# 1. Project or Company Profile

**Project Profile**

|  |  |
| --- | --- |
| **Project Title** | Stock Price Prediction using Machine Learning |
| **Project category** | Web Application |
| **Objective** | To Predict the stock price based on the past and upcoming events. |
| **Front End** | Django |
| **Back End** | Django, Python |
| **Tool** | Anaconda, Google Colab |
| **Server** | XAMP |
| **Documentation Tool** | Office 365 |
| **Company Name** | MADHDA BUSINESS SOLUTIONS PVT LTD |
| **Internal Guide** | Dr. Jigneshkumar A. Chauhan |
| **External Guide** |  |
| **Developed By** | Preyash Sanjay KaPatel (1803421003) |
| **Group No** | 19 |

**About the Company**

Company provides a complete solution with high performance with helping hand from startup level to the enterprise level.

We provide Microsoft Dynamics ERP solutions to our clients worldwide. We work closely to provide consulting, maintenance, Implementation support, Upgrades, Re-architect to our client. To make your company and products smarter are more successful.

* We provide custom solution to our client.
* An employee is the most valuable asset for an organization.
* We design, implement and manage your Microsoft Dynamic ERP.
* We utilize our expertise in providing sustainable enterprise solutions to customers and help their businesses maximize potential.
* For us every business is built on good relation with client.
* The leaders at Madhda about their employees’ well-being, happiness and success.
* We are driven by a set of guiding principles which dictate our decisions and responsibilities towards our people and customers.
* To provide businesses with the ERP software and knowledge they need to succeed in a competitive market.
* We conduct business ethically.
* We embrace positive change, innovation, and continuous improvement.

In return our employees feel recognised and appreciated for their contributions to the company.

# 2. Functional Requirement Specification

# 2.1 Module Specification

1. Login:
   1. Admin and Account Holder can access website by successfully login
2. Watchlist:
   1. User can add remove stock from watchlist.
3. Holdings:
   1. Here user can see there stock holding if they have.
4. Predictions:
   1. In this module user can predict the stock price.

# User Specification

1. Prediction:
   1. There should be option to predict the share price.
2. Recording of stock for Study:
   1. Learner should be able to add get latest price.
3. Buy and Sell:
   1. Learner should be able to buy and sell stock at any price for study purpose.

# 3.1 About Existing System

The existing system works as follow:

* Money related transaction require high alertness of statistical insights of history and future events, in such case taking decision of stake sale, hold or buy are difficult.
* Before taking decision, we need to look at the past data, stock patterns, Recent news and judging the price takes time and it might end up in slow decision, incomplete information etc.
* Taking the Stake sale/buy/Hold based on emotion and incomplete information may perform false prediction.

# 3.2 Need for new system

1. Rapid Decision:
   1. User can take decision rapidly as it is performed autonomous.
2. Improved Accuracy:
   1. User can use the result to take decision for stake sale or hold or buy
3. Based on Historic data:
   1. Prediction are based on historic data and past events.

# 4. Technical Requirement Specification

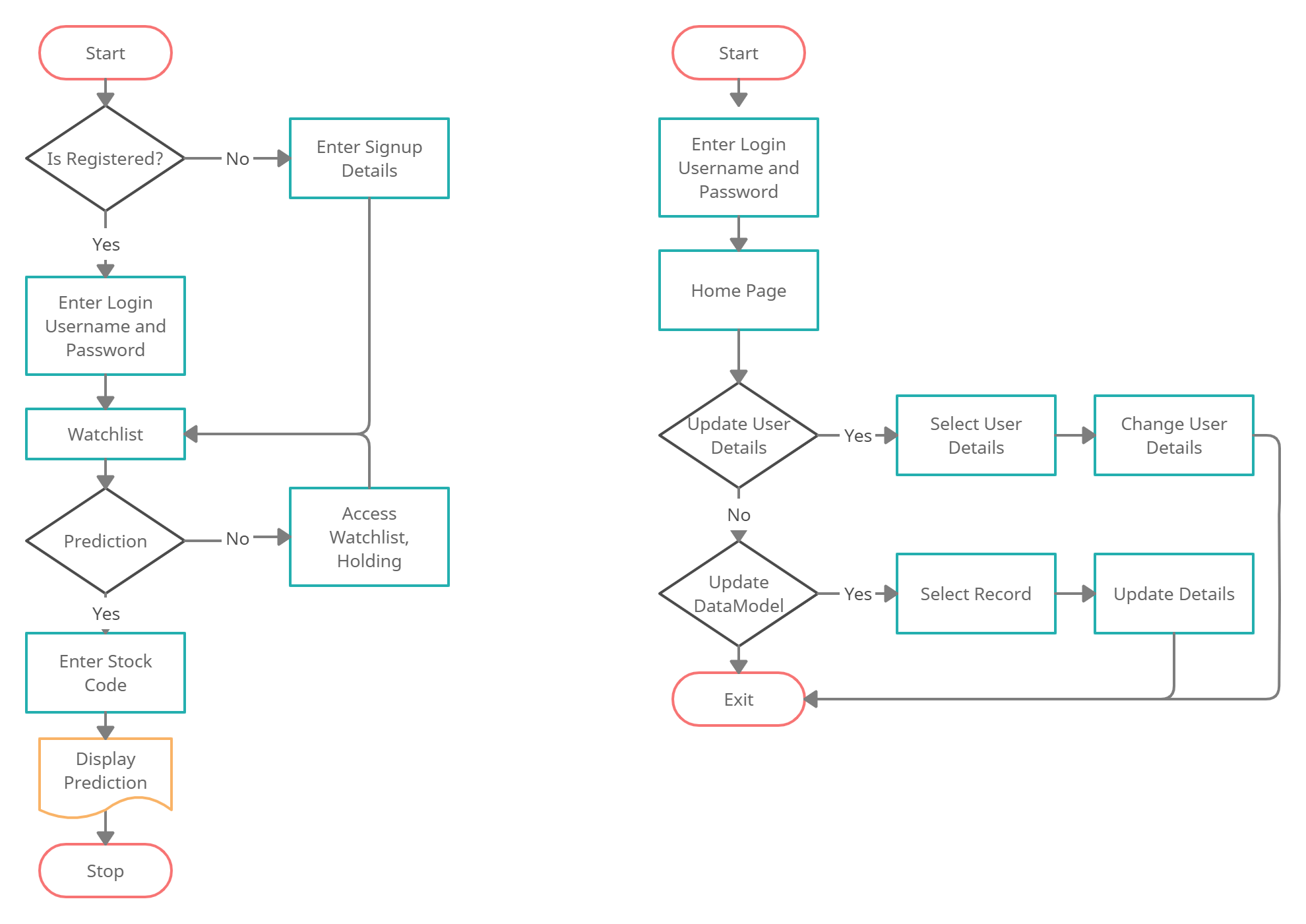
# 4.1 Hardware Requirement

* Client Side:
  + Hardware Requirement:
  + Basic CPU with 700 MHz Speed
  + 1 GB RAM
* Server Side:
  + Inter i3 10th generation
  + GB RAM
  + 250 mb Space in SSD

# 4.2 Software Requirement

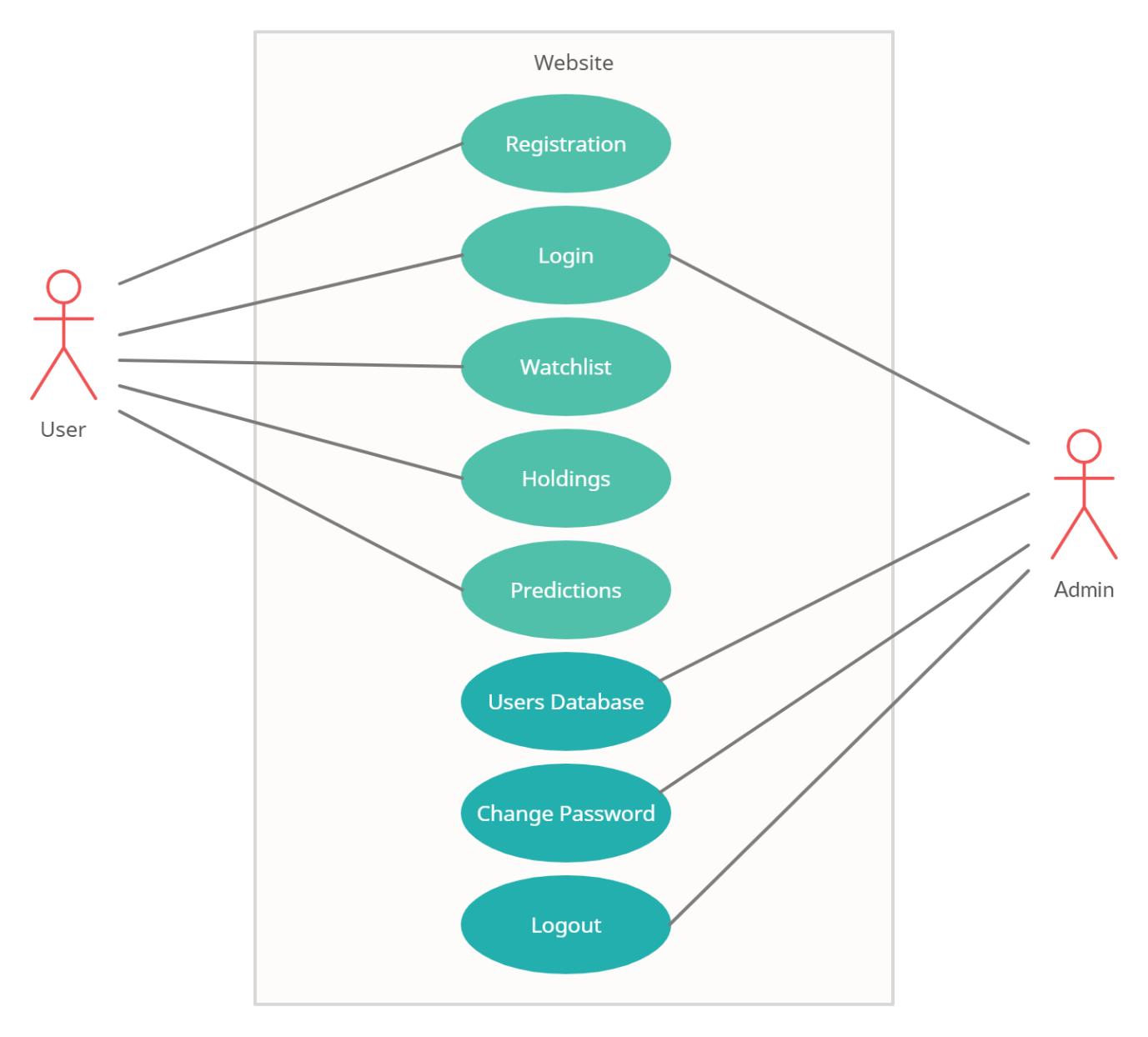
* Client Side:
  + Chrome with 68.0.3440.75 or above Version
  + Good Internet Speed
* Server Side:
  + Chrome with 68.0.3440.75 or above Version
  + High Speed Internet
  + Google Colab
  + Webpage IDE
  + Github Desktop

# 5. System Flow Chart



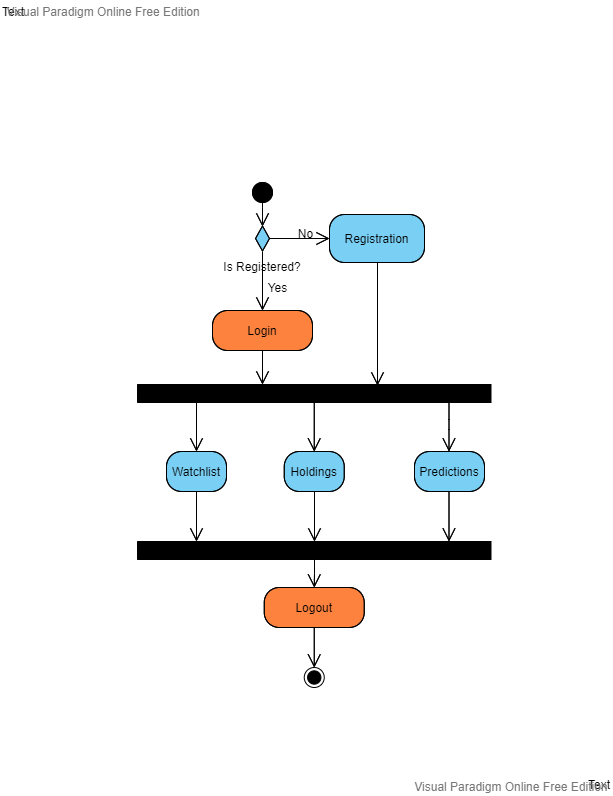
# 6. UML Diagrams

# 6.1 Use-case Diagram

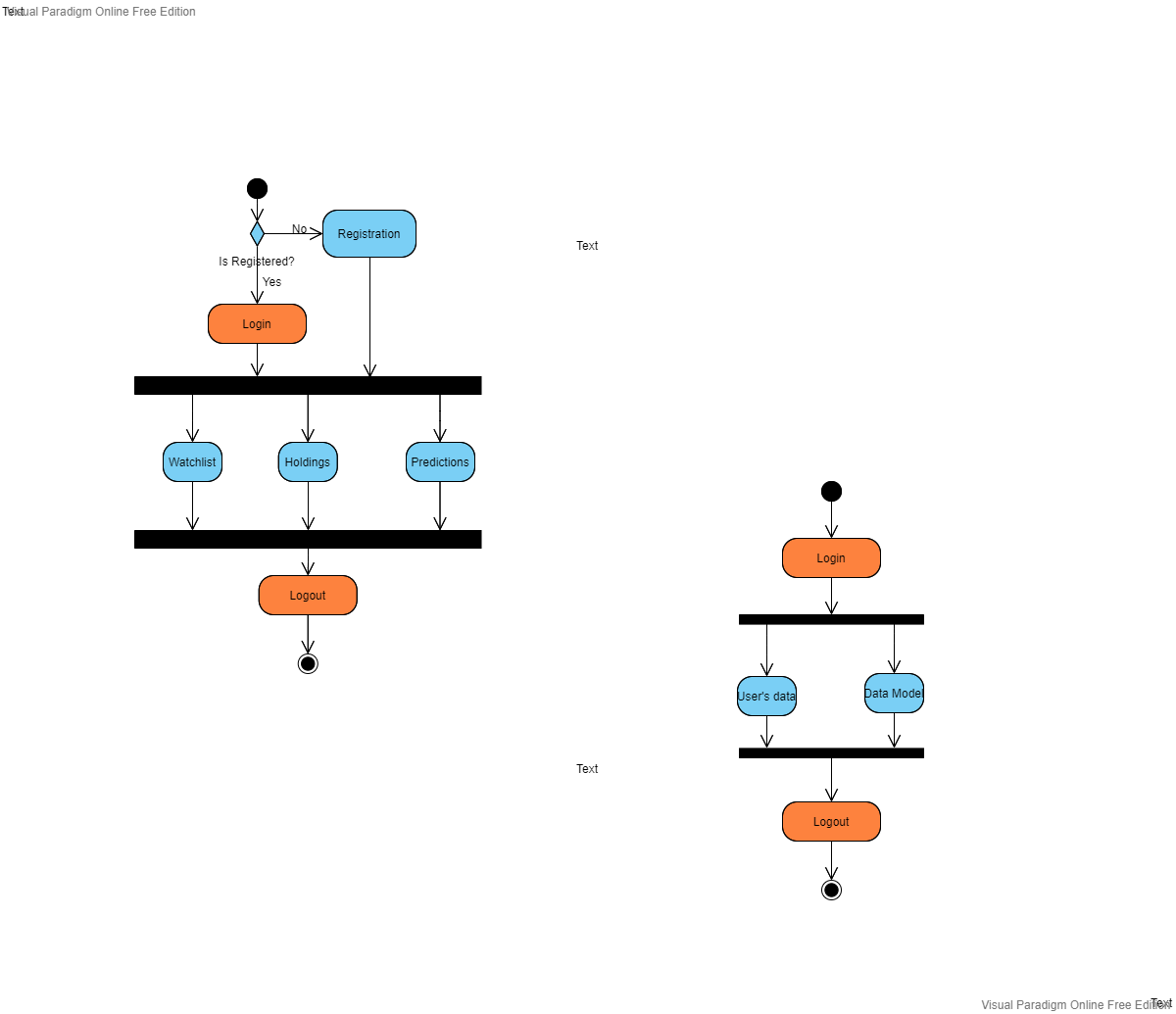
****

# 6.2 Activity Diagram

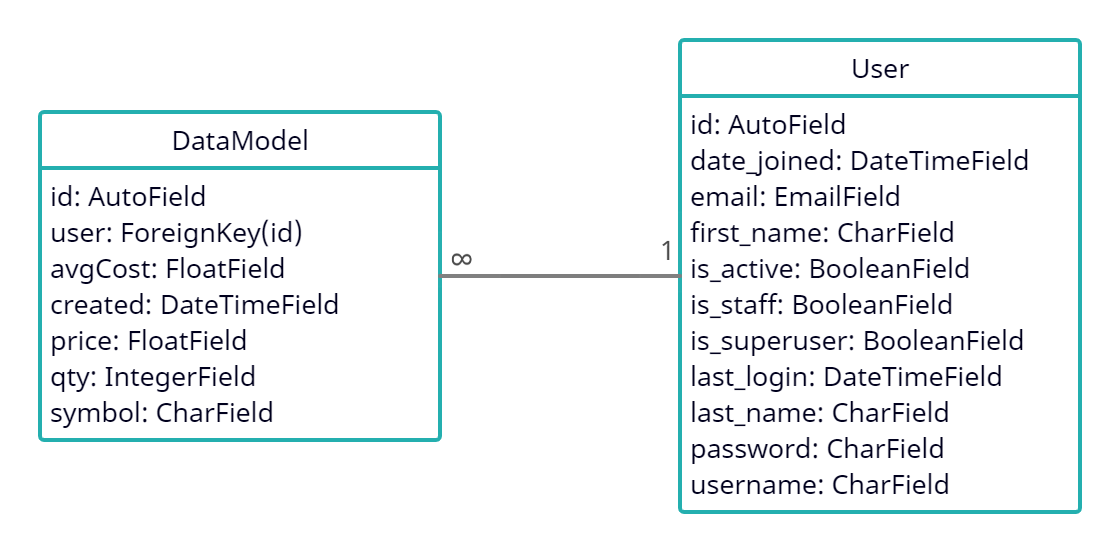
**Activity Diagram for User**

****

**Activity Diagram for Admin**

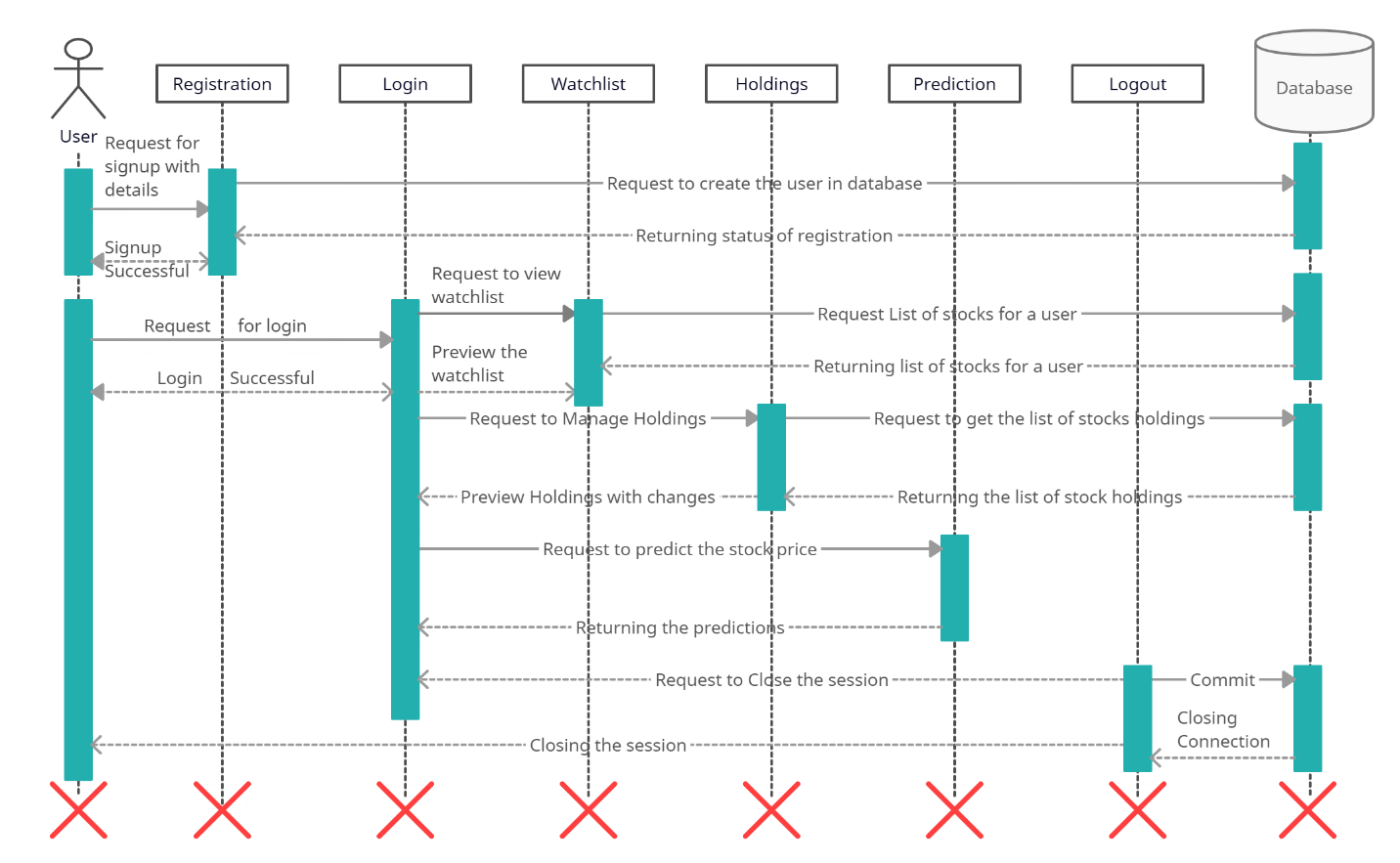
****

# 6.3 Class Diagram

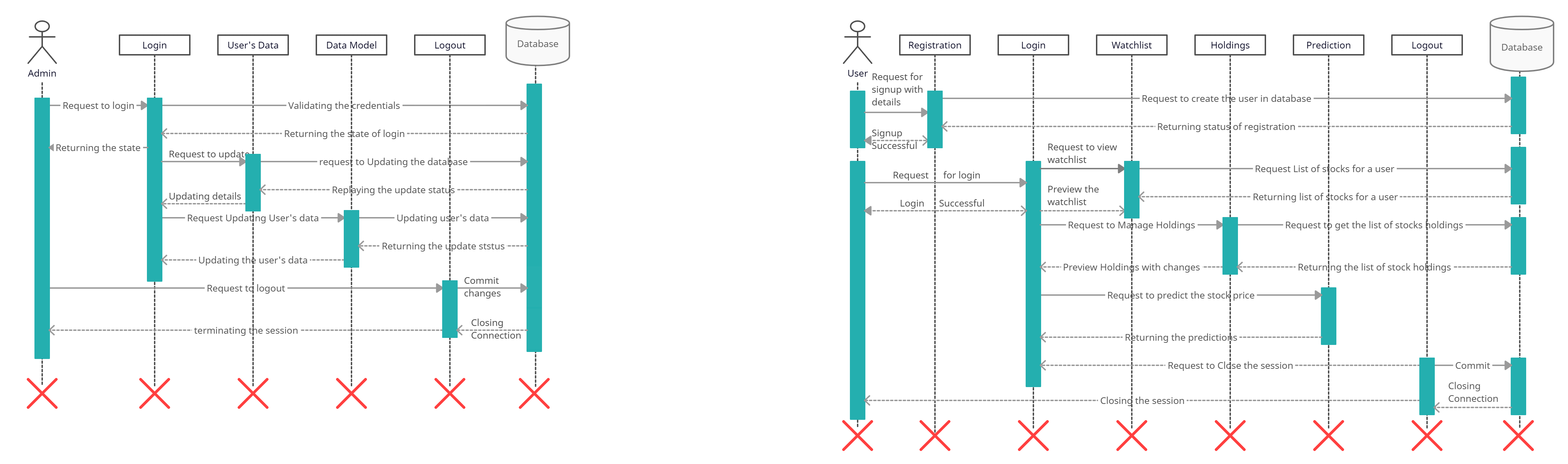
****

# 6.4 Sequence Diagram

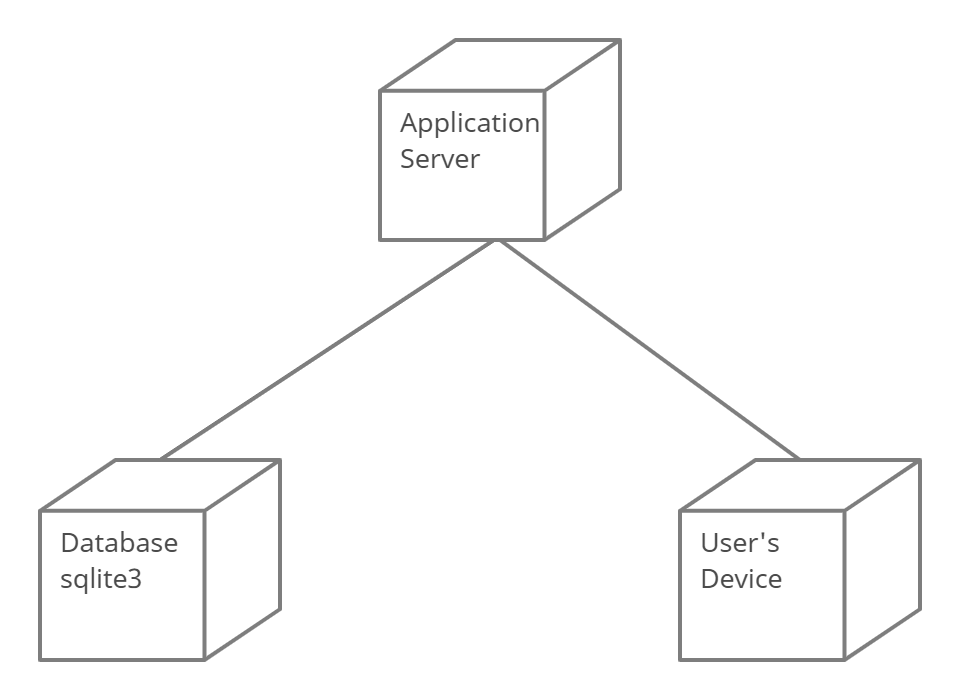
**Sequence Diagram for User**

****

**Sequence Diagram for Admin**

****

# 6.5 Deployment Diagram



# 7. Data Dictionary

**Name: DataModel**

**DESCRIPTION:- Represents stock holding and watchlist Primary Key: id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | id | IntegerField | Primary Key | Represent Record id |
| 2 | user | ForeignKey | ForeignKey | Represent registered User |
| 3 | avgCost | FloatField | Not Null | Represent Average cose |
| 4 | created | DateTimeField | Not Null | Represent stock addition date and time |
| 5 | price | FloatField | Not Null | Represent last updated price |
| 6 | qty | IntegerField | Not Null | Represent quenty |
| 7 | symbol | CharField | Not Null | Represent stock symbol |

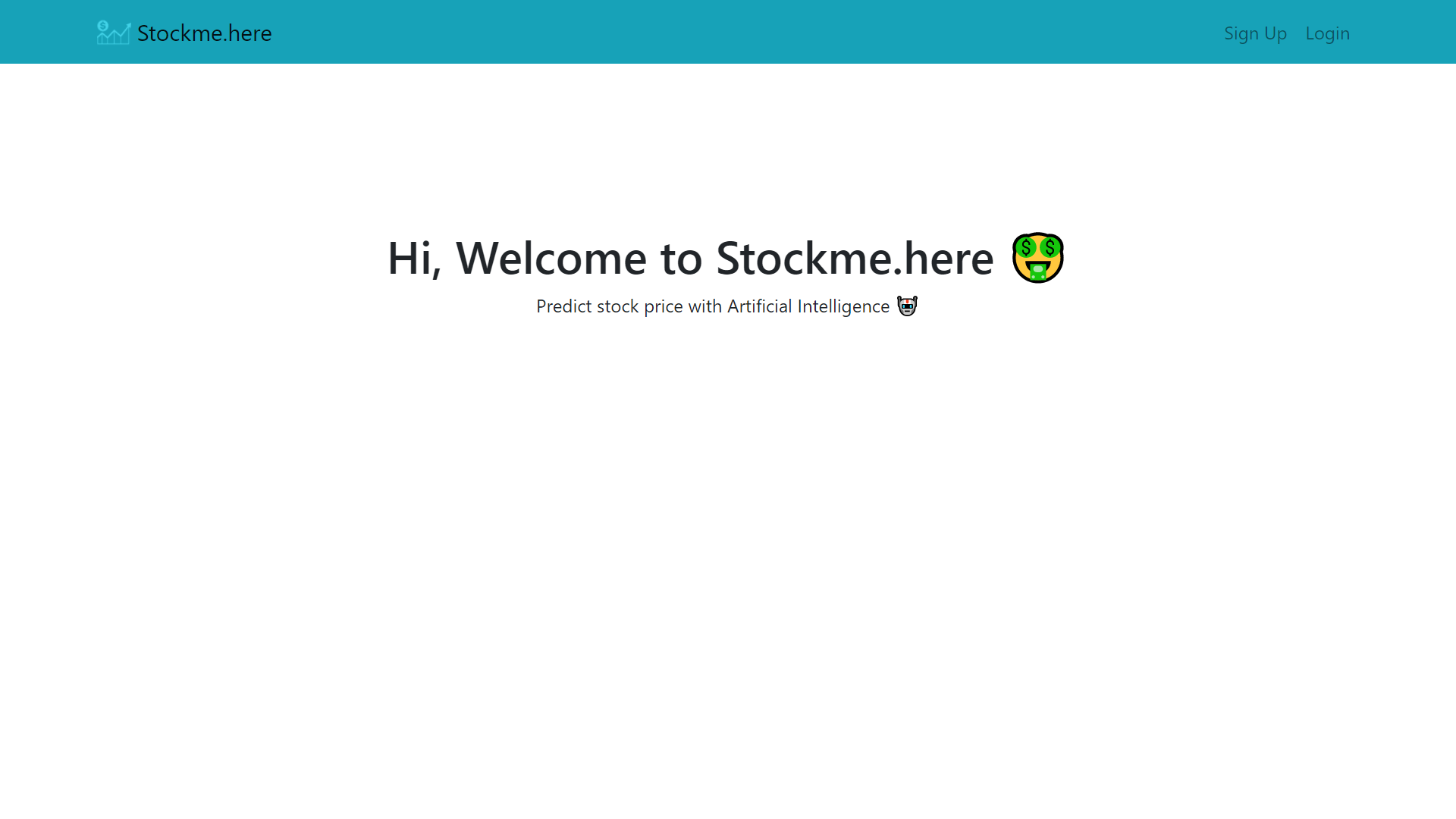
**Name: User**

**DESCRIPTION:- Represents users’s details Primary Key: id**

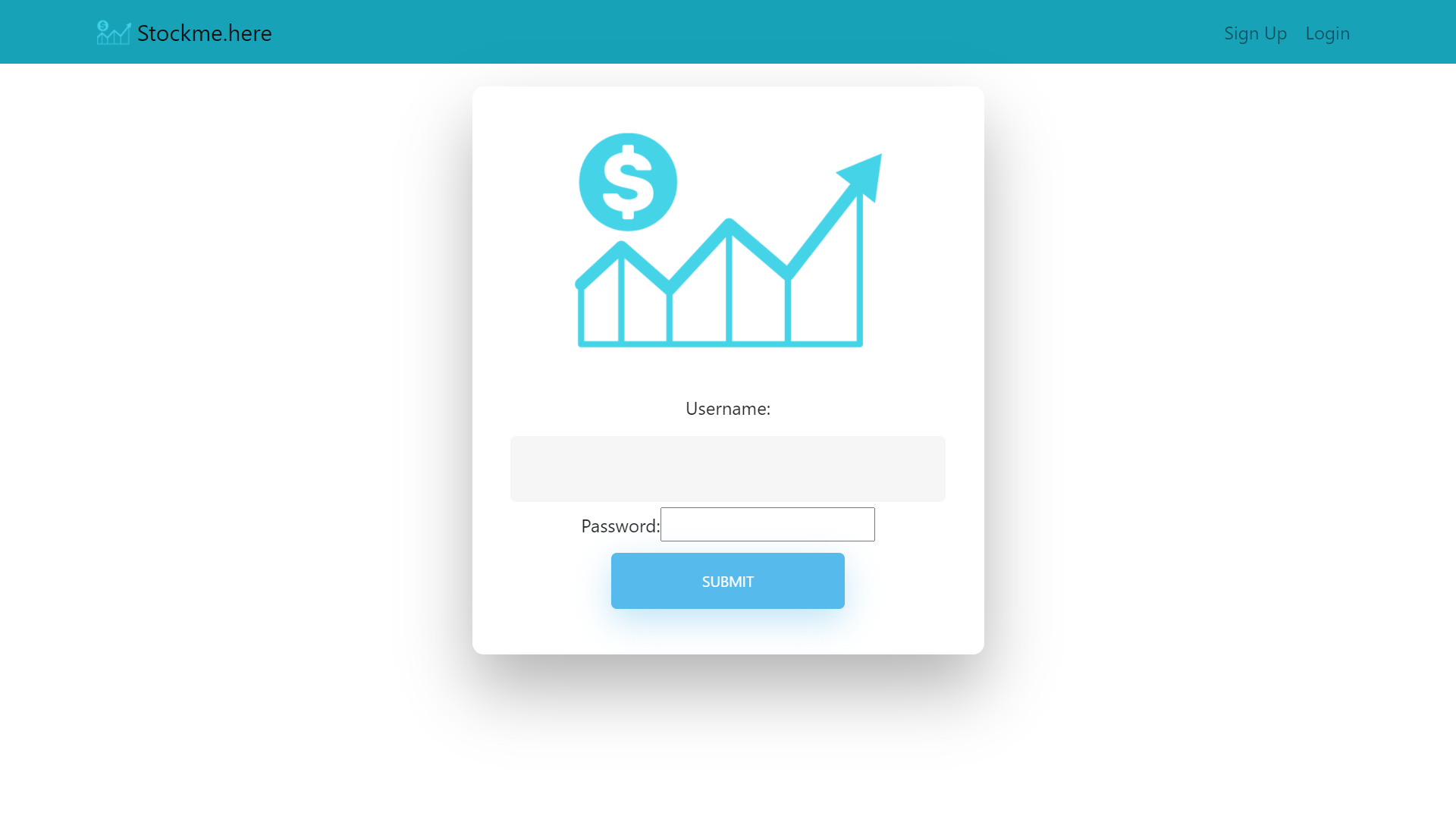
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | id | IntegerField | Primary Key | Represent user id |
| 2 | Date\_joined | DateTimeField | Not Null | Represent date of joining |
| 3 | email | EmailField | Not Null | Represent Email id of user |
| 4 | First\_name | CharField | Not Null | Represent First Name |
| 5 | Is\_active | BooleanField | Not Null | Represent user status |
| 6 | Is\_staff | BooleanField | Not Null | Represent user is staff member or not |
| 7 | Is\_superuser | BooleanField | Not Null | Represent user is super user or not |
| 8 | last\_login | DateTimeField | Not Null | Represent last login date and time |
| 9 | last\_name | CharField | Not Null | Represent user’s last name |
| 10 | Password | CharField | Not Null | Represent password |
| 11 | username | CharField | Not Null | Represent username |

# 8. Input & Output Design

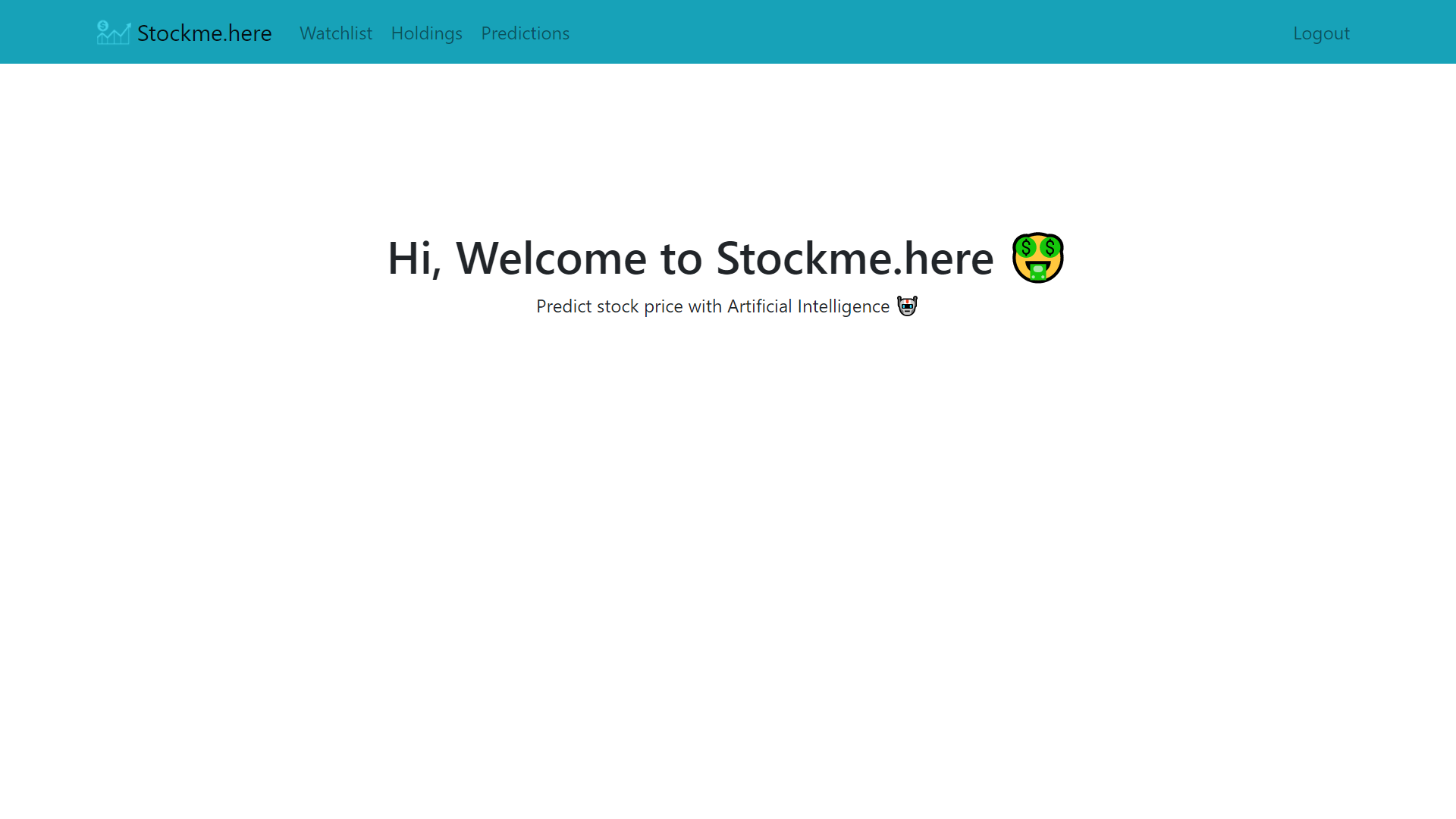
**Home Page**

****

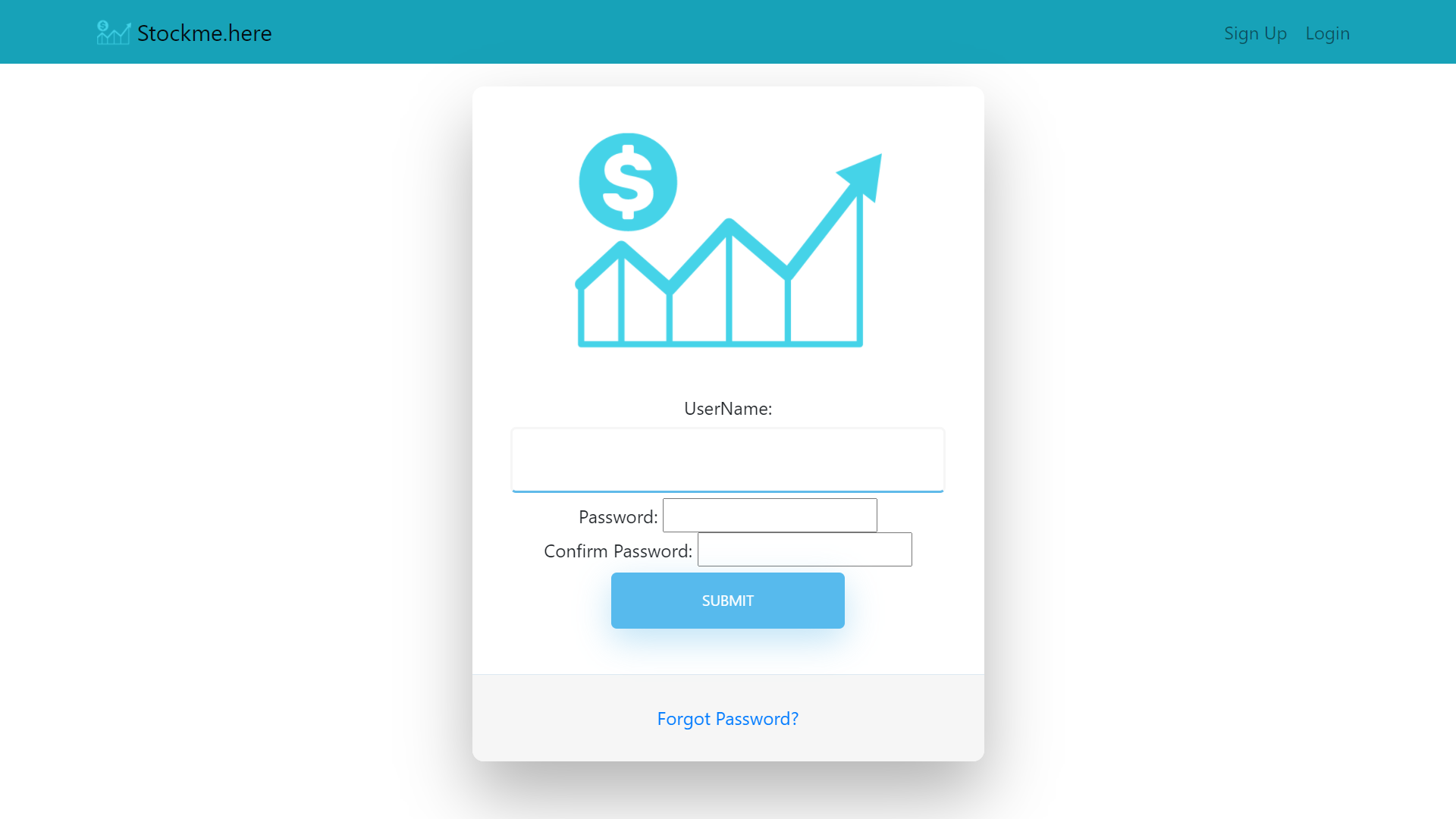
**Login Page:**

****

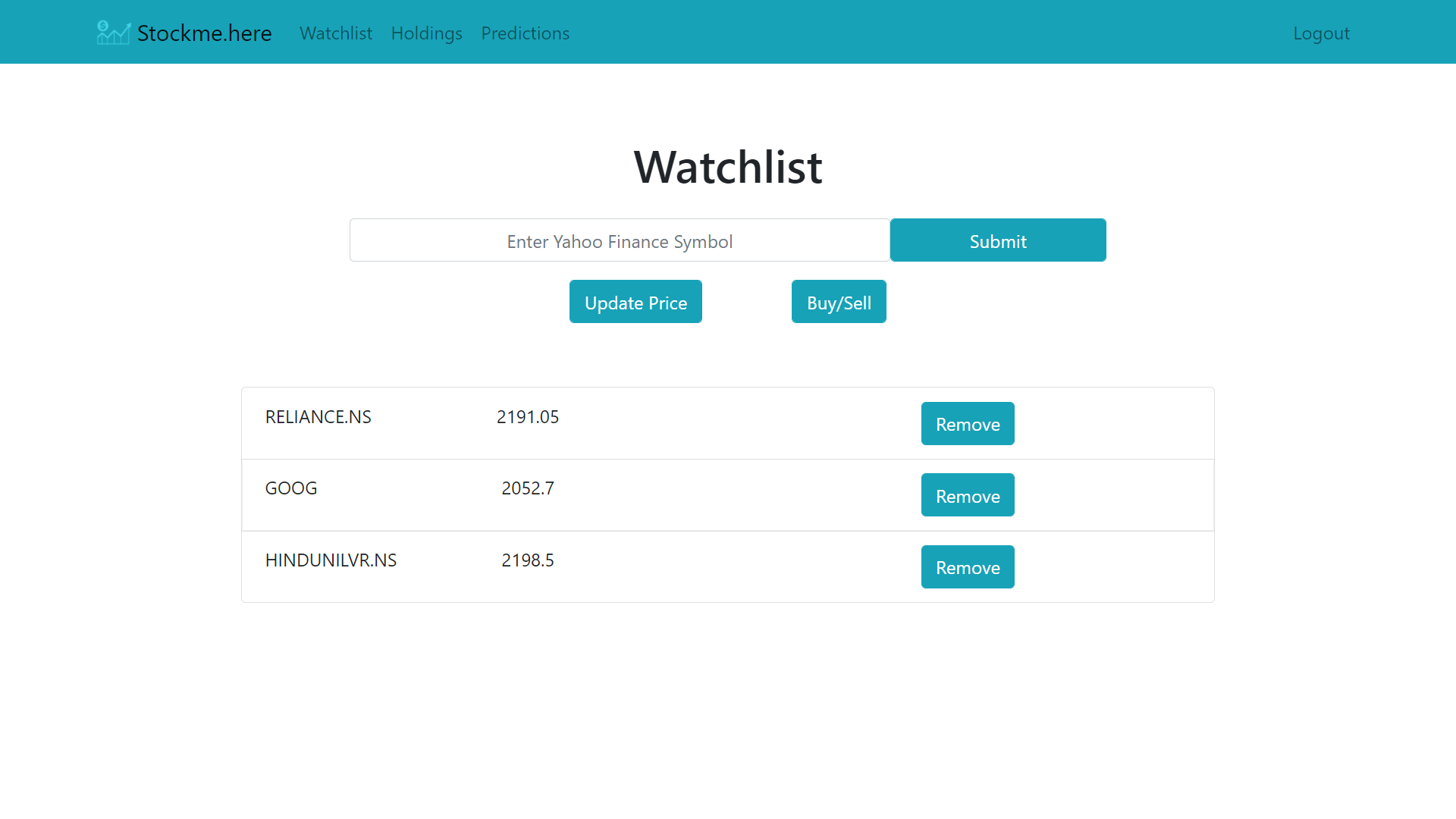
**Home after login**

****

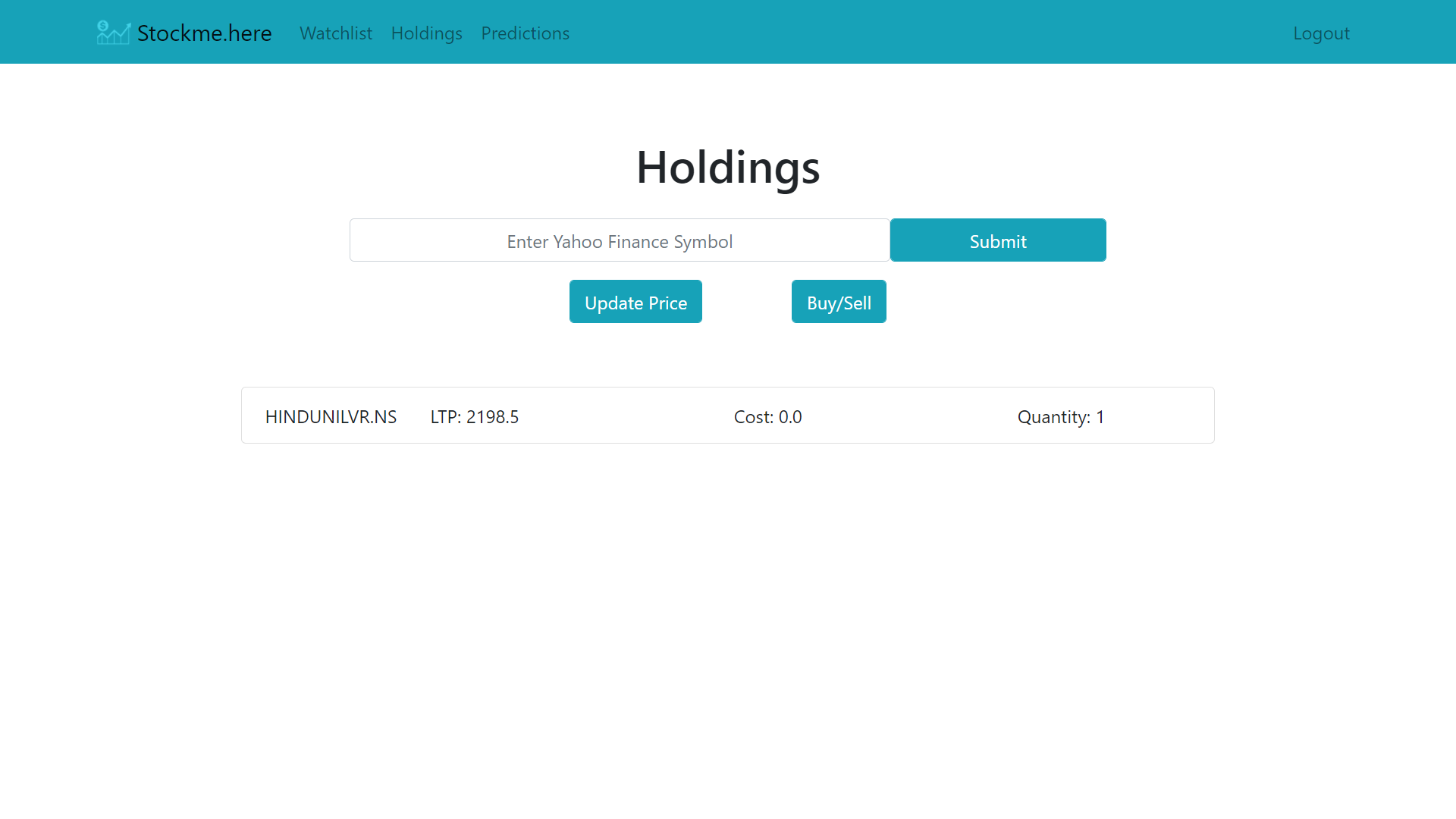
**Signup Page:**

****

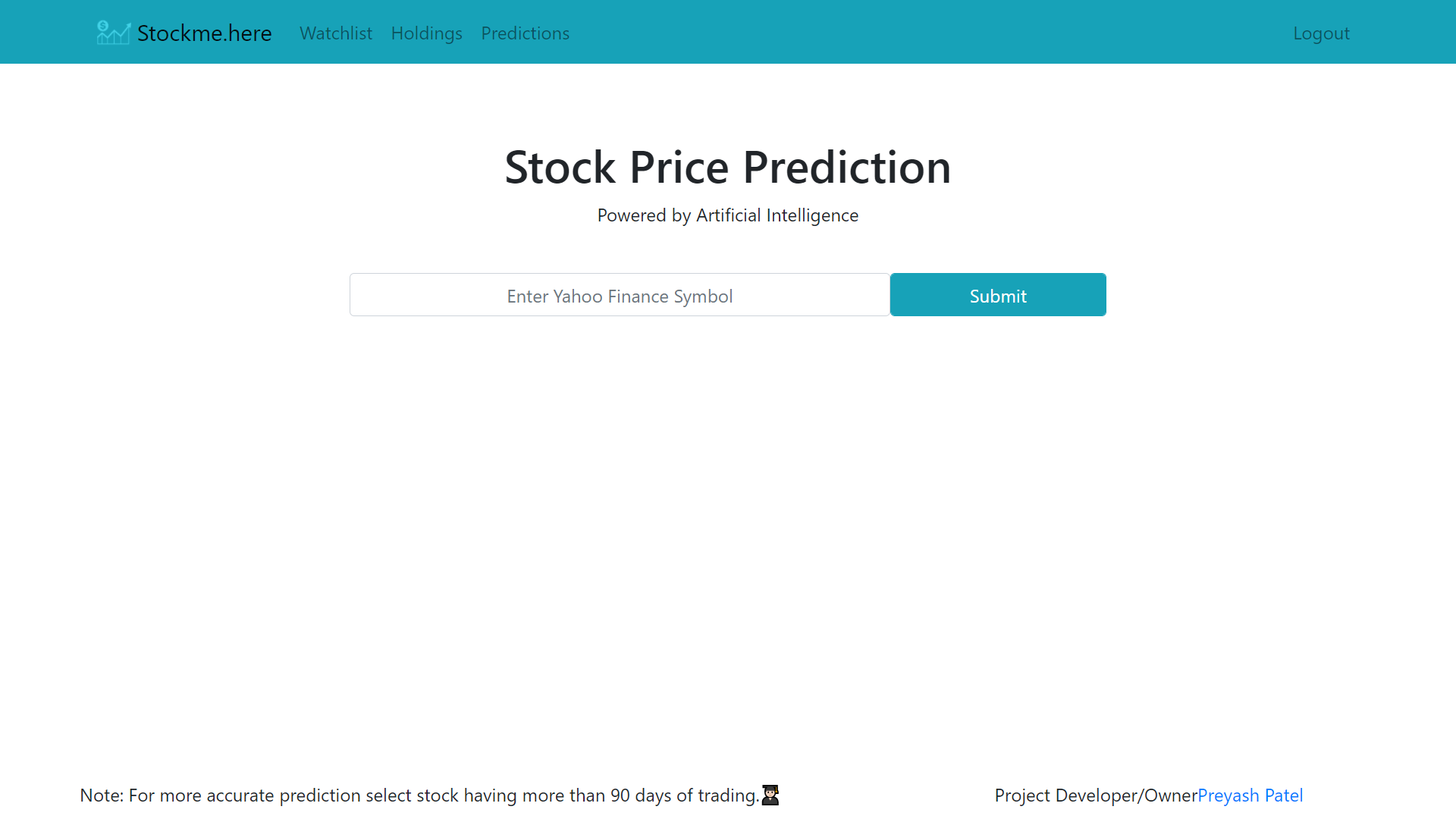
**Watchlist Page**

****

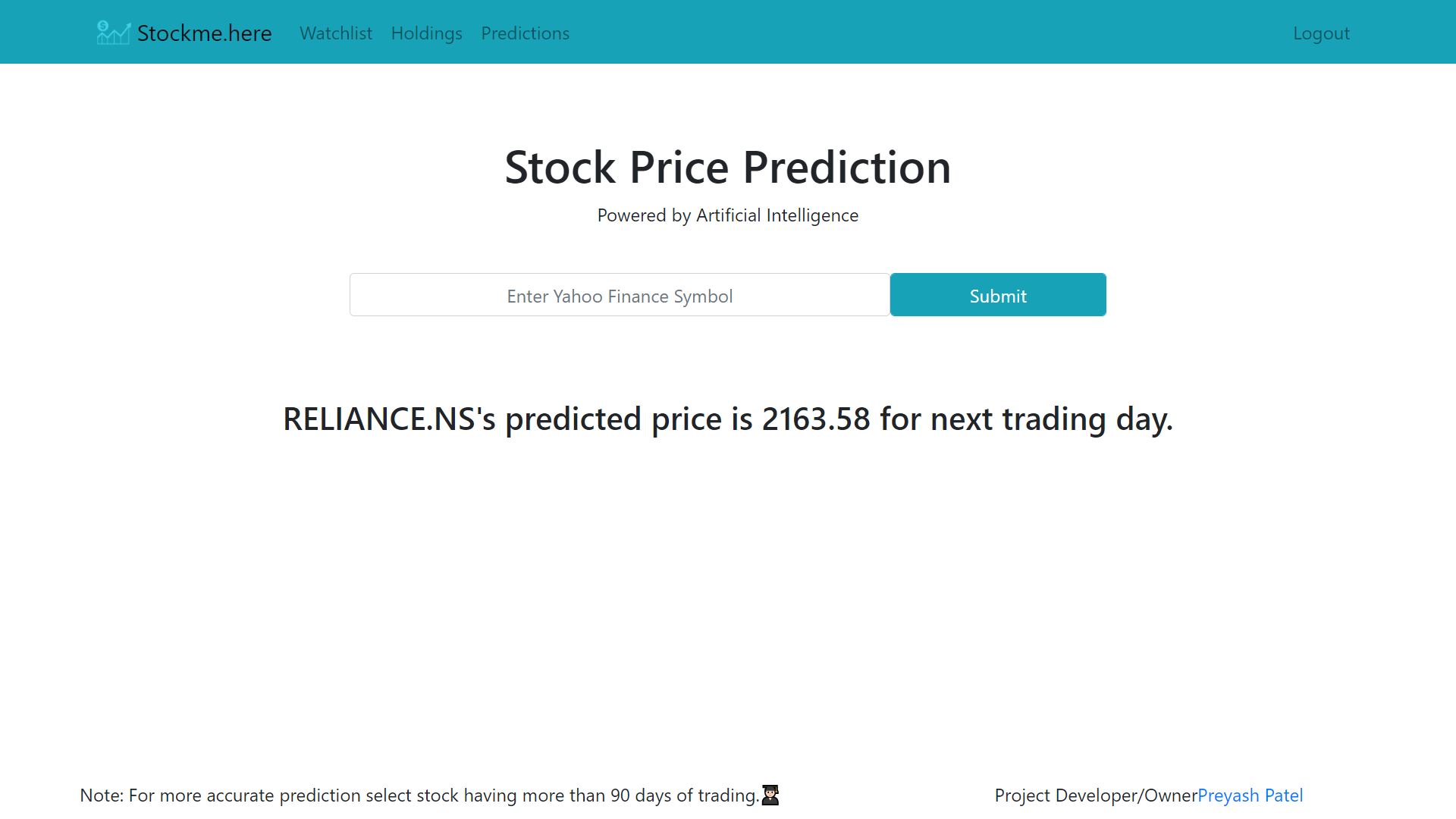
**Holdings Page**

****

**Prediction Page**

****

**Prediction Page**

****

# 9. Testing

**Testing For Login/Signup Validation**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No | Validation Checking | Excepted Result | Test Result |
| 1 | Usename | Not Null and Unique | Pass |
| 2 | Password | Not null | Pass |
| 3 | Confirm Password | Not null | Pass |

**Testing For Watchlist**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No | Validation Checking | Excepted Result | Test Result |
| 1 | Symbol | As per Yahoo Finance | Pass |

**Testing For Holding**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No | Validation Checking | Excepted Result | Test Result |
| 1 | Symbol | As per Yahoo Finance | Pass |
| 2 | Quantity | Greater then Zero | Pass |

**Testing For Prediction**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No | Validation Checking | Excepted Result | Test Result |
| 1 | Symbol | As per Yahoo Finance | Pass |

# 10. Post implementation review

Hereby, it can be proposed that no trading algorithm can be 100% effective, not only 100%, it will typically never be close to 70% but to attain even an accuracy of 40% or 35% is still good sufficient to get a good forecast spread. Although extreme attained accurateness was 39%, it was still able to closely forecast the predictable outcome and have coordinated against the company graph. To make our expectation more efficient, it can be done by including bulky data sets that have millions of entries and could train the machine more powerfully. Different activities of stocks can lead to diverse raises or lows in the forecast price, use these movements to magistrate whether a company should be traded in or not. No training Data can ever be stable, hence there are always some unevenness which can be seen in the above data spread, but to still forecast close to an consequence will also lead to a good approach if it has greater than 33% accuracy. While, developing a strategy trader should always think to always have nominal imbalance while still being above 33%accurate.

It can also be determined that in a stock market, there is probable that some companies might not be associated at all, and mostly can be associated to each other, and can help justice movements of stock accordingly, we can scale affairs and see how much in percentages they are correlated.

Including gigantic data sets, to increase more effectiveness, and in data set if had nan values in tables, because of two simple reasons either a specific company wasn’t opened during that time of year, or the data is not readily obtainable, in both the cases replace the null values with 0, which is somewhat that trader might want to change while developing a trading tactic.

Furthermore, there can be back testing of the trading strategy, using zip line and quantopian a python platform for testing trading strategies and can see how well can a model fit into some random data of stock, and can the model from this random data of stock develop relations and correlations, and predict on terms of change.

# 11. Future Enhancement

* Implement NLTK to have impact of News on Stock Price Prediction
* Speeding up the Model training time
* Maintain prebuilt model for faster response
* News update for respective stock
* Stock’s Fundamentals view

# 12. Bibliography / References

**Books References**

* Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems
* Python Machine Learning

**Web References**

* <https://www.djangoproject.com/>
* https://in.finance.yahoo.com/
* <https://www.python.org/>
* http://tensorflow.org/
* http://keras.io/