Prediction
Home
View Data
ML
Predict
Logout
Prediction Form
The Test Form Result is: 55.04 LACKS
Fields
Values
Location
Whitefield
Sqft
Bath
Please fill out this field.
BHK
Predict Result
Hot weather
Now
Search
ENG
IN
10:56
12-04-2025

6. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING CONCEPTS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product. It ensures that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests, each addressing a specific requirement of the system.

6.2 TESTING STRATEGIES

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at ex posing the problems that arise from the combination of components.

Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedure: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

User Acceptance testing

The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation testing

Validation checks are performed on the following fields.

- Text Field

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alpha numeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

- Numeric Field

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

White Box testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.3 TEST CASES

S.no	Test Case	Excepted Result	Result	Remarks (IF fails)
1	User Register	If User registration successfully.	Pass	If already user email exist then it fails.
2	User Login	If Username and password is correct then it will getting valid page.	Pass	Un Register Users will not logged in.
3	Linear regressor	The request will be accepted by the linear regressor	Pass	The request will be not accepted by the Decision tree otherwise its failed
4	ABR	The request will be accepted by the ABR	Pass	The request will be not accepted by the KNN otherwise its failed
5	Random forest	The request will be accepted by the random forest	Pass	The request will be not accepted by the Naïve bayes otherwise its failed
6	View dataset by user	Data set will be displayed by the user	Pass	Results not true failed
7	Calculate score	Score calculated	Pass	Score not displayed failed
8	Prediction results	Prediction results calculated and displayed	Pass	Results not true failed
9	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allowed here
10	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won't login.

7. CONCLUSION AND FUTURE SCOPE

The real estate sector plays a crucial role in economic development, with property values serving as key indicators of economic health. Accurate prediction of house prices enables informed decision-making for homeowners, investors, brokers, and policymakers. This project focused on leveraging machine learning techniques to forecast housing prices in Bangalore using a variety of regression models. Among the evaluated models, Modified XGBoost demonstrated the highest accuracy, showcasing its effectiveness in handling complex and feature-rich datasets. The results confirm that incorporating a wide range of relevant features—such as location, size, and amenities—can significantly enhance prediction accuracy. This model holds great promise for contributing to strategic property valuation and investment planning.

FUTURESCOPE

1. **Improved Machine Learning Algorithms:** Future improvements can integrate deep learning models like Neural Networks to make house price predictions even more accurate.
2. **Real-Time Data Updates:** The system can be enhanced to fetch real-time property prices, market trends, and economic indicators for more precise predictions.
3. **User-Friendly Mobile App:** A mobile app version can be developed, allowing users to check house prices anytime, anywhere with an easy interface.
4. **Smart Investment Suggestions:** Implement AI-driven investment suggestions based on future market trends, rental income potential, and location growth factors to help buyers make better decisions.
5. **Personalized Predictions**: Tailoring predictions based on user preferences such as budget, lifestyle needs, and preferred locations can improve user engagement and decision-making.
6. **Scalability to other cities and countries**: With suitable data, the model can be extended to predict house prices in different cities or countries, adapting to local factors and regional real estate trends

8. REFERENCES

[1] Gallent, Nick, Phoebe Stirling, and Iqbal Hamiduddin. (2022). Pandemic mobility, second homes and housing market change in a rural amenity area during COVID-19–The Brecon Beacons National Park, Wales. Progress in Planning, 100731.

[2] Gajalakshmi, R. K., et al. (2020). An Optimized ASM based Routing Algorithm for Cognitive Radio Networks. International Conference on System, Computation, Automation and Networking (ICSCAN), IEEE.

[3] Padmini, S. V. (2022). Socio-economic conditions of slum dwellers in Karnataka-a case study in Tumkur district. Archers & Elevators Publishing House.

[4] Raj, S. Gokul, N. Srinath, and T. Ananth Kumar (2019). Real-Time Trespasser Detection Using GPS based UAV. IEEE International Conference on Innovations in Communication, Computing and Instrumentation (ICCI).

[5] Tokunaga, Robert S. (2010). Following you home from school: A critical review and synthesis of research on cyberbullying victimization. Computers in Human Behavior, 26(3): 277-287.

[6] Arunmozhiselvi, S., et al. (2022). A Systematic Approach to Agricultural Drones Using a Machine Learning Model. Machine Learning Approaches and Applications in Applied Intelligence for Healthcare Data Analytics. CRC Press, Pages 41-60.

[7] Nguyen, Quang Hung, et al. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. Mathematical Problems in Engineering, 2021: 1-15.

[8] Rajmohan, R., et al. (2022). G-Sep: A Deep Learning Algorithm for Detection of Long-Term Sepsis Using Bidirectional Gated Recurrent Unit.

[9] Ford, Elizabeth, et al. (2019). Identifying undetected dementia in UK primary care patients: a retrospective case-control study comparing machine-learning and standard epidemiological approaches. BMC Medical Informatics and Decision Making, 19(1): 1-9.

[10] Muthukumarasamy, Sugumaran, et al. (2020). Machine learning in healthcare diagnosis. Blockchain and Machine Learning for E-Healthcare Sys., 343.

[11] Mair, Carolyn, et al. (2000). An investigation of machine learning based prediction systems. Journal of systems and software, 53(1): 23-29.

[12] Saravanan, V., et al. (2023). Design of deep learning model for radio resource allocation in 5G for massive IoT device. Sustainable Energy Technologies and Assessments, 56: 103054.

[13] Čeh, Marjan, et al. (2018). Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. ISPRS International Journal of Geo-Information, 7(5): 168.

[14] Hong, Yao, Jianhong Kang, and Ceji Fu. (2022). Rapid prediction of mine tunnel fire smoke movement with machine learning and supercomputing techniques. Fire Safety Journal, 127: 103492.

[15] Lu, Binbin, et al. (2014). Geographically weighted regression with a non-Euclidean distance metric: a case study using hedonic house price data. International Journal of Geographical Information Science, 28(4): 660-681.

[16] Alzain, Elham, et al. (2022). Application of Artificial Intelligence for Predicting Real Estate Prices: The Case of Saudi Arabia. Electronics, 11(21): 3448.

[17] Durganjali, P., and M. Vani Pujitha. (2019). House resale price prediction using classification algorithms. International Conference on Smart Structures and Systems (ICSSS), IEEE.

[18] Lu, Sifei, et al. (2017). A hybrid regression technique for house prices prediction. International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE.

[19] Prasanalakshmi, B. (2022). Deep Regression hybridized Neural Network in human stress detection. International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), IEEE.

[20] Fletcher, Mike, Jean Mangan, and Emily Raeburn. (2004). Comparing hedonic models for estimating and forecasting house prices. Property Management, (3): 189-200.

[21] Hallegatte, Stéphane, et al. (2012). Investment decision making under deep uncertainty-application to climate change. World Bank Policy Research Working Paper – 6193.

[22] Sun, Shiliang, et al. (2019). A survey of optimization methods from a machine learning perspective. IEEE Transactions on Cybernetics, 50(8): 3668-3681.

[23] Coenen, Lize, et al. (2022). Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods. Journal of the Operational Research Society, 73(1): 191-206.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program Educational Objectives (PEOs)

Graduates of B. Tech in Computer Science and Engineering programs are able to

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)	
PEO-1	Apply engineering knowledge in the chosen fields with ethics and professional values.
PEO-2	Continue to learn and solve real life problems inculcate with interdisciplinary teams.
PEO-3	Face the challenges in industry and purse higher studies.

Program Specific Outcomes (PSOs)

PROGRAMSPECIFICOUTCOMES(PSOS)	
PSO-1	Develop Computer Applications by applying Artificial Intelligence
PSO-2	Demonstrate the skills in the field of Networks, Web – Design, Cloud Computing and Data Analytics.

D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY

BALUSUMUDI, BHIMAVARAM, W.G. Dist., A.P., PIN-534202

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

(Accredited with A++ Grade by NAAC & Accredited by NBA)

Ph:08816-221238, Email: dnrcet@gmail.com, Website: https://dnrcet.org

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program Outcomes (POs)

PROGRAM OUTCOMES (POS)		
1	**Engineering knowledge:**	Apply knowledge of machine learning algorithms, data preprocessing techniques, and mathematical modeling to develop a predictive model for house price estimation.
2	**Problem analysis:**	Identify and analyze real-world housing datasets, determine critical features, and handle missing or noisy data to build a reliable solution.
3	**Design/development of solutions:**	Design an efficient predictive system using Modified Extreme Boosting (XG Boost), improving model accuracy and generalization.
4	**Conduct investigations of complex problems:**	Utilize data visualization, hyperparameter tuning, and evaluation metrics such as RMSE and R^2 to investigate and validate the model's performance.
5	**Modern tool usage:**	Employ tools like Python, scikit-learn, XG Boost library, Jupyter Notebook, and data visualization libraries (e.g., Matplotlib, Seaborn) for implementation and analysis.
6	**The engineer and society:**	Understand the economic and social impact of accurate property valuation systems in urban planning and real estate decisions.
7	**Environment sustainability:**	Recognize how smart housing analytics can contribute to sustainable urban development and housing resource allocation.
8	**Ethics:**	Ensure data privacy and ethical handling of sensitive information while sourcing and analyzing real-world housing data.
9	**Individual and team work:**	Collaborate effectively in a team environment to design, implement, and test the prediction system with shared responsibilities.
10	**Communication:**	Document and present findings using technical reports and visualizations to communicate insights effectively to both technical and non-technical stakeholders.
11	**Project management and finance:**	Manage project resources, plan stages of data analysis and model development, and assess cost-effectiveness of predictive analytics in housing.
12	**Life long learning**	Engage in continuous learning about evolving ML techniques, data sources, and software tools relevant to predictive modeling.

D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY

BALUSUMUDI, BHIMAVARAM, W.G. Dist., A.P., PIN-534202

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

(Accredited with A++ Grade by NAAC & Accredited by NBA)

Ph:08816-221238, Email: dnrcet@gmail.com, Website: https://dnrcet.org

Program Name:	B. Tech Computer Science &Engineering			**AY**	2024-25
Course Name:	Project	**Regulation**	R20	**Class/Sem**	IV-II

COURSE OUTCOMES (COs):

Up on completion of the course, students will be able to:

COS	Course Outcomes	Taxonomy
C5421.1	List the key components and features involved in house price prediction models	Remember (BTL-1)
C5421.2	Recall various datasets and parameters commonly used for predicting house prices	Remember (BTL-1)
C5422.3	Explain the importance of accurate prediction in real estate using machine learning techniques	Understand (BTL-2)
C5422.4	Interpret the working principles of Extreme Gradient Boosting (XG Boost) and its variations	Understand (BTL-2)
C5422.5	Apply data preprocessing and feature engineering techniques to improve model accuracy	Apply (BTL-3)
C5422.6	Implement a modified extreme boosting algorithm for enhanced prediction accuracy	Apply (BTL-3)
C5422.7	Analyze the impact of feature selection and parameter tuning on prediction performance	Analyze (BTL-4)
C5421.8	Evaluate the performance of the modified algorithm against traditional ML models	Evaluate (BTL-5)
C5422.9	Assess the model's generalizability using cross-validation and performance metrics	Evaluate (BTL-5)
C5422.10	Formulate improvements to enhance model interpretability and scalability	Evaluate (BTL-5)
C5422.11	Design a robust house price prediction system using modified extreme boosting techniques	Create (BTL-6)
C5422.12	Propose innovative uses of the enhanced model in domains such as urban planning	Create (BTL-6)

D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY
BALUSUMUDI, BHIMAVARAM, W.G. Dist., A.P., PIN-534202
(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)
(Accredited with A++ Grade by NAAC & Accredited by NBA)
Ph:08816-221238, Email: dnrcet@gmail.com, Website: https://dnrcet.org

Course Outcome mapping with Pos &PSOs

CO-PO/PSOMATRIX:(Level of Mapping-3: High; 2: Moderate;1-Low;-:Not mapped)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PS02
CO1	3	2	3	3	3	3	1	2	3	2	3	3	3	3
CO2	3	3	3	3	3	1	3	1	3	3	3	3	3	2
CO3	2	3	3	2	3	3	2	3	2	3	3	3	3	3
CO4	3	3	2	3	3	1	1	3	1	3	2	3	2	3
CO5	3	3	3	2	3	3	3	3	1	3	3	3	3	2
CO6	3	3	3	3	3	1	1	3	2	3	1	3	3	3
CO7	2	3	3	2	2	1	2	3	2	3	3	3	3	3
CO8	2	3	3	3	3	3	3	1	3	3	3	3	3	2
CO9	3	3	3	3	3	3	1	2	3	3	2	2	2	3
CO10	2	3	3	3	3	1	1	3	2	3	3	3	3	3
CO11	3	3	3	2	3	1	2	3	2	3	3	3	3	3
CO12	3	3	3	2	3	1	1	3	2	3	3	3	3	3
Average	**2.66**	**2.91**	**2.91**	**2.66**	**2.91**	**1.83**	**1.75**	**2.5**	**2.25**	**2.91**	**2.66**	**2.91**	**2.83**	**2.75**

PO1	Engineering Knowledge	PO7	Environment &Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of Solutions	PO9	Individual &Team Work
PO4	Conduct Investigations of complex problems	PO10	Communication Skills
PO5	Modern Tool usage	PO11	Project Management& Finance
PO6	Engineer &Society	PO12	Life-long Learning

PSO1: Develop computer applications to meet societal needs by applying the fundamental knowledge of Computer science.

PSO2: Familiarize in emerging technologies of Computer science & Engineering for successful career.

D.N.R. COLLEGE OF ENGINEERING & TECHNOLOGY

BALUSUMUDI, BHIMAVARAM, W.G. Dist., A.P., PIN-534202

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada) (Accredited with A++ Grade by NAAC & Accredited by NBA) Ph:08816-221238,

Email: dnrcet@gmail.com, Website: https://dnrcet.org

DECLARATION

By doing this project we gain the knowledge on the following POs and PSOs. They are

PO1: Engineering Knowledge

PO2: Problem Analysis

PO3: Design/ Development of Solutions

PO4: Conduct Investigation of Complex Problems

PO5: Modern Tool Usage

PO6: The Engineer and Society

PO7: Environment and Sustainability

PO8: Ethics

PO9: Individual and Team Work

PO10: Communication

PO11: Project Management Finance

PO12: Life Long Learning

PSO1: Develop Computer Applications by Applying Artificial Intelligence

PSO2: Demonstrate the skills in the field of Networks, Web-Design, Cloud and Data Analytics

M. LAKSHMI PRASANNA (219P1A0565)

G. LIKHITHA (219P1A0548)

D. JANAYATRI (219P5A0533)

J. YUVA RATNA (219P1A0554)