# STOCK PRICE FORECASTING USING MACHINE LEARNING

USING LSTM, GRU, STACKED LSTM, STACKED GRU, ARIMA.

# CONTENTS

- ABSTRACT
- INTRODUCTION
- EXISTING SYSTEM
- LIMITATIONS
- PROPOSED METHOD
- ADVANTAGES
- REQUIREMENTS
- MODULES
- UMLS
- SCREENSHOTS
- CONCLUSION
- REFERENCES

# ABSTRACT

This project explores stock price prediction using machine learning techniques to improve forecasting accuracy. We evaluated traditional algorithms such as Random Forest Regressor and Support Vector Regression (SVR), which yielded Mean Squared Errors (MSE) of 85.57 and 2878.20, respectively. To enhance performance, we implemented advanced models including Long Short-Term Memory (LSTM) networks, Stacked LSTM, Gated Recurrent Units (GRU), and Stacked GRU, alongside the Autoregressive Integrated Moving Average (ARIMA) model. The LSTM and its stacked variant achieved validation losses of 0.5048 and 0.9658, while the GRU and Stacked GRU models demonstrated even better results with validation losses of 0.0101 and 0.0361, respectively. ARIMA also performed notably well, with a Mean Squared Error of 11.86. The dataset utilized comprises 2417 entries with features including Adj Close, Open, High, Low, Close, and Volume. These results indicate significant improvements in prediction accuracy with the proposed models.

**KEYWORDS:** Stock price, Machine learning, LSTM, GRU, Mean square error, Forecasting Stock.

# INTRODUCTION

The project focuses on enhancing stock price prediction using advanced machine learning techniques to address the limitations of traditional forecasting methods. Accurate stock price prediction is crucial for investors, analysts, and financial institutions, as it significantly impacts decision-making and strategic planning. Traditional models, such as Random Forest Regressor and Support Vector Regression (SVR), often struggle to capture the intricate and non-linear patterns in financial data, leading to suboptimal forecasting performance.

To improve accuracy, this project explores the application of advanced machine learning algorithms, including Long Short-Term Memory (LSTM), Stacked LSTM, Gated Recurrent Units (GRU), and Stacked GRU. These models are well-suited for handling sequential and time-series data, allowing them to better capture temporal dependencies and complex market trends. Additionally, the project employs the Autoregressive Integrated Moving Average (ARIMA) model as a baseline for comparative analysis.

# EXISTING SYSTEM

The existing methods for stock price prediction primarily involve traditional machine learning algorithms such as Random Forest Regressor and Support Vector Regression (SVR). Random Forest Regressor utilizes ensemble learning to improve prediction accuracy through decision trees, while SVR applies support vector machines to handle non-linear relationships. Although these methods are widely used, they often struggle with capturing complex temporal dependencies in stock price data, leading to limited forecasting accuracy. The Mean Squared Error (MSE) values from these models indicate their constraints, highlighting the need for more advanced approaches to improve prediction performance.

# LIMITATIONS

**Limited Accuracy**: Traditional models like Random Forest Regressor and SVR often struggle to accurately predict stock prices due to their inability to capture complex patterns and trends in financial data.

**High Error Rates**: The Mean Squared Error (MSE) from these models can be relatively high, indicating that their predictions are not very precise and can be unreliable for making informed decisions.

**Inability to Handle Non-Stationarity**: Stock prices are influenced by various factors that change over time, and traditional methods may not effectively adapt to these dynamic changes.

**Difficulty with Temporal Patterns**: These models have challenges in understanding the sequential and time-dependent nature of stock price movements, which can lead to poor forecasting performance.

**Lack of Flexibility**: Traditional methods may not easily incorporate additional features or adapt to new data trends, limiting their effectiveness as market conditions evolve.

# PROPOSED SYSTEM

The proposed system aims to enhance stock price prediction accuracy by utilizing advanced machine learning models, specifically Long Short-Term Memory (LSTM), Stacked LSTM, Gated Recurrent Units (GRU), and Stacked GRU, alongside the Autoregressive Integrated Moving Average (ARIMA) model. Unlike traditional methods, these models are designed to capture complex temporal dependencies and dynamic patterns in stock price data. LSTM and its stacked variant are adept at remembering long-term dependencies, while GRU and Stacked GRU offer efficiency in learning sequential data. ARIMA serves as a benchmark to compare the effectiveness of these advanced models. By integrating these techniques, the proposed system aims to significantly reduce Mean Squared Error (MSE) and validation loss, thereby providing more reliable and accurate stock price forecasts **.**

# ADVANTAGES

**Improved Accuracy**

**Efficiency**

**Scalability**

**Real-time Predictions**

**Cost effective**

# SOFTWARE HARDWARE REQUIREMENTS

**S/W CONFIGURATION:**

- Operating System : Windows 7/8/10
- Server side Script : HTML, CSS, Bootstrap & JS
- Programming Language : Python
- Libraries : Flask, Pandas, MySQL. Connector, Scikit-learn
- IDE/Workbench : VS Code
- Technology : Python 3.8+
- Server Deployment : Xampp Server

# SOFTWARE HARDWARE REQUIREMENTS

**H/W CONFIGURATION:**

| | |
|---|---|
| **Processor** | **- I3/Intel Processor** |
| Hard Disk | - 160GB |
| Key Board | - Standard Windows Keyboard |
| Mouse | - Two or Three Button Mouse |
| Monitor | - SVGA |
| RAM | - 8GB |

# MODULES

**Data Collection**: This module is responsible for gathering historical stock price data from internet. It ensures the dataset includes relevant features such as Adj Close, Open, High, Low, Close, and Volume.

**Preprocessing**: In this stage, the collected data is cleaned and prepared for modeling. Tasks include handling missing values, normalizing or scaling data, and splitting it into training and testing sets to ensure accurate model evaluation.

**Modeling**: This module involves developing and training machine learning models. It includes implementing algorithms such as Random Forest Regressor, Support Vector Regression (SVR), LSTM, Stacked LSTM, GRU, and Stacked GRU. The models are tuned and optimized to achieve the best performance in predicting stock prices.

**Prediction**: Once models are trained, this module is used to generate predictions based on new or unseen data. It outputs forecasted stock prices and evaluates the model's performance using metrics like Mean Squared Error (MSE) and validation loss.

# MODULES

**User**

**Login**: This module allows users to securely access the system by verifying their credentials. It ensures that only registered users can access the platform.

**Register**: New users can create an account through this module by providing necessary details such as username, password, and email. It facilitates user onboarding and account management.

**User Home**: After logging in, users are directed to the home page, which provides an overview of their account and access to various features of the system.

**Prediction Page**: This page allows users to input data and initiate stock price predictions. Users can select the desired model and view the forecasted results based on their inputs.

**Result Page**: The result page displays the outcomes of the stock price predictions. It provides detailed information on the predicted values, model performance metrics, and any relevant visualizations to help users interpret the results effectively.

# MODULES

**User**

**1. Register:**

- Objective: User account creation.
- Description: Users, such as healthcare professionals, register with their credentials to create an account within the system.

**2. Login:**

- Objective: Secure system access.
- Description: Registered users log in with their credentials to access the system's diagnostic features.

**3. Input Data:**

- Objective: Upload facial images for diagnosis.
- Description: Users upload facial images into the system for Down syndrome diagnosis. The system preprocesses and prepares the images for model analysis.

# UMLS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

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 Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artefacts of software system, as well as for business modelling and other non-software systems.

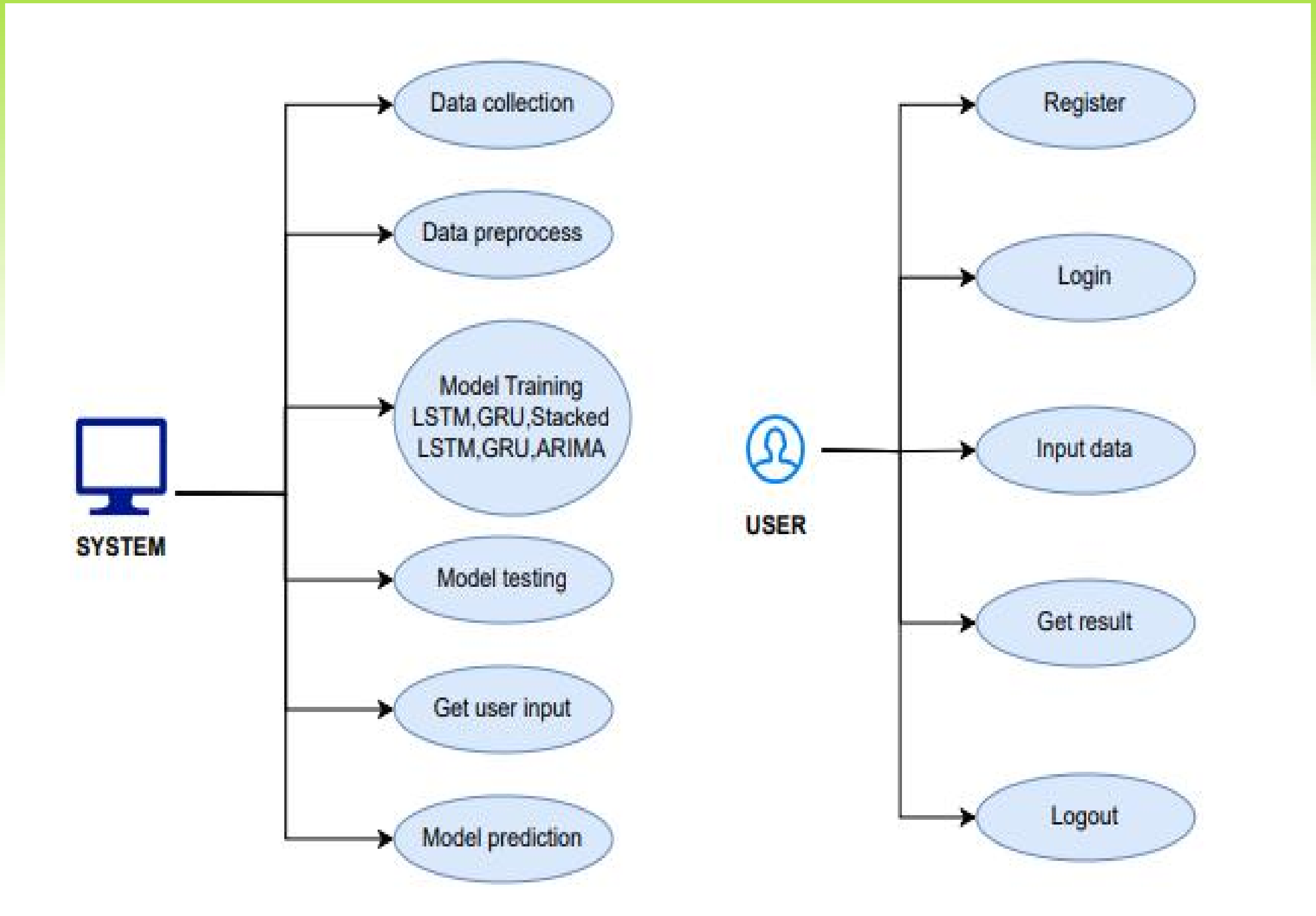
The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

## USE CASE DIAGRAM

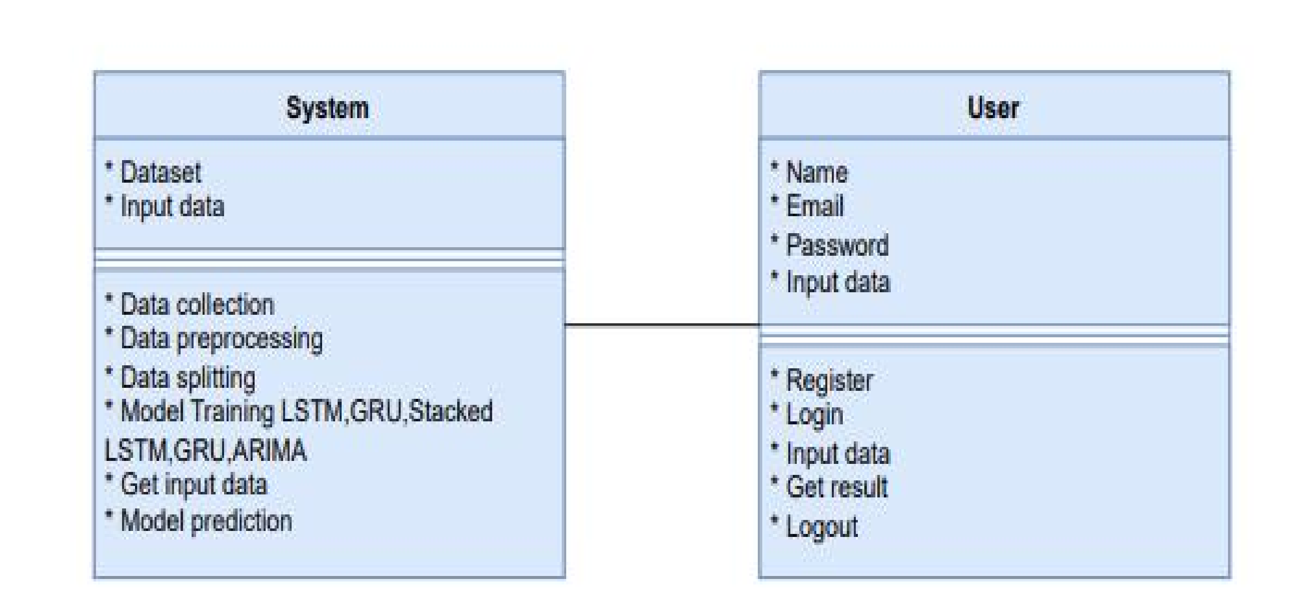
- A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.

# USE CASE DIAGRAM



## CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information



System

* Dataset
* Input data

* Data collection
* Data preprocessing
* Data splitting
* Model Training LSTM,GRU,Stacked LSTM,GRU,ARIMA
* Get input data
* Model prediction

User

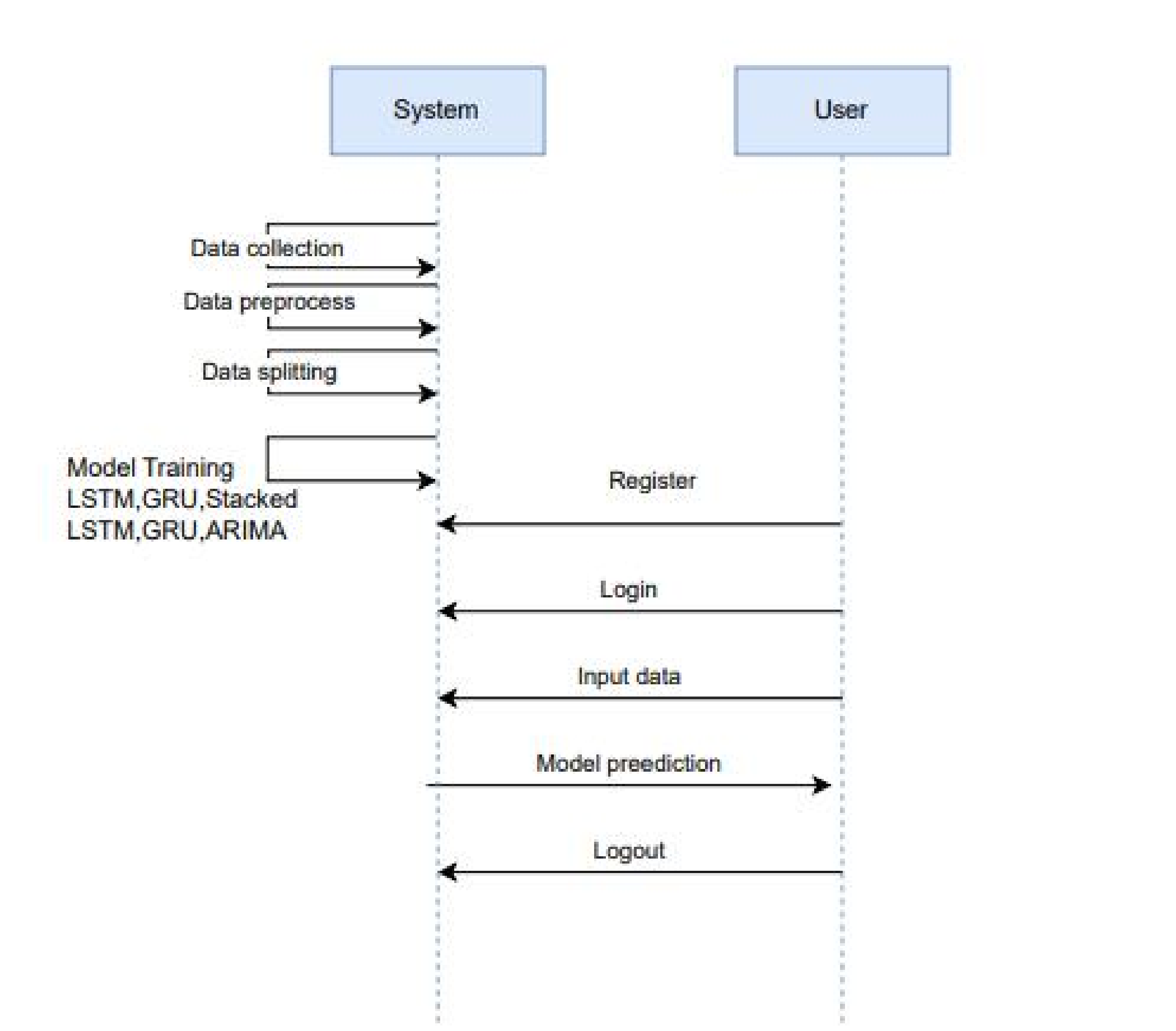
* Name
* Email
* Password
* Input data

* Register
* Login
* Input data
* Get result
* Logout

## SEQUENCE DIAGRAM

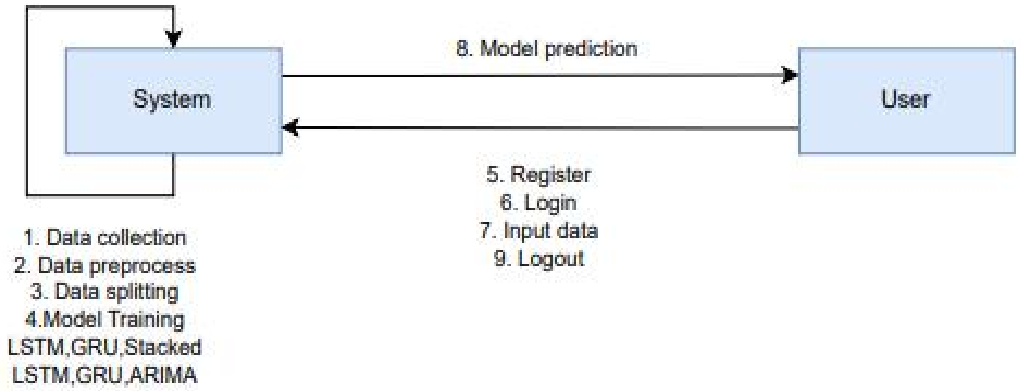
- A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.

- It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams



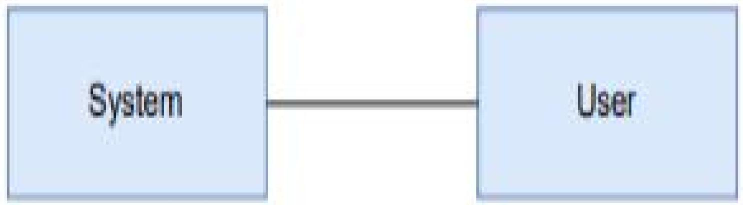
## COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



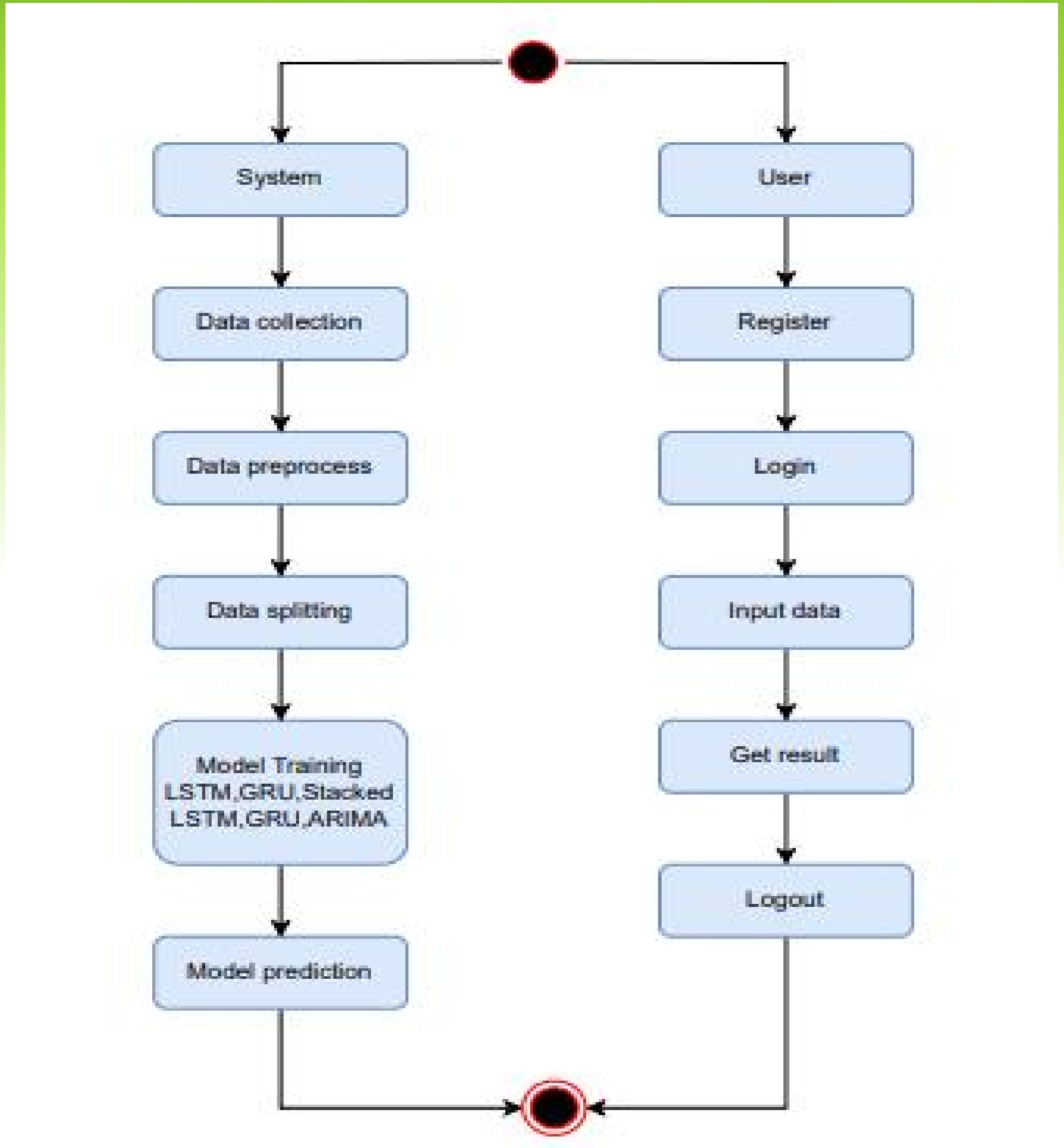
## DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.



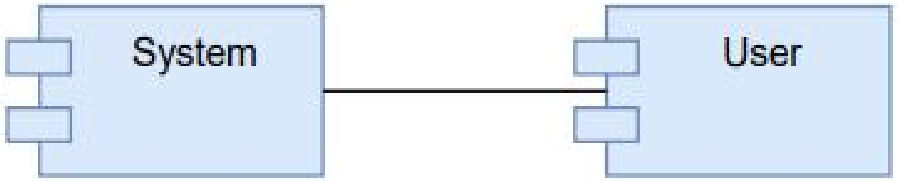
## ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**COMPONENT DIAGRAM**:

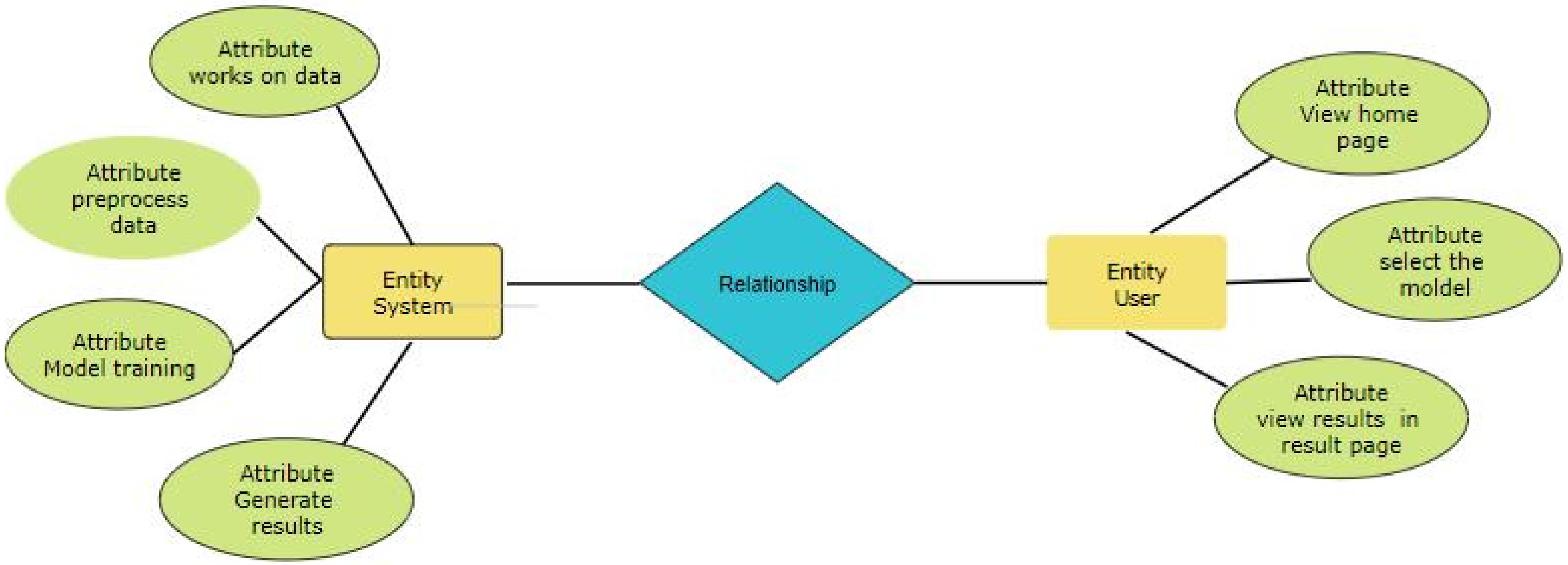
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



**ER DIAGRAM:**

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

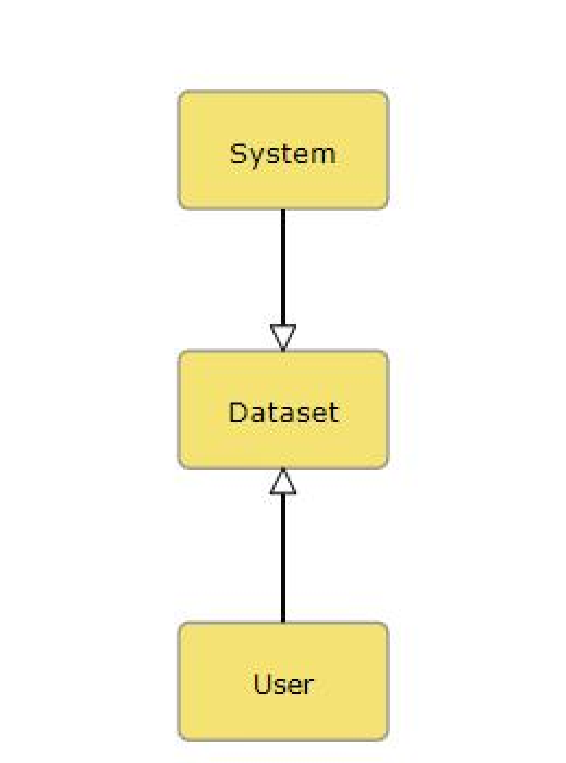
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.


Attribute works on data
Attribute preprocess data
Attribute Model training
Attribute Generate results
Entity System
Relationship
Entity User
Attribute View home page
Attribute select the moldel
Attribute view results in result page

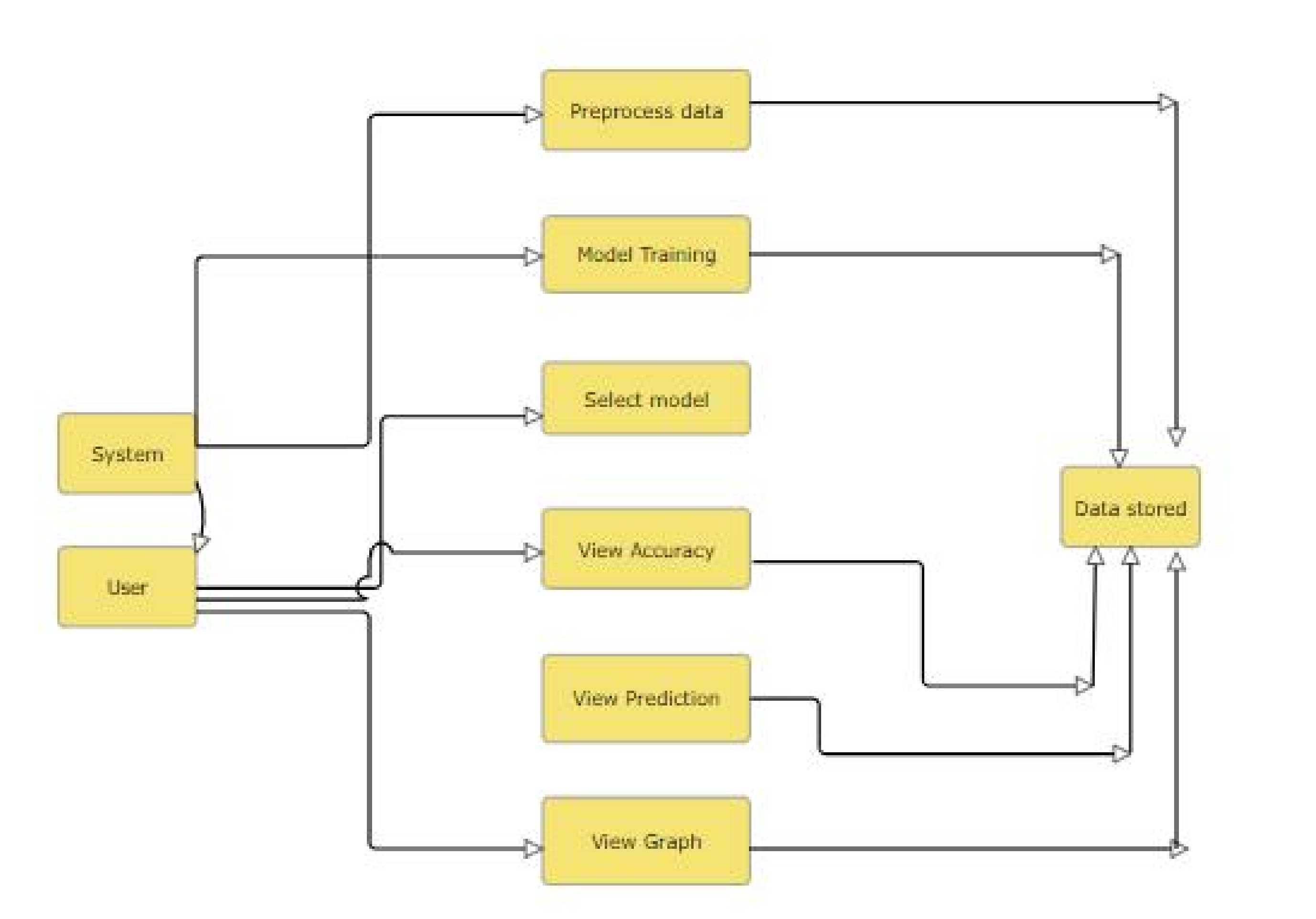
## DFD DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

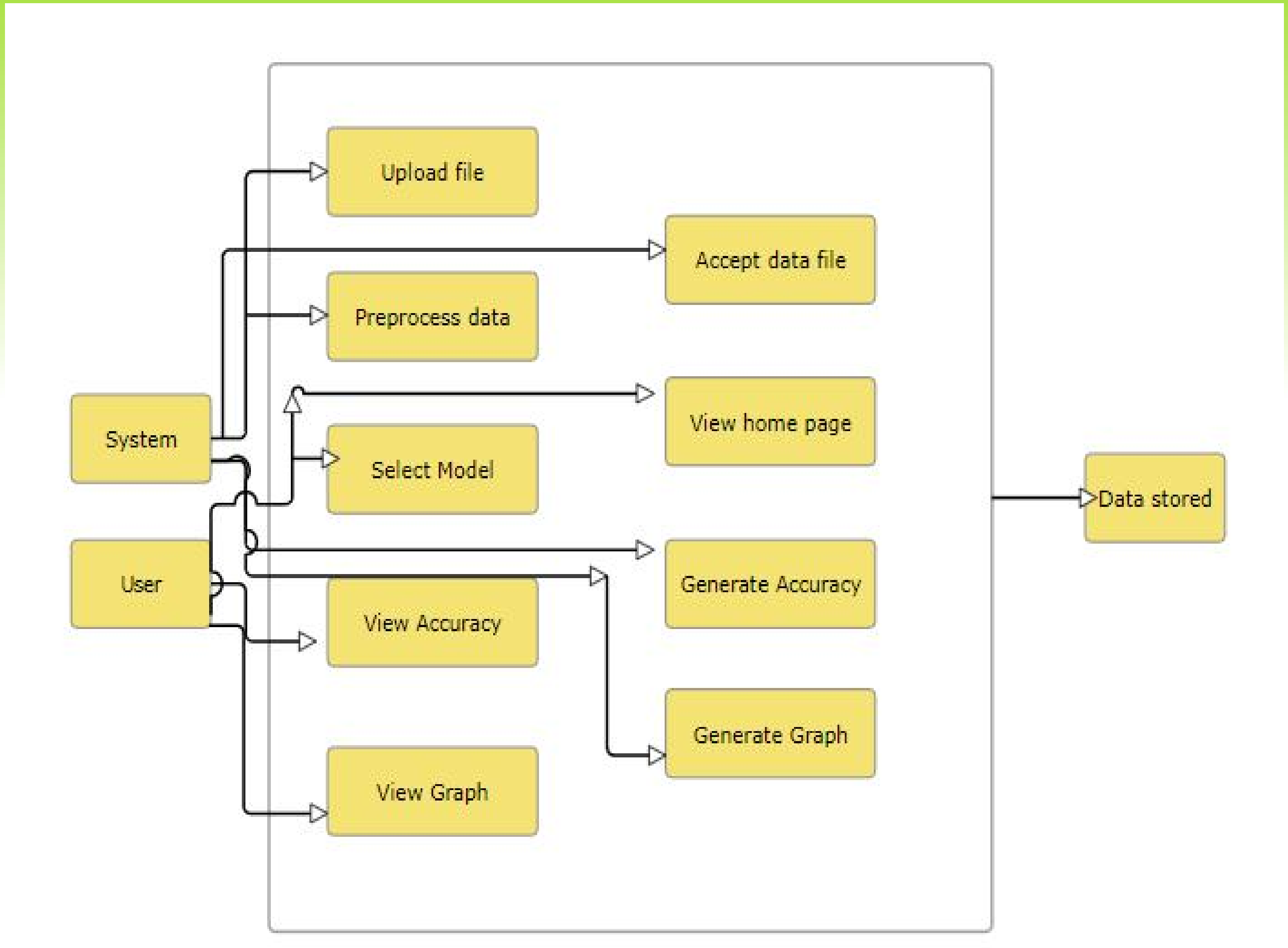
## Contrast Level:



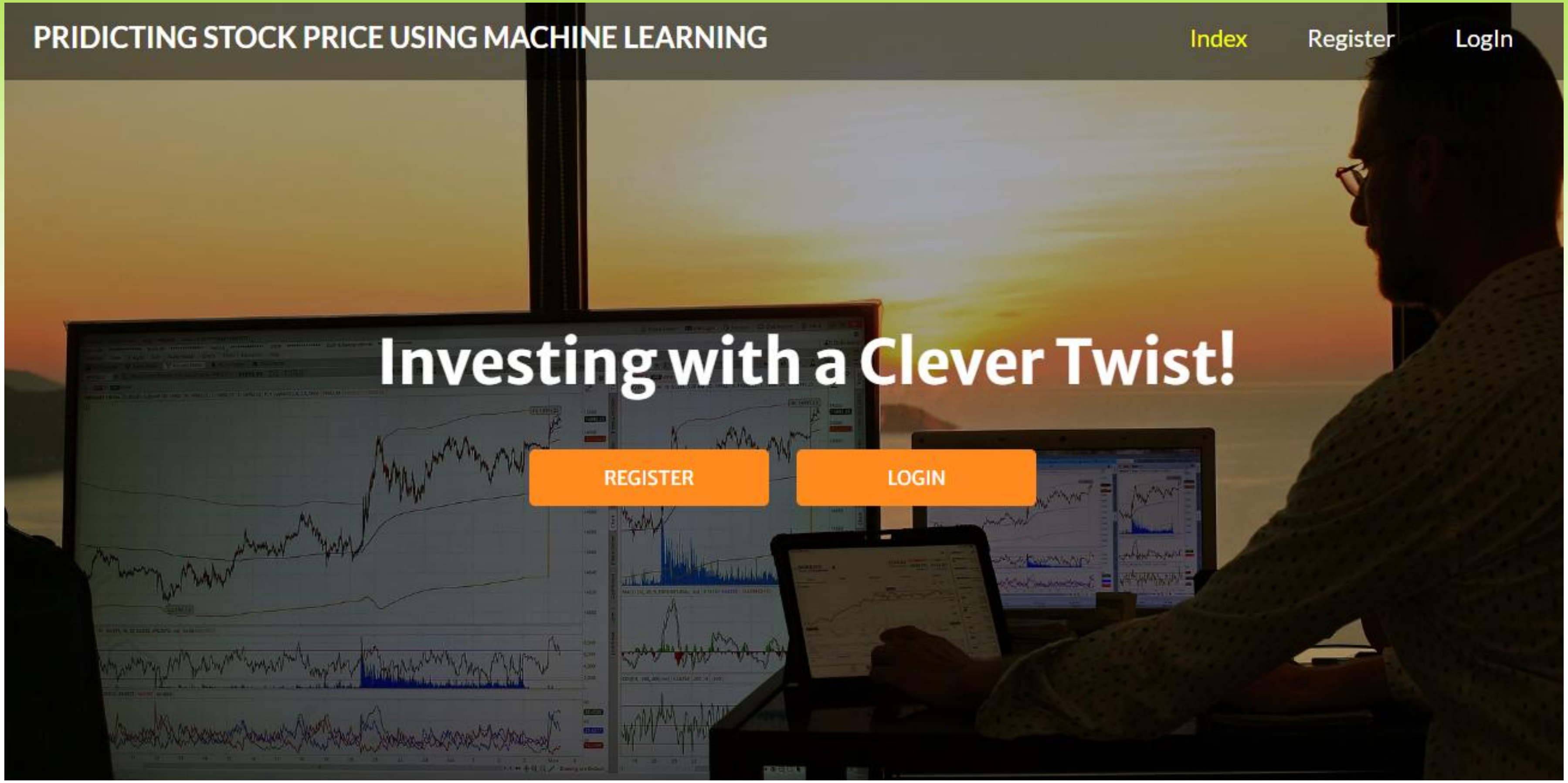
## Level 1 Diagram:



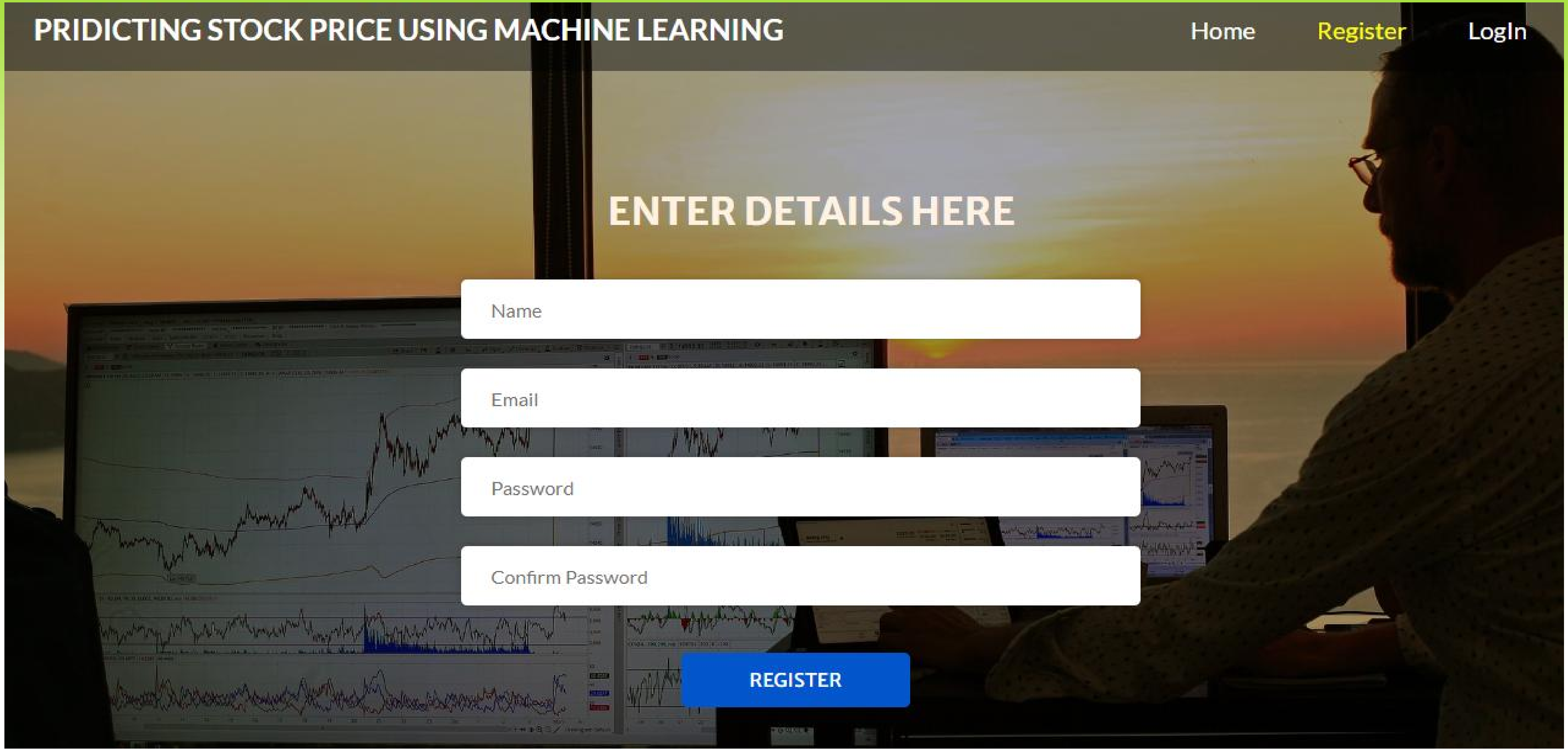
# Level 2 Diagram:



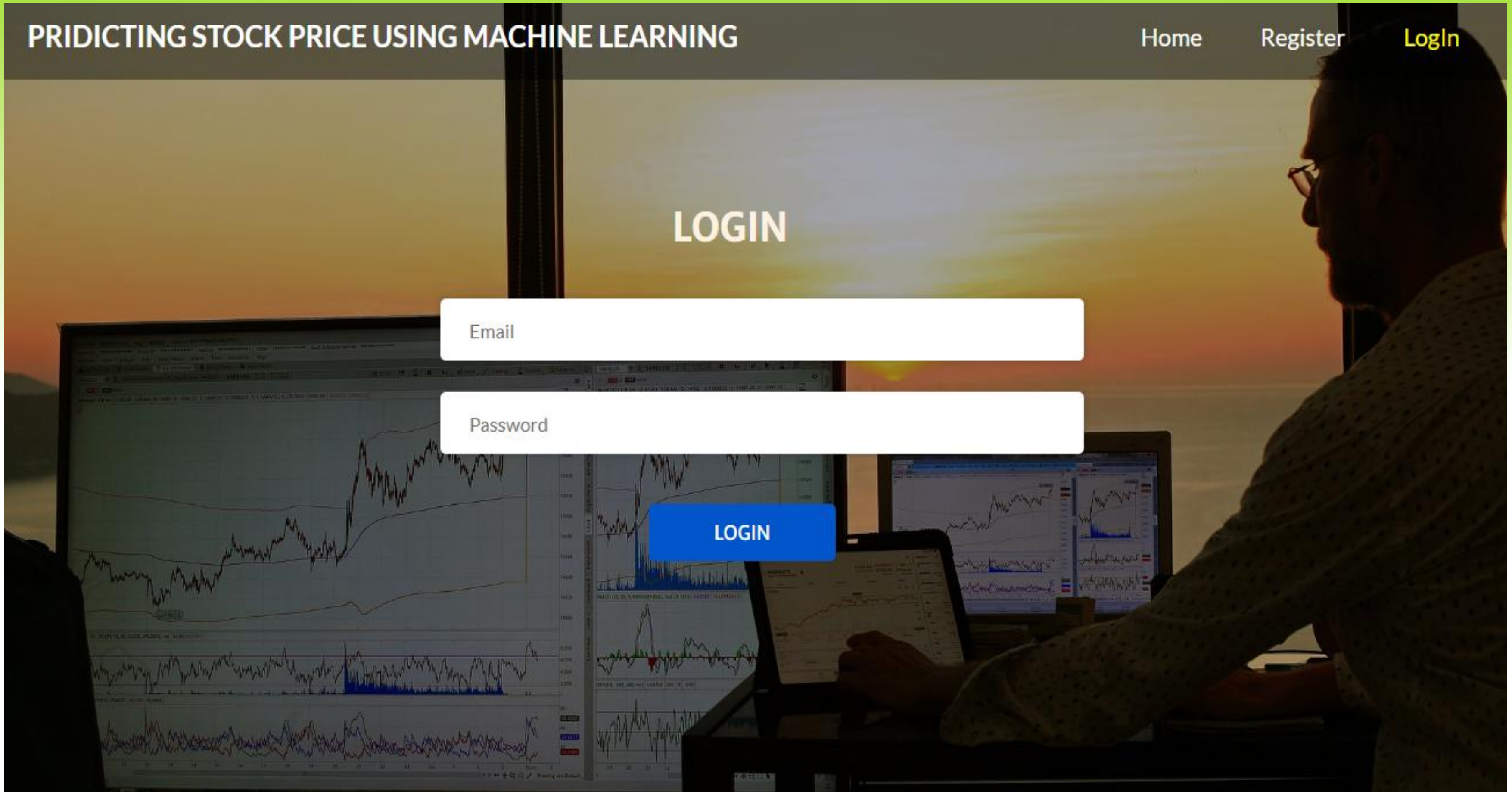
# SCREENSHOTS


PRIDICTING STOCK PRICE USING MACHINE LEARNING
Index
Register
LogIn
Investing with a Clever Twist!
REGISTER
LOGIN

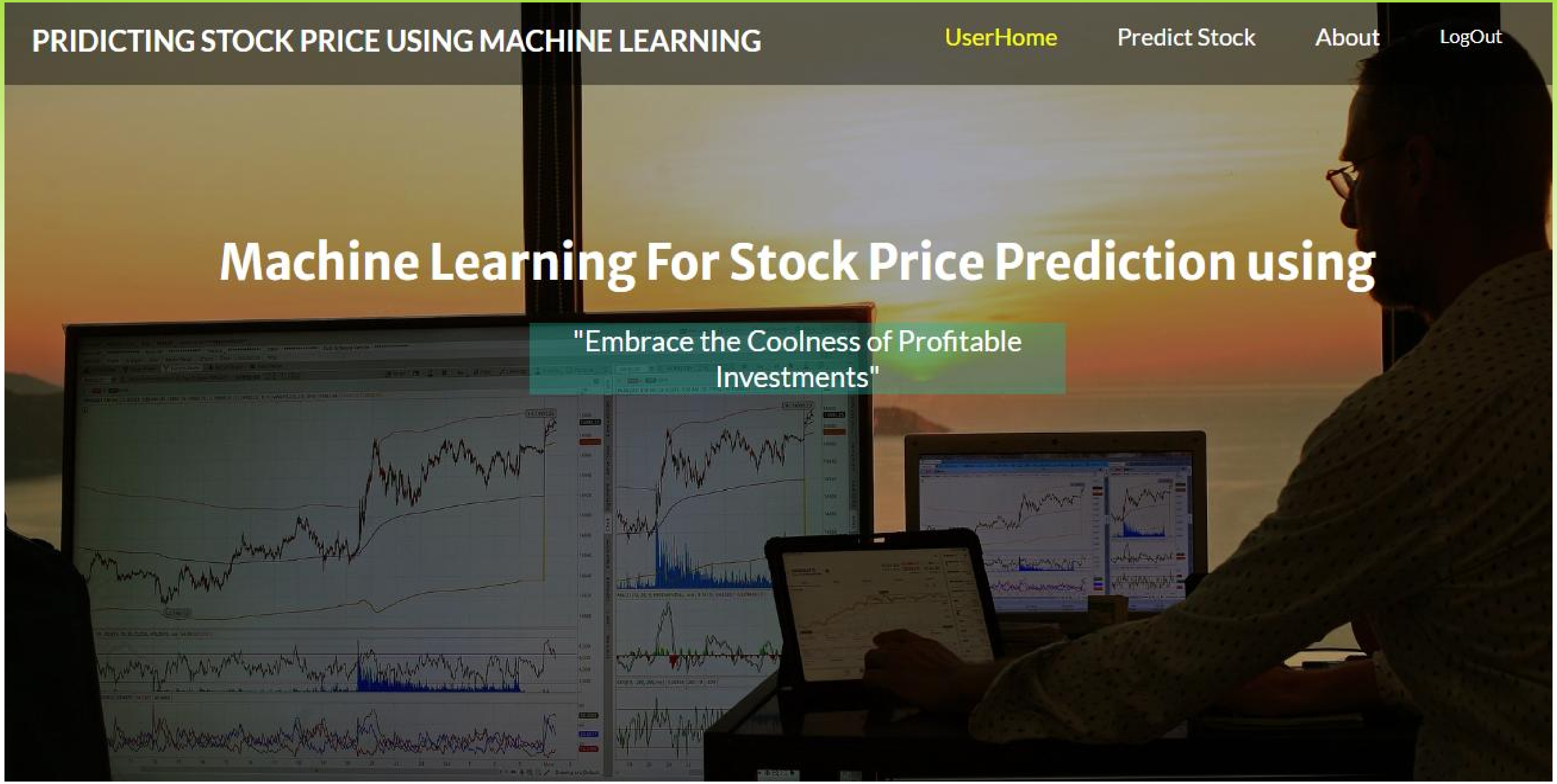
**Home page: This page will navigate user to register and login into the website.**



**Register page: This page will allow user to register and using the valid credentials.**



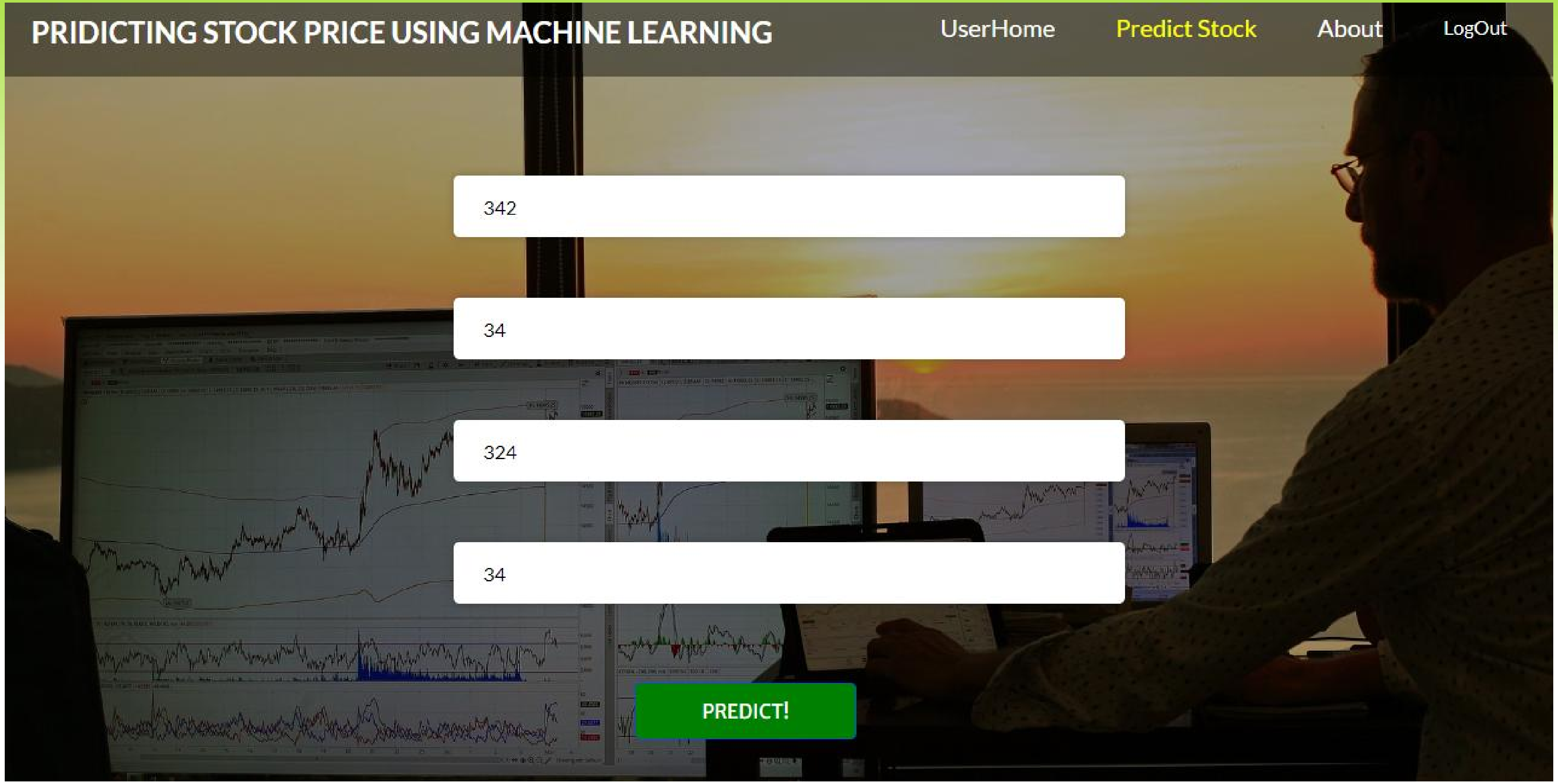
**Login page: This page will allow user to login into the user home page of the website.**



Machine Learning For Stock Price Prediction using

"Embrace the Coolness of Profitable Investments"

**User home page: This page allow user to navigate through the upload page, and logout page.**



PRIDICTING STOCK PRICE USING MACHINE LEARNING

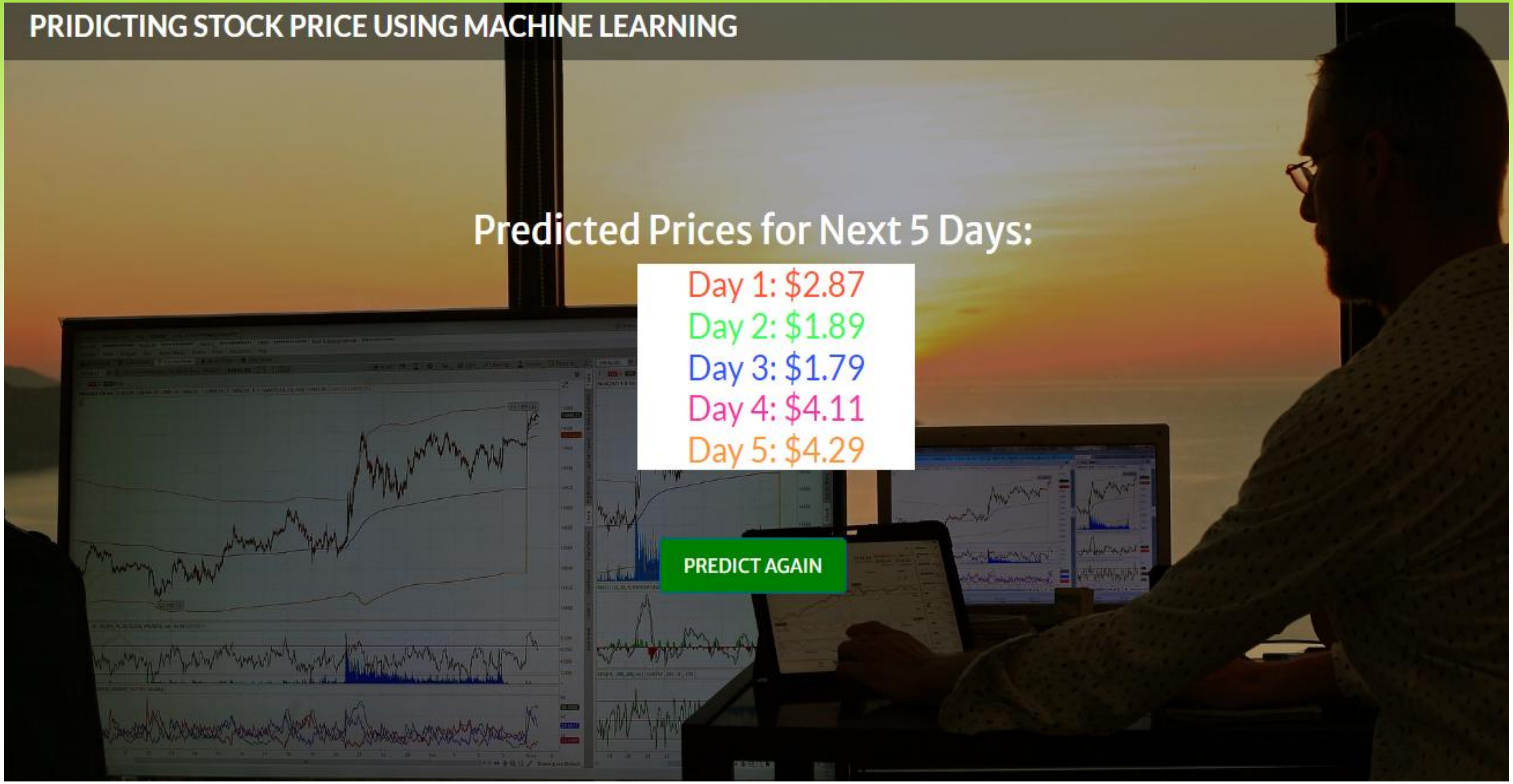
342

34

324

34

**Prediction page: This page will navigate user to upload the stock data and get the results.**



PRIDICTING STOCK PRICE USING MACHINE LEARNING

Predicted Prices for Next 5 Days:

Day 1: $2.87

Day 2: $1.89

Day 3: $1.79

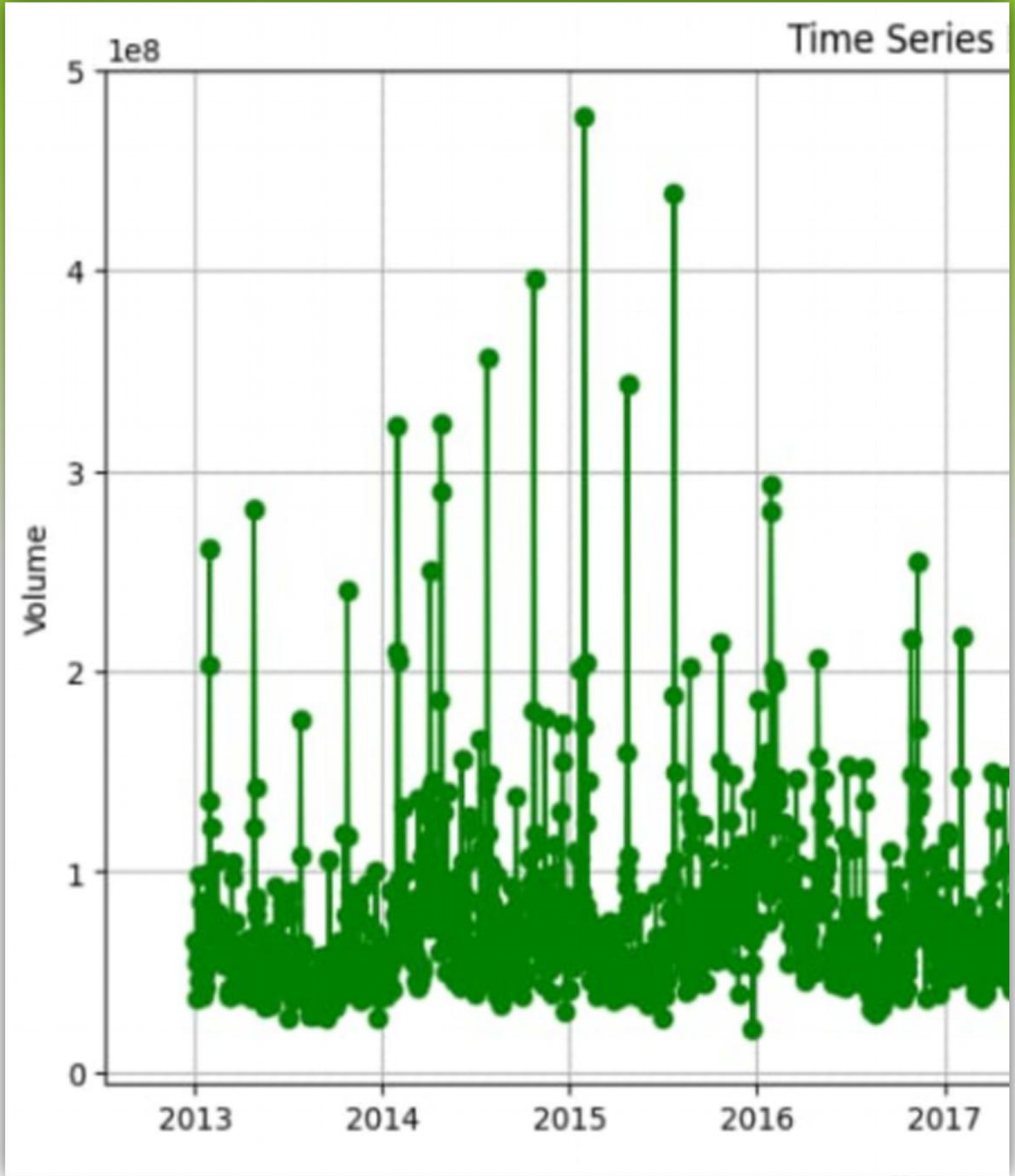
Day 4: $4.11

Day 5: $4.29

**Result page: This page will show you the result and prediction of future stock.**

# CONCLUSION

- Stock market price forecasting using machine learning enhances prediction accuracy, identifies complex patterns, and adapts to market changes in real-time. Advanced models like LSTM, GRU, and ARIMA improve decision-making by minimizing errors and optimizing trading strategies. Investors can better manage risks and capitalize on market opportunities, making stock price forecasting more efficient and reliable.



# REFERENCES

**M. Verma and R. Gupta**, "Deep Learning Techniques for Stock Price Prediction: A Survey," in *2018 IEEE 8th International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India, 2018, pp. 147-151. DOI: 10.1109/CSNT.2018.8820257.

**S. Biswas and A. Raj**, "Stock Market Forecasting Using Artificial Neural Networks," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2017, pp. 145-149. DOI: 10.1109/CCWC.2017.7868406.

**A. B. Thomas and P. Choudhary**, "An Improved Stock Price Prediction Model Using Hybrid Machine Learning Techniques," in *2020 IEEE 14th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2020, pp. 412-416. DOI: 10.1109/ICSESS51411.2020.9283025.

# Thank you