## ABSTRACT

The nature of stock market movement has always been ambiguous for investors because of various influential factors. This study aims to significantly reduce the risk of trend prediction with machine learning and deep learning algorithms. Four stock market groups, namely diversified financials, petroleum, non-metallic minerals and basic metals from Tehran stock exchange, are chosen for experimental evaluations. This study compares nine machine learning models (Decision Tree, Random Forest, Adaptive Boosting (Adaboost), eXtreme Gradient Boosting (XGBoost), Support Vector Classifier (SVC), Naïve Bayes, K-Nearest Neighbors (KNN), Logistic Regression and Artificial Neural Network (ANN)) and two powerful deep learning methods (Recurrent Neural Network (RNN) and Long short-term memory (LSTM).

Ten technical indicators from ten years of historical data are our input values, and two ways are supposed for employing them. Firstly, calculating the indicators by stock trading values as continuous data, and secondly converting indicators to binary data before using. Each prediction model is evaluated by three metrics based on the input ways. The evaluation results indicate that for the continuous data, RNN and LSTM outperform other prediction models with a considerable difference. Also, results show that in the binary data evaluation, those deep learning methods are the best; however, the difference becomes less because of the noticeable improvement of models' performance in the second way.

## INDEX

### ACKNOWLEDGEMENT iii

**ABSTRACT** iv

### INDEX

**LIST OF FIGURES**

### LIST OF TABLES

**LIST OF SCREENSHOTS**

v vii vii vii

**1.INTRODUCTION**

1.1 ABOUT PROJECT 8

1.2 EXISTING SYSTEM 10

1.3 PROPOSED SYSTEM 10

### 2.REQUIREMENT SPECIFICATIONS

2.1 REQUIREMENT ANALYSIS 12

2.2 FEASIBILITY STUDY 13

2.3 SPECIFICATION PRINCIPLES 13

### 3.SYSTEM DESIGN

3.1 INPUT DESIGN 17

3.2 OUTPUT DESIGN 17

3.3 ARCHITECTURE DIAGRAM 18

3.4 FLOW CHART: REMOTE USER 19

3.5 FLOW CHART: SERVICE PROVIDER 20

3.6 DATA FLOW DIAGRAM 21

3.7 USE CASE DIAGRAM 22

3.8 SEQUENCE DIAGRAM 23

### 4.IMPLEMENTATION

4.1 PROJECT MODULES 24

4.2 MACHINE LEARNING ALGORITHMS 25

4.3 DEEP LEARNING ALGORITHMS 26

4.4 SOURCE CODE 27

### 5.TESTING

5.1 SOFTWARE TESTING 33

5.2 TYPES OF TESTS 33

5.3 SYSTEM TESTING 35

### 6.RESULTS 38

### 7.CONCLUSION 42

### 8.REFERENCES 43

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Particulars** | **Page No.** |
| 3.3 | Architecture Diagram | 18 |
| 3.4.1 | Flow Chart: Remote User | 19 |
| 3.4.2 | Flow Chart: Service Provider | 20 |
| 3.5 | Data Flow Diagram | 21 |
| 3.6 | Use Case Diagram | 22 |
| 3.7 | Sequence Diagram | 23 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Particulars** | **Page No.** |
| 2.1 | Hardware Requirements | 12 |
| 2.2 | Software Requirements | 12 |

## LIST OF SCREENSHOTS

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Particulars** | **Page No.** |
| 6.1 | User Login Page | 38 |
| 6.2 | Service Provider Login Page | 38 |
| 6.3 | Uploading Dataset | 39 |
| 6.4 | Uploaded Dataset | 39 |
| 6.5 | Searching for Company Stock Market trend | 40 |
| 6.6 | Output | 40 |
| 6.7 | All Remote Users | 41 |
| 6.8 | Dataset | 41 |

* 1. **ABOUT PROJECT**

# 1. INTRODUCTION

The task of stock prediction has always been a challenging problem for statistics experts and finance. The main reason behind this prediction is buying stocks that are likely to increase in price and then selling stocks that are probably to fall. Generally, there are two ways for stock market prediction. Fundamental analysis is one of them and relies on a company's technique and fundamental information like market position, expenses and annual growth rates. The second one is the technical analysis method, which concentrates on previous stock prices and values. This analysis uses historical charts and patterns to predict future prices [1], [2]. Stock markets were normally predicted by financial experts in the past time. However, data scientists have started solving prediction problems with the progress of learning techniques.

Also, computer scientists have begun using machine learning methods to improve the performance of prediction models and enhance the accuracy of predictions. Employing deep learning was the next phase in improving prediction models with better performance [3], [4]. Stock market prediction is full of challenges, and data scientists usually confront some problems when they try to develop a predictive model. Complexity and nonlinearity are two main challenges caused by the instability of stock market and the correlation between investment psychology and market behavior [5]. It is clear that there are always unpredictable factors such as the public image of companies or political situation of countries, which affect stock markets trend.

Therefore, if the data gained from stock values are efficiently preprocessed and suitable algorithms are employed, the trend of stock values and index can be predicted. In stock market prediction systems, machine learning and deep learning approaches can help investors and traders through their decisions. These methods intend to automatically recognize and learn patterns among big amounts of information. The algorithms can be effectively self-learning, and can tackle the predicting task of price puctuations in order to improve trading strategies [6].

Since recent years, many methods have been improved to predict stock market trends. The implementation of a model combination with Genetic Algorithms (GA), Artificial Neural Networks and Hidden Markov Model (HMM) was proposed by Hassan et al. [7]; the purpose was transforming the daily stock values to independent groups of prices as inputs to HMM. The predictability of financial trend with SVM model by evaluating the weekly trend of NIKKEI 225 index was investigated by Huang et al. [8]. A comparison between SVM, Linear Discriminant method, Elman Back propagation Neural Networks and Quadratic Discriminant method was their goal. The results indicated that SVM was the best classifier method. New financial prediction algorithm based on SVM ensemble was proposed by Sun et al. [9].

The method for choosing SVM ensemble' s base classifiers was proposed by deeming both diversity analysis and individual prediction. Final results showed that SVM ensemble was importantly better than individual SVM for classification. Ten data mining methods were employed by Ou et al. [10] to predict value trends of Hang index from Hong Kong market.

The methods involved Tree based classification, K-nearest neighbor, Bayesian classification, SVM and neural network. Results indicated that the SVM outperformed other predictive models. The value puctuations by a developed Legendre neural network was forecasted by Liu et al. [11] by assuming investors' positions and their decisions by analyzing the prior data on the values. Indeed, they examined a random function (time strength) in the prediction model. Araújo et al. [12] proposed the morphological rank linear forecasting approach to compare its results with time-delay added evolutionary forecasting approach and multilayer perceptron networks. From the above research background, it is clear that each of the algorithms can effectively solve stock prediction problems. However, it is vital to notice that there are specific limitations for each of them. The prediction results not only are affected by the representation of the input data but also depend on the prediction method. Moreover, using only prominent features and identifying them as input data instead of all features can noticeably develop the accuracy of the prediction models.

Employing tree-based ensemble methods and deep learning algorithms for predicting the stock and stock market trend is a new area of research activities. In light of employing bagging and majority vote methods, Tsai et al. [13] used two different kinds of ensemble classifiers, such as heterogeneous and homogeneous methods. They also consider macroeconomic features and financial ratios from Taiwan stock market to examine the performance of models. The results demonstrated that with respect to the investment returns and prediction accuracy, ensemble classifiers were superior to single classifiers. Ballings et al. [14] compared the performance of Ada Boost, Random Forest and kernel factory versus single models involving SVM, KNN, Logistic Regression and ANN. They predict European company's prices for one year ahead.

The results showed that Random Forest outperformed among all models. Basak et al. [15] employed XGBoost and Random Forest methods for the classification problem to forecast the stock increase or decrease based on previous values. Results showed that the prediction performances have advanced for several companies in comparison with the existing ones. For examining macroeconomic indicators to accurately predict stock market for one-month ahead, Weng et al. [16] improved four ensemble models, boosting regressor, bagging regressor, neural network ensemble regressor and random forest regressor. Indeed,another aim was employing a hybrid way of LSTM to prove that the macroeconomic features are the most successful predictors for stock market. Moving on using deep learning algorithms, a deep neural network algorithm with the transaction records and public market data was investigated by Long et al. [17] to assess stock price trends. Their nal results indicated that bidirectional LSTM could forecast the future of market for investors, and the technique attained the greatest performance.

The employment of RNN and CNN algorithms was examined by Rekha et al. [18] to compare the accuracy of those with real values from stock markets. LSTM with an automatic encoder and LSTM with an embedded layer were utilized by Pang et al. [19] to acquire better stock market estimations. The result of experimental works indicated that LSTM with an embedded layer outperformed for the Shanghai composite index with 57.2% accuracy. The deep convolutional LSTM algorithm was employed by Kelotra and Pandey [20] to efficiently calculate stock market movements. They used a model with Rider-based monarch buttery optimization method and gained the RMSE and MSE of 2.6923 and 7.2487. A forecasting LSTM model and an overating prevention LSTM module were suggested by Baek and Kim [21] to predict stock market

## EXISTING SYSTEM

Stock market trends can be affected by external factors such as public sentiment and political events. The goal of this research is to find whether or not public sentiment and political situation on a given day can affect stock market trends of individual companies or the overall market. For this purpose, the sentiment and situation features are used in a machine learning model to find the effect of public sentiment and political situation on the prediction accuracy of algorithms for 7 days in future. Besides, interdependencies among companies and stock markets are also studied. For the sake of experimentation, stock market historical data are downloaded from Yahoo! Finance and public sentiments are obtained from Twitter. Important political events data of Pakistan are crawled from Wikipedia. The raw text data are then pre-processed, and the sentiment and situation features are generated to create the final data sets. Ten machine learning algorithms are applied to the final data sets to predict the stock market future trend. The experimental results show that the sentiment feature improves the prediction accuracy of machine learning algorithms by 0–3%, and political situation feature improves the prediction accuracy of algorithms by about 20%. Furthermore, the sentiment attribute is most effective on day 7, while the political situation attribute is most effective on day 5. SMO algorithm is found to show the best performance, while ASC and Bagging show poor performance. The interdependency results indicate that stock markets in the same industry show a medium positive correlation with each other.

## DISADVANTAGES

1. In the existing work, the system in which Stock market prediction is full of challenges, and data scientists usually confront some problems when they try to develop a predictive model.
2. This system is less performance in which it is clear that there are always unpredictable factors such as the public image of companies or political situation of countries, which affect stock markets trend.
3. The quality of the input data is crucial for the performance of machine learning models. Noisy, incomplete, or inaccurate data can lead to suboptimal predictions. Preprocessing continuous and binary data might involve handling missing values, outliers, and normalizing features, which can introduce complexities and uncertainties.
4. Machine learning and deep learning models often have hyperparameters that need to be tuned for optimal performance. The sensitivity of these models to hyperparameters may lead to challenges in finding the right configuration, and suboptimal choices could impact predictive accuracy.
5. Market sentiment, which can play a crucial role in stock price movements, is challenging to quantify and incorporate effectively into machine learning models. Sentiment analysis from news articles, social media, or other sources may introduce noise and subjectivity.

### PROPOSED SYSTEM

In the proposed system, the system concentrates on comparing prediction performance of nine machine learning models (Decision Tree, Random Forest, Adaboost, XGBoost, SVC, Naïve Bayes, KNN, Logistic Regression and ANN) and two deep learning methods (RNN and LSTM) to predict stock market movement. Ten technical indicators are utilized as inputs to our models.

The proposed study includes two different approaches for inputs, continuous data and binary data, to investigate the effect of preprocessing; the former uses stock trading data (open, close, high and low values) while the latter employs preprocessing step to convert continuous data to binary one. Each technical indicator has its specific possibility of up or down movement based on market inherent properties.

The performance of the mentioned models is compared for the both approaches with three classification metrics, and the best tuning parameter for each model (except Naïve Bayes and Logistic Regression) is reported. All experimental tests are done with ten years of historical data of four stock market groups (petroleum, diversified financials, basic metals and non-metallic minerals), that are totally crucial for investors, from Tehran stock exchange. We believe that this study is a new research paper that incorporates multiple machine learning and deep learning methods to improve the prediction task of stock groups' trend and movement.

### ADVANTAGES

1. Machine learning and deep learning algorithms can analyze large datasets and identify complex patterns that may be difficult for traditional methods to recognize.
2. Machine learning models can adapt to changing market conditions, making them more robust in handling dynamic and unpredictable stock market trends. This adaptability allows the system to continuously learn and update its predictions based on new data.
3. Advanced algorithms can automatically extract relevant features from continuous and binary data, identifying key factors that influence stock market trends. This helps in focusing on the most important information for making predictions.
4. Machine learning models can process data in real-time, enabling timely decision-making. This is crucial in the stock market, where quick responses to changing conditions can be advantageous.
5. The inclusion of a comparative analysis between different machine learning and deep learning algorithms allows for a better understanding of the strengths and weaknesses of each approach. This can guide future improvements and optimizations.

# 2.REQUIREMENTS SPECIFICATIONS

## REQUIREMENT ANALYSIS

* + 1. **HARDWARE REQUIREMENTS**

|  |  |
| --- | --- |
| Processor | Pentium-IV |
| Hard Disk | 20 GB |
| Laptop with - | External mouse and keyboard |
| Monitor | SVGA |
| RAM | 4 GB |

## Table 2.1

* + 1. **SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| Operationg system | Windows 7 Ultimate |
| Coding Language | Python |
| Front-End | Python |
| Back-End | Django-ORM |
| Designing | Html, CSS, Java Script |
| Data Base | MySQL (WAMP Server) |

## Table 2.2

* 1. **FEASIBILITY STUDY**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

### Operational Feasibility

* **Economic Feasibility**

### Technical Feasibility

**Operational Feasibility**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

### Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

### Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

## SPECIFICATION PRINCIPLES

### SOFTWARE DESCRIPTION Python:

Python is a **high-level, interpreted**,**interactive** and **object-oriented scripting language**. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages

.

* **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

**Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games

### History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

* 6 It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

### Python Features

Python's features include:

* **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
* **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases:** Python provides interfaces to all major commercial databases.
* **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable:** Python provides a better structure and support for large programs than shell scripting.

### 2.3.1 SOFTWARE REQUIREMENTS

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development.

### Operating system

The operating system (OS) manages all of the software and hardware on the computer. It performs basic tasks such as file, memory and process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

The operating system we have used is Windows 10.

### Front-end

Front end is the layer or element that the user has the ability to use, see, and interact with through buttons, images, interactive elements, navigational menus, and text. It is the main interface with which the user experiences the project. It includes programming as well as scripting languages for designing. Html, CSS and Javascript are used for designing the user interface of the project.

The programming language used for the front end is Python and for the designing part we have used HTML, CSS and JavaScript.

### Back-end

The backend (or “server side”) is the portion of the website you don't see. It's responsible for storing and organizing data, and ensuring everything on the client-side actually works. The backend communicates with the frontend, sending and receiving information to be displayed as a web page.

The back-end of our project is done with the help of Django-ORM.

### Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management

system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just a database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

We have used the WampServer, which is a Windows web development environment that allows users to create web applications with a MySQL database.

### Coding language

The programming language used for this project is python. It brings an exceptional amount of power and versatility to machine learning environments. The language's simple syntax simplifies data validation and streamlines the scraping, processing, refining, cleaning, arranging and analyzing processes, thereby making collaboration with other programmers less of an obstacle.

The coding language we have used is Python with the latest version of 3.12.0.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

1. Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
2. Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
3. Python is Object-Oriented**:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
4. Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

# SYSTEM DESIGN AND DEVELOPMENT

## INPUT DESIGN

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

## OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project wilrun on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

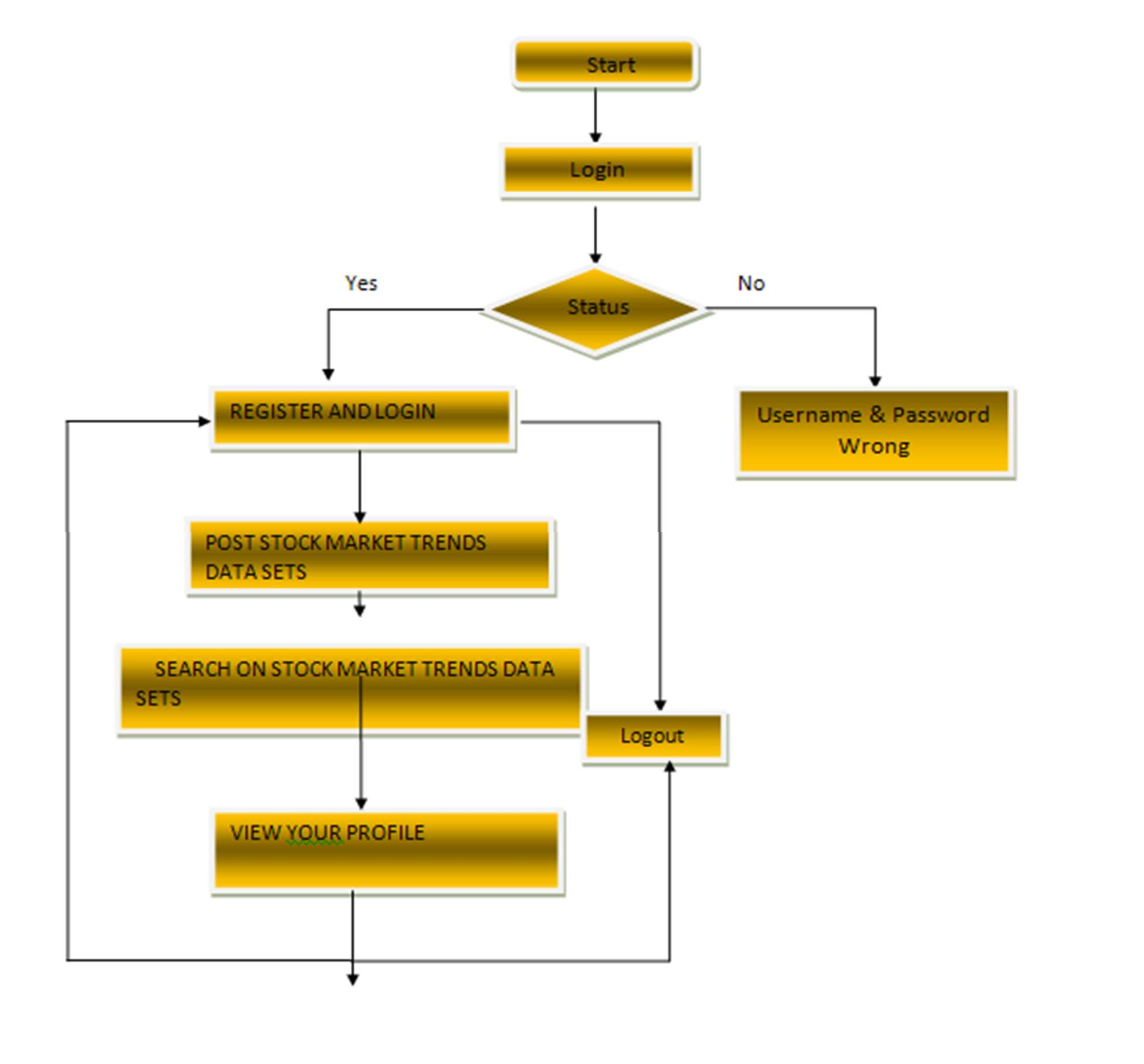
## ARCHITECTURE DIAGRAM

**Figure.3.3. Architecture Diagram**

### REGISTER AND LOGIN

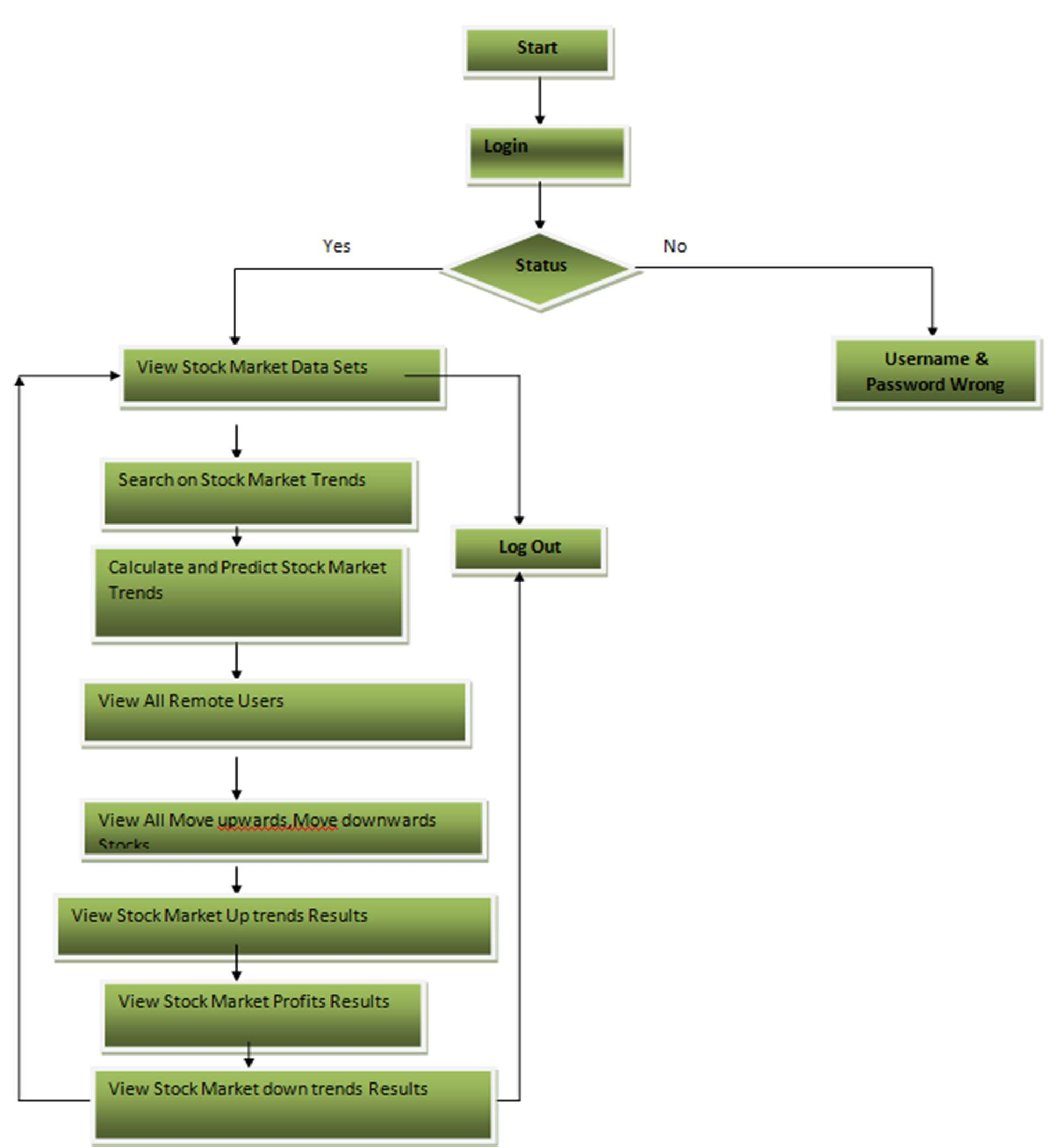
* **PREDICT URLs TYPE**
* **VIEW YOUR PROFIL**

## FLOW CHART: REMOTE USER



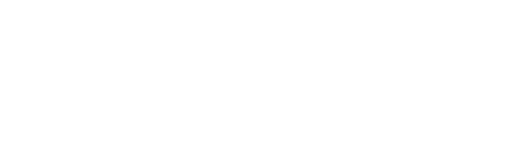
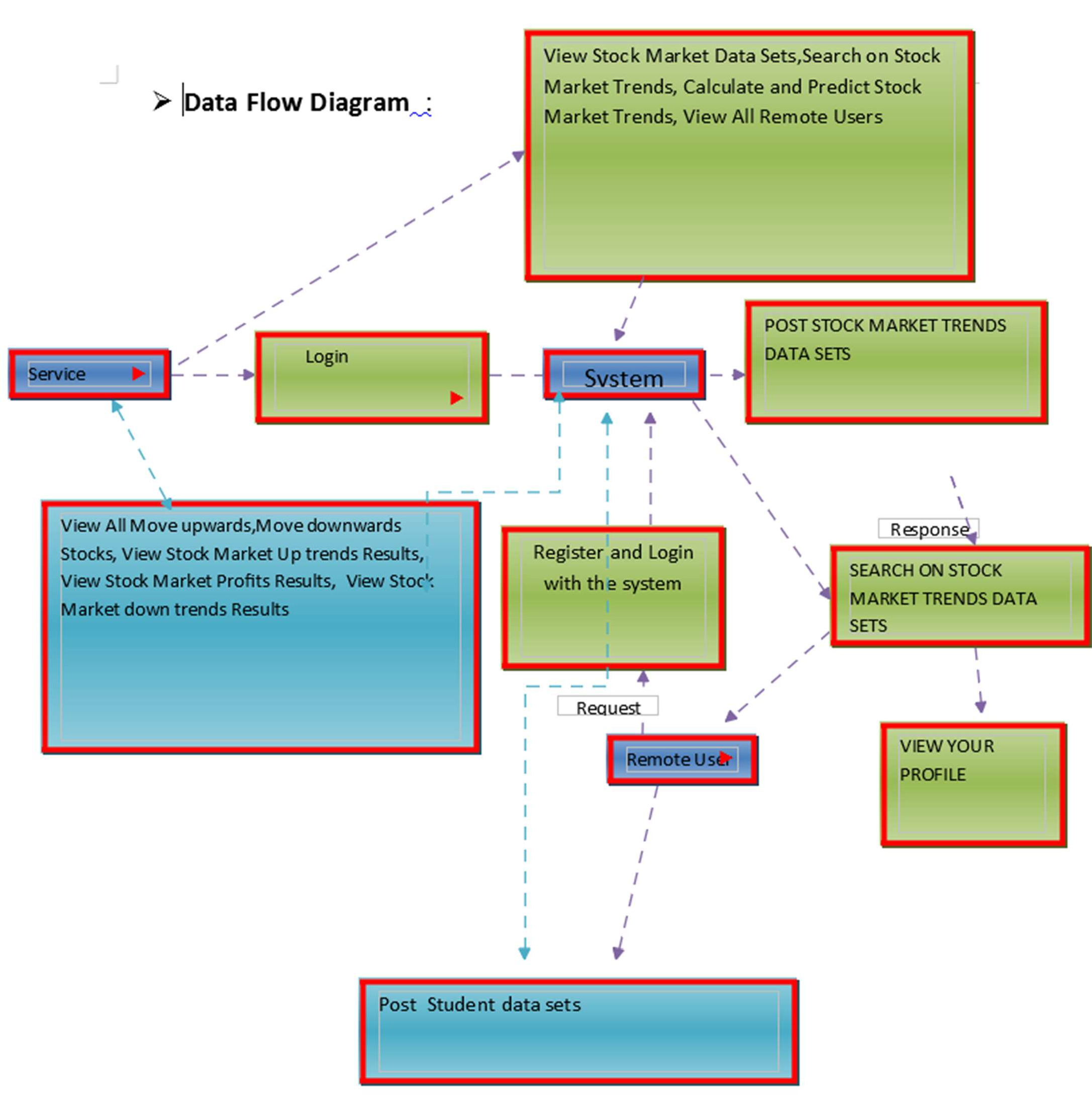
**Figure.3.4.1. Flow Chart of Remote User**

### FLOW CHART: SERVICE PROVIDER



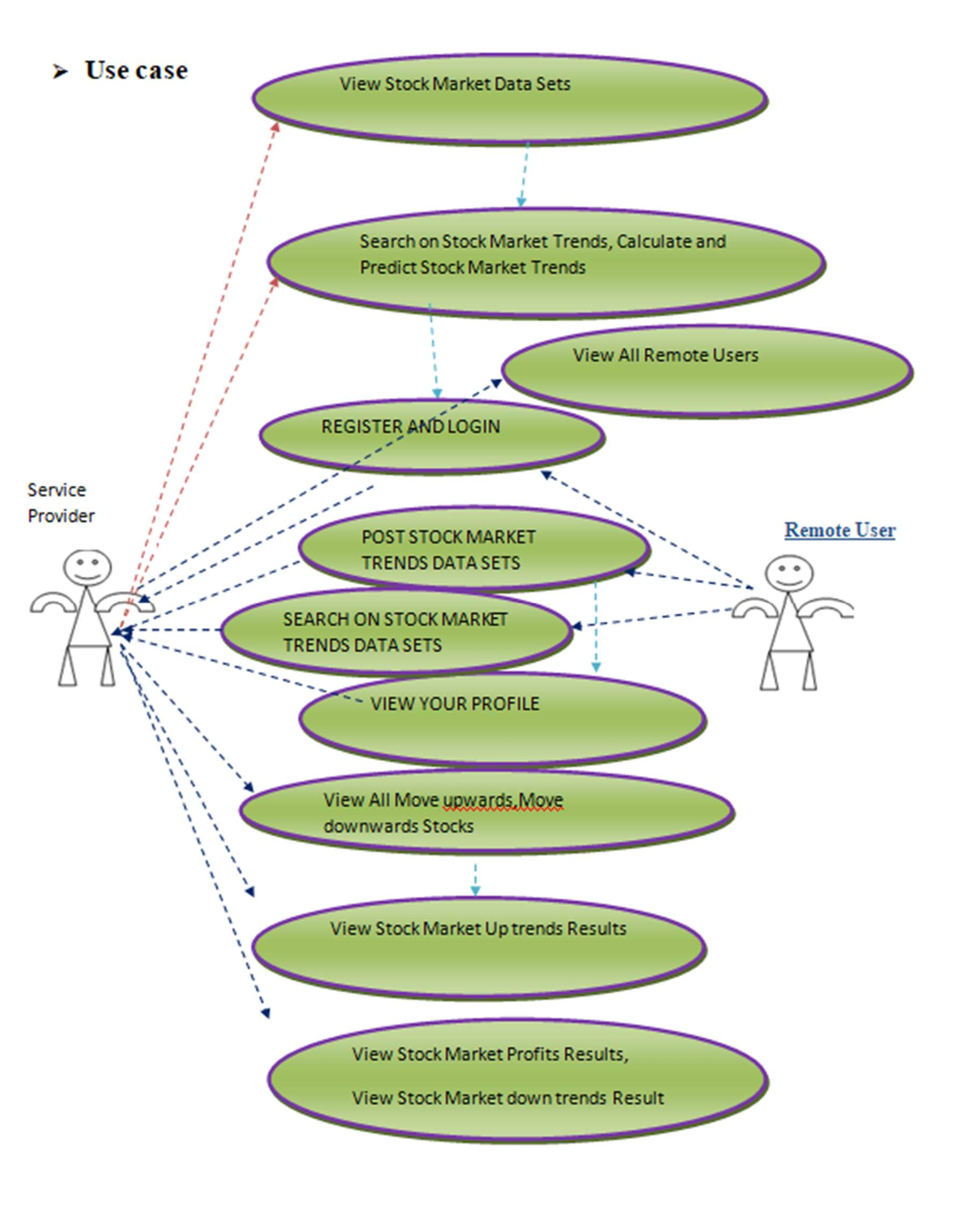
**Figure.3.4.2.Flow Chart of Service Provider**

## DATA FLOW DIAGRAM



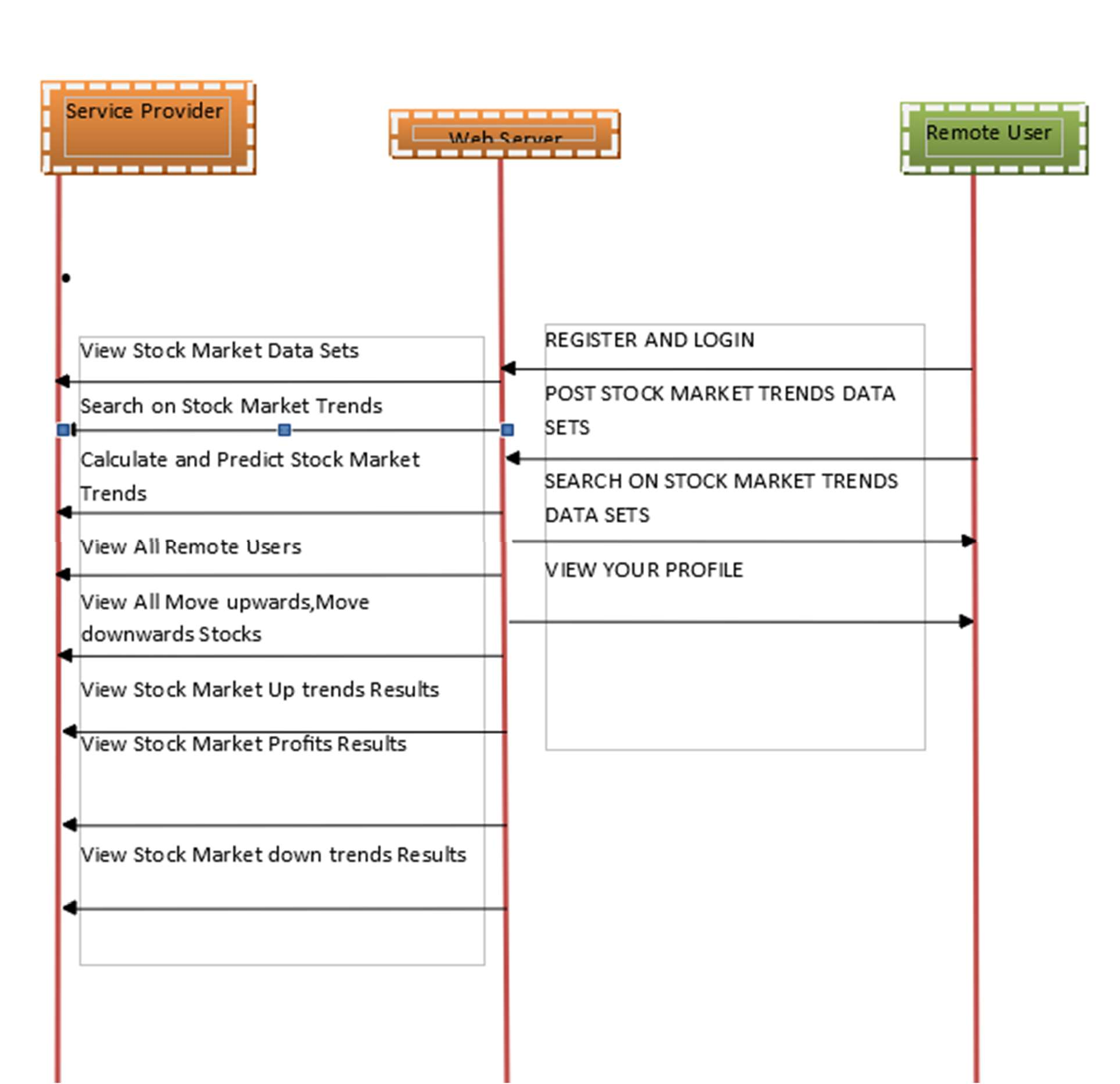
**Figure.3.5.Data Flow Diagram**

## USE CASE DIAGRAM



**Figure.3.6. Use Case Diagram**

## SEQUENCE DIAGRAM



**Figure .3.7. Sequence Diagram**

* 1. **PROJECT MODULES**

# IMPLEMENTATION

In this project there are modules as follows:

* + Service Provider
  + View and Authorize Users
  + Remote User

### Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as View Stock Market Data Sets, Search on Stock Market Trends, Calculate and Predict Stock Market Trends, View All Remote Users, View All Move upwards, Move downwards Stocks, View Stock Market Up trends Results.

View Stock Market Profits Results, View Stock Market down trends Results.

### View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

### Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like POST STOCK MARKET TRENDS DATA SETS, SEARCH ON STOCK MARKET TRENDS DATA SETS, and VIEW YOUR PROFILE

* 1. **MACHINE LEARNING ALGORITHMS**

**Artificial Neural Networks (ANN):**

Artificial Neural Networks are computational models inspired by the structure and functioning of the human brain. Comprising layers of interconnected nodes (neurons), ANNs learn patterns and relationships from data. Input data is processed through the network's layers, with each connection assigned a weight. Through training, the network adjusts these weights to minimize the difference between predicted and actual outcomes. ANNs are widely used for pattern recognition, classification, and prediction tasks, making them applicable in stock market analysis for capturing complex relationships in financial data.

**Hidden Markov Model (HMM):**

A Hidden Markov Model is a statistical model that represents a system with hidden states, each associated with observable outcomes. The "hidden" states are not directly observable but influence the observed data. In the context of stock market analysis, HMM can be applied to model underlying states (e.g., market conditions) affecting observable data (e.g., stock prices). The model assumes a Markovian property, where the future state depends only on the present state, making it useful for capturing temporal dependencies. HMMs have been employed in finance for tasks such as market regime identification and predicting trends based on hidden states.

## Linear Regression:

To understand the working functionality of Linear Regression, imagine how you would arrange random logs of wood in increasing order of their weight. There is a catch; however – you cannot weigh each log. You have to guess its weight just by looking at the height and girth of the log (visual analysis) and arranging them using a combination of these visible parameters. This is what linear regression in machine learning is like.

## Logistic Regression:

Logistic Regression is used to estimate discrete values (usually binary values like 0/1) from a set of independent variables. It helps predict the probability of an event by fitting data to a logit function. It is also called logit regression.

## Decision Tree:

Decision Tree algorithm in machine learning is one of the most popular algorithm in use today; this is a supervised learning algorithm that is used for classifying problems. It works well in classifying both categorical and continuous dependent variables. This algorithm divides the population into two or more homogeneous sets based on the most significant attributes/ independent variables.

## SVM (Support Vector Machine) Algorithm:

SVM algorithm is a method of a classification algorithm in which you plot raw data as points in an n-dimensional space (where n is the number of features you have). The value of each featureis then tied to a particular coordinate, making it easy to classify the data. Lines called classifiers

can be used to split the data and plot them on a graph.

## Naive Bayes Algorithm:

A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features are related to each other, a Naive Bayes classifier would consider all of these properties independently when calculating the probability of a particular outcome. A Naive Bayesian model is easy to build and useful for massive datasets. It's simple and is known to outperform even highly sophisticated classification methods.

## KNN (K- Nearest Neighbors) Algorithm:

This algorithm can be applied to both classification and regression problems. Apparently, within the Data Science industry, it's more widely used to solve classification problems. It’s a simple algorithm that stores all available cases and classifies any new cases by taking a majority vote of its k neighbors. The case is then assigned to the class with which it has the most in common. A distance function performs this measurement.

## K-Means Algorithm:

It is an unsupervised learning algorithm that solves clustering problems. Data sets are classified into a particular number of clusters (let's call that number K) in such a way that all the data points within a cluster are homogenous and heterogeneous from the data in other clusters.

## Random Forest Algorithm:

A collective of decision trees is called a Random Forest. To classify a new object based on its attributes, each tree is classified, and the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

## DEEP LEARNING ALGORITHMS Convolutional Neural Networks (CNNs):

CNNs process the data by passing it through multiple layers and extracting features to exhibit convolutional operations. The Convolutional Layer consists of Rectified Linear Unit (ReLU) that outlasts to rectify the feature map. The Pooling layer is used to rectify these feature maps into the next feed. Pooling is generally a sampling algorithm that is down-sampled and it reduces the dimensions of the feature map. Later, the result generated consists of 2-D arrays consisting of single, long, continuous, and linear vector flattened in the map. The next layer i.e., called Fully Connected Layer which forms the flattened matrix or 2-D array fetched from the Pooling Layer as input and identifies the image by classifying it.

## Long Short Term Memory Networks (LSTMs)

LSTMs can be defined as Recurrent Neural Networks (RNN) that are programmed to learn and adapt for dependencies for the long term. It can memorize and recall past data for a greater period and by default, it is its sole behavior. LSTMs are designed to retain over time and henceforth they are majorly used in time series predictions because they can restrain memory or previous inputs. This analogy comes from their chain-like structure consisting of four interacting layers that communicate with each other differently. Besides applications of time series prediction, they can be used to construct speech recognizers, development in pharmaceuticals, and composition of music loops as well.

## SOURCE CODE

from django.db.models import Count, Avg from django.shortcuts import render, redirect from django.db.models import Count

from django.db.models import Q import datetime

# Create your views here.

from Remote\_User.models import ClientRegister\_Model,stock\_market\_model,predicting\_stock\_markettrends\_model

def serviceproviderlogin(request): if request.method == "POST":

admin = request.POST.get('username') password = request.POST.get('password')

if admin == "SProvider" and password =="SProvider": stock\_market\_model.objects.all().delete() predicting\_stock\_markettrends\_model.objects.all().delete() return redirect('View\_Remote\_Users')

return render(request,'SProvider/serviceproviderlogin.html') def viewtreandingquestions(request,chart\_type):

dd = {} pos,neu,neg =0,0,0 poss=None

topic=predicting\_stock\_markettrends\_model.objects.values('ratings').annotate(dcount=Count('rati ngs')).order\_by('-dcount')

for t in topic: topics=t['ratings']

pos\_count=predicting\_stock\_markettrends\_model.objects.filter(topics=topics).values('names').an notate(topiccount=Count('ratings'))

poss=pos\_count

for pp in pos\_count: senti= pp['names']

if senti == 'positive': pos= pp['topiccount']

elif senti == 'negative': neg = pp['topiccount']

elif senti == 'nutral': neu = pp['topiccount']

dd[topics]=[pos,neg,neu] return

render(request,'SProvider/viewtreandingquestions.html',{'object':topic,'dd':dd,'chart\_type':chart\_t ype})

def Search\_StockMarket(request): # Search if request.method == "POST":

kword = request.POST.get('keyword') print(kword)

obj = stock\_market\_model.objects.all().filter(Company\_Name contains=kword)

obj1 = stock\_market\_model.objects.get(Company\_Name contains=kword) opening=int(obj1.Opening\_Price)

closing=int(obj1.Closing\_Price) trends=closing-opening

if(opening<closing): val='Profit'

Stock\_Market\_Trends='Uptrends' if(opening>closing):

val = 'prices drop' Stock\_Market\_Trends = 'downtrends'

if (opening == closing): val = 'Horizontal'

Stock\_Market\_Trends = 'HorizontalTrends'

return render(request, 'SProvider/Search\_StockMarket.html', {'objs': obj,'trends': trends,'val':

val,'Stock\_Market\_Trends': Stock\_Market\_Trends})

return render(request, 'SProvider/Search\_StockMarket.html') def View\_All\_StockMarket\_Prediction\_Details(request):

pl=0 pl1=0

obj1 = stock\_market\_model.objects.values( 'Company\_Name',

'Company\_Category', 'Opening\_Price', 'Date\_Of\_Opening', 'Closing\_Price', 'Date\_Of\_Closing', 'volume',

'Profit', 'prices\_drop',

'Stock\_Market\_Trends', 'Stock\_Exchange\_By')

predicting\_stock\_markettrends\_model.objects.all().delete() for t in obj1:

Company\_Name=t['Company\_Name'] Company\_Category=t['Company\_Category'] Opening\_Price=int(t['Opening\_Price']) Date\_Of\_Opening=t['Date\_Of\_Opening'] Closing\_Price=int(t['Closing\_Price']) Date\_Of\_Closing=t['Date\_Of\_Closing'] volume=t['volume']

Profit=t['Profit'] prices\_drop=t['prices\_drop']

Stock\_Market\_Trends=t['Stock\_Market\_Trends'] Stock\_Exchange\_By=t['Stock\_Exchange\_By']

Total =Closing\_Price-Opening\_Price

if (Opening\_Price < Closing\_Price): val = 'Profit'

Stock\_Market\_Trends = 'Up trends' finalstr=val+':'+Stock\_Market\_Trends

if (Opening\_Price > Closing\_Price): val = 'prices drop'

Stock\_Market\_Trends = 'down trends'

finalstr = val + ':' + Stock\_Market\_Trends if (Opening\_Price == Closing\_Price):

val = 'Horizontal'

Stock\_Market\_Trends = 'Horizontal Trends' finalstr = val + ':' + Stock\_Market\_Trends

if (Total > 0): pl = Total

predicting\_stock\_markettrends\_model.objects.create(names=Company\_Name, Company\_Category=Company\_Category,

Opening\_Price=Opening\_Price, Date\_Of\_Opening=Date\_Of\_Opening, Closing\_Price=Closing\_Price, Date\_Of\_Closing=Date\_Of\_Closing, volume=volume, Profit=pl, prices\_drop=0, Stock\_Market\_Trends=finalstr, Stock\_Exchange\_By=Stock\_Exchange\_By)

if (Total < 0): pl1 = Total

predicting\_stock\_markettrends\_model.objects.create(names=Company\_Name, Company\_Category=Company\_Category,

Opening\_Price=Opening\_Price, Date\_Of\_Opening=Date\_Of\_Opening,

Closing\_Price=Closing\_Price, Date\_Of\_Closing=Date\_Of\_Closing,

volume=volume, Profit=0, prices\_drop=pl1, Stock\_Market\_Trends=finalstr, Stock\_Exchange\_By=Stock\_Exchange\_By)

if (Total == 0): pl1 = Total

predicting\_stock\_markettrends\_model.objects.create(names=Company\_Name, Company\_Category=Company\_Category,

Opening\_Price=Opening\_Price, Date\_Of\_Opening=Date\_Of\_Opening,

Closing\_Price=Closing\_Price, Date\_Of\_Closing=Date\_Of\_Closing,

volume=volume, Profit=0, prices\_drop=0, Stock\_Market\_Trends=finalstr, Stock\_Exchange\_By=Stock\_Exchange\_By)

obj = predicting\_stock\_markettrends\_model.objects.all()

return render(request, 'SProvider/View\_All\_StockMarket\_Prediction\_Details.html', {'objs': obj})

def View\_Remote\_Users(request): obj=ClientRegister\_Model.objects.all()

return render(request,'SProvider/View\_Remote\_Users.html',{'objects':obj})

def ViewTrendings(request): topic=

predicting\_stock\_markettrends\_model.objects.values('topics').annotate(dcount=Count('topics')). order\_by('-dcount')

return render(request,'SProvider/ViewTrendings.html',{'objects':topic})

def negativechart(request,chart\_type): dd = {}

pos, neu, neg = 0, 0, 0 poss = None

topic= predicting\_stock\_markettrends\_model.objects.values('ratings').annotate(dcount=Count('ratings'). order\_by('-dcount')

for t in topic:

topics = t['ratings'] pos\_count=

predicting\_stock\_markettrends\_model.objects.filter(topics=topics).values('names').annotate (topiccount=Count('ratings'))

poss = pos\_count for pp in pos\_count:

senti = pp['names'] if senti == 'positive':

pos = pp['topiccount'] elif senti == 'negative':

neg = pp['topiccount'] elif senti == 'nutral':

neu = pp['topiccount'] dd[topics] = [pos, neg, neu]

return render(request,'SProvider/negativechart.html',{'object':topic,'dd':dd,'chart\_type':chart\_type})

def charts(request,chart\_type): chart1=predicting\_stock\_markettrends\_model.objects.values('names').annotate(dcount=Avg('Pro fit'))

return render(request,"SProvider/charts.html", {'form':chart1, 'chart\_type':chart\_type})

def charts1(request,chart\_type): chart1=predicting\_stock\_markettrends\_model.objects.values('names').annotate(dcount=Avg('pric es\_drop'))

return render(request,"SProvider/charts1.html", {'form':chart1, 'chart\_type':chart\_type})

def View\_StockMarket\_Details(request): obj =stock\_market\_model.objects.all()

return render(request, 'SProvider/View\_StockMarket\_Details.html', {'list\_objects': obj})

def likeschart(request,like\_chart): charts

=predicting\_stock\_markettrends\_model.objects.values('names').annotate(dcount=Avg('Profit')) return render(request,"SProvider/likeschart.html", {'form':charts, 'like\_chart':like\_chart})

def View\_StockMarketUpDown(request):

obj = predicting\_stock\_markettrends\_model.objects.all()

return render(request, 'SProvider/View\_StockMarketUpDown.html', {'objs': obj})

* 1. **SOFTWARE TESTING**

# TESTING

Software testing is the process of validating and verifying that a software application meets the technical requirements which are involved in its design and development. It is also used to uncoverany defects/bugs that exist in the application. It assures the quality of the software. There are manytypes of testing software viz., manual testing, unit testing, black box testing, performance testing,stress testing, regression testing, white box testing etc. Among these performance testing and loadtesting are the most important one for an android application and next sections deal with some ofthese types.

## TYPES OF TESTS

### 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.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actuallyrun as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. 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 exposing the problems that arise from the combination of components.

### Functional test

Functional tests provide systematic demonstrations that functions tested are available as specifiedby 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/Procedures : 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 testsis determined.

### System Test

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 andflows, emphasizing pre-driven process links and integration points.

### 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 totest 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, structureor 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 withoutconsidering how the software works.

### Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinctphases.

## SYSTEM TESTING

### TESTING METHODOLOGIES

The following are the Testing Methodologies:

### Unit Testing.

* **Integration Testing.**

### User Acceptance Testing.

* **Output Testing.**
* **Validation Testing.**

## Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

## Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

### The following are the types of Integration Testing:

1. **Top Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

### Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

* The low-level modules are combined into clusters into clusters that perform a specific Software

sub-function.

* A driver (i.e.) the control program for testing is written to coordinate test case input and output
* The cluster is tested.
* Drivers are removed and clusters are combined moving upward in the program structure
* The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

## User Acceptance Testing:

User Acceptance of a system is the key factor for the success of any system. 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 Checking:

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 alphanumeric 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 message. 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.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

## Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

## Using Live Test Data

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

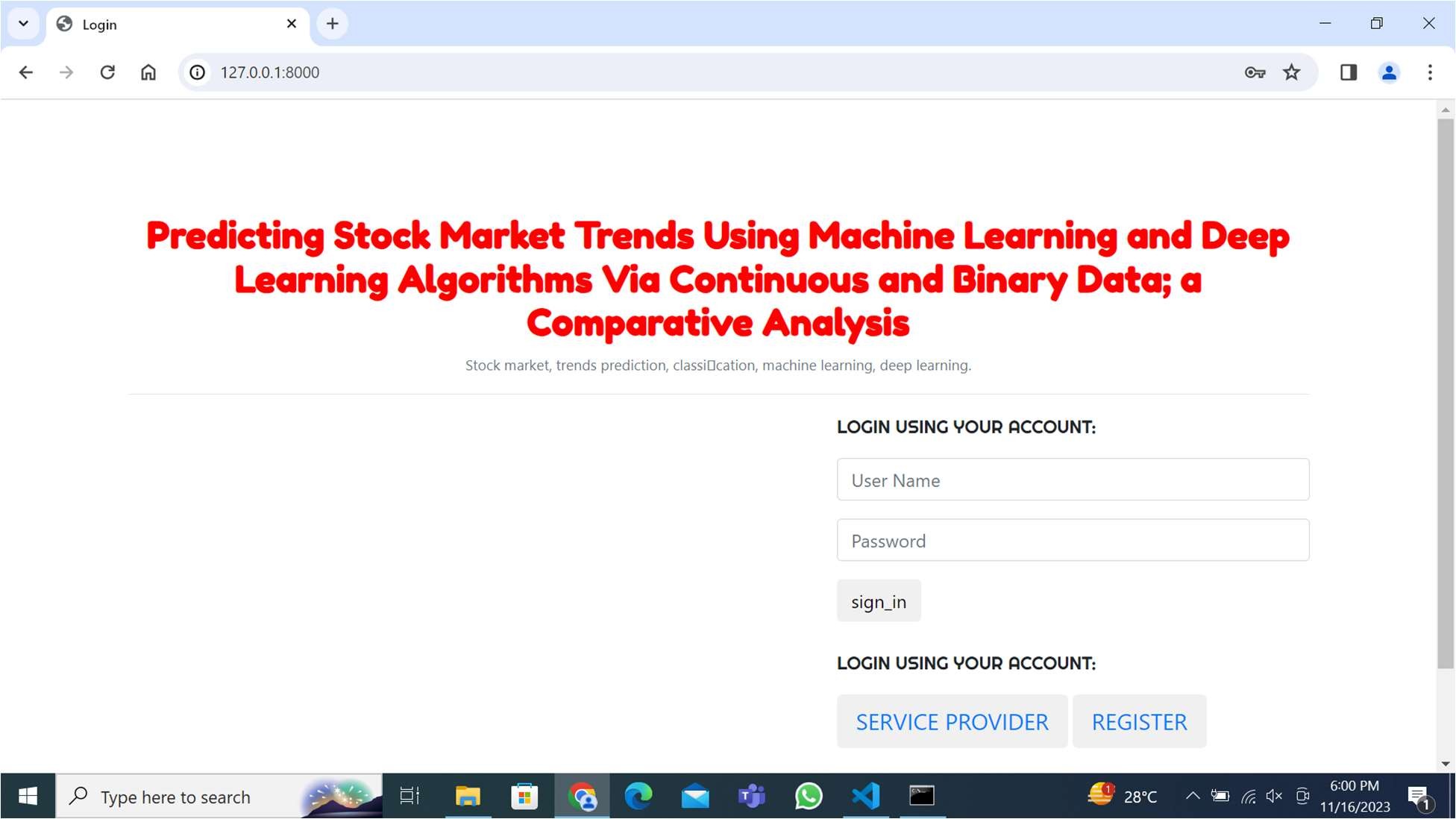
## Using Artificial Test Data

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

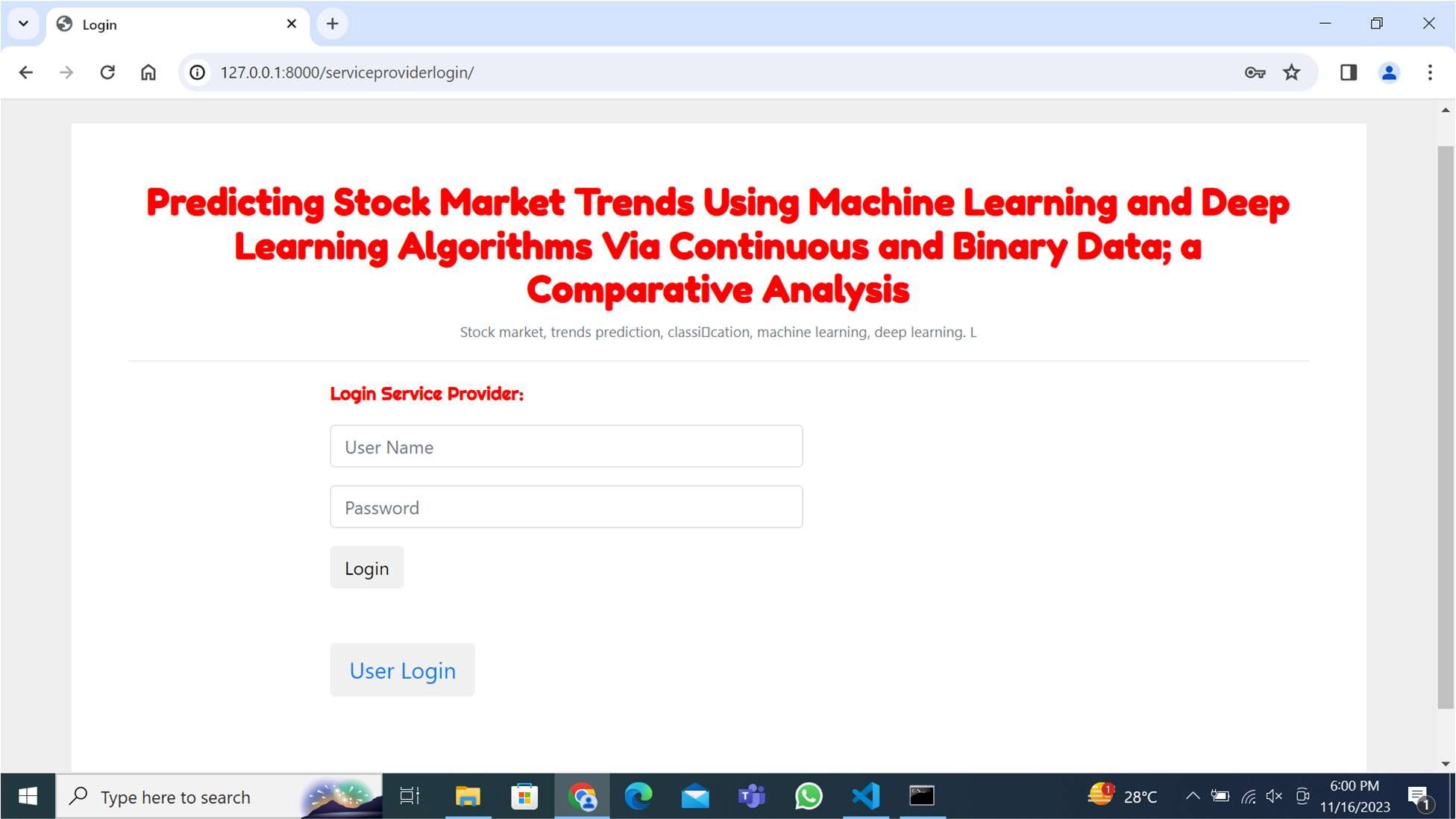
The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

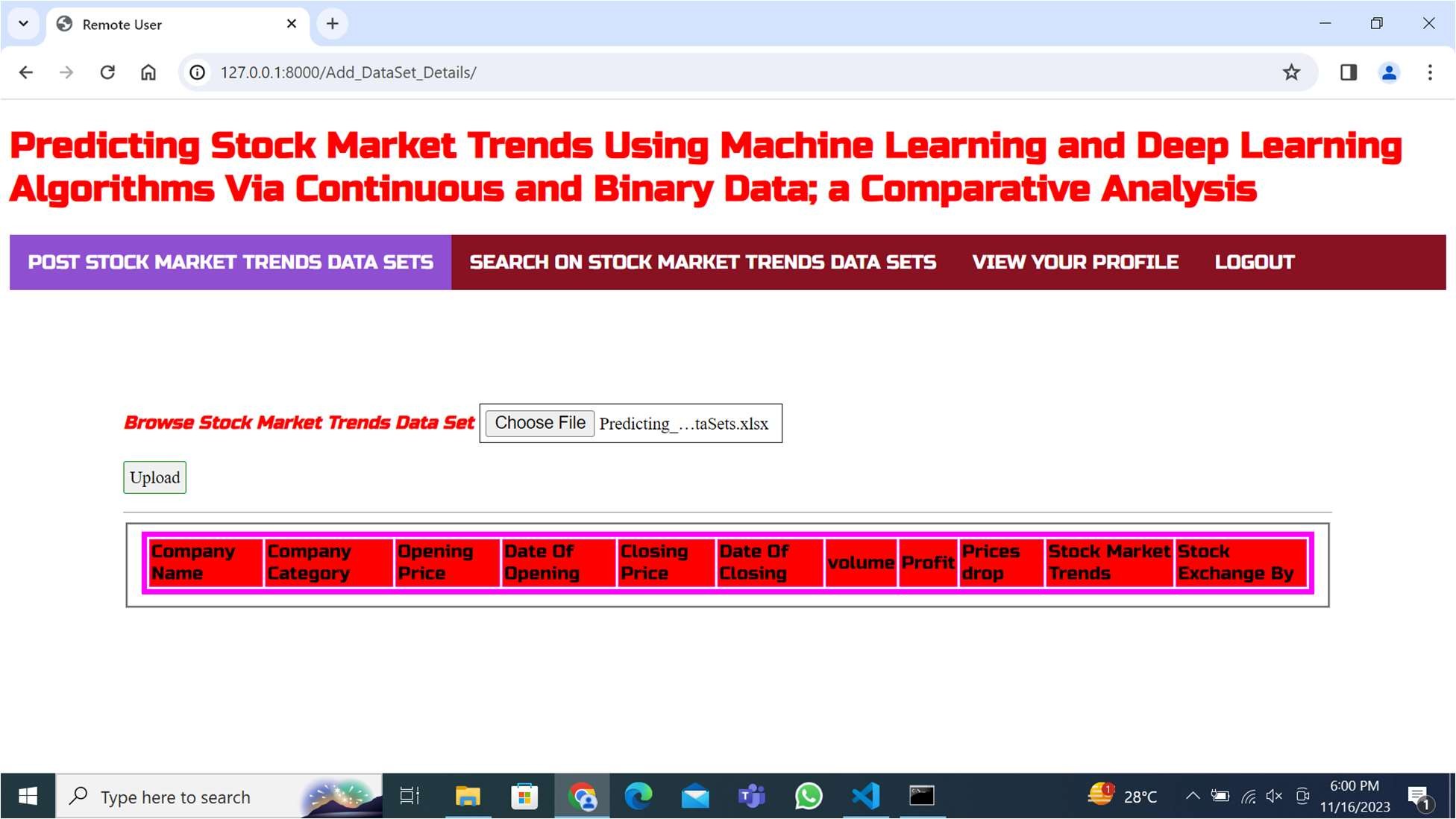
# RESULTS



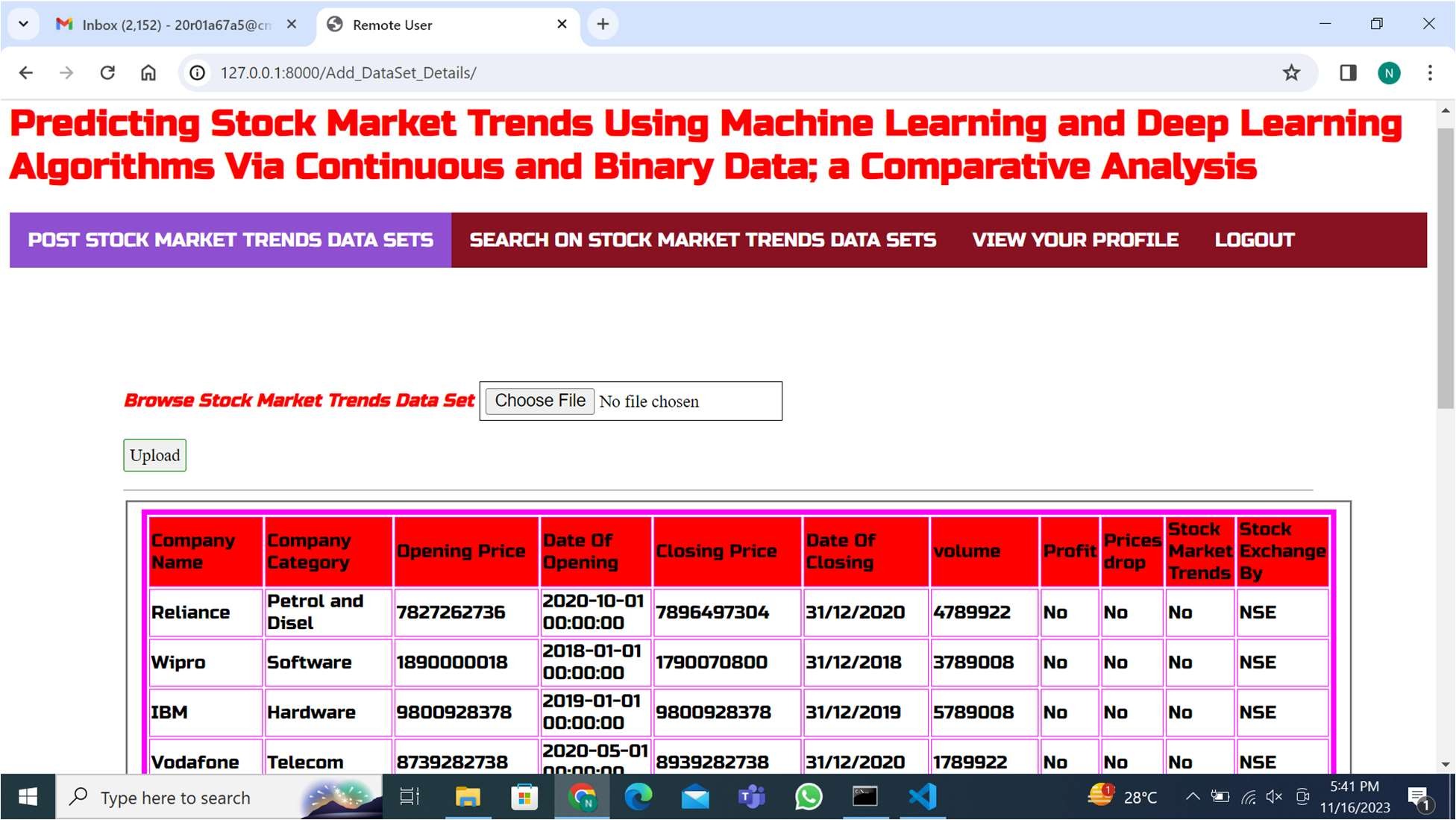
**Figure.6.1 User Login Page**



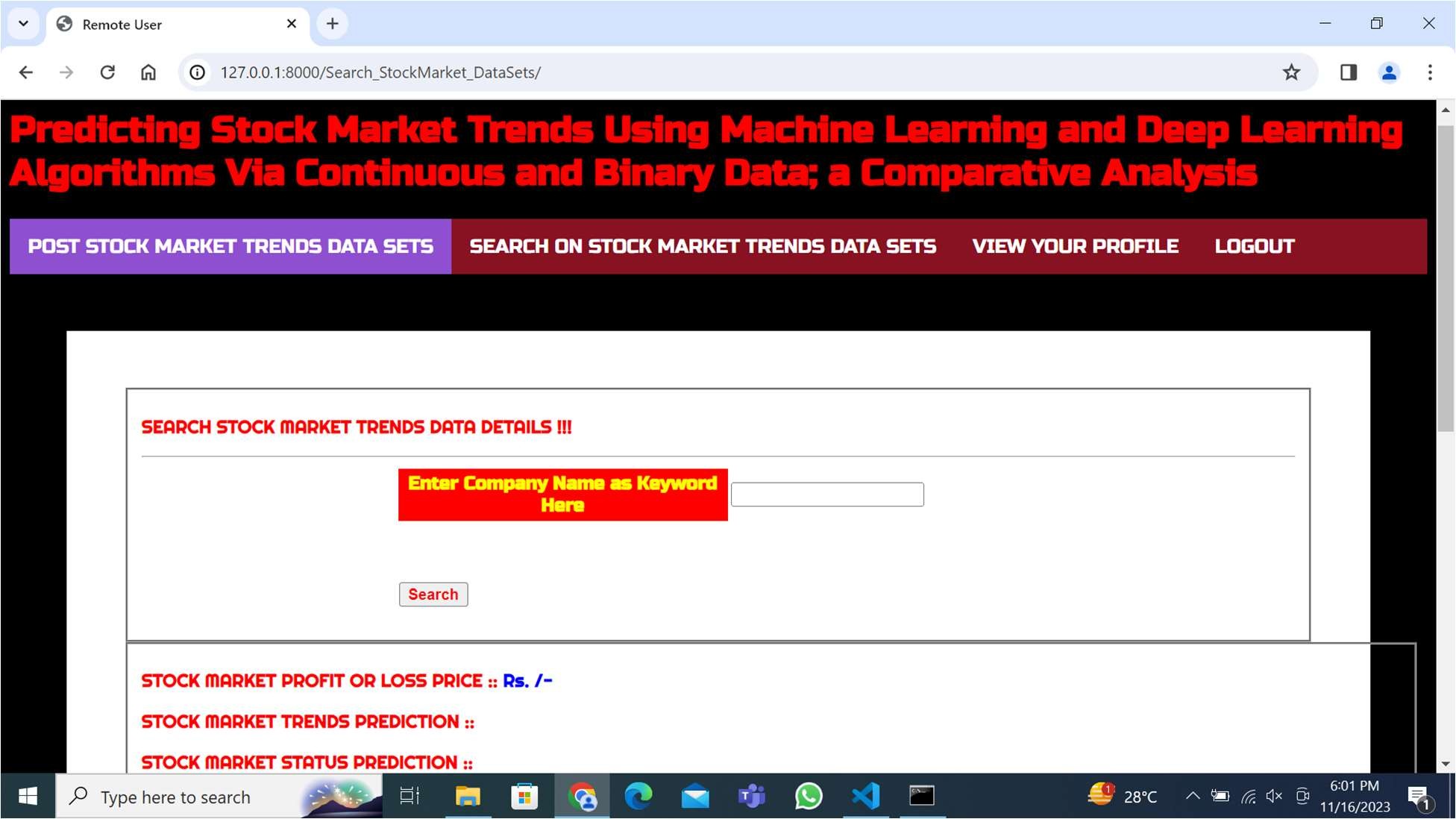
**Figure.6.2. Service Provider Login Page**



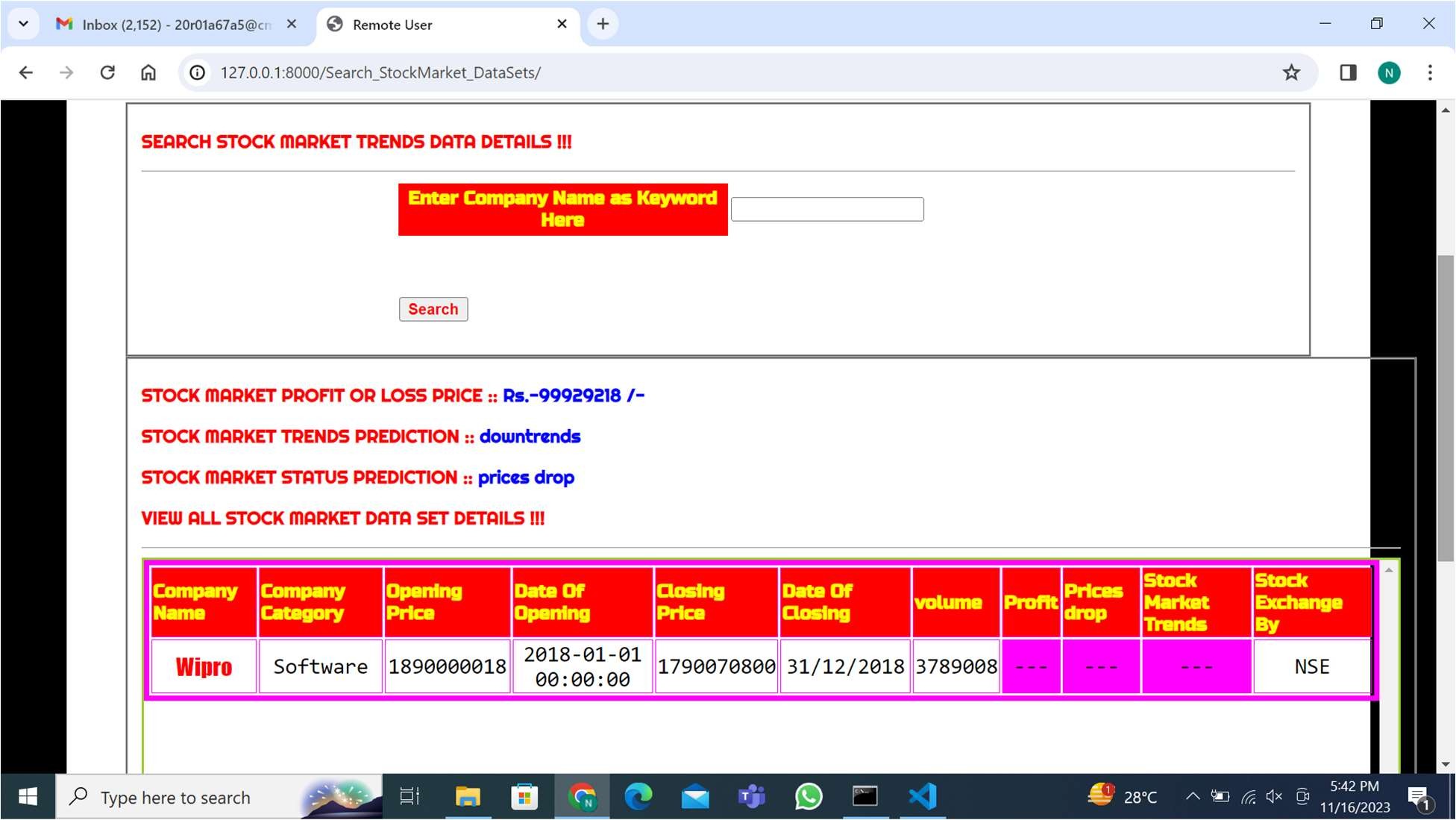
**Figure.6.3. Uploading Dataset**



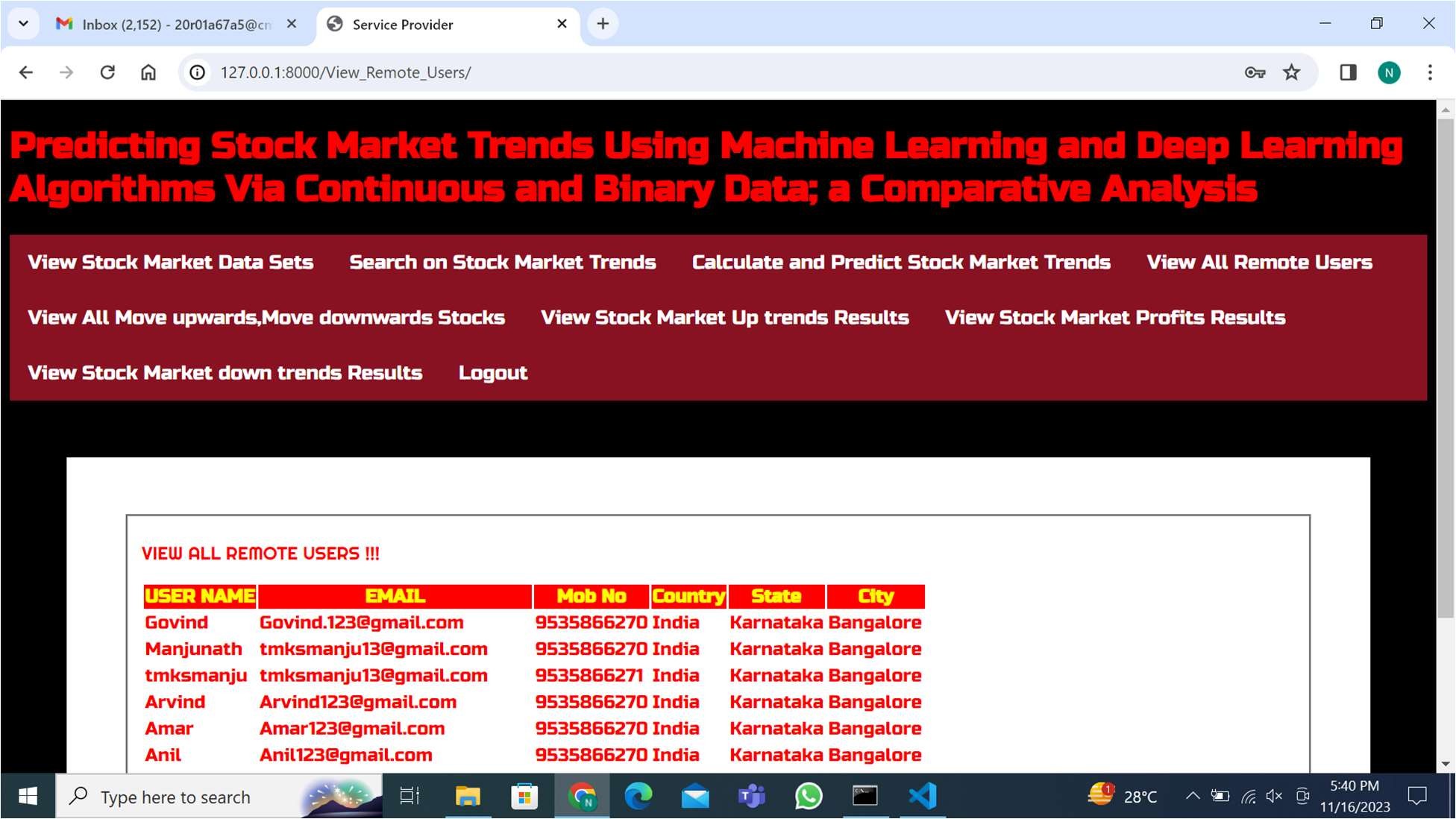
**Figure.6.4. Uploaded Dataset**



**Figure.6.5. Searching for Company Stock Market Trend**



**Figure.6.6. Output**



**Figure.6.7. All Remote Users**



**Figure.6.8. Dataset**

# CONCLUSION

The study aimed to predict stock market movements using machine learning and deep learning algorithms. To achieve this, four distinct stock market groups—diversified financials, petroleum, non-metallic minerals, and basic metals—listed on the Tehran Stock Exchange were selected. The dataset utilized for analysis spanned a decade of historical records and included ten technical features for each group.

In order to assess the predictive capabilities, a total of nine machine learning models (Decision Tree, Random Forest, Ada boost, XG Boost, SVC, Naïve Bayes, KNN, Logistic Regression, and ANN) and two deep learning methods (RNN and LSTM) were employed as predictors. The study considered two different approaches for input values—continuous data and binary data. Evaluation of model performance was conducted using three classification metrics. Notably, the experimental results revealed a significant enhancement in model performance when utilizing binary data instead of continuous data. Interestingly, both deep learning algorithms (RNN and LSTM) emerged as superior models in both input approaches.

The selection of stock market groups from diverse sectors and the utilization of a comprehensive dataset spanning ten years contribute to the robustness of the study. The inclusion of a wide range of machine learning and deep learning models, along with the comparison of input approaches, adds depth to the analysis, revealing the superiority of binary data for enhanced predictive performance. The findings underscore the efficacy of deep learning algorithms, specifically RNN and LSTM, in predicting stock market movements, positioning them as promising tools for future financial forecasting endeavors.

# REFERENCES

[1]. Murphy, Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. Penguin, 1999.

1. T. Turner, A Beginner's Guide To Day Trading Online, 2nd ed. New York, NY, USA: Simon and Schuster, 2007.
2. H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil,M. M. Selim, and K. Muhammad, ``A local and global event sentimentbased efcient stock exchange forecasting using deep learning,'' Int. J. Inf.Manage., vol. 50, pp. 432451, Feb. 2020.
3. W. Long, Z. Lu, and L. Cui, ``Deep learning-based feature engineering for stock price movement prediction,'' Knowl.-Based Syst., vol. 164,pp. 163173, Jan. 2019.
4. J. B. Duarte Duarte, L. H. Talero Sarmiento, and K. J. Sierra Juárez,``Evaluation of the effect of investor psychology on an artical stock market through its degree of efficiency,'' Contaduria y Administration, vol. 62, no. 4, pp. 13611376, Oct. 2017.
5. Lu, Ning, A Machine Learning Approach to Automated Trading. Boston, MA, USA: Boston College Computer Science Senior, 2016.
6. M. R. Hassan, B. Nath, and M. Kirley, ``A fusion model of HMM, ANNand GA for stock market forecasting,'' Expert Syst. Appl., vol. 33, no. 1,pp. 171180, Jul. 2007.
7. W. Huang, Y. Nakamori, and S.-Y. Wang, ``Forecasting stock market movement direction with support vector machine,'' Compute. Oper. Res. vol. 32, no. 10, pp. 25132522, Oct. 2005.
8. J. Sun and H. Li, ``Financial distress prediction using support vector machines: Ensemble vs. Individual,'' Appl. Soft Compute., vol. 12, no. 8, pp. 22542265, Aug. 2012.
9. P. Ou and H. Wang, ``Prediction of stock market index movement by ten data mining techniques,'' Modern Appl. Sci., vol. 3, no. 12, pp. 2842, Nov. 2009.
10. F. Liu and J.Wang, ``Fluctuation prediction of stock market index by legend neural network with random time strength function,'' Neuro computing, vol. 83, pp. 1221, Apr. 2012.
11. C.-F. Tsai, Y.-C. Lin, D. C. Yen, and Y.-M. Chen, ``Predicting stock returns by classier ensembles,'' Appl. Soft Compute., vol. 11, no. 2, pp. 24522459, Mar. 2011.
12. R. D. A. Arajo and T. A. E. Ferreira, ``A Morphological-Rank-Linear evolutionary method for stock market prediction,'' Inf. Sci., vol. 237, pp. 317, Jul. 2013.
13. M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, ``Evaluating multiple classifiers for stock price direction prediction,'' Expert Syst. Appl.