# INDEX

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **TITLE** | **PAGENO** |
|  | **Abstract** | 1 |
| **1.** | **Introduction** | 2–6 |
|  | * 1. Module Description.   1.2) Objective of the Project. |  |
| **2.** | **Literature Survey** | 7-8 |
|  | * 1. Existing System   2. Feasibility Study   3. Technical Feasibility   4. Economical Feasibility   5. Operational Feasibility |  |
| **4.** | **Software Requirement Specifications** | 13–16 |
|  | * 1. Introduction.   2. Overview of the System.   3. Functional and Non-Functional Requirements.   4. Domain Requirements   5. System Analysis |  |
| **5.** | **System Design** | 17–24 |
|  | * 1. Introduction To Design Part.   2. Use Case Diagram   3. UML Diagram   4. Sequence Diagram   5. Activity Diagram.   6. Data Flow Diagram |  |
| **6.** | **Database Design** | 25–31 |
| **7.** | IMPLEMENTATION | 32-44 |
| **8.** | SCREENSHOTS | 45-52 |
| **9.** | SYSTEM TESTING | 53-57 |
| **10.** | CONCLUSION | 58-60 |
| **11.** | FUTURE ENHANCEMENT | 61-64 |
| **12.** | BIBLOGRAPHY | 65-66 |

**ABSTRACT**

Stock price forecasting is a popular and important topic in financial and academic studies. Share Market is anuntidy place for predicting since there are no significant rules to estimate or predict the price of share in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis, etc. are all used to attempt to predict the price in the share market. In this project we attempt to implement a Predictive Modeling and Technical Indicators Analysis approach topredict stock market prices by developing an automated stock data collection and predictive analysis tool. Predictive Modeling is very effectively implemented in forecasting stock prices, returns, and stock modeling and the most frequent methodologies are the Decision Tree algorithm and the Regression Algorithm. This project is for Indian users as the prediction is done on the listed companies of National Stock Exchange of India’s NIFTY index. We outline the design of the Predictive models with its salient features and customizable parameters, and design visually interactive trend charts for stock technical indicators analysis. We select a certain group of parameters with relatively significant impact on the share price of a company. With the help of statistical analysis,the relation between the selected factors and share price is formulated which can help in forecasting accurate results. Although, share market can never be predicted, due to its vague domain, this project aims at applying Predictive Modeling Machine Learning techniques and stock indicator concepts in forecasting the stock prices.

1. **INTRODUCTION**

**Introduction:-**

Stock Market prediction and analysis is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. Stock market is the important part of economy of the country and plays a vital role in the growth of the industry and commerce of the country that eventually affects the economy of the country. Both investors and industry are involved in stock market and wants to know whether some stock will rise or fall over certain period of time. The stock market is the primary source for any company to raise funds for business expansions. It is based on the concept of demand and supply. If the demand for a company's stock is higher, then the company share price increases and if the demand for company's stock is low then the company share price decrease.

The National Stock Exchange of India Limited (NSE) is the leading stock exchange of India, locatedin Mumbai. The NSE was established in 1992 as the first demutualized electronic exchange in the country. NSE was the first exchange in the country to provide a modern, fully automated screen-based electronic trading system which offered easy trading facility to the investors spread across the length and breadth of the country.

The NIFTY 50 index is National Stock Exchange of India's benchmark broad based stock market index for the Indian equity market. It represents the weighted average of 50 Indian company stocks in 12 sectors and is one of the two main stock indices used in India, the other being the BSE Sensex.

Due to involvement of many number of industries and companies, it contain very large sets of data from which it is difficult to extract information and analyse their trend of work manually. The application developed in this project, not only helps in prediction the future movement if the stock in the market, but also automate the data retrieval, trend analysis, predictive analysis and insights generation of a stock, just at the click of a button. Stock market analysis and prediction will reveal the market patterns and predict the time to purchase stock. The successful prediction of a stock's future price could yield significant profit. This is done using large historic market data of 12 months in this project, to represent varying conditions and confirming that the time series patterns have statistically significant predictive power for high probability of profitable trades and high profitable returns for the competitive business investment.

Statement about the problem: The Stock Market is a complex and dynamical system, & is influenced by many factors that are subject to uncertainty. So, it is a difficult task to forecast stock price movements. Due to technology and globalization of business & financial markets it is important to predict the stock prices more quickly & accurately. Automated User friendly Trading application can be developed based on financial predictive indicator algorithms & machine learning techniques to predict the performance of stocks in NSE’s Nifty 50 Index.

My role in the Project

I will be involved in the end to end activities for the creation of this project under the guidance of my supervisor and serve as an “Individual Contributor” for this project fulfilling the following roles:

Idea Generation – Research on Indian Stock Market and Prediction Analysis of the NIFTY stocks

Planning – Design blueprint of the methodology for Stock Data Retrieval & Buy/Sell Prediction

Design and Development – Developing prediction models based on the planned methodology and designing the prediction application tool for stock data retrieval and stock movements analysis

Testing & Documentation – Validating the accuracy of prediction models, reducing the error rate and documenting the result

What contribution would the Project make?

Making predictions is an interesting exercise, but the real fun is looking at how well these forecasts would play out in the actual market. Using the Stock Movement Predictor application, developed in this project, an investor can “play” the stock market using our in-built prediction models (Decision Tree & Regression Analysis) over an evaluation period. The investors will use a strategy informed by our model which they can then compare to the simple strategy of buying and holding the stock over an entire period.

Movement of Stock Market can also be predicted by Technical Analysis. The main two component of Technical Analysis are price and volume and on these two data whole stock market can predicted. Stock Market movement is nothing but a mix and match of Mathematics and Human psychology and Technical Analysis is all about these two attributes.

When market moves positive people invest expecting a further positive movement but when small downtrend is seen people book profits expecting, small generated profits would be lost.

When market goes down, people hold thinking that they would exit if the market falls further. They average it out and thus increase their losses.

The correct way is to cut the losses by placing correct stop loss and add up more stocks with the profits with maintaining trailing stop losses but unfortunately 97% traders and investors do it exactly opposite. Only 3% people do it in correct way and hence they are called as smart investors

This project aims to contribute to the smart investors by easing out their market movement analysis. The tool developed in this project generates trend charts and prediction models using latest data for a NIFTY stock at the click of a button, thus speeding up their investing analysis and decision making process.

**Objectives:-**

The objectives of the system are-

* Improved communication
* Better engagement than audio conferencing
* Helps build relationships
* Saves money
* Saves time
* Streamlines collaboration
* Improves efficiency
* Increases productivity
* Makes scheduling meetings easier
* Eliminates geographical barriers

This software package can be readily used by non-programming persons avoiding human handled chances of error. This project can be easily used by all sorts of users.

**Objective and Scope of the Project:**

**Objective:**

To add to the academic understanding of stock market prediction:

With a greater understanding of how the market moves, investors will be better equipped to prevent another financial crisis.

Evaluate some existing strategies from a rigorous scientific perspective and provide a quantitative evaluation of new strategies.

Provide an automated Stock Prediction Tool to Traders to:

Make Buy/Sell Decisions

Distinguish between conservative and risky stocks

1. **LITERATURE SURVEY**

**Literature Survey:-**

"What other people think” has always been an important piece of information for most of us during the decision-making process. The Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that is driving force for this area of interest. And there are many challenges involved in this process which needs to be walked all over in order to attain proper outcomes out of them. In this survey we analysed basic methodology that usually happens in this process and measures that are to be taken to overcome the challenges being faced.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

Stock Market Prediction Using Machine Learning The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India. In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools. Two techniques are used to benchmark the AI techniques, namely, Autoregressive Moving Average (ARMA) which is linear modelling technique and random walk (RW) technique. The experimentation was performed on data obtained from the Johannesburg Stock Exchange. The data used was a series of past closing prices of the All Share Index. The results showed that the three techniques have the ability to predict the future price of the Index with an acceptable accuracy. All three artificial intelligence techniques outperformed the linear model. However, the random walk method out performed all the other techniques. These techniques show an ability to predict the future price however, because of the transaction costs of trading in the market, it is not possible to show that the three techniques can disprove the weak form of market efficiency. The results show that the ranking of performances support vector machines, neuro- fuzzy systems, multilayer perceptron neural networks is dependent on the accuracy measure used.

#### Feasibility Study

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running

system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operational Feasibility
* Economical

#### 3.3 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

#### Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

#### 3.5 Economical Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and Economical feasibility for certain.

**4. SOFTWARE**

**REQUIREMENT**

**SPECIFICATION**

**4.1 Introduction**

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed fictional description, a representation of system behaviour, an indication of performance requirement and design constraints appropriate validation criteria, and other information pertinent to requirement.

The introduction to software requirements specification states the goals and objectives of the software, describing it in the context of the computer-based system.

The Information Description provides a detailed description of the problem that the software must solve. Information content, flow and structure are documented.

A description of each function required to solve the problem is presented in the Functional Description.

Validation Criteria is probably the most important and ironically the most often neglected section of the software requirement specification.

Software requirement specification can be used for different purpose. Here are the major uses, not clearly understood by the developer. If this is the case, a careful analysis involving much interaction with the user should be devoted to reaching a clear statement of requirements, in order to avoid possible misunderstandings.

Sometimes at the beginning of a project, even the user has no clear idea of what exactly the desired product Think for instance of user interface, a' user with no previous experience with computer products may not appreciate the difference between, say menu driven interaction and a command line interface. Even an exact formation of system functions and performance may be missing an initial description produced by an in experienced user.

**4.2 Overview of the System**

This SRS document contains the complete software requirements for the Code-For-Coders and describes the design decisions, architectural design and the detailed design needed to implement the system. It provides the visibility in the design and provides information needed for software support. This Code-For-Coders system is a new self constrained software product which will be produce by the project team in order to overcome the problem that have occurred due to the current manual system. The newly introduced system will provide and easy access to the system and it will contain user friendly function with attractive interfaces. The system will give a better obtains for the problem of handling large scale of physical file system, for the errors occurring in calculation and all the require task that has been specified by the client. The final outcome of this project will increase the efficiency of almost all the task done at the site in a much convenient manner.

**4.3 Functional and Non-Functional Requirements**

**Functional requirements**

It defines functions of a system or its components. A function is described as a set of input, the behaviour, and the outputs. Behavioural requirements describing all the cases where the system uses the functionality requirements are captured in use cases. The user shall be able to search either all the initial set of database. The system shall provide appropriate viewers for the user to read documents in document shop. Every reservation shall be allowed a unique identifier which shall be copy to account permanent storage area.

Some of the functional requirements of our project are as follows

* The system shall provide appropriate viewers for the user to read documents in document.
* Every reservation shall be allocated a unique identification.

**Non-Functional requirements**

Non-functional requirements are not directly concerned with the specific functions delivered by the system. They may relate to reliability, response time and store occupancy. They relate to system as a whole rather than to individual system features.

• **Product Requirements**

Requirements specify that the delivered product behave in a particular way.

E.g. execution speeds, reliability etc.

**• Organizational Requirements**

Requirements which are a consequence of organizational policies and procedures.

E.g. process standards used implementation requirements etc.

**• External Requirements**

Requirements which arise from factors which are external to the system and its development process.

E.g. Interoperability requirements, legislative requirements etc.

This system provides security, changing of the password and the facility of notifications for the attendance.

**4.4 Domain Requirements**

Requirements that come from the application domain of the system that reflect the characteristics of that domain

* Describe the system characteristics send features that reflect the domain.
* If domain requirement are not satisfied, the system may be unworkable.
* Requirements are expressed in the language of the application domain; this is often not understood by software engineers developing the system.
* Domain specialists understand the area so well that they do not think of making the domain requirements explicit.
* To use domain and host our website, we have used Vercel, to make it live.

**4.5 System Requirements :**

##### **Hardware Specification**:

* **Processor:** Intel
* **RAM:** 4 GB OR MORE
* **Hard Disk:** 100 GB

##### **Software Specification:**

* **Front End:** React JS
* **Language**: PYTHON AND MACHINE LEARNING
* **Back End:** SQL Server
* **Server:** Web
* **Framework:** Flask

**5. SYSTEM DESIGN**

**5.1. Introduction**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analysed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**5.2 UML Diagram**

UML stands for Unified Modelling Language which is used in object oriented software engineering .Although typically used in software engineering it is a rich language that can be used to model an application structure, behaviour and even business process. There are 14 UML diagram types to help you model this behaviour.

They can be divided into two main categories; structure diagrams and behavioural diagrams.

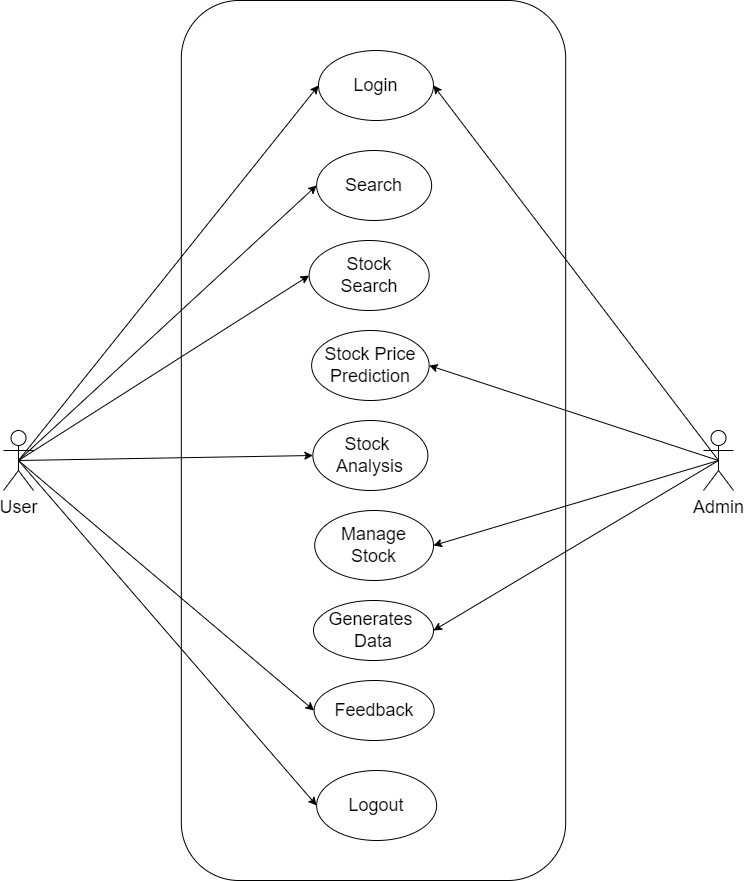
Few diagrams are listed below:

1. Use Case Diagram
2. Sequence Diagram
3. Activity Diagram

**5.2.1 Use Case Diagram:**

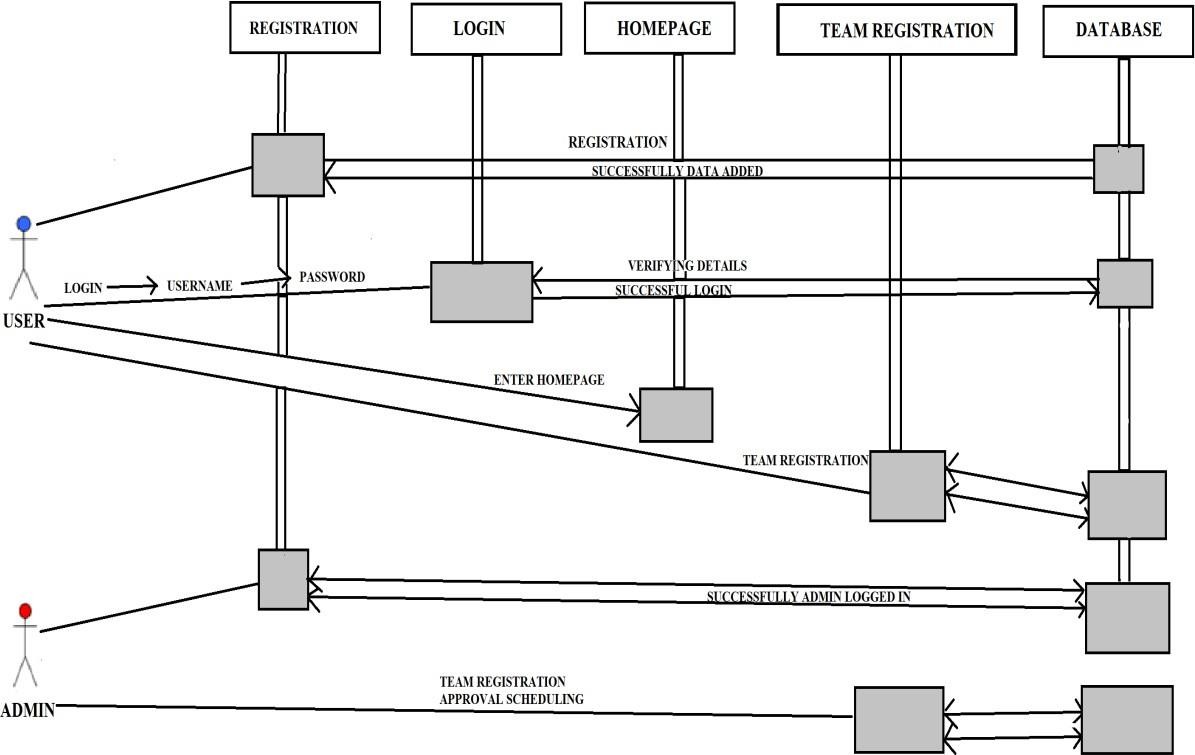
As the most known Diagram type of the behavioural UML diagram ,Use case diagram gives a graphical overview of the actors involved in a system, different functions needed by those actors and how this different factions are interacted .It’s a great starting point for any project

discussion, because you can easily identify the main actors involved and the main processes of the system.



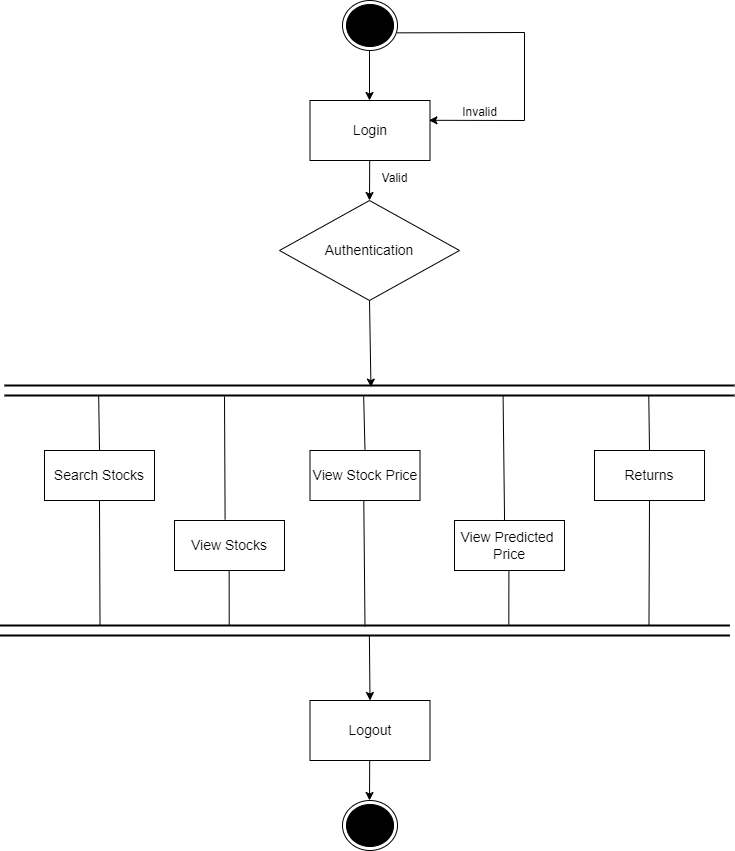
**5.2.2 Sequence Diagram**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the [logical view](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model) of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios

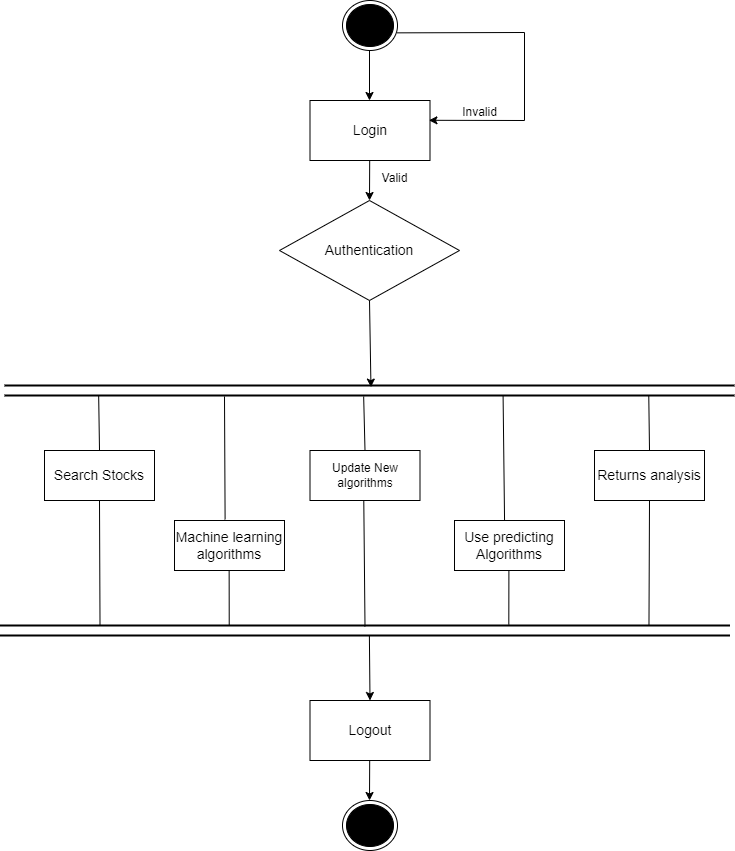


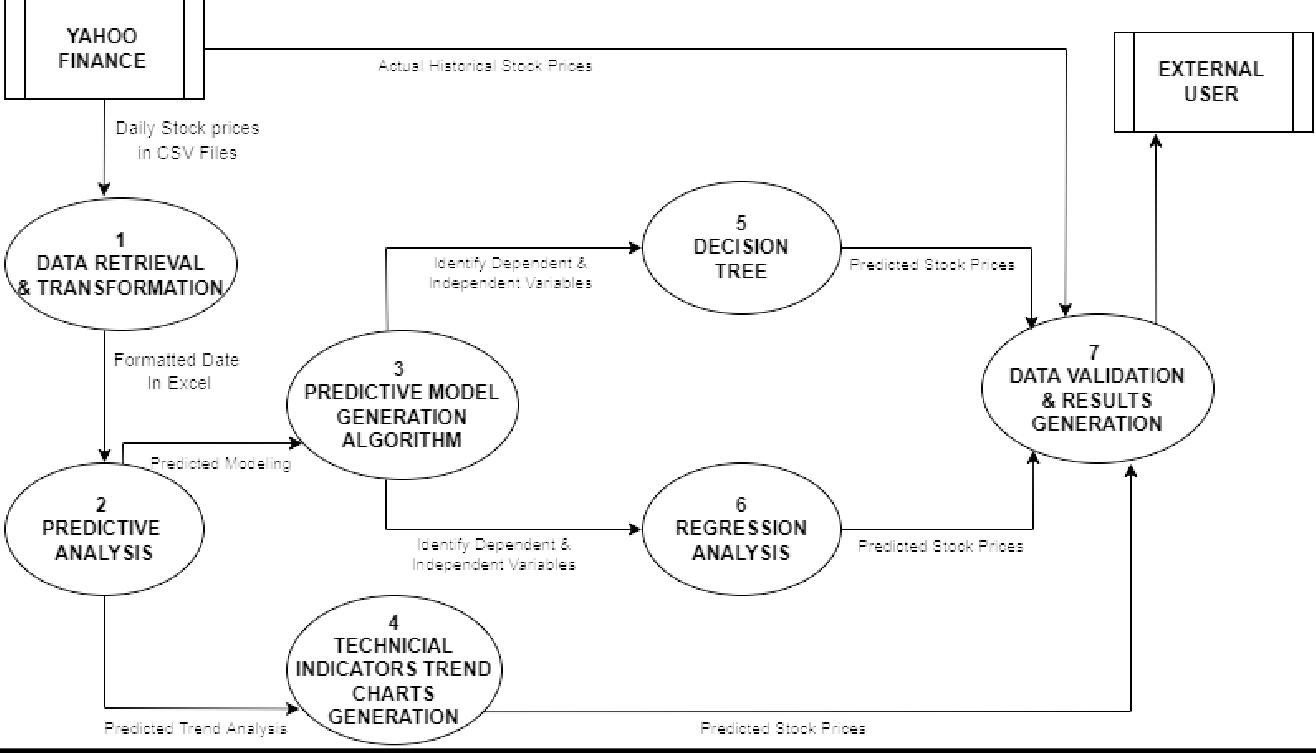
**5.2.3 Activity Diagram**

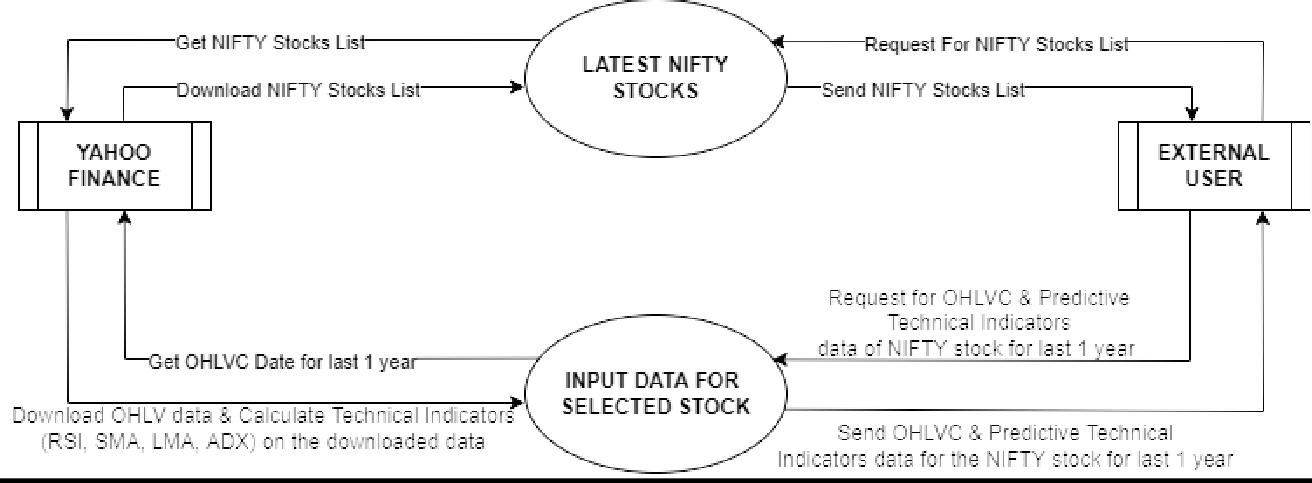
Activity diagrams are graphical representations of [workflows](https://en.wikipedia.org/wiki/Workflow) of stepwise activities and actions with support for choice, iteration and concurrency. In the [Unified Modelling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language), activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.

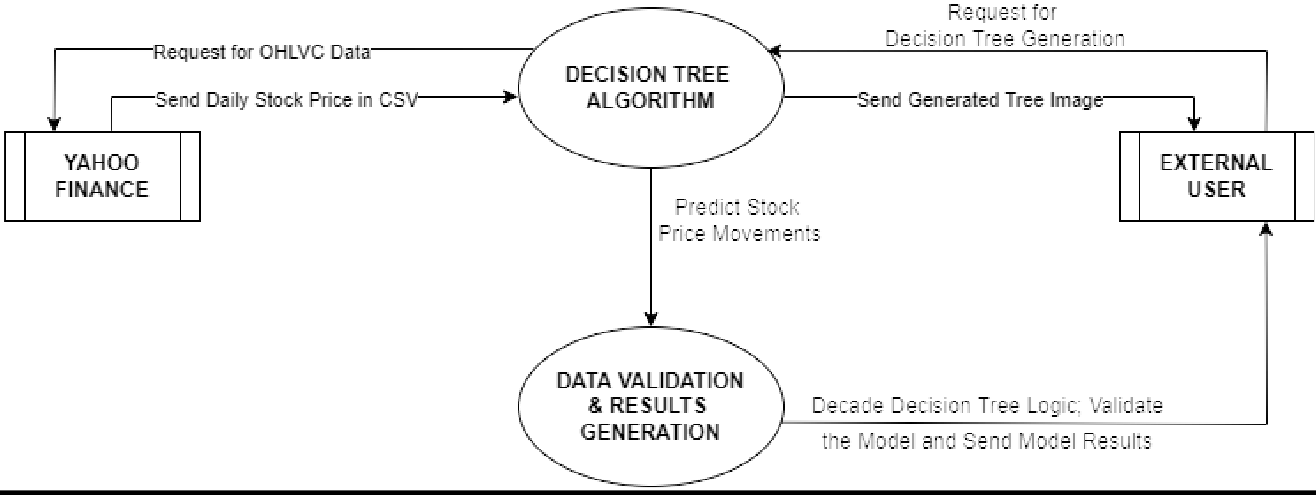


**Admin**

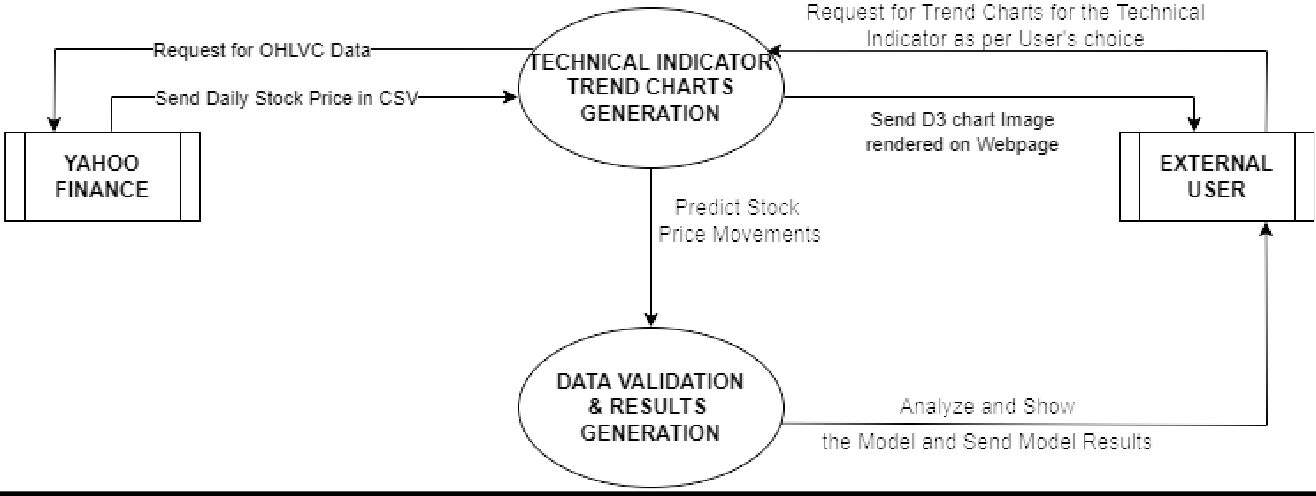


**DFD LEVEL 0**

**DFD LEVEL 1**

**DFD LEVEL 3 (DECISION TREE)**

**XDFD LEVEL (Technical Indicators Trend Charts Generation)**



**6. DATABASE DESIGN**

**DATABASE DESIGN:**

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database application within the database management system (DBMS).

The process of doing database design generally consists of number of steps which will be carried out by the database designer. Usually, the designer must:

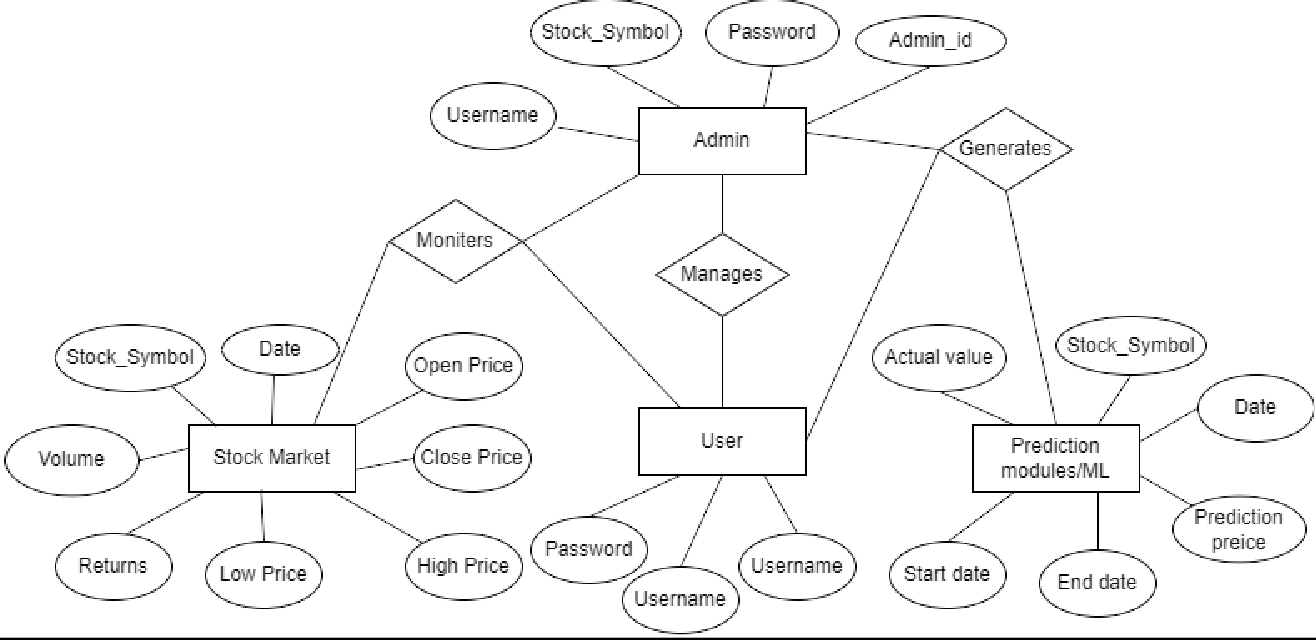
* Determine the data to be stored in the database.
* Determine the relationship between the different data elements.

**6.1 E-R Diagram**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. An **entity-relationship model** (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modelling is a relational schema database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database

* The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
* The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modelling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
* The set of primary components that are identified by the ERD are
* Data object
* Relationships
* Attributes
* Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.



**Normalization**

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

#### First Normal Form (1NF):

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

Consider the following example-

The above table satisfies both the rules and hence the above table is in 1NF.

|  |  |
| --- | --- |
| **STOCK NAME** | **STOCK PRICES** |
| WIPRO Ltd | 410,440 |
| TATA POWER | 212,218 |

**Rule 1-** column with atomic data cannot have several values of same type in it.

In the above table there are multiple values for **Course** column, hence we create multiple columns.

|  |  |  |
| --- | --- | --- |
| **STOCKNAME** | **HIGH PRICE** | **LOW PRICE** |
| WIPRO Ltd | 440 | 410 |
| TATA POWER | 218 | 212 |

It satisfies Rule 1, but the table is not in 1NF

**Rule 2-** A table with atomic data cannot have multiple columns with same type of data.

|  |  |
| --- | --- |
| **STOCK NAME** | **PRICE** |
| WIPRO Ltd | 440 |
| WIPRO Ltd | 410 |
| TATA POWER | 218 |
| TATA POWER | 212 |

The above table satisfies both the rules and hence the above table is in 1NF.

**Second Normal Form (2 NF):**

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

Primary key is a not a composite primary key

No non key attributes are present

Every non key attribute is fully functionally dependent on full set of primary key. Consider the following example-

In the above example Notes depends on Semester, similarly User name depends on

Password. Therefore, the tables are split into two smaller tables as shown below.

As the tables are divided into two separate tables, now Notes is fully functionally dependent on Semester and User name is fully functionally dependent on Password.

|  |  |  |  |
| --- | --- | --- | --- |
| **STOCK NAME** | **CURRENT PRICE** | **QUANTIT Y** | **RETURN S**  **ASSUMP TION** |
| WIPRO  Ltd | 415 | 500 | 5% |
| TATA POWER | 215 | 400 | 4.8% |

As the tables are divided into two separate tables, now TEAM CAPTAIN is fully functionally dependent on TEAM NAME and User name is fully functionally dependent on Password.

Third Normal Form:

A relation is said to be in third normal form if their exits no transitive dependencies.

Transitive Dependency: If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STOCK NAME** | **STOCK ID** | **CURRENT PRICE** | **QUANTIT Y** | **RETUR NS**  **ASSUM PTION** | **INVENTORY ID** |
| WIPRO  Ltd | 1 | 415 | 500 | 5% | 22 |
| TATA  POWER | 2 | 215 | 400 | 4.8% | 24 |

**7. IMPLEMENTATION**

**APP.PY(main file):**

import streamlit as st

import pandas as pd

import numpy as np

from keras.models import load\_model

import matplotlib.pyplot as plt

import yfinance as yf

from sklearn.preprocessing import MinMaxScaler

from datetime import datetime

def plot\_graph(figsize, values, full\_data, extra\_data=None, extra\_dataset=None, title="Stock Data"):

    fig, ax = plt.subplots(figsize=figsize)

    ax.plot(values, 'orange', label='MA')

    ax.plot(full\_data['Close'], 'b', label='Close Price')

    if extra\_data and extra\_dataset is not None:

        ax.plot(extra\_dataset, label='Additional MA')

    ax.set\_title(title)

    ax.set\_xlabel('Date')

    ax.set\_ylabel('Price')

    ax.legend()

    return fig

# Initialize session state for recent stocks

if 'recent\_stocks' not in st.session\_state:

    st.session\_state.recent\_stocks = []

if 'search\_triggered' not in st.session\_state:

    st.session\_state.search\_triggered = False

# Main App Page

def main\_page():

    st.set\_page\_config(page\_title="Stock Maps", layout="wide")

    st.title("Stock Maps")

    # Sidebar for user input

    st.sidebar.header("User Input")

    # Text input for stock ID

    stock\_input = st.sidebar.text\_input("Enter the Stock ID", "GOOG", key="stock\_input")

    # Button to trigger search

    search\_button = st.sidebar.button("Search")

    # Date range selection

    default\_start\_date = datetime(2022, 3, 2)

    end = datetime.now()

    start = st.sidebar.date\_input("Start Date", default\_start\_date)

    end = st.sidebar.date\_input("End Date", end)

    if start > end:

        st.sidebar.error("Start date must be before end date.")

        st.stop()

    # Trigger search when the button is clicked or when Enter is pressed

    if search\_button or st.session\_state.get('search\_triggered', False):

        selected\_stock = stock\_input

        # Fetch stock data

        try:

            google\_data = yf.download(selected\_stock, start, end)

            if google\_data.empty:

                st.error("No data found for the stock ID. Please try a different one.")

                st.stop()

        except Exception as e:

            st.error(f"Error fetching data: {e}")

            st.stop()

        # Load the model

        try:

            model = load\_model("Latest\_stock\_price\_model.keras")

        except Exception as e:

            st.error(f"Error loading the model: {e}")

            st.stop()

        # Display stock data

        st.subheader("Stock Data")

        st.write(google\_data)

        # Add the current stock to the recent stocks list

        if selected\_stock not in st.session\_state.recent\_stocks:

            st.session\_state.recent\_stocks.insert(0, selected\_stock)

            if len(st.session\_state.recent\_stocks) > 5:

                st.session\_state.recent\_stocks.pop()

        # Display recent stocks

        st.sidebar.subheader("Recently Viewed Stocks")

        for recent\_stock in st.session\_state.recent\_stocks:

            if st.sidebar.button(f"Load {recent\_stock}"):

                st.session\_state.search\_triggered = True

                st.session\_state.search\_triggered\_stock = recent\_stock

        # Sidebar for moving average selection

        ma\_options = st.sidebar.multiselect(

            "Select Moving Averages",

            ["100 days", "200 days", "250 days"],

            default=["100 days", "200 days", "250 days"]

        )

        # Plot moving averages based on user selection

        for days in ["100 days", "200 days", "250 days"]:

            if days in ma\_options:

                days\_int = int(days.split()[0])

                ma\_col = f'MA\_for\_{days\_int}\_days'

                google\_data[ma\_col] = google\_data['Close'].rolling(days\_int).mean()

                st.subheader(f'Original Close Price and MA for {days}')

                st.pyplot(plot\_graph((15, 6), google\_data[ma\_col], google\_data, title=f"MA for {days}"))

        # Plot combined moving averages

        if "100 days" in ma\_options and "250 days" in ma\_options:

            st.subheader('Original Close Price and MA for 100 days and MA for 250 days')

            st.pyplot(plot\_graph((15, 6), google\_data['MA\_for\_100\_days'], google\_data, 1, google\_data['MA\_for\_250\_days'], title="MA for 100 and 250 days"))

        # Prepare data for predictions

        splitting\_len = int(len(google\_data) \* 0.7)

        x\_test = pd.DataFrame(google\_data['Close'][splitting\_len:])

        # Normalize data

        scaler = MinMaxScaler(feature\_range=(0, 1))

        scaled\_data = scaler.fit\_transform(x\_test[['Close']])

        x\_data = []

        y\_data = []

        for i in range(100, len(scaled\_data)):

            x\_data.append(scaled\_data[i-100:i])

            y\_data.append(scaled\_data[i])

        x\_data, y\_data = np.array(x\_data), np.array(y\_data)

        # Predict

        try:

            predictions = model.predict(x\_data)

            inv\_pre = scaler.inverse\_transform(predictions)

            inv\_y\_test = scaler.inverse\_transform(y\_data)

        except Exception as e:

            st.error(f"Error making predictions: {e}")

            st.stop()

        # Plot predictions

        plotting\_data = pd.DataFrame({

            'original\_test\_data': inv\_y\_test.reshape(-1),

            'predictions': inv\_pre.reshape(-1)

        }, index=google\_data.index[splitting\_len+100:])

        st.subheader("Original values vs Predicted values")

        st.write(plotting\_data)

        # Plot original vs predicted close price

        fig = plt.figure(figsize=(15, 6))

        plt.plot(pd.concat([google\_data['Close'][:splitting\_len+100], plotting\_data], axis=0))

        plt.legend(["Data - not used", "Original Test data", "Predicted Test data"])

        plt.title("Original vs Predicted Close Prices")

        plt.xlabel('Date')

        plt.ylabel('Price')

        st.pyplot(fig)

        # Reset search trigger

        st.session\_state.search\_triggered = False

# Main logic to display content

def main():

    main\_page()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Config.py:**

# config.py

USER\_CREDENTIALS = {

    "qwerty": "qwerty",

    "qwerty": "Mustu@123"

}

# Function to hash a password

def hash\_password(password):

    import hashlib

    return hashlib.sha256(password.encode()).hexdigest()

**Login.py:**

# pages/login.py

import streamlit as st

from config import USER\_CREDENTIALS, hash\_password

def authenticate(username, password):

    hashed\_password = hash\_password(password)

    if username in USER\_CREDENTIALS and hash\_password(USER\_CREDENTIALS[username]) == hashed\_password:

        return True

    return False

def login\_page():

    st.subheader('Login')

    username = st.text\_input('Username')

    password = st.text\_input('Password', type='password')

    if st.button('Login'):

        if authenticate(username, password):

            st.session\_state.logged\_in = True

            st.session\_state.username = username

            st.session\_state.page = "Main Content"

            st.session\_state.flag = True  # Trigger page reload

        else:

            st.error('Username/password is incorrect')

    if st.button('Sign Up'):

        st.session\_state.page = "Sign Up"

        st.session\_state.flag = True  # Trigger page reload

    # Redirect if the flag is set

    if 'flag' in st.session\_state and st.session\_state.flag:

        st.session\_state.flag = False

        st.experimental\_rerun()  # Refresh to show the updated page

**Signup.py:**

# pages/signup.py

import streamlit as st

import time  # Add this import

from config import USER\_CREDENTIALS, hash\_password

# Function to add a new user

def add\_user(username, password):

    USER\_CREDENTIALS[username] = hash\_password(password)

def signup\_page():

    st.subheader('Sign Up')

    username = st.text\_input('Username')

    password = st.text\_input('Password', type='password')

    confirm\_password = st.text\_input('Confirm Password', type='password')

    if st.button('Sign Up'):

        if password != confirm\_password:

            st.error('Passwords do not match')

        elif username in USER\_CREDENTIALS:

            st.error('Username already exists')

        else:

            add\_user(username, password)

            st.success('Account created successfully! You can now log in.')

            st.session\_state.page = "Login"

            st.rerun()  # Refresh the app to show the login page

**Style.css:**

/\* assets/styles.css \*/

body {

  background-color: white;

  color: #333;

  font-family: Arial, sans-serif;

}

.streamlit-expanderHeader {

  color: #333;

}

.stButton>button {

  background-color: #007bff;

  color: white;

}

.stButton>button:hover {

  background-color: #0056b3;

}

.logout-button {

  position: absolute;

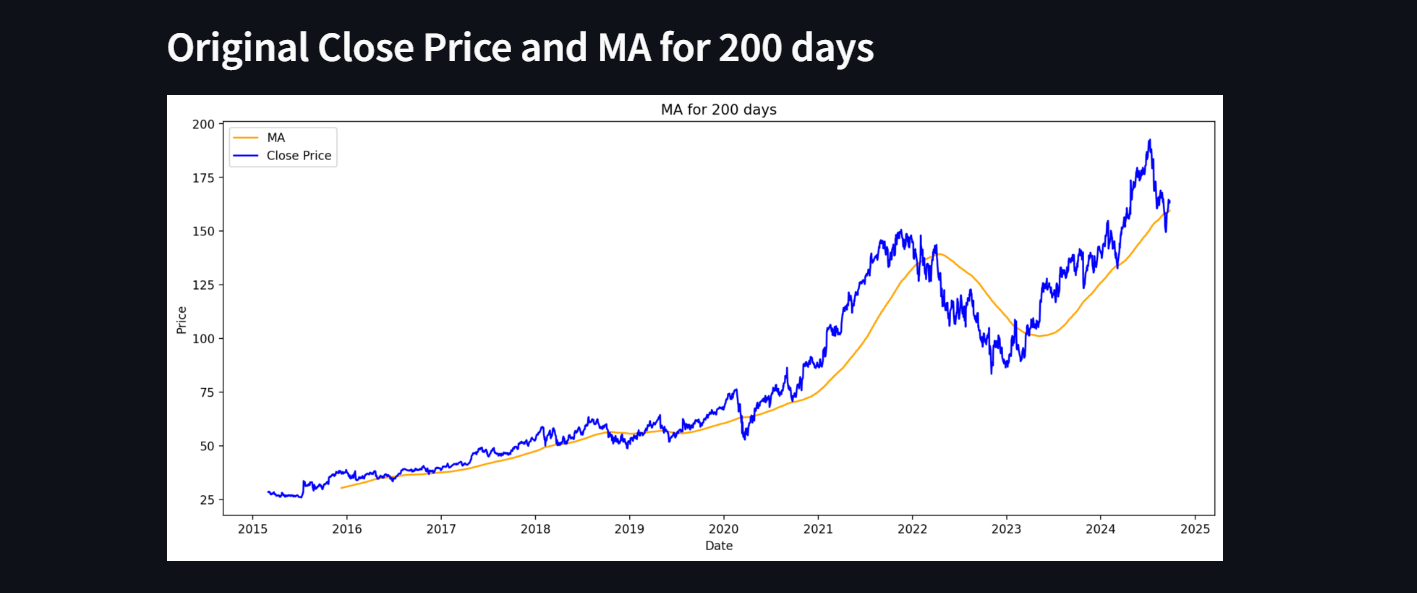
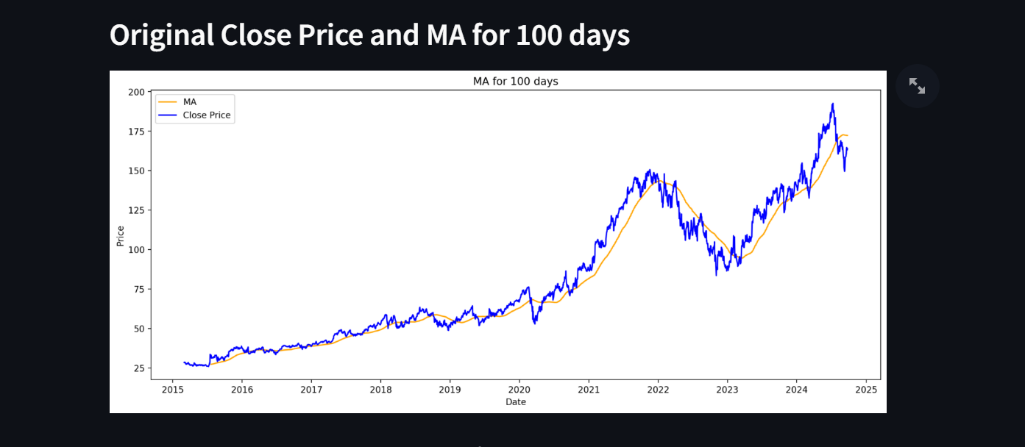
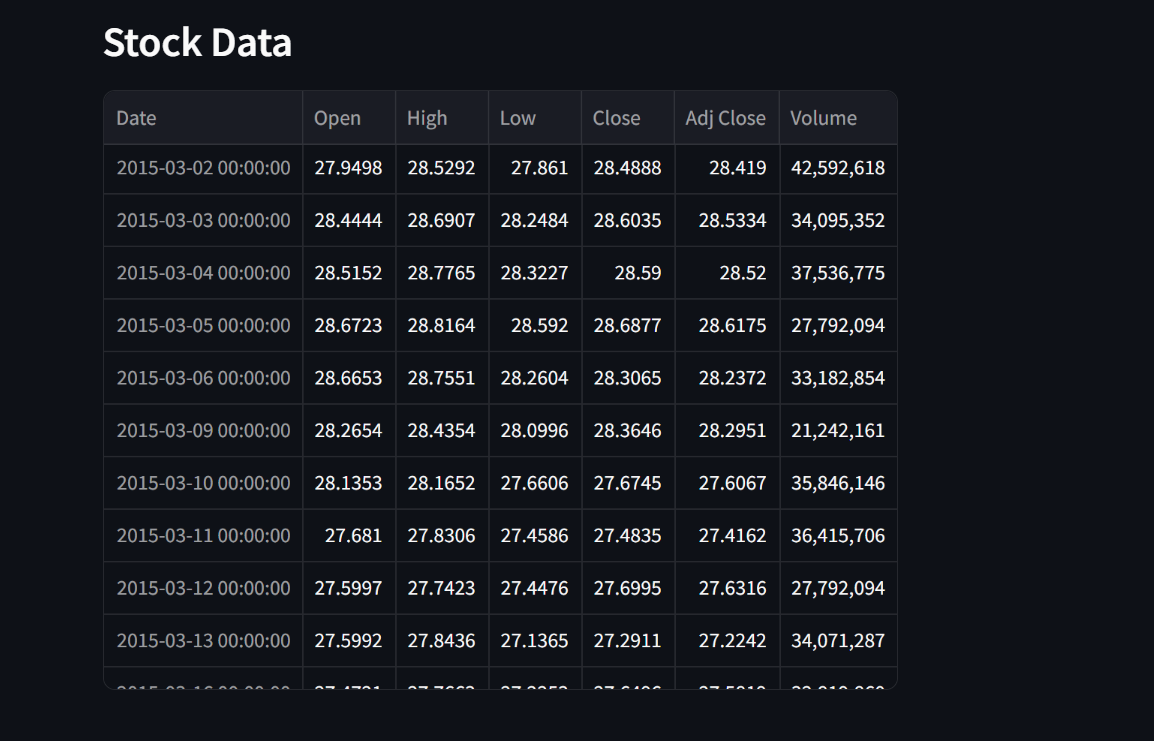
  top: 10px;

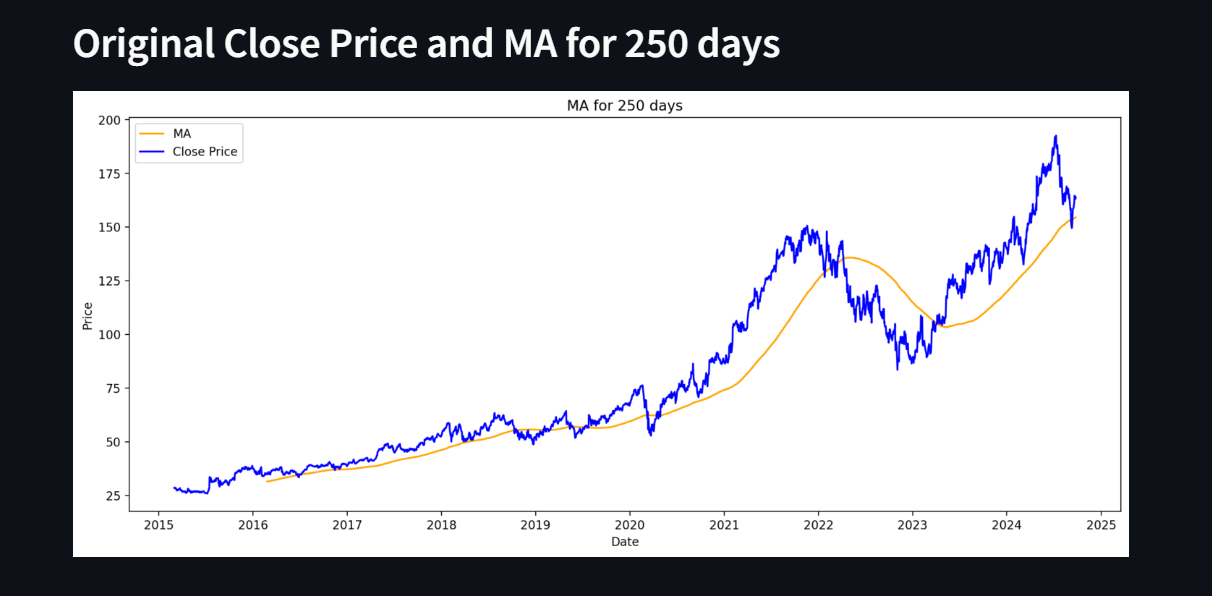
  right: 10px;

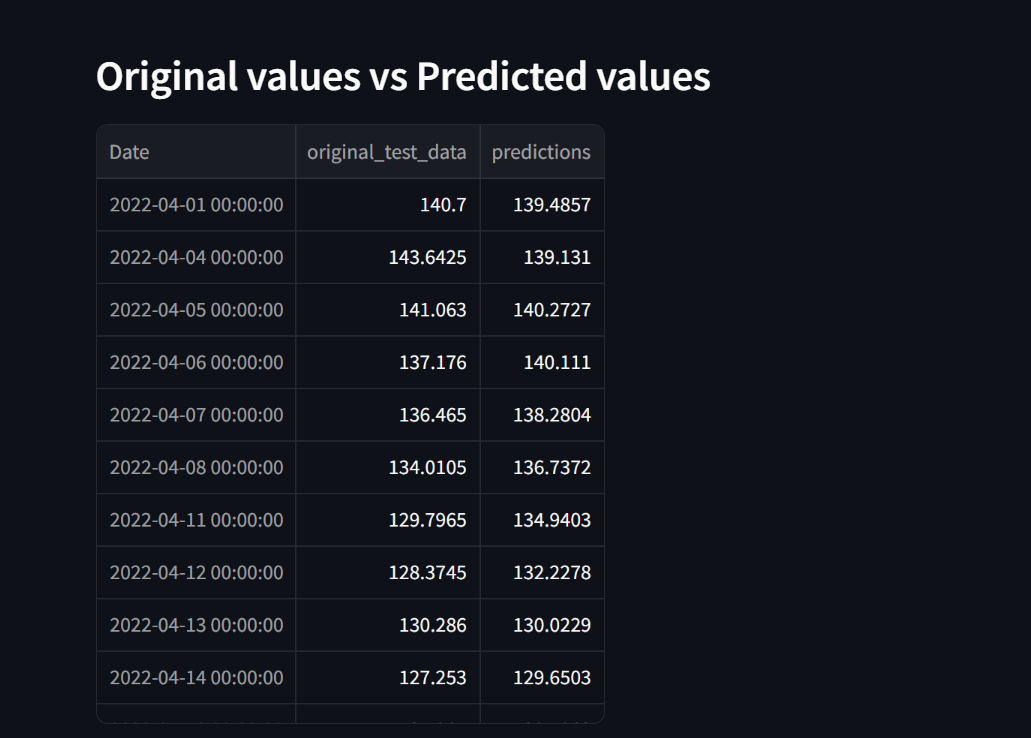
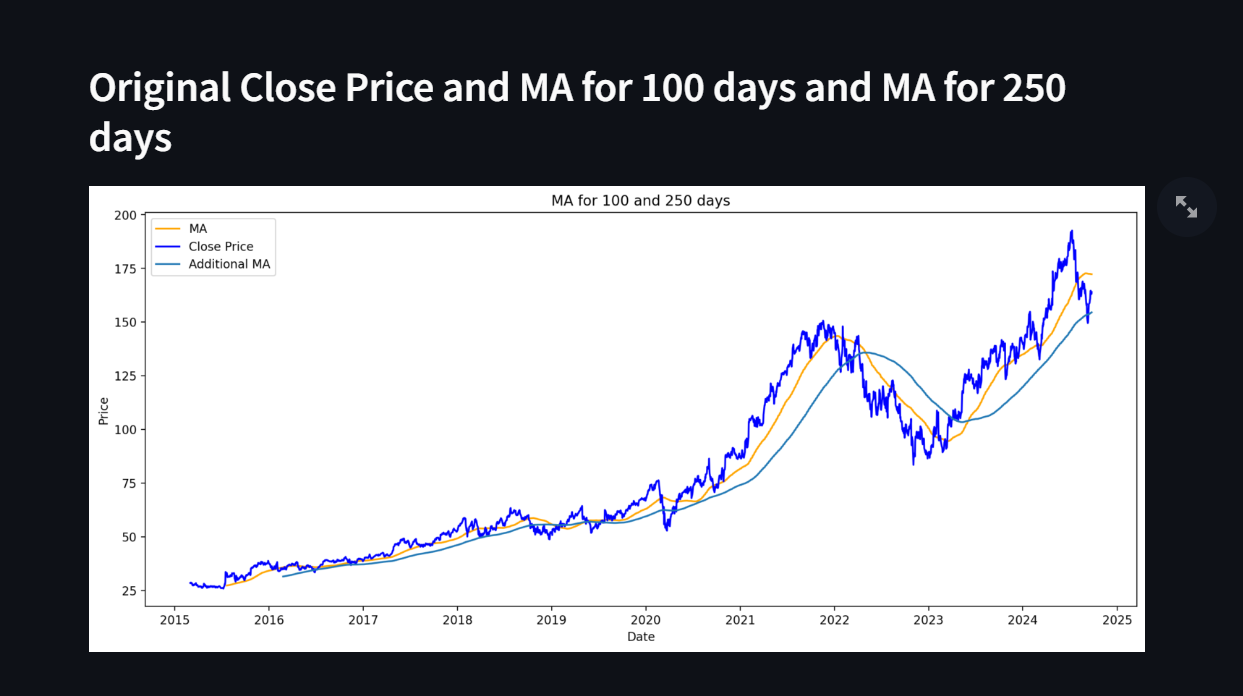
  z-index: 1000;

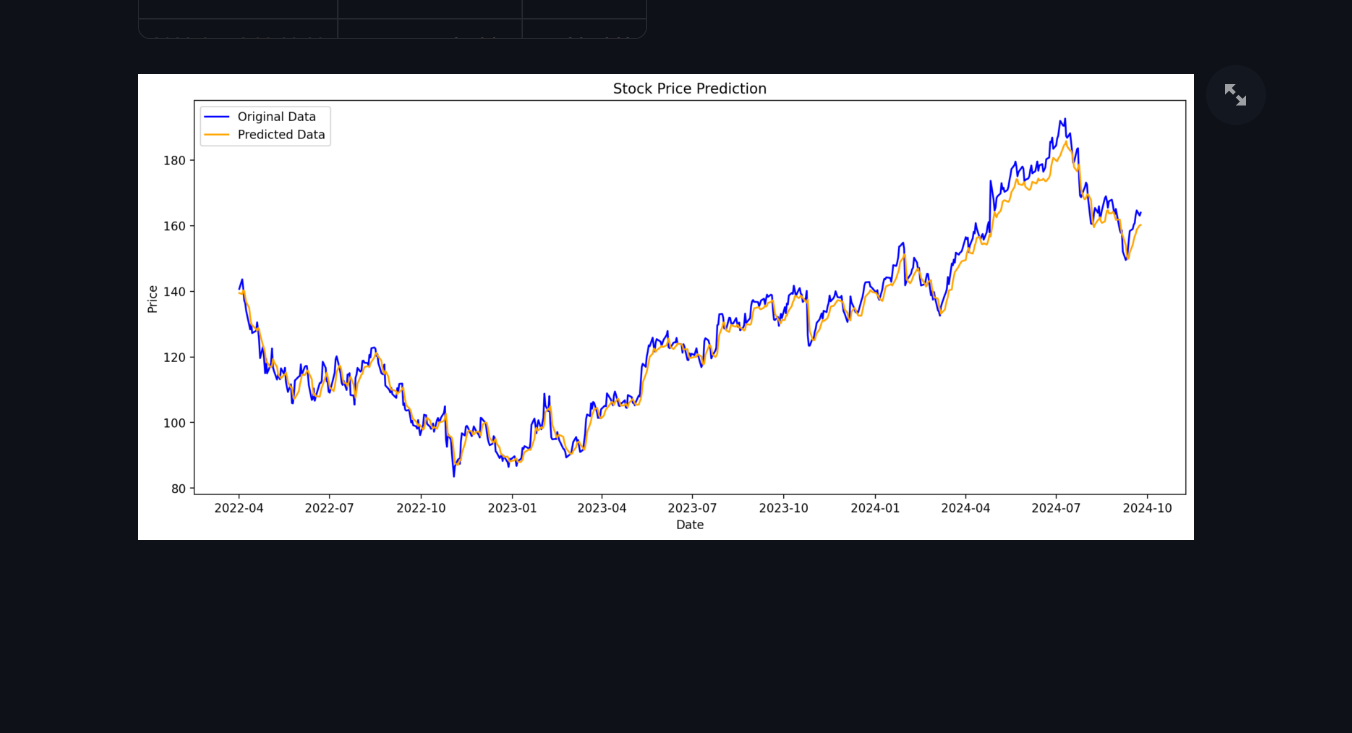
}

**8. SCREENSHOTS**

****

****

****

****

**9. SYSTEM TESTING**

**Introduction:-**

Testing is the process of finding errors in the software. Testing presents an interesting anomaly for the software engineer. During earlier software engineering activities, the engineer attempts to build software from an abstract concept to a successful product. Now comes testing. The engineer creates a series of test cases that are intended to “demolish” the software has been built. In fact, testing is the one step in software process that could be viewed as destructive rather than constructive.

* To ensure that during operation the system will perform as per specification.
* To make sure that system meets the user requirements during operation.
* To make sure that during the operation, incorrect input, processing and output will be detected
* To see that when correct inputs are fed to the system the outputs are correct
* To verify that the controls incorporated in the same system as intended
* Testing is a process of executing a program with the intent of finding a error
* A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error. Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

**TYPES OF TESTING**

**9.1 White box testing**

White box testing Is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

**9.2 Black box testing**

Black box testing attempts to find errors in following area or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

**9.3 LEVELS OF TESTING**

The following are the different tests at various level:

**9.3.1 Unit Testing:**

Unit testing is essentially for the verification of the code procedure during the coding phase and the goal is test the internal logic of the module/program. In the generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

**9.3 Integration Testing:**

All the tested modules are combined into sub systems, which are then tested, the goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

**9.3.3 Validation Testing**

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

**9.3.4 System Testing**

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

Implementing the system in a simulated production environment and testing it.

Introducing errors and testing for error handling.

**9.4 Implementation of testing schedule**

Testing is done at different levels such as,

* Low level:

This is done by same developer

* Medium level:

This is done by same developer

* High level:

This is done by independent body (guide).

**User Login:-**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case  Number | Check  Item | Test Case  Objective | Step To Execute | Test Data | Expected Result | Actual Result |
| 1 | Login Page | Leave all fields empty | Click Submit | \*\*\*\*\*\*\*\* | Prompt Error Message | Yes |
| 2 | User Page | Enter Invalid username | Click login | 1234  Admin | Prompt Error Message | Yes |
| 3 | Username | Enter a valid username & password | Click Submit | Srkashif028  srkashif | Prompt login successful | Yes |

**Register Page:-**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case  Number | Check  Item | Test Case  Objective | Step To Execute | Test Data | Expected Result | Actual Result |
| 1 | Register Page | Leave all fields empty | Click Submit | \*\*\*\*\*\*\*\* | Prompt Error Message | Yes |
| 2 | Register Page | Enter Invalid username | Click login | Hello.com | Any Type Of Email is Accepted | Yes |
| 3 | Email | Enter a valid email | Click Submit | [hello@gmail.com](mailto:hello@gmail.com) | Prompt no error | Yes |
| 4 | Password | Enter Password | Click Submit | 12345 | Prompt no error  (It can have any value) | Yes |

**10. CONCLUSION**

**Conclusion:-**

the development of a stock market prediction app holds tremendous potential to empower investors with actionable insights and tools for informed decision-making. By leveraging advanced technologies like AI and machine learning, the app can analyze vast amounts of historical data, identifying patterns that human analysts may overlook. Integrating real-time market data, sentiment analysis from news and social media, and customizable dashboards will provide users with a holistic view of market trends and shifts.

Moreover, the incorporation of risk assessment tools enables users to evaluate their portfolios critically, helping them make strategic decisions based on potential market scenarios. Educational resources, such as in-app tutorials and webinars featuring market experts, will further enhance user knowledge, fostering a more informed investment community.

Community features, such as forums and social trading options, can facilitate collaboration and knowledge sharing among users, promoting a sense of belonging and support. Visualization tools, including interactive charts and heat maps, will make complex data more accessible and easier to interpret.

Additionally, integrating features that focus on sustainability, such as ESG ratings, can attract socially conscious investors, aligning financial goals with ethical considerations. The inclusion of voice commands and chatbot assistance can streamline user interactions, making the app more intuitive and user-friendly.

As the market continues to evolve, the importance of continuous updates and user feedback cannot be overstated. Regular enhancements based on user experiences will ensure the app remains relevant, accurate, and trustworthy.

Ultimately, this app can serve as a vital companion in navigating the complexities of the stock market, equipping users with the tools and insights they need to achieve their financial goals with confidence. By fostering a more educated, engaged, and connected investment community, the app can contribute to a more dynamic and informed market environment.

This expanded conclusion aims to encapsulate the app's features while emphasizing its potential impact on users and the broader investment community.

Video conferencing is one of the best ways of communication for large organizations as they provide an

instant and reliable method through which the entire organization can connect, communicate, and

collaborate. Not only do video conferencing tools make communication efficient, more comfortable, and

cheaper for the organization but they also provide a lot of intangible benefits such as increasing the

productivity of the employees in the organization in general

Video conferencing is one of the best ways of communication for large organizations as they provide an

instant and reliable method through which the entire organization can connect, communicate, and

collaborate. Not only do video conferencing tools make communication efficient, more comfortable, and

cheaper for the organization but they also provide a lot of intangible benefits such as increasing the

productivity of the employees in the organization in general

Video conferencing is one of the best ways of communication for large organizations as they provide an

instant and reliable method through which the entire organization can connect, communicate, and

collaborate. Not only do video conferencing tools make communication efficient, more comfortable, and

cheaper for the organization but they also provide a lot of intangible benefits such as increasing the

productivity of the employees in the organization in general

**11. FUTURE ENHANCEMENT**

Enhancing a stock market prediction app can significantly improve user experience and accuracy. Here are some future enhancement ideas:

**1. AI-Powered Insights**

* **Machine Learning Algorithms**: Implement advanced algorithms to analyze historical data and identify patterns, improving prediction accuracy.
* **Sentiment Analysis**: Use natural language processing to gauge market sentiment from news articles, social media, and financial reports.

**2. Personalized Dashboards**

* **Customizable Interfaces**: Allow users to create personalized dashboards that highlight their preferred stocks, news, and metrics.
* **Alerts and Notifications**: Enable customizable alerts for price changes, news events, and significant market movements.

**3.** **Real-Time Data Integration**

* **Live Market Data**: Provide real-time updates on stock prices, trading volume, and market trends.
* **News Aggregation**: Integrate live feeds from financial news sources to keep users informed about relevant events.

**4. Risk Assessment Tools**

* **Portfolio Analysis**: Offer tools to analyze the risk and performance of users’ portfolios based on historical data and predictive analytics.
* **Scenario Simulation**: Allow users to simulate different market scenarios and their potential impact on investments.

**5. Educational Resources**

* **In-App Tutorials**: Include resources and tutorials on stock market concepts, investment strategies, and risk management.
* **Webinars and Expert Insights**: Host webinars with market experts to provide users with additional insights and strategies.

**6. Social Trading Features**

* **Community Forums**: Create a platform for users to discuss strategies, share insights, and seek advice.
* **Copy Trading**: Enable users to follow and replicate the trades of experienced investors.

**7. Enhanced Visualization Tools**

* **Interactive Charts**: Implement advanced charting tools that allow users to visualize data with multiple indicators and overlays.
* **Heat Maps**: Use heat maps to display market performance, helping users quickly identify trends.

**8. Integration with Financial Tools**

* **Tax Optimization Features**: Offer tools to help users optimize their portfolios for tax efficiency.
* **Budgeting Tools**: Integrate budgeting features that allow users to manage their investment contributions effectively.

**9. Multi-Asset Coverage**

* **Beyond Stocks**: Expand to include predictions and analytics for ETFs, cryptocurrencies, commodities, and other asset classes.

**10. Sustainability Metrics**

* **ESG Ratings**: Include environmental, social, and governance (ESG) ratings to help users make socially responsible investment choices.

**11. Voice and Chatbot Integration**

* **Voice Commands**: Enable voice commands for quick queries and commands within the app.
* **Chatbot Assistance**: Implement a chatbot for user queries, providing immediate responses and assistance.

**12. User Feedback Loop**

* **Continuous Improvement**: Create mechanisms for users to provide feedback on predictions and features to improve the app continuously.

Implementing these enhancements could make your stock market prediction app more comprehensive, user-friendly, and valuable to both novice and experienced investors.

**12. BIBLIOGRAPHY**

**Bibliography:-**

**The following books were referred during the analysis and execution phase of the project.**

[**https://tailwindcss.com/**](https://tailwindcss.com/)

[**https://nextjs.org/**](https://nextjs.org/)

[**https://streamlit.io/**](https://streamlit.io/)