CPSC 471 Database Management Systems

Final Report

Group Number: *21*
Stock Scanner

Laura Timm

30064647

Abhay Khosla

30085789

Hassan Khan

3006680

TABLE OF CONTENTS

Abstract

The project we had built throughout the semester using MySQL as the chosen database since SQL was covered heavily during the course it was our primary reason to use this technology and design the database using the language. The languages which were mostly used in the project were HTML/CSS and Python. These languages were combined with a framework called Flask which had a lower learning curve compared to Django, NodeJS, etc. Extra learning was required during the semester as not all group members were familiarized with this framework. The website is targeted towards individuals who are just stepping their foot into the financial stock market or professionals who are tracking stocks for a business or another person. Analysts can view the stocks and analyze them using our platform and can also register as private users. We also have the option to register as an admin with controlling all of the access to the database, which they can add/update/delete business, analysts, users, stocks, and exchanges from filling out the forms. This project's aim is to eliminate the need for a day stock trader to use excel sheets to keep track of all this information that is attached to a stock. This setup presents a modern solution to users shuffling through slow spreadsheets files to keep a track of all stocks. Our website allows users to have all that information at their fingertips.

Introduction

Stock Scanner is a website designed to ease the life of a stock trader, eliminating the usage of notepads or electronic Excel sheets. After completing the web interface it looks pretty smooth to be even used on a tablet size screen which is beneficial for a stock trader working in an exchange because the accessibility is easier if this was hosted publicly on a domain. The platform is free of cost with user authentication implemented to make sure nobody else is able to view other users' watchlist. The purpose of our database is to store the stocks and events for individual stocks since it's convenient to store all of this in one place. The user can view all of the stocks on the website on the main page when they log in successfully. If any searching is required it's mostly done through the backend through the API calls we have designed. With our design stage, we had a lot of diagrams to draw and plan out our implementation component for the project this helped further and refine our coding process. The code is structured with a similar model view controller design pattern. With the model being the python files with our code in them interacting with the database which acts as a controller as well. The view is our templates folder which has all of the HTML files which are displayed on the website. Overall, the concepts we learned through the semester were effectively applied in this final project to resolve the problem which was mentioned earlier about the confusion that day stock traders have to face in order to work in a high-paced environment in an exchange or a firm.

Walkthrough of System Requirements

The system we have created entails a web application that is locally hosted and does not live in the public domain. We tried to keep this project as simple as we could due to the experience levels of all of the group members, we used the framework Flask which requires you to run some pip commands to have it on your device and be able to run it. We also used MySQL for the database which again requires installation but the steps could be found from the Oracle official website with support available to guide through the installation. For the database, you will need to set up a local user on your end and put your credentials in the db.yaml file in the fields "mysql_user" and "mysql_password" in order successfully for the web application to communicate with the database. Need to make sure you have a Python version also installed only on your local machine since it will be required to run Flask and the web application. Flask also makes SQL injections pretty easily with the libraries offered. To check out more on Flask here is the official documentation. Each URL path is mapped to a View, which is a function or class that handles the back end logic, and sends it to the front end HTML "template". Class-based views can be used to automatically generate code that is reused multiple times, whereas in function-based views everything needs to be handled explicitly. The other libraries which were installed were using the Python modules for example REQUEST, JSONIFY which were imported in order to make the API calls work. The website will be running locally on a development server on this link: http://127.0.0.1:5000/. In order for you to use this link, you need to make sure all the things mentioned above are working correctly. Note other imports may be required to do based on the user's missing software. This system also has multiple additional features which are seen in the web interface, for example, three types of users can access this website when registering they are able to choose this option. After they register they are taken to a login page to prevent unauthorized users from logging in to view the stocks currently in the database. You can view stock events related to that stock, headlines about all of the stock, and also an extra addition to this be able to see the 52-week table with the high/low for a stock for the whole year. It also allows authorized users to add stocks to their watchlist as well for easier access to their favourite stocks. Each user is assigned

Different Users In System

There are three various users which are designed to access our application as mentioned above. The first one is a private user which is for individuals and analysts who are just exploring the market and browsing the internet. The second type of user is for professional users who actually have a good grasp of the stocks and can handle on their own, this can be for a company as well to manage their stock as well as contacting the admin to make changes to their 52 weeks high and low. The last user type is an admin which is basically the main boss of the website and can manage the API requests and from the web interface make easy changes rather than using the MySQL command-line client or coding a few changes which will be required as needed be.

Detailed Users Access In The System

Private Users can:

- Register as an authorized user from the login page.
- Login via the login page with their username and password making sure the same credentials are used to prevent any random person from entering our system.
- View Stocks on one page that are currently in the database.

- Access the stock page in detail and also be able to see the news headlines related to the stock on that page.
- Add stocks to their watchlist and view it at any time.
- Ability to see the headlines which are present in the database on the PR page from the right-hand side navbar.
- Access the 52-week table and view the stocks which are present in the database in a tabular form and be able to see their highs and lows of stock prices throughout the year.
- See the events of all stocks inputted in by the database admin, and the events page has any price change which could affect the stock negatively/positively.

Professional Users can:

- Register as an authorized user from the login page.
- Login via the login page with their username and password making sure the same credentials are used to prevent any random person from entering our system.
- View Stocks on one page that are currently in the database.
- Access the stock page in detail and also be able to see the news headlines related to the stock on that page.
- Add stocks to their watchlist and view it at any time.
- Ability to see the headlines which are present in the database on the PR page from the right-hand side navbar.
- Access the 52-week table and view the stocks which are present in the database in a tabular form and be able to see their highs and lows of stock prices throughout the year.
- See the events of all stocks inputted in by the database admin, and the events page has any price change which could affect the stock negatively/positively.
- Can make offerings to the business about buying a certain amount of stocks from them and makes sure to check the status of it as well.

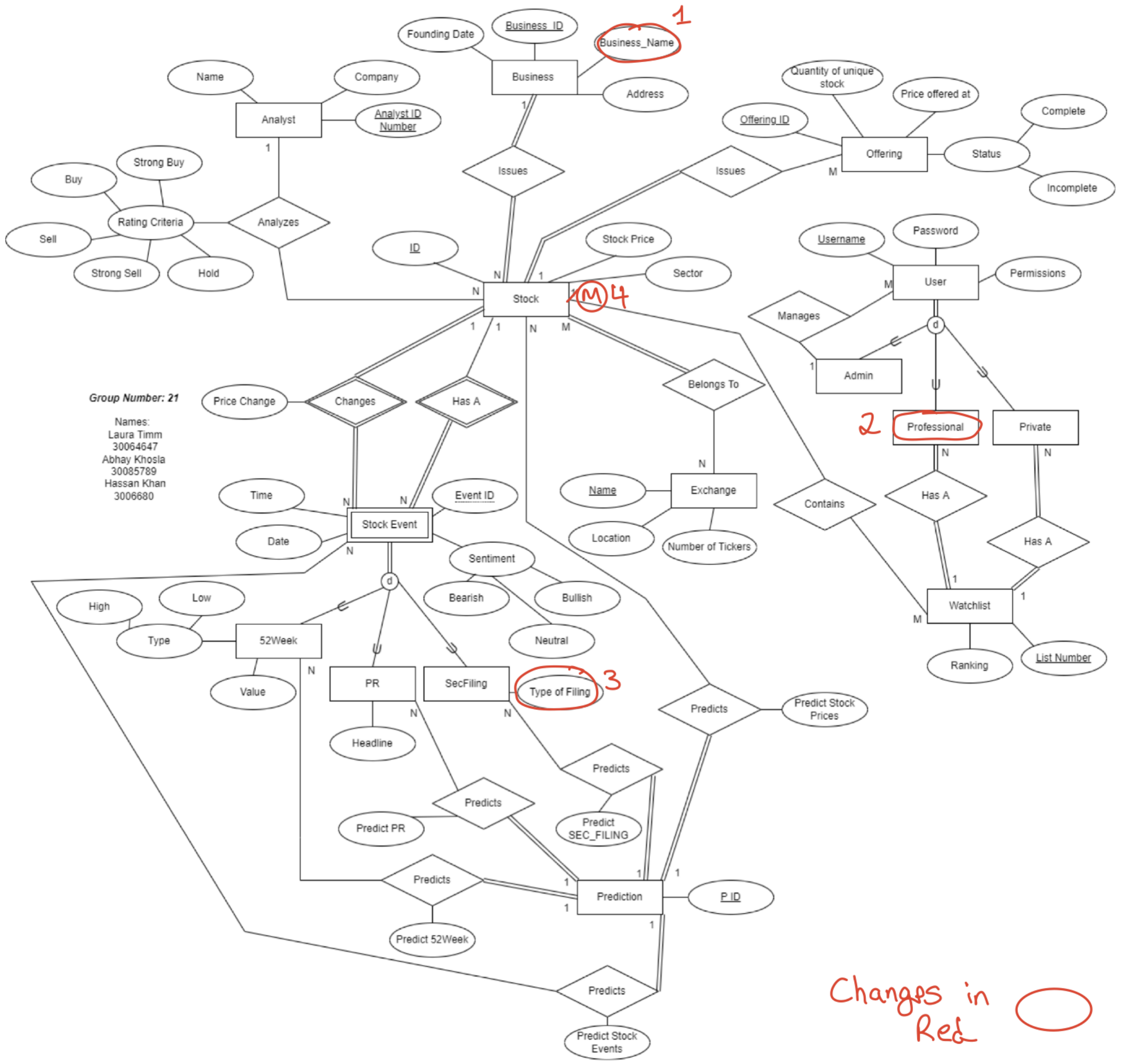
Admin Users can:

- There are three admin users which include all of the group members who can only alter the database and add anything which is required by the two users above.
- An admin can register through the register page as well no limits on how many admins there can be but for this, we will assume there is a max of three.
- The admins will still need to log in to access their view of the website.
- They do not see the stocks on all one page but instead in a tabular form.
- Admins can add/delete/update/ multiple things which include stocks, businesses, analysts, exchanges, and users who are present in the database.
- Admins can also see stocks, businesses, analysts, exchanges, and users in a tabular form by pressing the respective buttons in the row with all of the other functionalities.
- Admins will need to also add images of stocks which will be needed to be later added into the database since we do not store them into it as it’s not a best practice to do so.
- Admins need to update any other headlines as well manually by adding the article information so they will require past programming knowledge which we all possess.

- Will also need to know how to use Postman in order to make any changes that are not on the admin page for example editing/creating a stock event or any other tables which are not seen on the Admin homepage.
- Admin can also view the headlines and 52-week table through the homepage as it's available to anyone regardless of whether you are logged in or not.

Note: A detailed user manual will be provided in the later stages of this report. This is a brief quick rundown of what the API calls did and how it relates back to our users.

Extended Entity Relationship Diagram



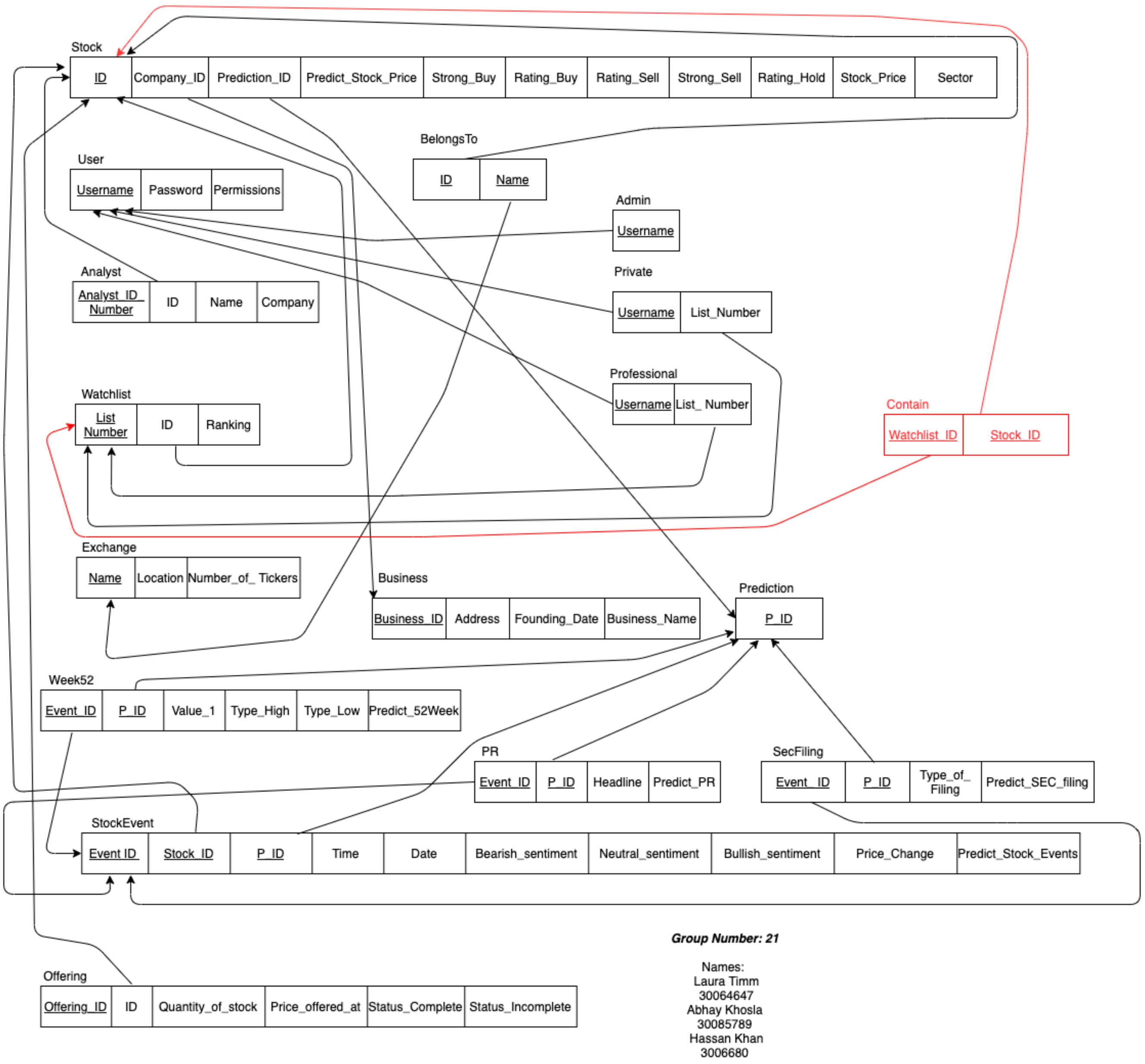
Changes in Red

Explaining Extended Entity Relationship Diagram

This diagram helped a lot when designing since we were clearly able to see the entities we will need for the user to modify or use. This was also created to resolve the issues that day stock traders were facing after getting in contact with some real-life experts, we were motivated to make this website to achieve a good balance for this solution. This evolved our understanding of our project in which we were keeping a track of in our database the attributes and with all of this information from the client as the professor mentions that try to ask many questions from your client. Using this advice we were able to construct a well-thought EER diagram. The EER diagram takes into account multiple roles that a user can for example a user must be private, professional or an admin. Private and professional do not differ a whole lot in the access, just simply the user's choice and their comfort with the market knowledge. Stocks must be added by the database admins through the admin view which will later be shown. We also have a future implementation of artificial intelligence in our EER diagram with the ability of this website to use web scraping from famous financial news articles to determine stock events and create them automatically into the database.

Implementation Section

We followed the algorithm for converting an entity-relationship diagram to a relational schema diagram and came up with the following relational model.


Stock
ID | Company_ID | Prediction_ID | Predict_Stock_Price | Strong_Buy | Rating_Buy | Rating_Sell | Strong_Sell | Rating_Hold | Stock_Price | Sector
BelongsTo
ID | Name
User
Username | Password | Permissions
Admin
Username
Analyst
Analyst_ID_Number | ID | Name | Company
Private
Username | List_Number
Professional
Username | List_ Number
Contain
Watchlist_ID | Stock_ID
Watchlist
List Number | ID | Ranking
Exchange
Name | Location | Number_of_ Tickers
Business
Business_ID | Address | Founding_Date | Business_Name
Prediction
P_ID
Week52
Event_ID | P_ID | Value_1 | Type_High | Type_Low | Predict_52Week
PR
Event_ID | P_ID | Headline | Predict_PR
SecFiling
Event_ID | P_ID | Type_of_ Filing | Predict_SEC_filing
StockEvent
Event ID | Stock_ID | P_ID | Time | Date | Bearish_sentiment | Neutral_sentiment | Bullish_sentiment | Price_Change | Predict_Stock_Events
Offering
Offering_ID | ID | Quantity_of_stock | Price_offered_at | Status_Complete | Status_Incomplete
Group Number: 21
Names:
Laura Timm
30064647
Abhay Khosla
30085789
Hassan Khan
3006680

1

During this process, we began to think about how each entity will behave on the website. From the Relational Model we began to draft an outline of the webpages we wanted our website to include. We discussed the tables that would be used on each page and how they would interact with each other. Next we moved on to creating the API calls. Creating GET, DELETE, PUT, POST calls for each interaction on our website. Once we started writing the API for the website we discovered that for the Watchlist we needed to make some alterations. We split the watchlist into another table called CONTAINS, as our previous Relational Model implementation would only allow us to have once stock per watchlist and we implemented a many to many relationship. Since we wanted each user to be able to add multiple stocks to their watchlist, adding the CONTAIN table made that possible. WATCHLIST then only has list number as an entity now, and CONTAIN has watchlist number and stock_id as entities. Each watchlist is assigned to a user who can now add as many stocks to their list as they would like. We also had to make an alteration to the STOCKEVENT, PR, 52WEEK, and SECFILING tables as we were having issues with creating API calls that used date and time. We changed this so that it would use a varchar instead. Once we made this change we were able to create all the API calls successfully. Once we finished the API calls we moved on to creating a nonfunctional website. This went smoothly. Lastly, we began creating a fully functional website. We made some changes to the API calls and simplified our Watchlist page. The Watchlist page now just contains each stock id where the user can click to see more information. We decided to do this as the back end for this page is quite complicated as it scans the database first for the user in PRIVATE then PROFESSIONAL if not found, next it uses the watchlist number from the previous scan to find all the stocks found in that user's watchlist. We agreed that creating another call could lead to errors for the website so we discussed alternative ways for the user to view the stock information, this is seen on our current implementation for watchlist. We also updated the PR, SECFILING and 52WEEK tables when working on the Stock Information Page so that only the PR table has a primary key. This was done so that the news articles were able to show up on the Stock Information Page. Lastly, on the login page we were not able to implement the remember me button, but most modern browsers have this feature so we agreed that we should spend our time implementing other features instead.

We decided to use MySQL for our database selection, we chose this over MongoDB which is a NoSQL database as we were more experienced with using SQL from class and tutorials. Another reason that we decided to use my SQL was based on the research we did when choosing Flask as our framework. There was quite a bit of support using Flask with MySQL, since we are not familiar with using Flask we decided that it made sense to use the database which had the most documentation for integration with Flask. Overall, we are happy with the choice that we made for our database selection, as we referred to online resources that can be found in our reference section of our report. We also used our notes from class and tutorials that we attended during the semester.

The SQL statements for each of the transactions implemented:

Registration page:

```
"SELECT * FROM USER WHERE USERNAME = %s", ([username])

"INSERT INTO USER(username, password, permissions) VALUES(%s, %s, %s)",
        (username, password, permissions)

"SELECT * FROM Watchlist WHERE List_Number = %s", ([watchlistId])

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
        ([watchlistIdNew])

"INSERT INTO PRIVATE(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
        (username, watchlistId, permissions)

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
        ([watchlistId])
```

```
"INSERT INTO PRIVATE(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
        (username, watchlistId, permissions)

"INSERT INTO PROFESSIONAL(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
              (username, watchlistId, permissions)

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
           ([watchlistId])

"INSERT INTO PROFESSIONAL(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
           (username, watchlistId, permissions)
```

All Users Page:

```
"SELECT * FROM USER"
```

Login Page:

```
"SELECT * FROM USER Where Username = %s And Password = %s And Permissions= %s",
        (usernameForm, passwordForm, userAdmin)

"SELECT * FROM USER Where Username = %s And Password = %s", (usernameForm,
passwordForm)
```

Forgot Page:

```
"SELECT * FROM USER WHERE USERNAME = %s", ([username])

"UPDATE USER SET username = %s, password = %s WHERE username = %s",
        (username, newPassword, username)
```

Admin Add User:

```
"SELECT * FROM USER WHERE USERNAME = %s", ([username]))

"INSERT INTO USER(username, password, permissions) VALUES(%s, %s, %s)",
        (username, password, permissions)
"SELECT * FROM Watchlist WHERE List_Number = %s", ([watchlistId]))

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
              ([watchlistIdNew])

"INSERT INTO PRIVATE(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
              (username, watchlistId, permissions)

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
           ([watchlistId])
```

```
"INSERT INTO PRIVATE(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
            (username, watchlistId, permissions)

"SELECT * FROM Watchlist WHERE List_Number = %s", ([watchlistId])

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
               ([watchlistIdNew])

"INSERT INTO PROFESSIONAL(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
               (username, watchlistId, permissions)

"INSERT INTO Watchlist(List_Number) VALUES(%s)",
            ([watchlistId])

"INSERT INTO PROFESSIONAL(username, List_Number, Role_Type) VALUES(%s, %s, %s)",
(username, watchlistId, permissions)
```

Admin Delete User:

```
"SELECT * FROM USER WHERE USERNAME = %s", ([username])

"DELETE FROM USER WHERE USERNAME = %s", ([username])
```

Admin Update User:

```
"SELECT * FROM USER WHERE USERNAME = %s", ([username])

"UPDATE USER SET username = %s, password = %s, permissions = %s WHERE username = %s",
        (username, password, permissions, username)
```

Admin Add Exchange:

```
"SELECT * FROM EXCHANGES WHERE NAME = %s", ([name]))

"INSERT INTO EXCHANGES(Name, Location, Number_of_Tickers) VALUES(%s, %s, %s)",
        (name, location, numberOfTickers)
```

Admin Delete Exchange:

```
"SELECT * FROM EXCHANGES WHERE NAME = %s", ([name])
```

Admin Update Exchange:

```
"SELECT * FROM EXCHANGES WHERE NAME = %s", ([name])

"UPDATE EXCHANGES SET Name = %s, Location = %s, Number_of_Tickers = %s WHERE Name =
%s",
        (name, location, numberOfTickers, name)
```

Admin Show Exchanges:
"SELECT * FROM EXCHANGES"

Admin Add Business:
"SELECT * FROM BUSINESS WHERE Business_ID = %s", ([business_id])

"INSERT INTO BUSINESS(Business_ID, Address, Founding_Date, Business_Name) VALUES(%s, %s, %s, %s)",
(business_id, address, founding_date, business_name)

Admin Delete Business:
"SELECT * FROM BUSINESS WHERE Business_ID = %s", ([business_id])

Admin Update Business:
"SELECT * FROM BUSINESS WHERE Business_ID = %s", ([business_id])

"UPDATE BUSINESS SET Business_ID=%s, Address=%s, Founding_Date=%s, Business_Name=%s WHERE Business_ID=%s",
(business_id, address, founding_date, business_name, business_id)

Admin Show Businesses:
"SELECT * FROM BUSINESS"

Admin Add Analyst:
"SELECT * FROM ANALYST WHERE Analyst_ID_Number = %s", ([analyst_id_number])

"INSERT INTO ANALYST(Analyst_ID_Number, ID, Name, Company) VALUES(%s, %s, %s, %s)",
(analyst_id_number, stock_id, analyst_name, analyst_company)

Admin Delete Analyst:
"SELECT * FROM ANALYST WHERE Analyst_ID_Number = %s", ([analyst_id_number])

Admin Update Analyst:
"SELECT * FROM ANALYST WHERE Analyst_ID_Number = %s", ([analyst_id_number])

"UPDATE ANALYST SET Analyst_ID_Number=%s, ID=%s, Name=%s, Company=%s WHERE Analyst_ID_Number=%s",
(analyst_id_number, stock_id, analyst_name, analyst_company, analyst_id_number)

Admin Show Analysts:

```
"SELECT * FROM ANALYST"
```

Admin Add Stock:

```
"SELECT * FROM STOCK WHERE ID = %s", ([stock_id])

"INSERT INTO STOCK(ID, Company_ID, Prediction_ID, Predict_Stock_Price, Strong_Buy,
Rating_Buy, Rating_Sell, Strong_Sell, Rating_Hold, Stock_Price, Sector) VALUES(%s, %s, %s, %s, %s,
%s, %s, %s, %s, %s, %s)",
(stock_id, company_id, prediction_id, predict_stock_price, strong_buy, rating_buy, rating_sell,
strong_sell, rating_hold, stock_price, sector)
```

Admin Delete Stock:

```
"SELECT * FROM STOCK WHERE ID = %s", ([stock_id])
```

Admin Update Stock:

```
"SELECT * FROM STOCK WHERE ID = %s", ([stock_id])

"UPDATE STOCK SET ID=%s, Company_ID=%s, Prediction_ID=%s, Predict_Stock_Price=%s,
Strong_Buy=%s, Rating_Buy=%s, Rating_Sell=%s, Strong_Sell=%s, Rating_Hold=%s,
Stock_Price=%s, Sector=%s WHERE ID=%s",
(stock_id, company_id, prediction_id, predict_stock_price, strong_buy, rating_buy, rating_sell,
strong_sell, rating_hold, stock_price, sector, stock_id)
```

Admin & User Show Stock:

```
"SELECT * FROM STOCK"
```

Show Stock Information:

```
"SELECT * FROM STOCK WHERE ID = %s", ([ID])

"SELECT * FROM STOCKEVENT WHERE STOCK_ID = %s", ([ID])

"SELECT * FROM PR WHERE P_ID = %s", ([newPID])

"SELECT * FROM ANALYST WHERE ID = %s", ([ID])
```

Show Watchlist:

```
"SELECT List_Number FROM PRIVATE WHERE Username = %s"
resultValue = cur.execute(select_stmt, (new_User,)

select_stmt = "SELECT List_Number FROM PROFESSIONAL WHERE Username = %s"
resultValue = cur.execute(select_stmt, (new_User,)

"SELECT * FROM CONTAIN WHERE Watchlist_ID = %s AND Stock_ID=%s "
```

```
cur.execute(select_stmt, (newWatchlist, post_id)

"INSERT INTO CONTAIN(Stock_ID, Watchlist_ID) VALUES(%s, %s)",
        (post_id, newWatchlist)

"SELECT * FROM CONTAIN WHERE Watchlist_ID = %s", ([newWatchlist])
```

Delete from Watchlist:

```
"DELETE FROM CONTAIN WHERE Watchlist_ID = %s AND Stock_ID=%s",
                ([watchlist, stock_id]))
```

Show Events:

```
"SELECT * FROM STOCKEVENT"
```

Show PR Details:

```
"SELECT HEADLINE FROM PR"
```

Show 52 Week Details:

```
"SELECT * FROM BUSINESS"

"SELECT * FROM Week52"
```

Show SecFiling Page:

```
"SELECT * FROM SECFILING"
```

Show Offering Page:

```
"SELECT * FROM OFFERING"
```

Add Offering in Professional:

```
"SELECT * FROM OFFERING WHERE Offering_ID = %s", ([offeringID])

"INSERT INTO OFFERING(Offering_ID, ID, Quantity_of_stock, Price_offered_at, Status_Complete,
Status_Incomplete) VALUES(%s, %s, %s, %s, %s, %s)",
            (offeringID, iD, offer_quant, offer_price, statusComplete, statusNotComplete)
```

Update Offering in Professional:

```
"SELECT * FROM OFFERING WHERE Offering_ID = %s", ([offeringID])

"UPDATE OFFERING SET Status_Complete=%s, Status_Incomplete=%s WHERE Offering_ID=%s",
            (statusComplete, statusNotComplete, offeringID)
```

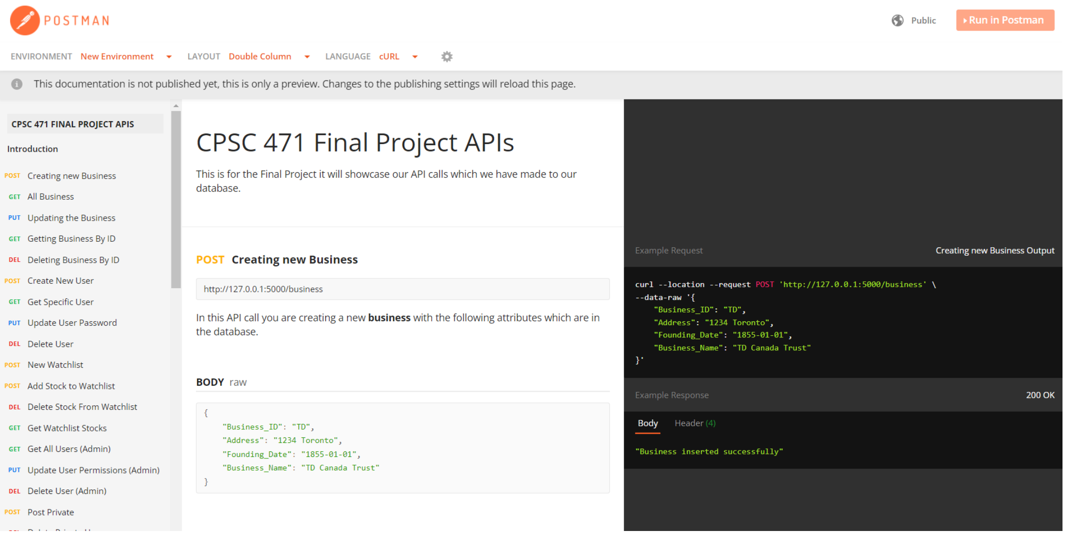
Delete Offering in Professional:

```
"SELECT * FROM OFFERING WHERE Offering_ID = %s", ([offerID])
```

"DELETE FROM OFFERING WHERE Offering_ID = %s", ([offerID])

API Postman Documentation

The API was tested and documented using Postman. The documentation which was generated can be found here: https://documenter.getpostman.com/view/15869453/UVRAH6qM
Since data in Flask is served to the front end directly instead of through back-end APIs, the APIs views are separate from the views used for the website and were developed using the Flask REST Framework. The APIs allow in the future for a mobile application to be developed, which will get and serve data through interacting with these APIs. A sample screenshot of the preview of the documentation:


POSTMAN
Public
Run in Postman
CPSC 471 FINAL PROJECT APIS
Introduction
POST Creating new Business
GET All Business
PUT Updating the Business
GET Getting Business By ID
DEL Deleting Business By ID
POST Create New User
GET Get Specific User
PUT Update User Password
DEL Delete User
POST New Watchlist
POST Add Stock to Watchlist
DEL Delete Stock From Watchlist
GET Get Watchlist Stocks
GET Get All Users (Admin)
PUT Update User Permissions (Admin)
DEL Delete User (Admin)
POST Post Private
Body
Header (4)

ENVIRONMENT New Environment LAYOUT Double Column LANGUAGE cURL

This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

CPSC 471 Final Project APIs

This is for the Final Project it will showcase our API calls which we have made to our database.

POST Creating new Business

http://127.0.0.1:5000/business

In this API call you are creating a new **business** with the following attributes which are in the database.

BODY raw

```
{
    "Business_ID": "TD",
    "Address": "1234 Toronto",
    "Founding_Date": "1855-01-01",
    "Business_Name": "TD Canada Trust"
}
```

Example Request — Creating new Business Output

```
curl --location --request POST 'http://127.0.0.1:5000/business' \
--data-raw '{
    "Business_ID": "TD",
    "Address": "1234 Toronto",
    "Founding_Date": "1855-01-01",
    "Business_Name": "TD Canada Trust"
}'
```

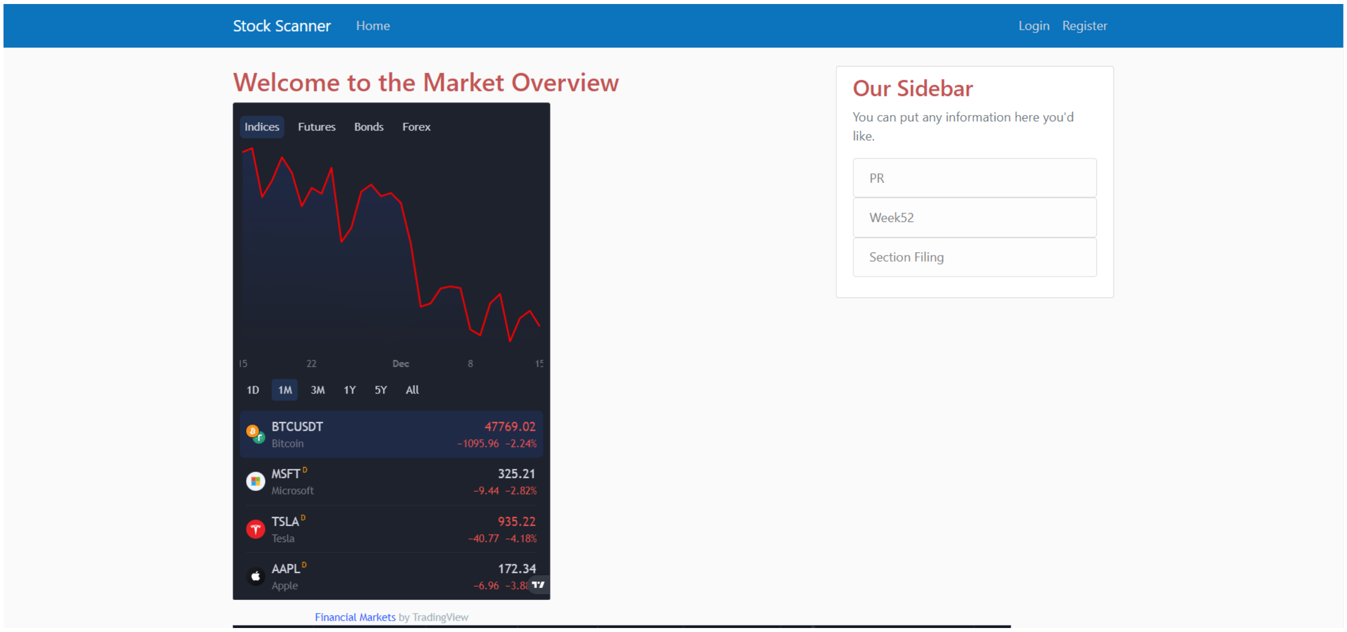
Example Response — 200 OK

```
"Business inserted successfully"
```

User Guide

This section of the report is a walkthrough of the web interface with some explanation that is different from what was included in the functional planning document submitted a few weeks ago. This takes into account any notes that the TA should have when using our program or anyone else on the internet is using it when this is deployed live onto a public domain. Acts like a manual for operating a car but instead, it's a website. There are also screenshots that have been attached to provide better visualization of the current steps of a user's path. The screenshots of the database successfully show the connection between user interface is completed in the backend of the MySQL database with the help of the framework Flaska and its SQL injections which are being performed in our Python programming files which will also be attached as a .zip folder alongside with our .sql file.

Home Page(URL: http://127.0.0.1:5000 or http://127.0.0.1:5000/home)



Stock Scanner Home Login Register

Welcome to the Market Overview

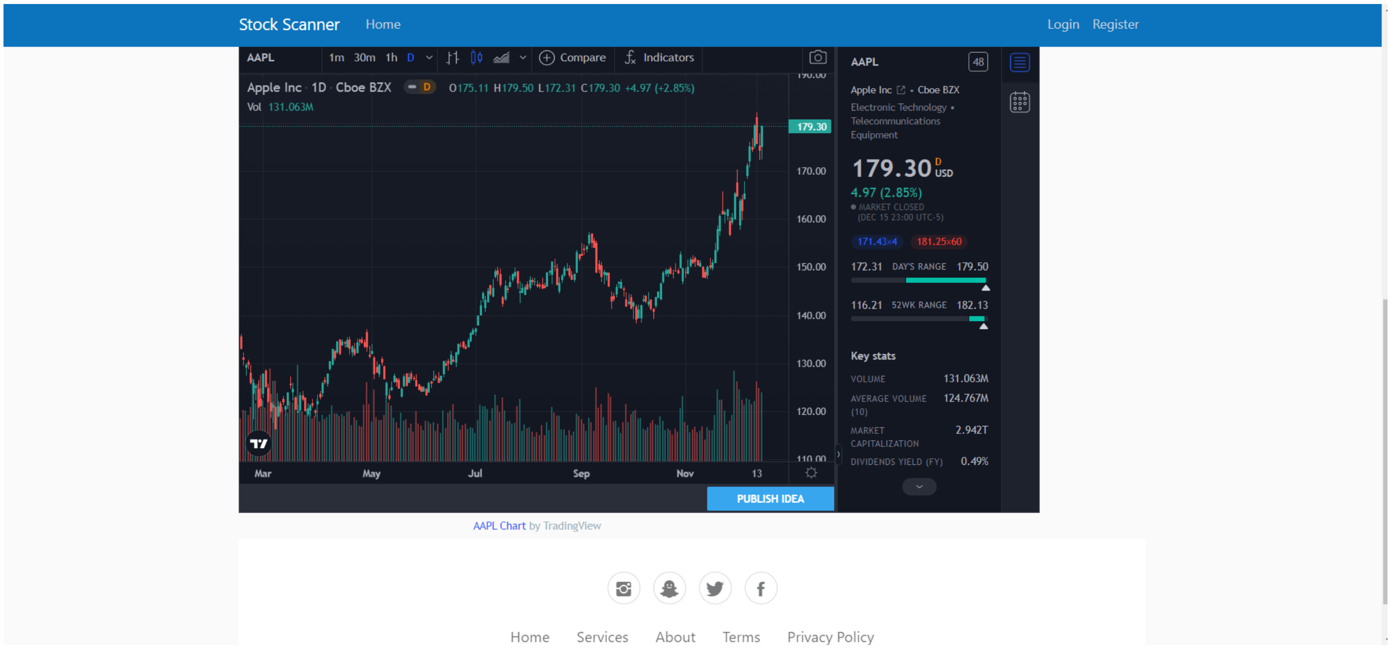
Our Sidebar

You can put any information here you'd like.

PR

Week52

Section Filing

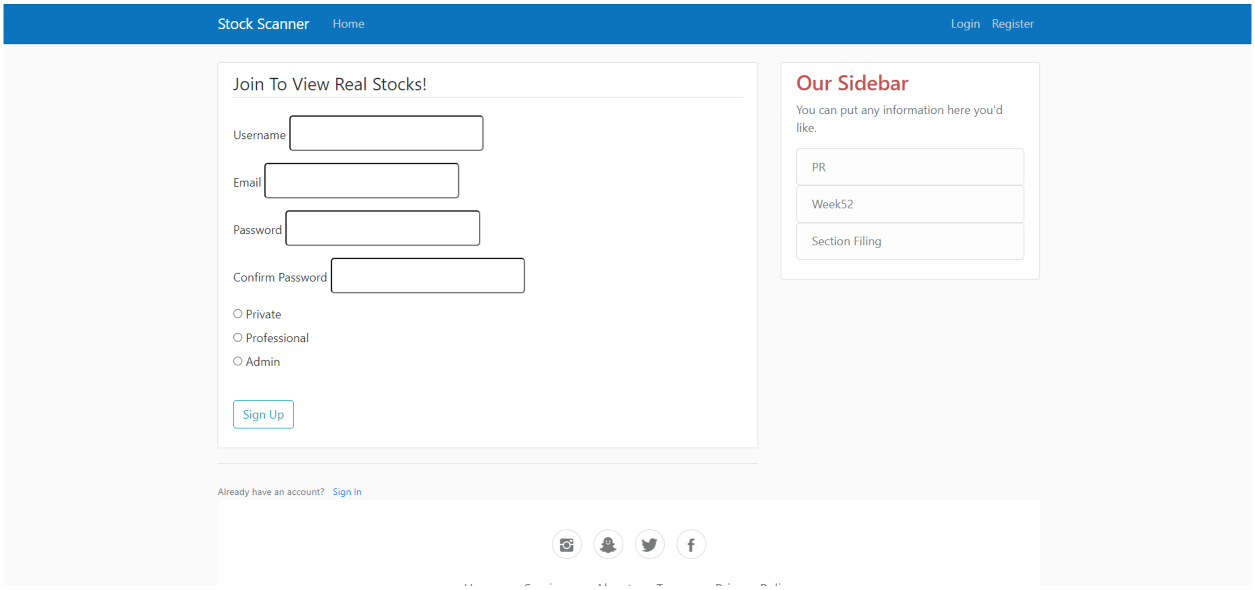


Stock Scanner Home Login Register

AAPL
1m
30m
1h
D
Compare
Indicators
Apple Inc · 1D · Cboe BZX
O175.11 H179.50 L172.31 C179.30 +4.97 (+2.85%)
Vol 131.063M
179.30
170.00
160.00
150.00
140.00
130.00
120.00
110.00
Mar
May
Jul
Sep
Nov
13
PUBLISH IDEA
AAPL
Apple Inc · Cboe BZX
Electronic Technology · Telecommunications Equipment
179.30 USD
4.97 (2.85%)
MARKET CLOSED (DEC 15 23:00 UTC-5)
171.43×4
181.25×60
172.31 DAY'S RANGE 179.50
116.21 52WK RANGE 182.13
Key stats
VOLUME 131.063M
AVERAGE VOLUME (10) 124.767M
MARKET CAPITALIZATION 2.942T
DIVIDENDS YIELD (FY) 0.49%
AAPL Chart by TradingView
Home
Services
About
Terms
Privacy Policy

When anyone visits this website they will start off with this page and can view some widgets and check out what's currently happening in the stocks market by playing around with the widgets which were referenced from TradingView. After the user is done browsing they can be presented with some choices in the navbar either register or login depending on what their current state of authorization is they can take the branch of pages. For example, in this case, we will start the user by registering with the next two screenshots.

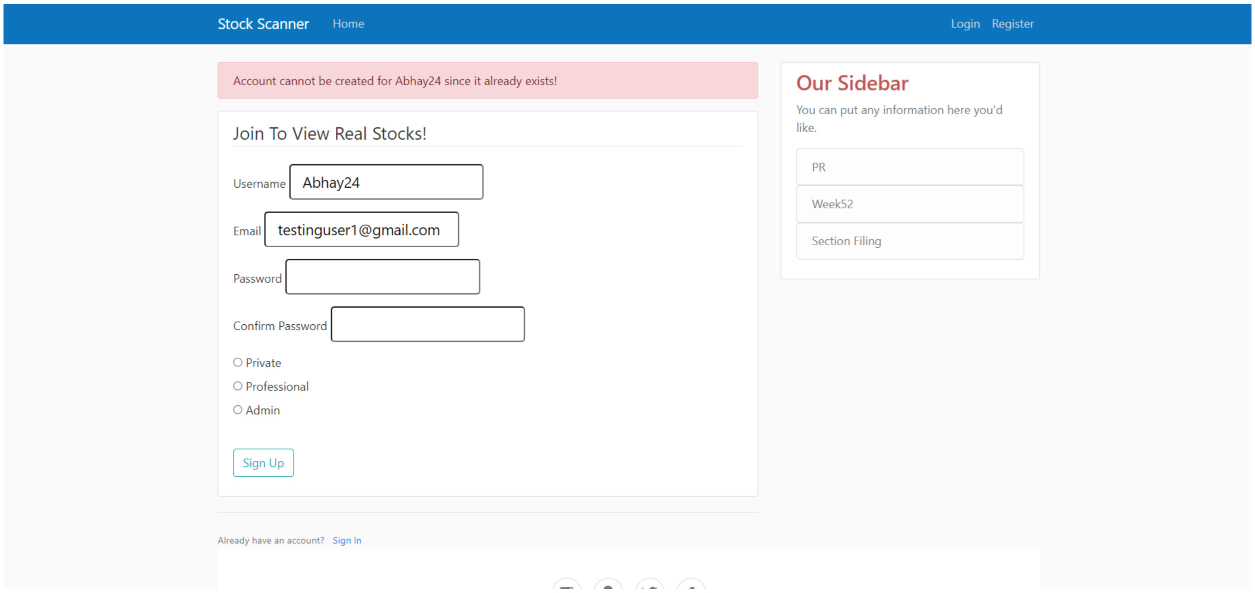
Register Page(URL: http://127.0.0.1:5000/register)



You can put any information here you'd like.

This is the platform where users can register for free and choose which type of user they are, it requires them to input a valid email and a username that is not taken already in the system it will prompt them if this is the case. The password has to be a minimum of eight characters long or else it will fail the registration of the user. If the user directly wants to have the login credentials they would need to contact the admin in real life and be able to grant them access.

Failed Register Page(URL: http://127.0.0.1:5000/register)



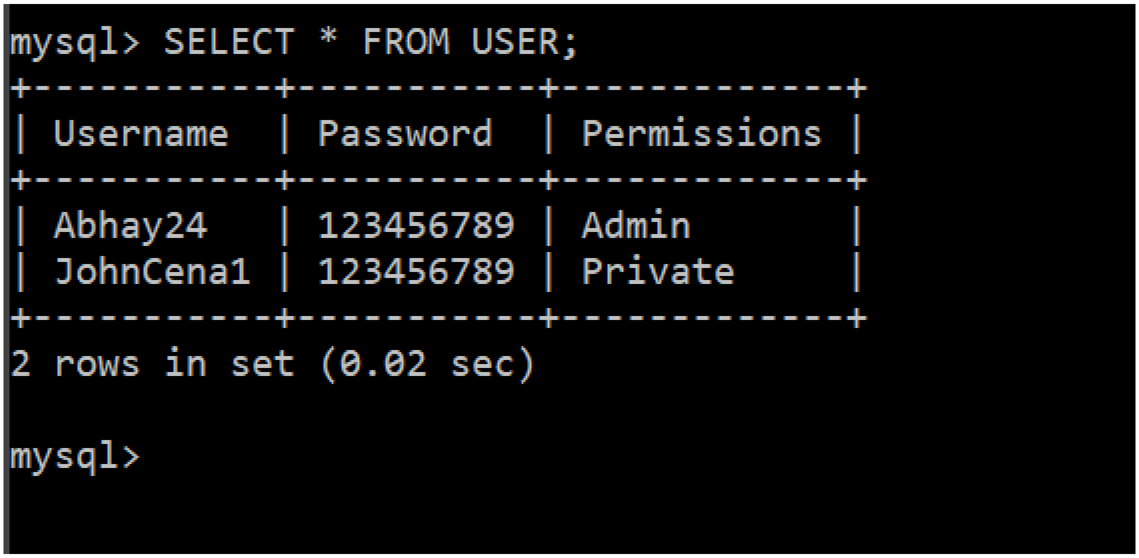
Account cannot be created for Abhay24 since it already exists!

Username: Abhay24

Email: testinguser1@gmail.com

You can put any information here you'd like.

As you can see the username with Abhay24 exists in the database hence the application will throw a flash message to the user to create a different username following this message the user should be able to come up with a new username. We can also confirm by attaching a screenshot of the current state of the database:

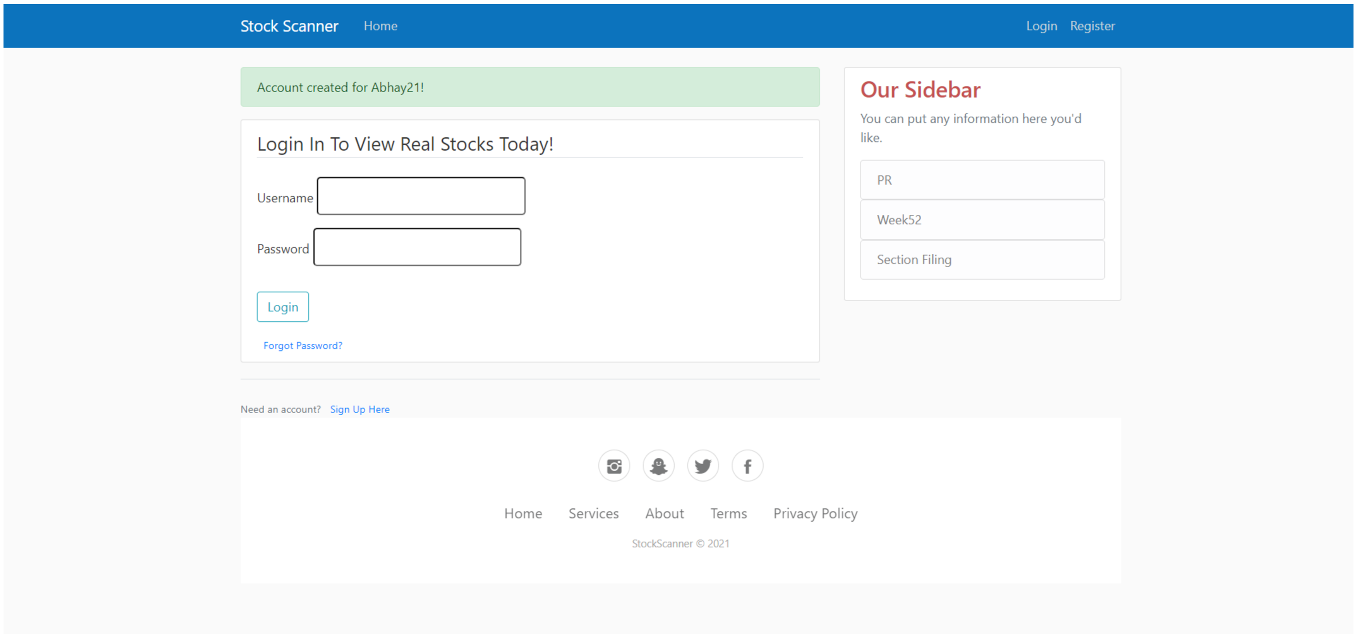


```
mysql> SELECT * FROM USER;
+-----------+-----------+-------------+
| Username  | Password  | Permissions |
+-----------+-----------+-------------+
| Abhay24   | 123456789 | Admin       |
| JohnCena1 | 123456789 | Private     |
+-----------+-----------+-------------+
2 rows in set (0.02 sec)

mysql>
```

Confirmed with the command line client of MySQL that Abhay24 exists in the database.

Successful Register Page Redirected to Login(URL: http://127.0.0.1:5000/login)



Account created for Abhay21!

Login In To View Real Stocks Today!

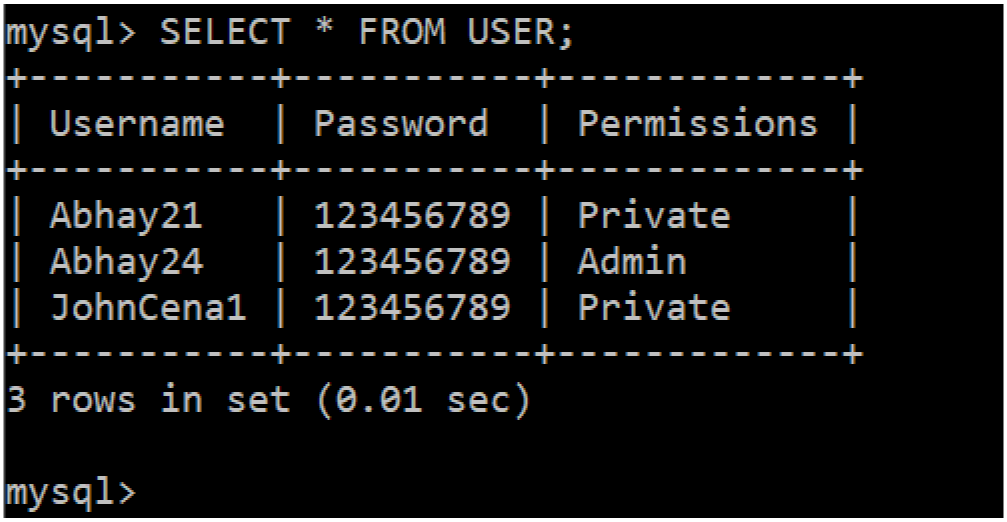
Username

Password

Our Sidebar

You can put any information here you'd like.

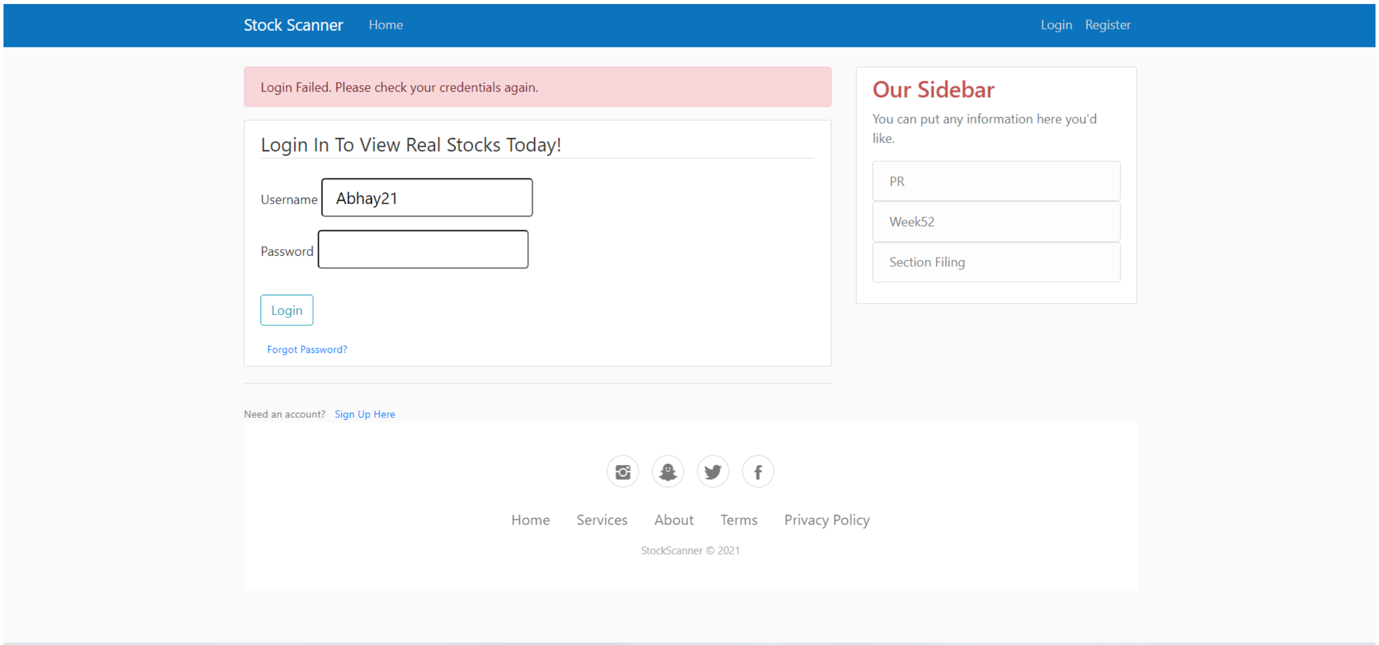
As the user here was successful in registering they will be redirected to the login page directly to login with their newly created credentials notice we don't use the email address here since it seemed a bit harder to remember as users could have multiple emails instead of opting out for the username login approach. It also flashes the user with their username that the account has been created and we can see the updated state of the database:



```
mysql> SELECT * FROM USER;
+-----------+-----------+-------------+
| Username  | Password  | Permissions |
+-----------+-----------+-------------+
| Abhay21   | 123456789 | Private     |
| Abhay24   | 123456789 | Admin       |
| JohnCena1 | 123456789 | Private     |
+-----------+-----------+-------------+
3 rows in set (0.01 sec)

mysql>
```

Failed to Login(URL: http://127.0.0.1:5000/login)



Login Failed. Please check your credentials again.

Login In To View Real Stocks Today!

Username: Abhay21

Password:

Our Sidebar

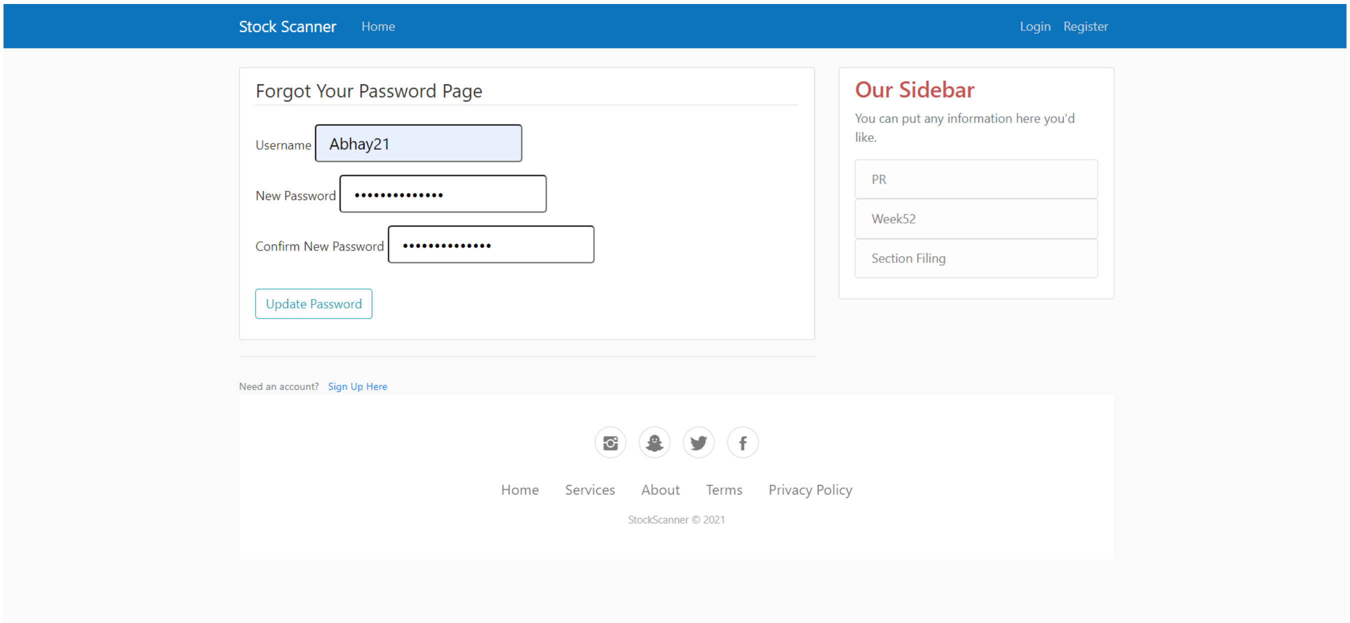
You can put any information here you'd like.

Need an account? Sign Up Here

StockScanner © 2021

If the user manages to not correctly input their password or username it flashes you a message to check your credentials again, the forgot password will take you to a page where you can input your username and make a new password and it will update your information in the database if you do not remember it correctly for example, since I have forgotten my password I will go to the page and update my password.

Forgot Password Page(URL:http://127.0.0.1:5000/forgot)



Forgot Your Password Page

Username: Abhay21

New Password: ••••••••••••••

Confirm New Password: ••••••••••••••

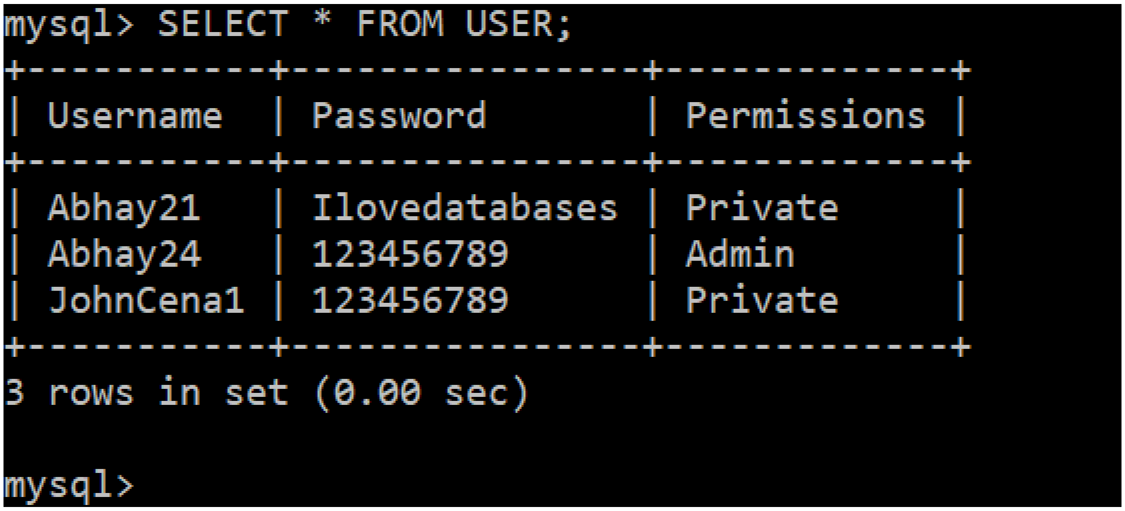
Our Sidebar

You can put any information here you'd like.

- PR
- Week52
- Section Filing

It resembles the register page, instead, it will ask you to input your correct username, and your new password and confirm that again to make sure there is no issue with the passwords. If it successfully works it will redirect you to the login page and will flash a message informing the user that their credentials have been updated.

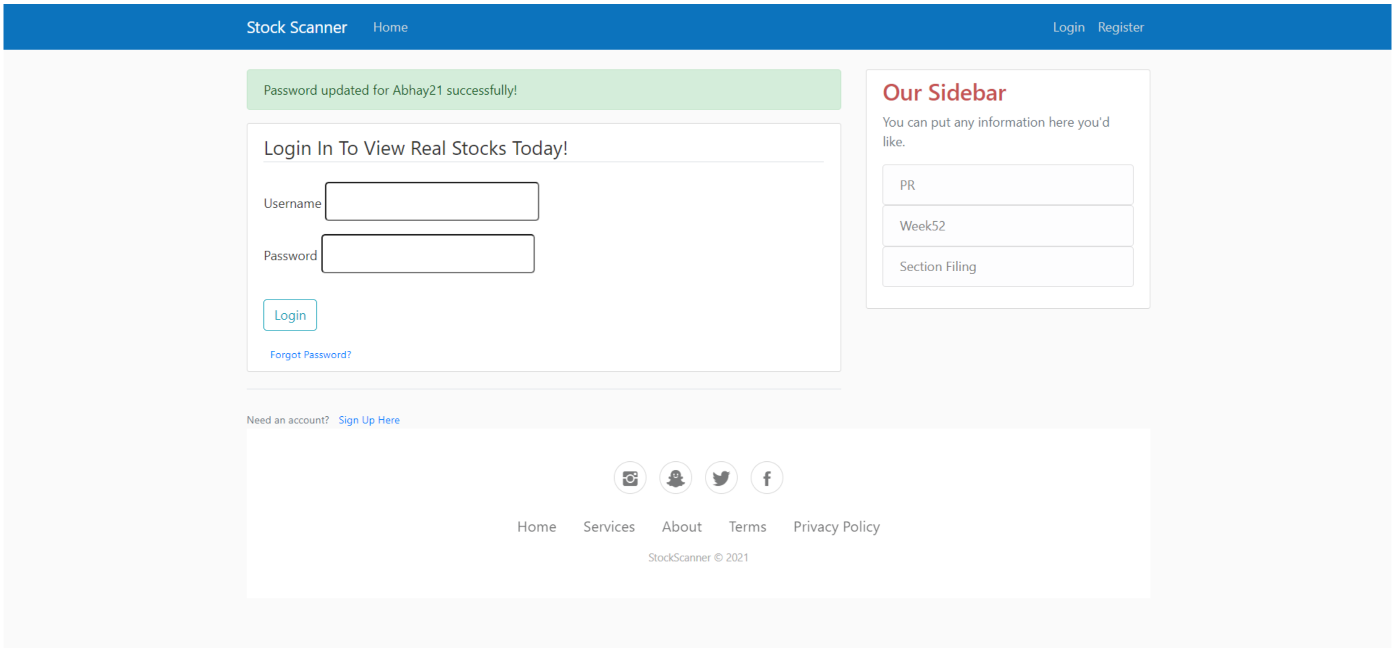
The database is updated to this afterwards:



```
mysql> SELECT * FROM USER;
+-----------+----------------+-------------+
| Username  | Password       | Permissions |
+-----------+----------------+-------------+
| Abhay21   | Ilovedatabases | Private     |
| Abhay24   | 123456789      | Admin       |
| JohnCena1 | 123456789      | Private     |
+-----------+----------------+-------------+
3 rows in set (0.00 sec)

mysql>
```

Forgot Password Successful Redirect to Login(URL:http://127.0.0.1:5000/login)



Password updated for Abhay21 successfully!

Login In To View Real Stocks Today!

Username

Password

Our Sidebar

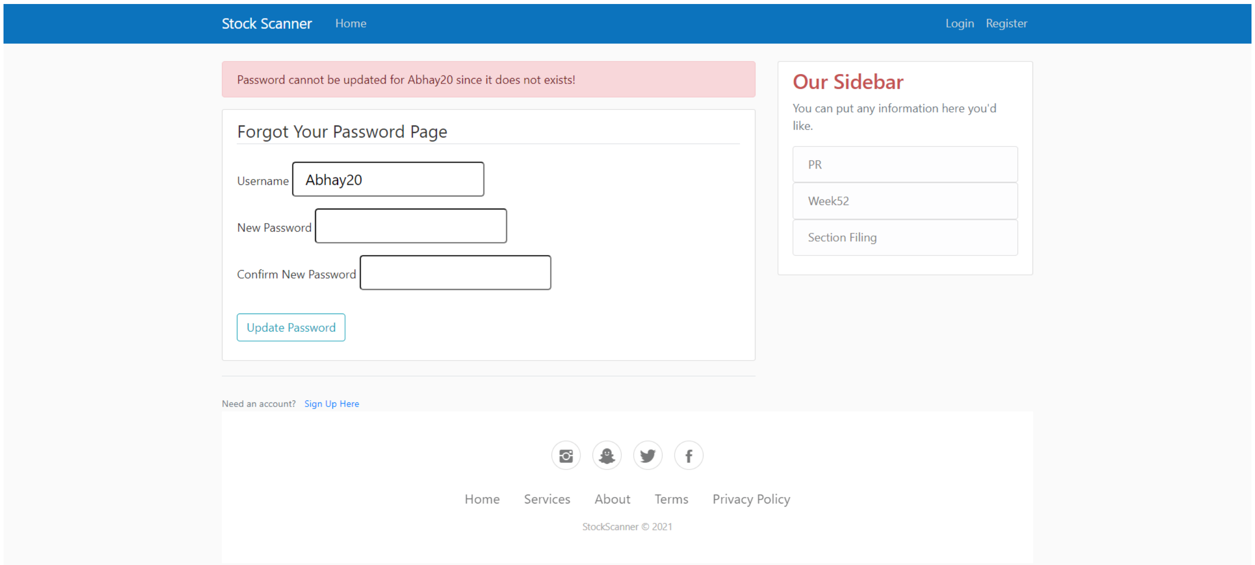
You can put any information here you'd like.

Need an account? Sign Up Here

StockScanner © 2021

After successfully updating your password the user can finally login into the application to view the stocks and access their watchlist.

Forgot Password Unsuccessful(URL:http://127.0.0.1:5000/forgot)



Password cannot be updated for Abhay20 since it does not exists!

Forgot Your Password Page

Username: Abhay20

New Password

Confirm New Password

Our Sidebar

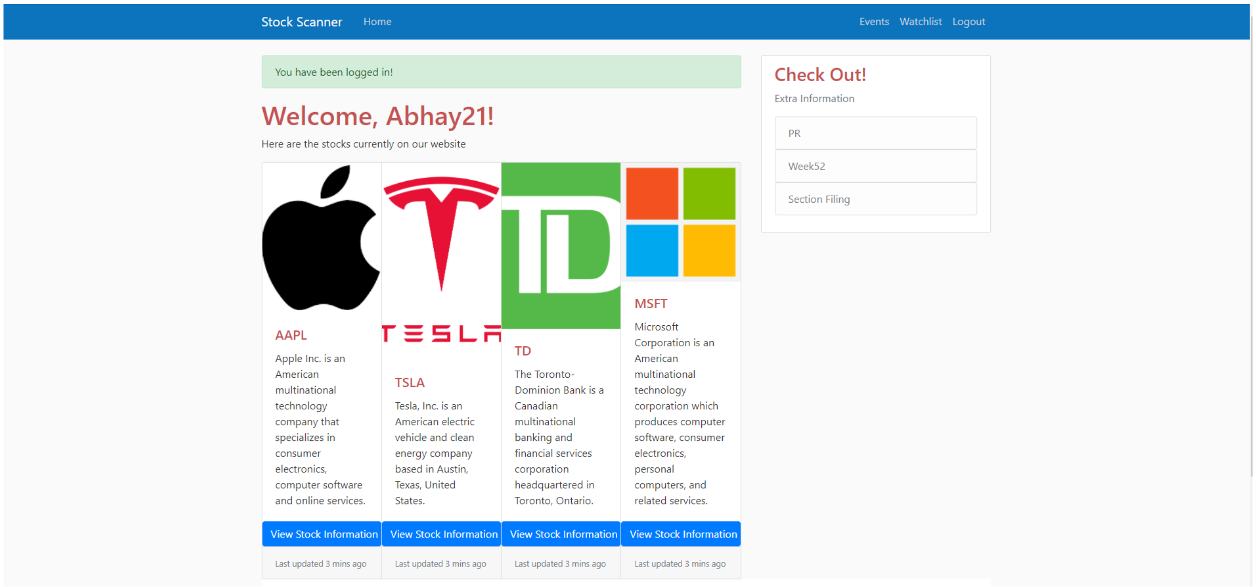
You can put any information here you'd like.

Need an account? Sign Up Here

StockScanner © 2021

If the user inputs a wrong username it flashes them informing that this username does not exist in the database which is correct as well because the current state of the database is like above after successfully updating the password. This page is for any user who might not remember their password and can use this tool to update it.

Show Stocks Page For Private User(URL:http://127.0.0.1:5000/showStocks)



You have been logged in!

Welcome, Abhay21!

Here are the stocks currently on our website

AAPL

Apple Inc. is an American multinational technology company that specializes in consumer electronics, computer software and online services.

Last updated 3 mins ago

TSLA

Tesla, Inc. is an American electric vehicle and clean energy company based in Austin, Texas, United States.

Last updated 3 mins ago

TD

The Toronto-Dominion Bank is a Canadian multinational banking and financial services corporation headquartered in Toronto, Ontario.

Last updated 3 mins ago

MSFT

Microsoft Corporation is an American multinational technology corporation which produces computer software, consumer electronics, personal computers, and related services.

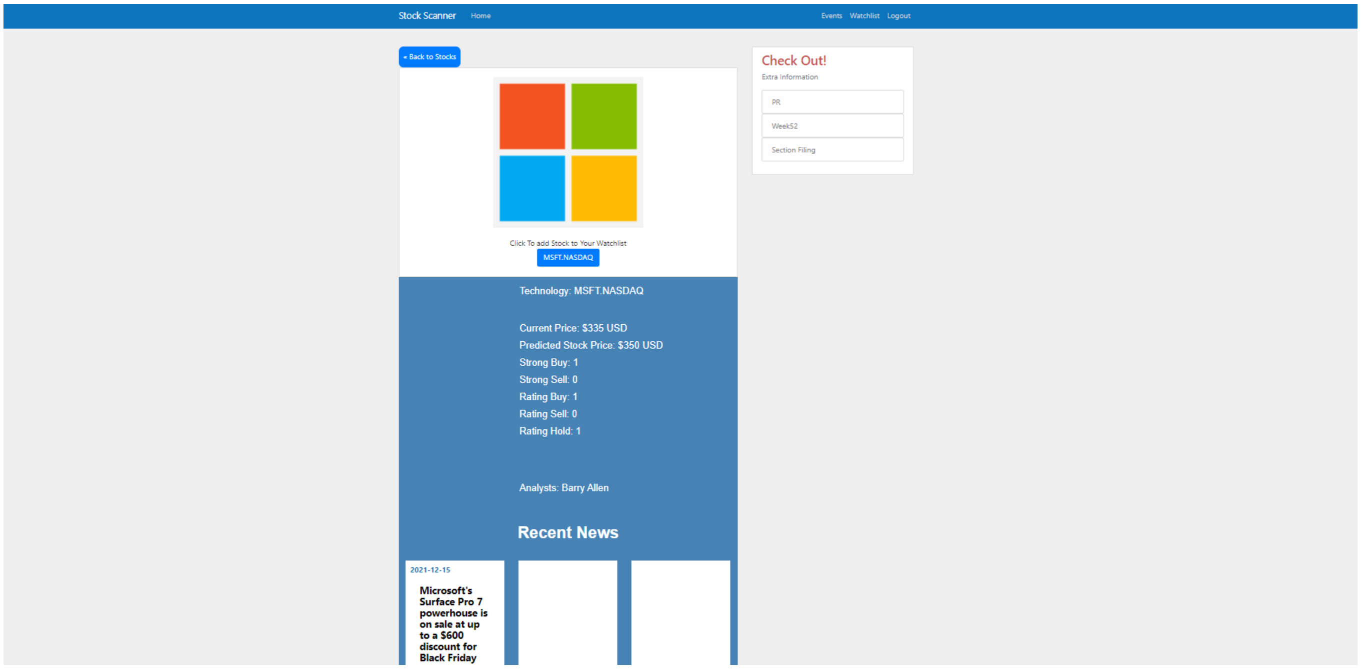
Last updated 3 mins ago

Check Out!

Extra Information

After a successful login, you will be redirected to the show stocks page where all of the stocks exist for the database, note this going through as a private user in which they can view the stocks and add them to their watchlist. This page also greets the user by also having their username on this page. As a user, I am really interested in Microsoft stock so I will click on "View Stock Information" button.

Show Stock Information Page(URL:http://127.0.0.1:5000/stockInformation/MSFT.NASDAQ)



Check Out!

Extra Information

Click To add Stock to Your Watchlist

Technology: MSFT.NASDAQ

Current Price: $335 USD

Predicted Stock Price: $350 USD

Strong Buy: 1

Strong Sell: 0

Rating Buy: 1

Rating Sell: 0

Rating Hold: 1

Analysts: Barry Allen

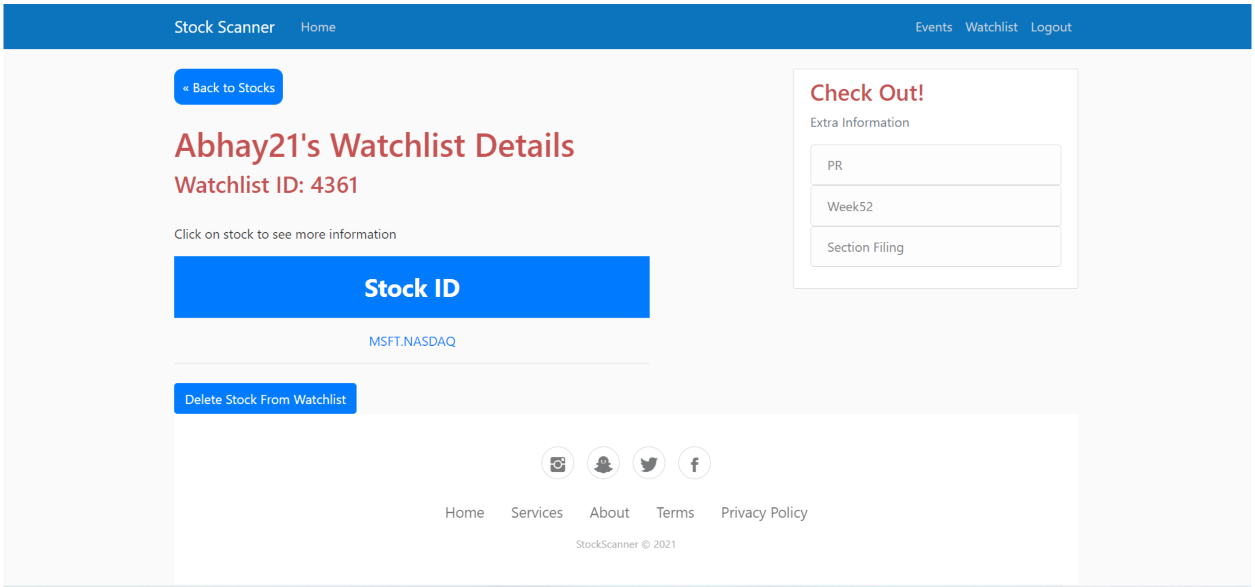
Recent News

2021-12-15

Microsoft's Surface Pro 7 powerhouse is on sale at up to a $600 discount for Black Friday

On this page, the user has quite a few options to explore, the first one being that they can add this stock to their watchlist or look at the attributes of this stock for example strong buy, strong sell, etc. Alongside at the bottom we have recent news about the stock this uses the PR table in the database to display these here. Since I am really keen on tracking this stock I will go ahead and add this stock to my watchlist. If I change my mind I can always go back to the stocks and view others from the top left.

Personal Watchlist Page(URL:http://127.0.0.1:5000/watchlistDetails)



Abhay21's Watchlist Details

Watchlist ID: 4361

Click on stock to see more information

Stock ID
MSFT.NASDAQ

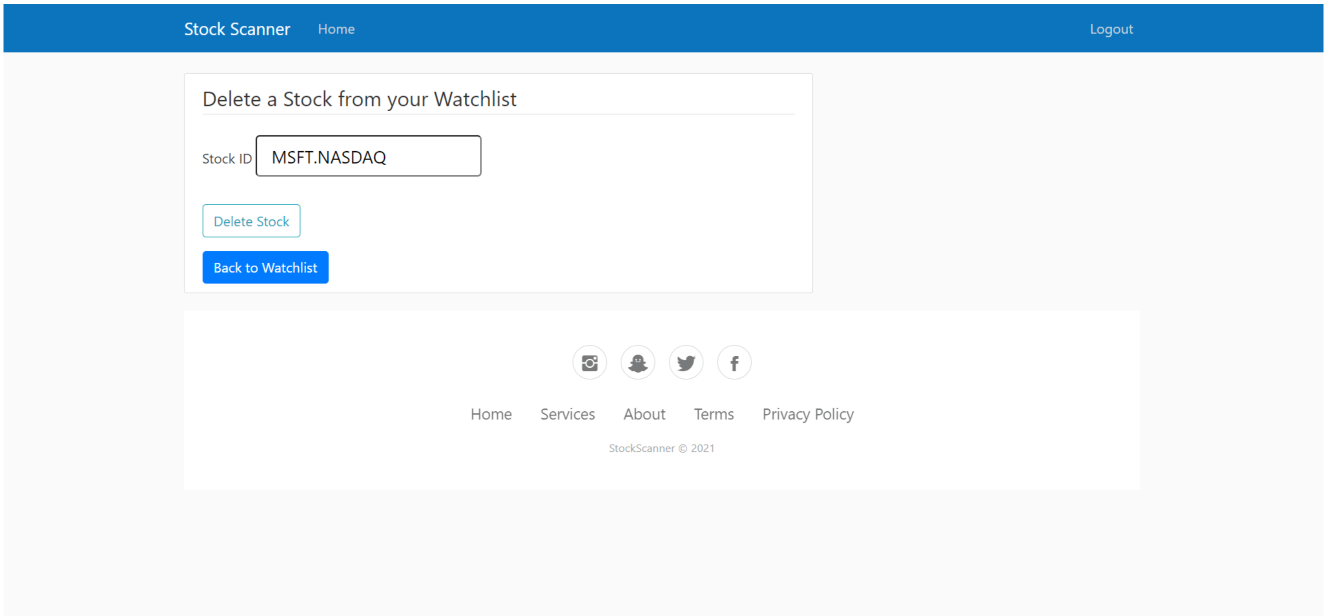
Check Out!

Extra Information

- PR
- Week52
- Section Filing

On this page, you can see the stock id's which also show the exchange to which the stock belongs, this page is unique to you and the watchlist id is assigned to you randomly when you are registering for an account and is saved for you. If you click on the ID it will take you back to the stock detailed page for that specific stock. Or you can go browse other stocks as well by pressing the "Back to Stocks" button from the top left corner. This is a unique page for every user regardless of whether you are a private/professional user. For example, if I am done tracking the stock from my watchlist I can delete it from my watchlist as well.

Delete Stock From Watchlist Page(URL:http://127.0.0.1:5000/deleteStockWatchlist/4361)

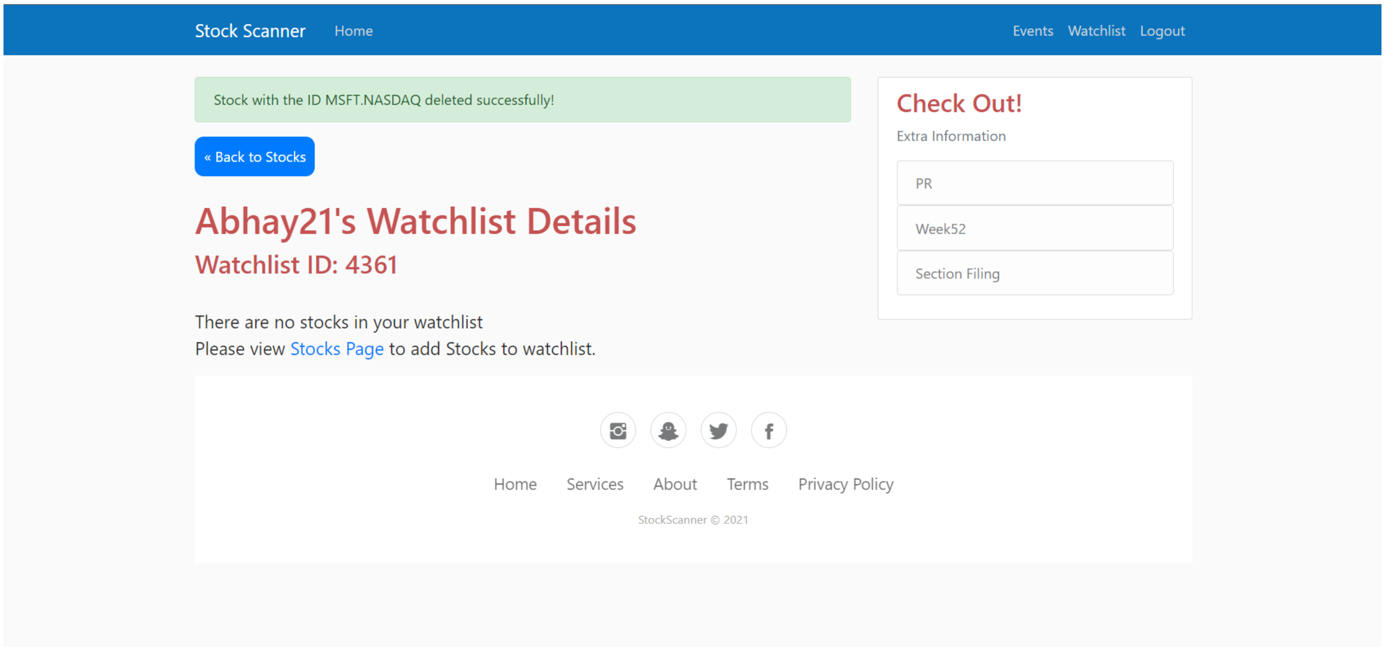


Delete a Stock from your Watchlist

Stock ID: MSFT.NASDAQ

On this page the user will need to input the Stock ID for which they want to remove from the watchlist in this guide there was the only one which was the Microsoft stock so we will press the "Delete Stock" button.

Redirect After Successful Deletion of Stock From Watchlist Page (URL:http://127.0.0.1:5000/watchlistDetails)



Stock with the ID MSFT.NASDAQ deleted successfully!

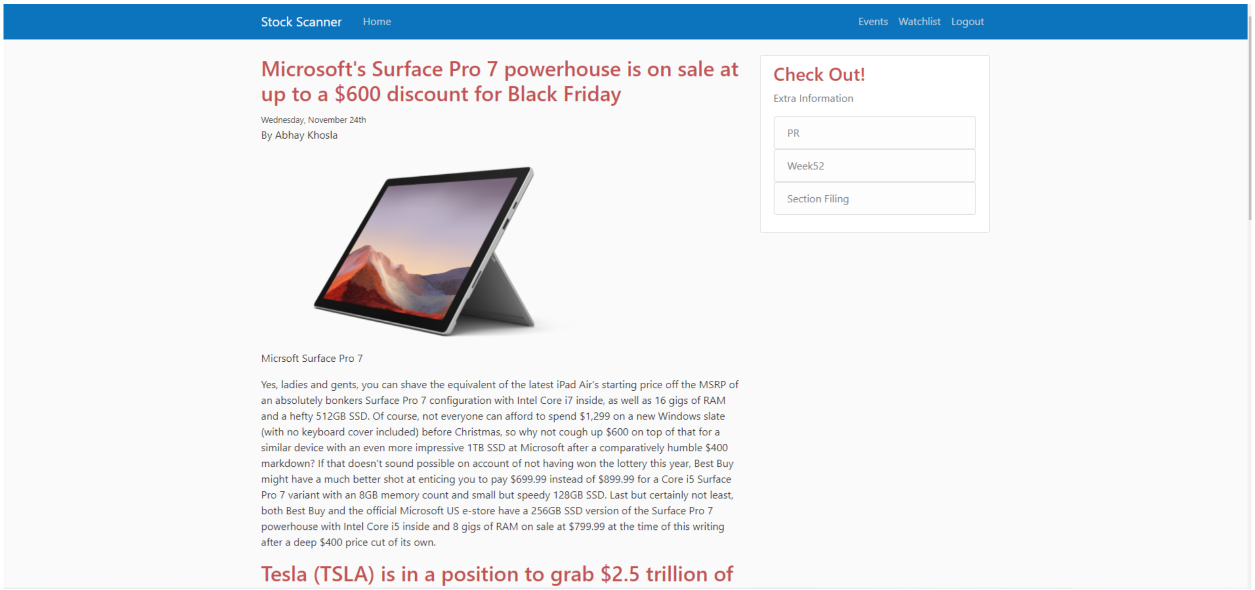
Abhay21's Watchlist Details

Watchlist ID: 4361

There are no stocks in your watchlist
Please view Stocks Page to add Stocks to watchlist.

Informs the user it has been deleted from their watchlist, and then prompts them to go back to the stocks page to view others, and then makes you also want to go explore the right-hand sidebar as well with other details on the website it holds. We will start by exploring the PR extra information.

PR Page (URL:http://127.0.0.1:5000/prDetails)



Microsoft's Surface Pro 7 powerhouse is on sale at up to a $600 discount for Black Friday

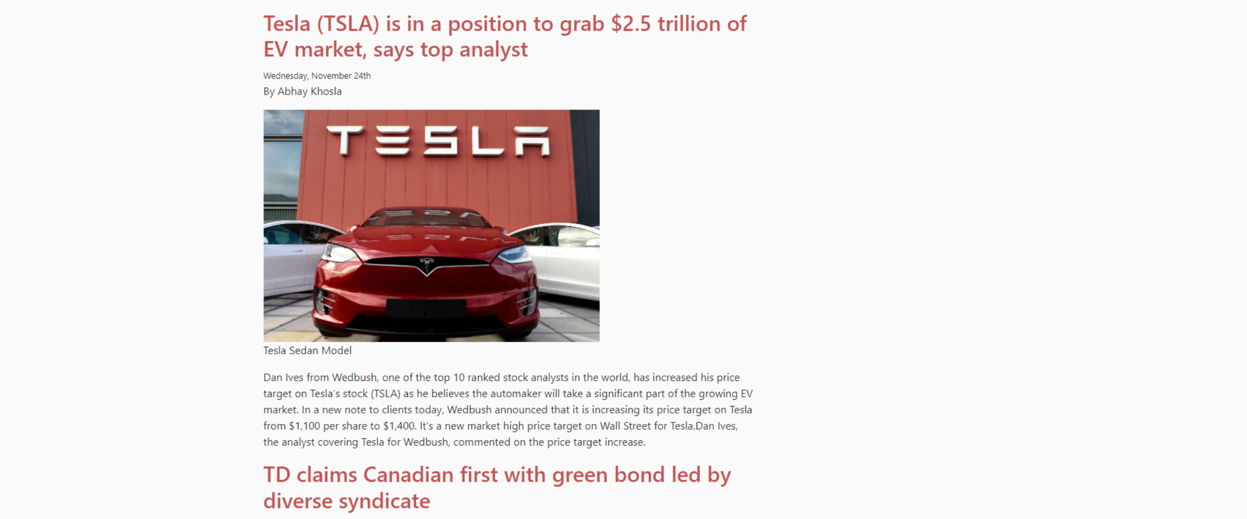
Wednesday, November 24th

By Abhay Khosla

Micrsoft Surface Pro 7

Yes, ladies and gents, you can shave the equivalent of the latest iPad Air's starting price off the MSRP of an absolutely bonkers Surface Pro 7 configuration with Intel Core i7 inside, as well as 16 gigs of RAM and a hefty 512GB SSD. Of course, not everyone can afford to spend $1,299 on a new Windows slate (with no keyboard cover included) before Christmas, so why not cough up $600 on top of that for a similar device with an even more impressive 1TB SSD at Microsoft after a comparatively humble $400 markdown? If that doesn't sound possible on account of not having won the lottery this year, Best Buy might have a much better shot at enticing you to pay $699.99 instead of $899.99 for a Core i5 Surface Pro 7 variant with an 8GB memory count and small but speedy 128GB SSD. Last but certainly not least, both Best Buy and the official Microsoft US e-store have a 256GB SSD version of the Surface Pro 7 powerhouse with Intel Core i5 inside and 8 gigs of RAM on sale at $799.99 at the time of this writing after a deep $400 price cut of its own.

Tesla (TSLA) is in a position to grab $2.5 trillion of



Tesla (TSLA) is in a position to grab $2.5 trillion of EV market, says top analyst

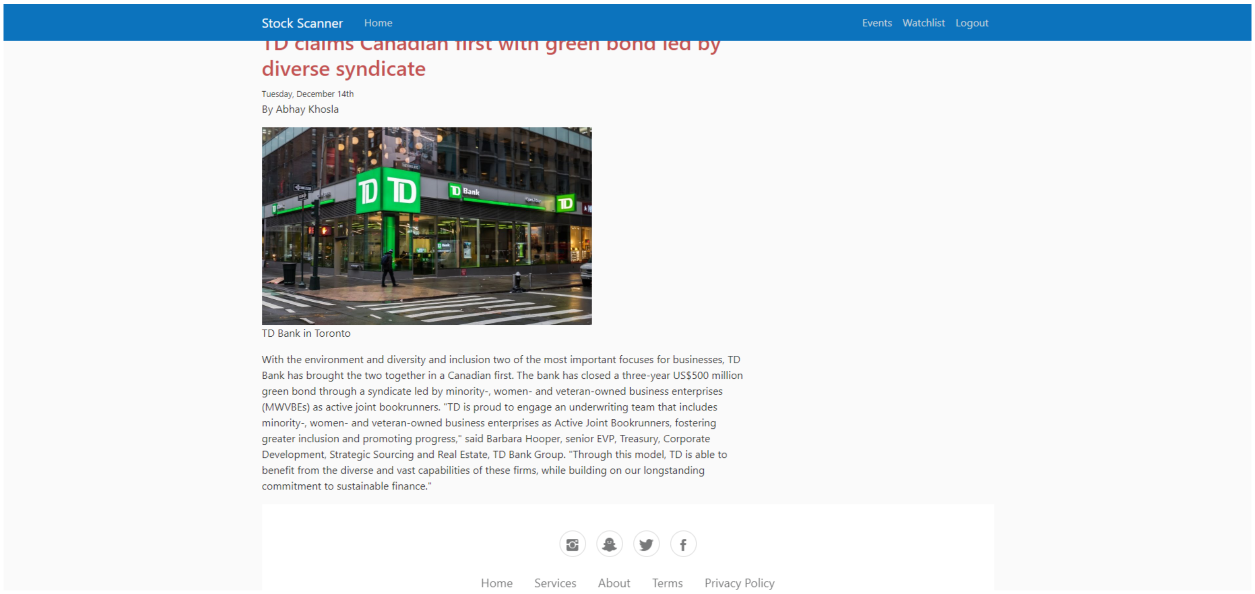
Wednesday, November 24th

By Abhay Khosla

Tesla Sedan Model

Dan Ives from Wedbush, one of the top 10 ranked stock analysts in the world, has increased his price target on Tesla's stock (TSLA) as he believes the automaker will take a significant part of the growing EV market. In a new note to clients today, Wedbush announced that it is increasing its price target on Tesla from $1,100 per share to $1,400. It's a new market high price target on Wall Street for Tesla.Dan Ives, the analyst covering Tesla for Wedbush, commented on the price target increase.

TD claims Canadian first with green bond led by diverse syndicate



TD claims Canadian first with green bond led by diverse syndicate

Tuesday, December 14th

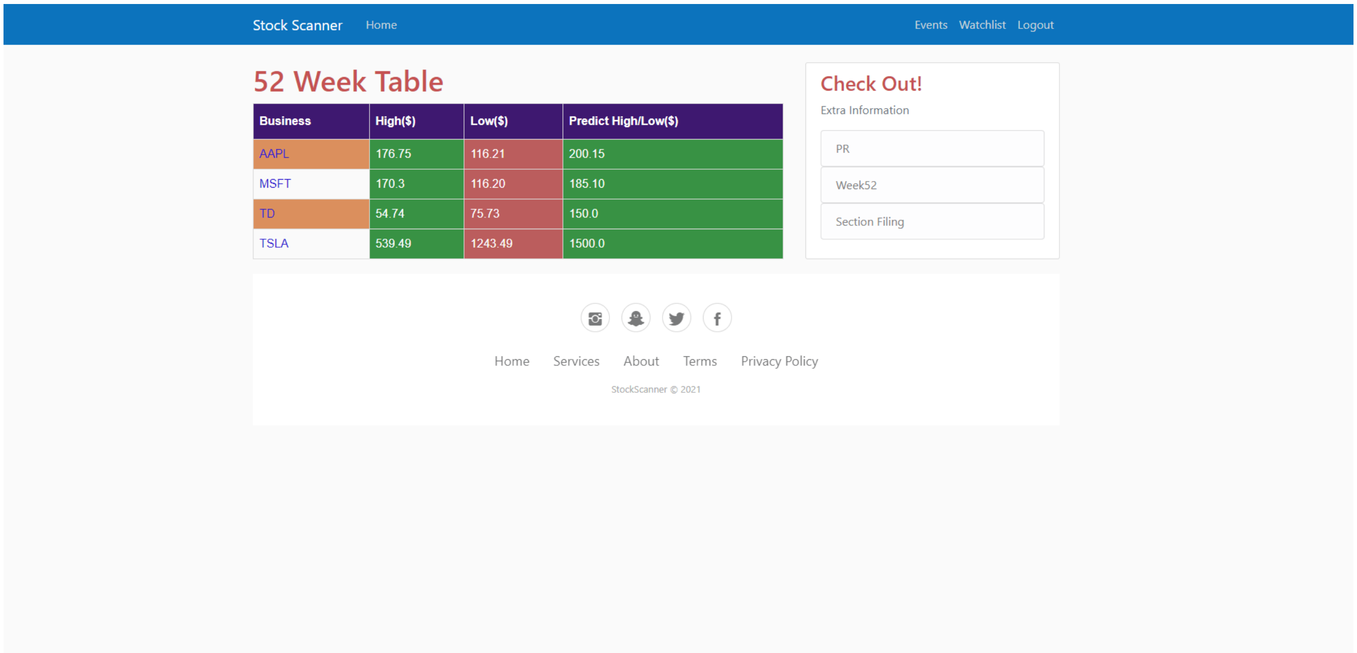
By Abhay Khosla

TD Bank in Toronto

With the environment and diversity and inclusion two of the most important focuses for businesses, TD Bank has brought the two together in a Canadian first. The bank has closed a three-year US$500 million green bond through a syndicate led by minority-, women- and veteran-owned business enterprises (MWVBEs) as active joint bookrunners. "TD is proud to engage an underwriting team that includes minority-, women- and veteran-owned business enterprises as Active Joint Bookrunners, fostering greater inclusion and promoting progress," said Barbara Hooper, senior EVP, Treasury, Corporate Development, Strategic Sourcing and Real Estate, TD Bank Group. "Through this model, TD is able to benefit from the diverse and vast capabilities of these firms, while building on our longstanding commitment to sustainable finance."

On this page, the users can catch up on recent news articles about all of the stocks in one place which is easier to do rather than going one by one and clicking on the showing stocks page. These articles and images need to be manually entered by the admin which is keeping it up to date daily with brand news, for this demo we only have some recent news but if this was fully developed as a website application to the public we would have someone monitoring these pages.

Week52 Page (URL:http://127.0.0.1:5000/week52Details)



52 Week Table

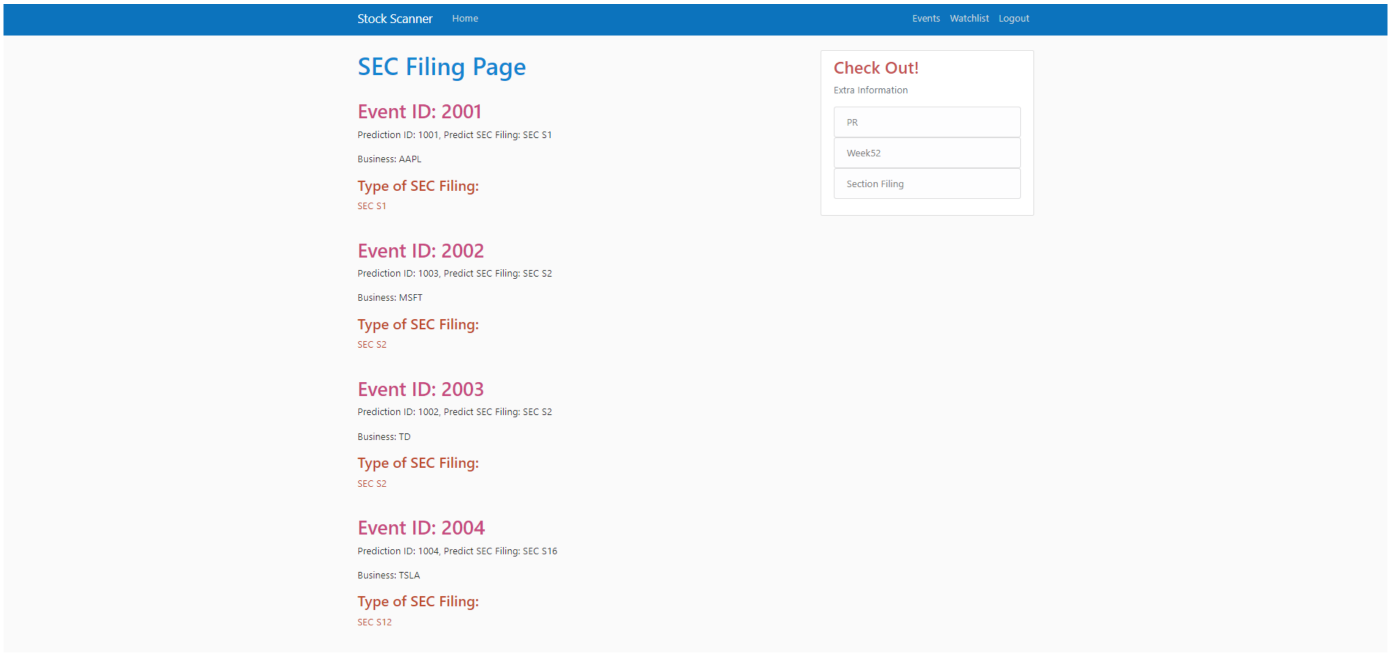
Business	High($)	Low($)	Predict High/Low($)
AAPL	176.75	116.21	200.15
MSFT	170.3	116.20	185.10
TD	54.74	75.73	150.0
TSLA	539.49	1243.49	1500.0

Check Out!

Extra Information

On this page, users can look at the stocks and their yearly highs or lows and the prediction of the highs/lows made by the analysts who sign up as private users and inform the admin about their predictions about the stock for the next year. The future implementation would be to have this automatically predict these values but for now, this is done manually by the database administrators with the help of the analysts.

Section Filing Page (URL:http://127.0.0.1:5000/showSecFiling)



SEC Filing Page

Event ID: 2001

Prediction ID: 1001, Predict SEC Filing: SEC S1

Business: AAPL

Type of SEC Filing:

SEC S1

Event ID: 2002

Prediction ID: 1003, Predict SEC Filing: SEC S2

Business: MSFT

Type of SEC Filing:

SEC S2

Event ID: 2003

Prediction ID: 1002, Predict SEC Filing: SEC S2

Business: TD

Type of SEC Filing:

SEC S2

Event ID: 2004

Prediction ID: 1004, Predict SEC Filing: SEC S16

Business: TSLA

Type of SEC Filing:

SEC S12

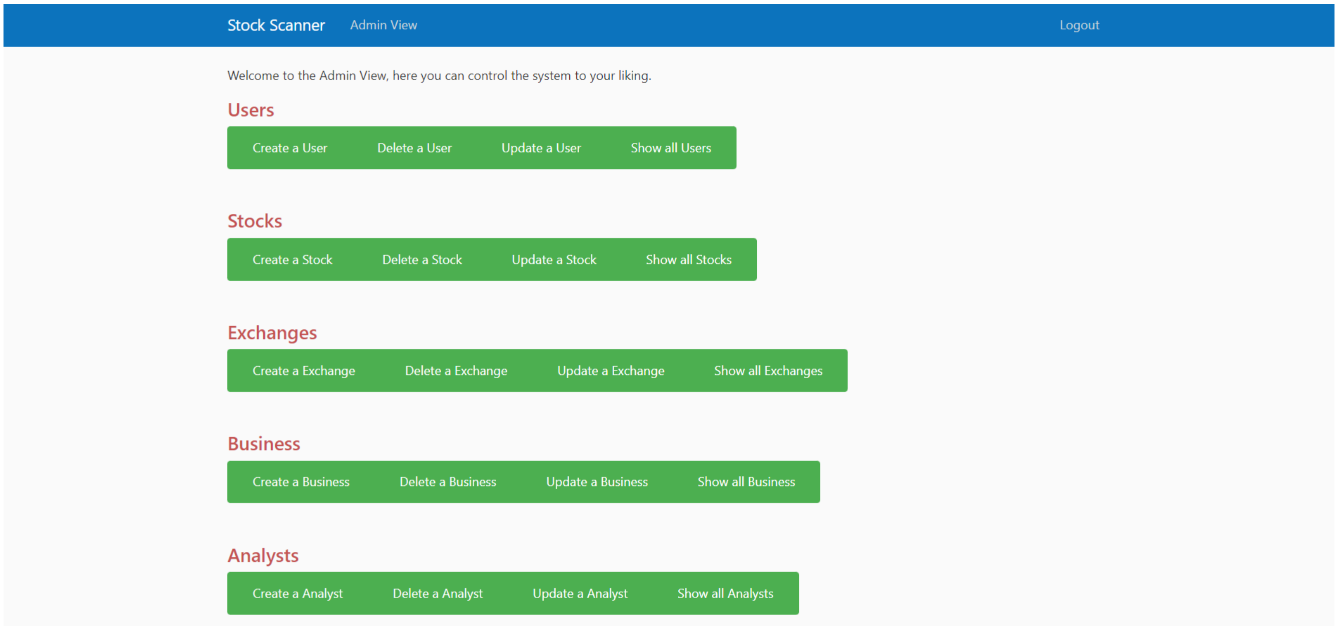
Check Out!

Extra Information

On this page, the users can see the SEC filing for each event and what type it is, this is similar to the Week 52 page where it is manually being entered in by the database administrators but with the help of the analysts researching the event and informing us to make these changes in the database. This is also for the future where the AI will predict all of these things and we will not need to make these changes. But this is for later expansion after the semester ends and we gain more industry experience.

We will now press the logout which will redirect us to the Login page and where we will walk through as an admin:

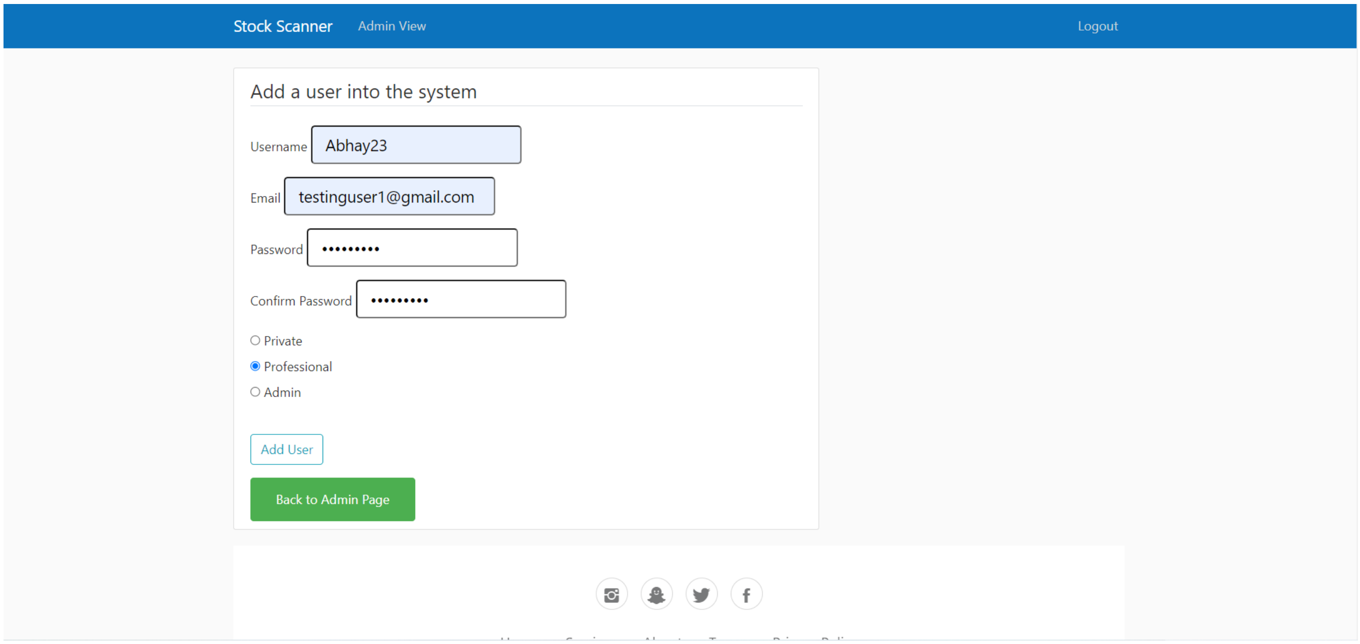
Admin Home Page(URL:http://127.0.0.1:5000/showAdminView)



Welcome to the Admin View, here you can control the system to your liking.

As an admin, you really don't have many viewing options but do have more accessibility to the database from the web interface where you can create stocks, exchanges, businesses, users, and analysts. The navbar is pretty empty with the option to go log out or go to the homepage which is the admin view.

Admin Create User Page(URL:http://127.0.0.1:5000/addUserAdmin)



Add a user into the system

Username Abhay23

Email testinguser1@gmail.com

Password •••••••••

Confirm Password •••••••••

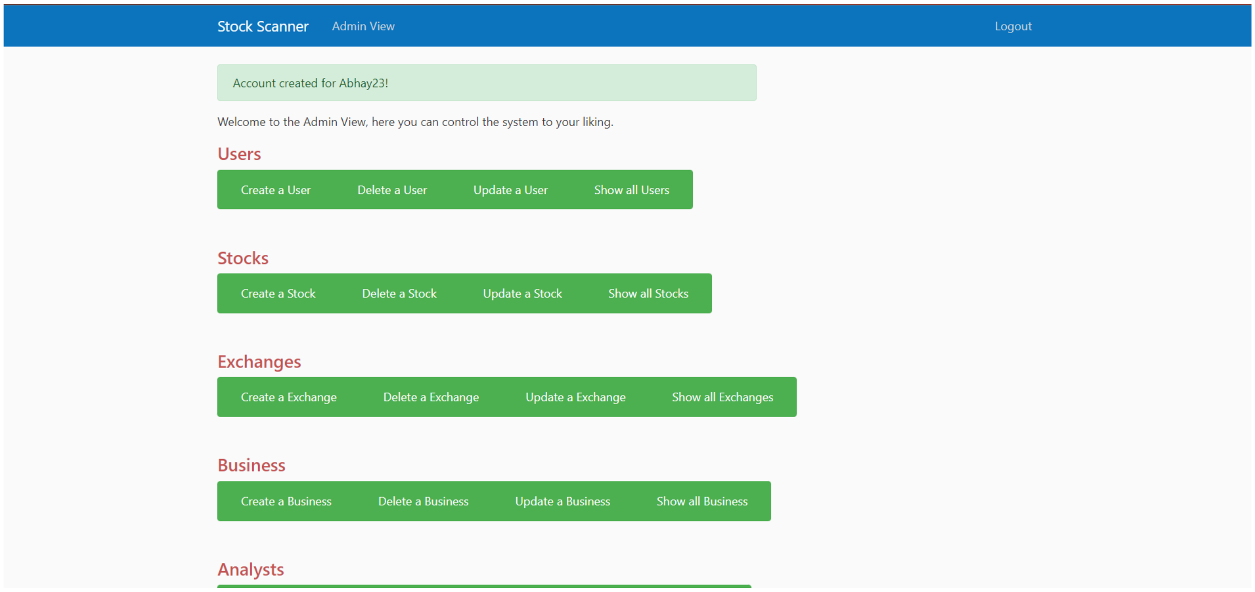
○ Private

● Professional

○ Admin

Here we are creating a brand new user straight from the website with no help from the Postman app, this will redirect us back to the homepage if the user has successfully been created. If we were to use the same username it will flash at us that this username has been taken please choose another one.

Redirect to Admin View Page After Creating a User(URL: http://127.0.0.1:5000/showAdminView)

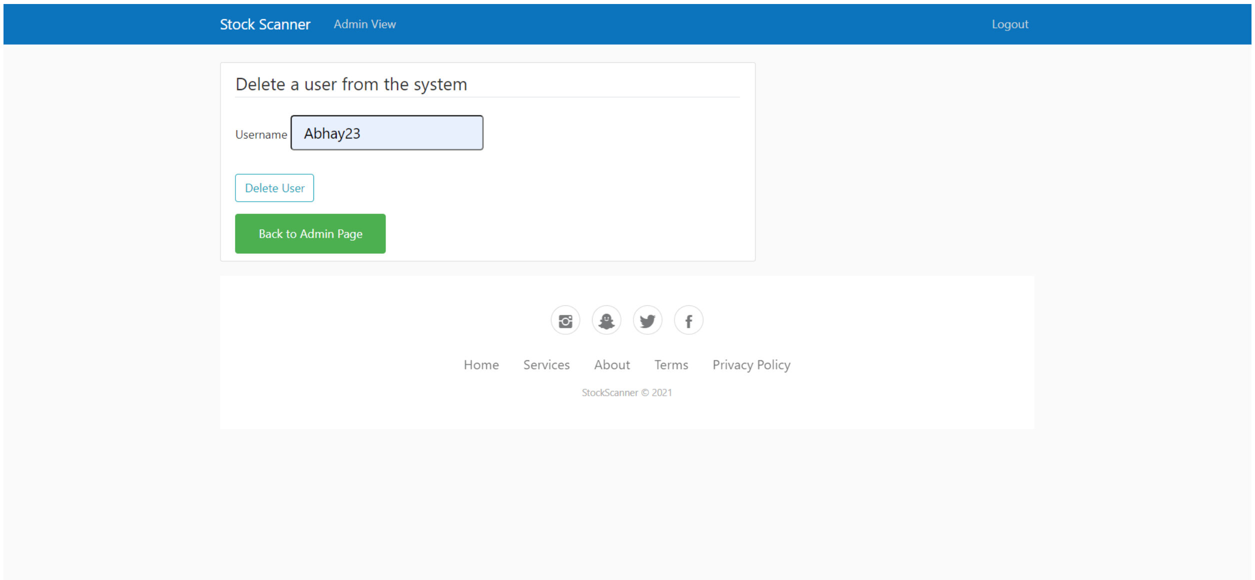


Account created for Abhay23!

Welcome to the Admin View, here you can control the system to your liking.

The message is flashed to the admin notifying them that this has action has been completed and we can continue from here to make other changes to the database. For the other future changes, we will just show the forms and not show the image of us redirecting back to the admin view as it will become redundant.

Admin Delete User Page(URL:http://127.0.0.1:5000/deleteUserAdmin)

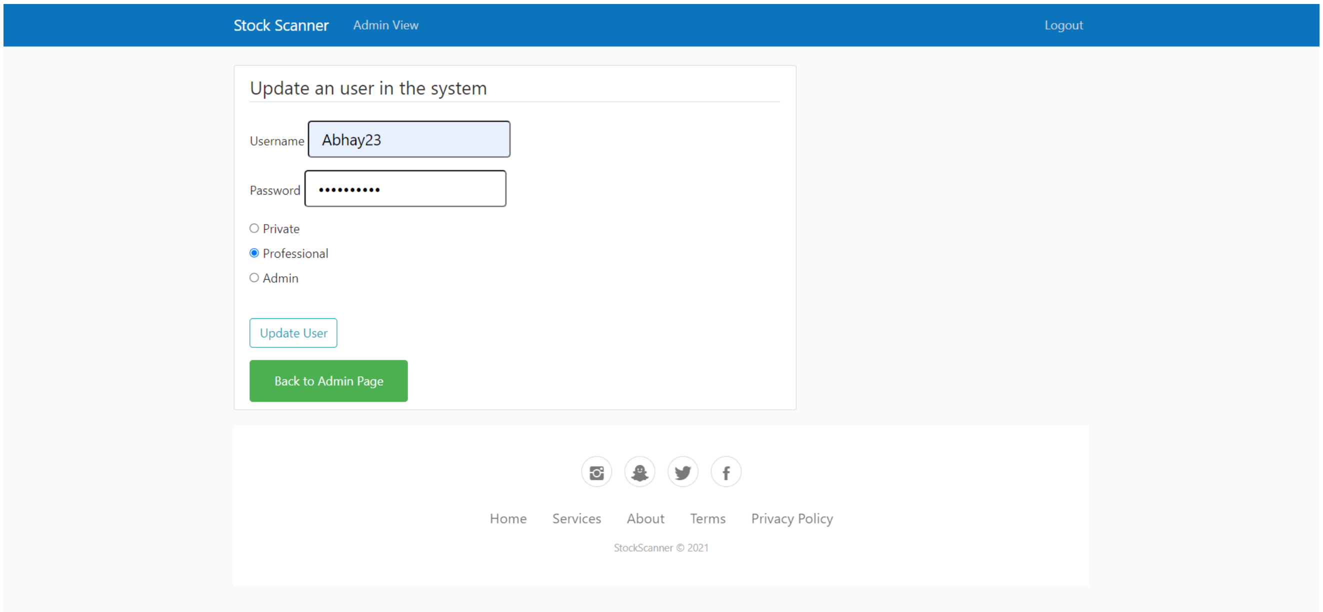


Delete a user from the system

Username Abhay23

Similar to adding the user this uses the username which is a primary key in the database, using this username the Admin can delete the user, for this demo we will just leave this user in as we will need it later to demonstrate a feature for the professional user.

Admin Update User Page(URL:http://127.0.0.1:5000/updateUserAdmin)



Update an user in the system

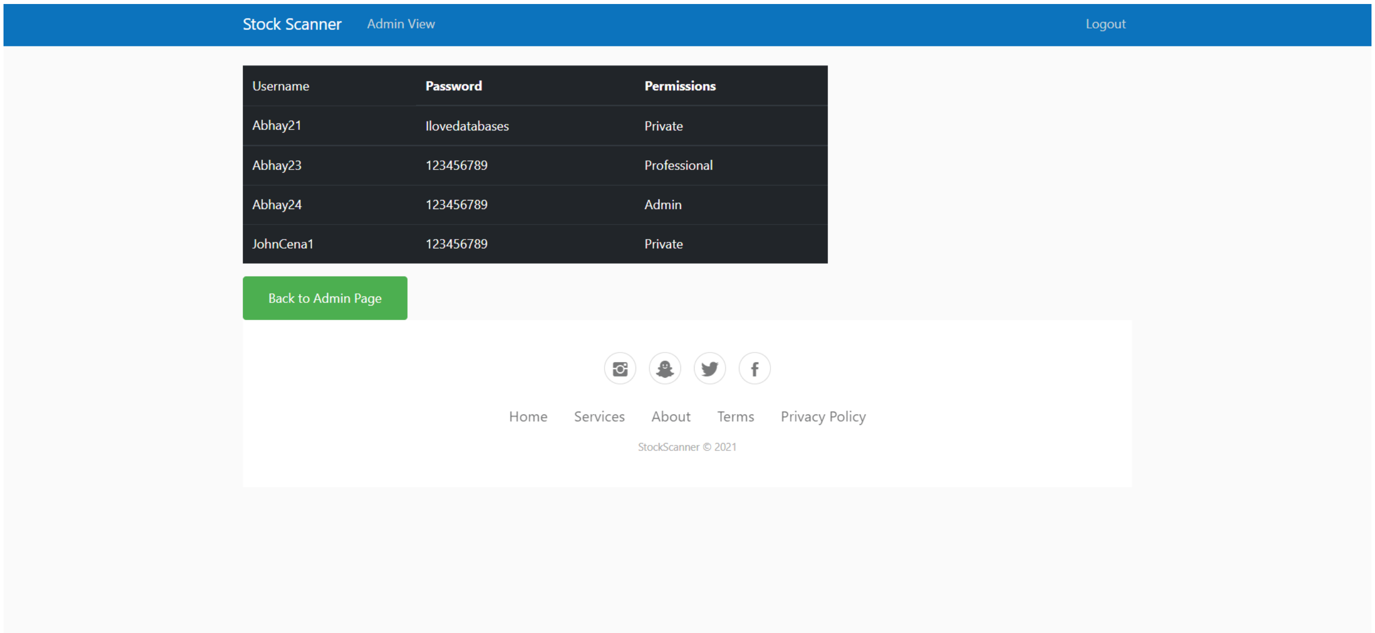
Username: Abhay23

Password: ••••••••••

○ Private
● Professional
○ Admin

On this page the Admin can update a user, the only thing which cannot be updated is the username since it’s restricted but the password or the role type can be updated. This is to ensure that the username does not change and the credentials are not messed up.

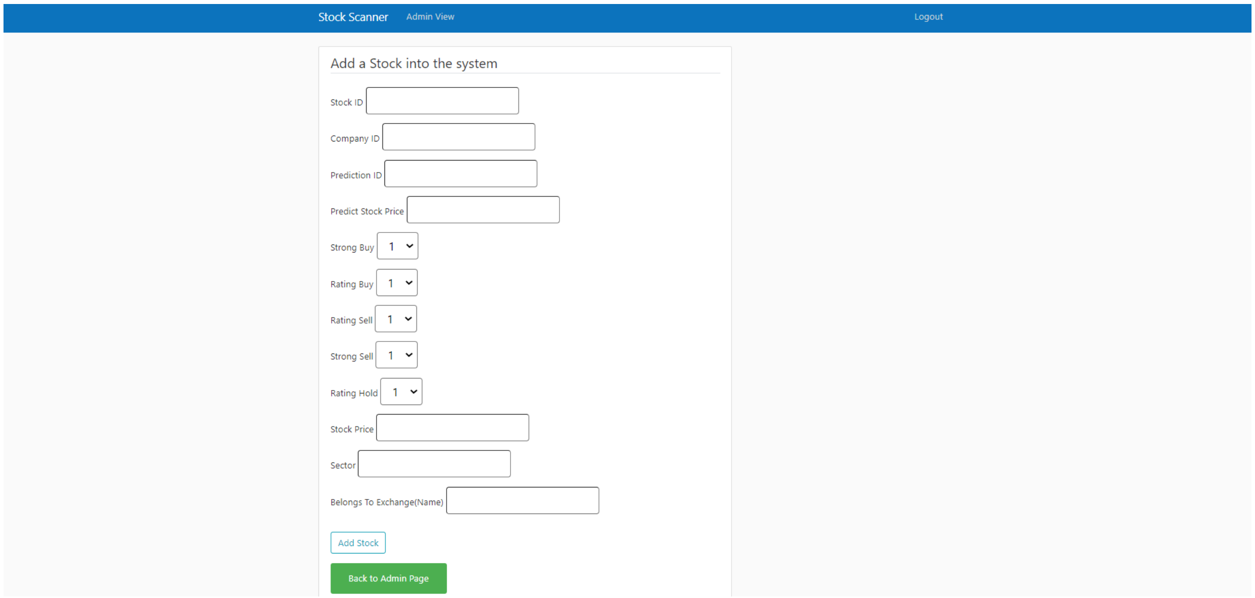
Show All Users Page(URL:http://127.0.0.1:5000/usersall)



Username	Password	Permissions
Abhay21	Ilovedatabases	Private
Abhay23	123456789	Professional
Abhay24	123456789	Admin
JohnCena1	123456789	Private

This page gives a little preview of what the database would look like instead, we just display all of the current users in the system in a tabular form which is a table. Also an option to go back to the admin view and page to make other changes.

Admin Create Stock Page(URL:http://127.0.0.1:5000/addStockAdmin)



Add a Stock into the system

Stock ID

Company ID

Prediction ID

Predict Stock Price

Strong Buy 1

Rating Buy 1

Rating Sell 1

Strong Sell 1

Rating Hold 1

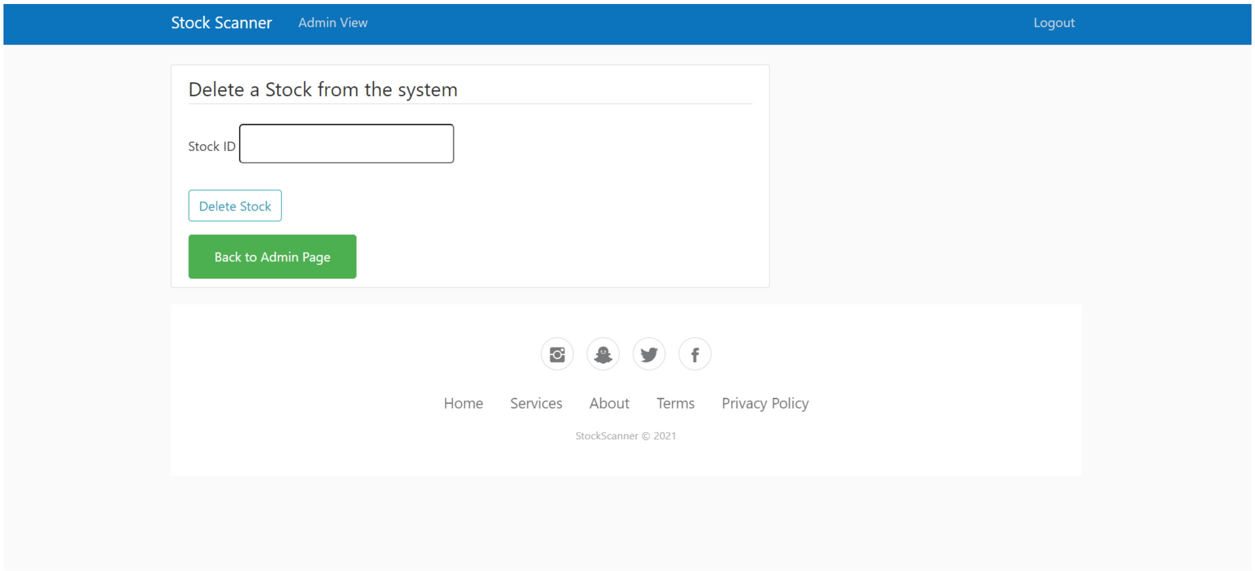
Stock Price

Sector

Belongs To Exchange(Name)

Here the admin can insert a stock into the system by filling out this form and inputting all correct fields, there are some restrictions on these fields that should be noted. The stock id should be in the format of the ticker name and the exchange combined together, for example, MSFT.NASDAQ and then the company id will just be four letters long MSFT in this case. The prediction ID has to be 4 string characters as well no more than that. Before adding the stock you need to make sure the company(business) exists in the database or else it will not work. The belongs to is inputting the exchange name which should also exist before adding a stock into the database.

Admin Delete Stock Page(URL:http://127.0.0.1:5000/deleteStockAdmin)

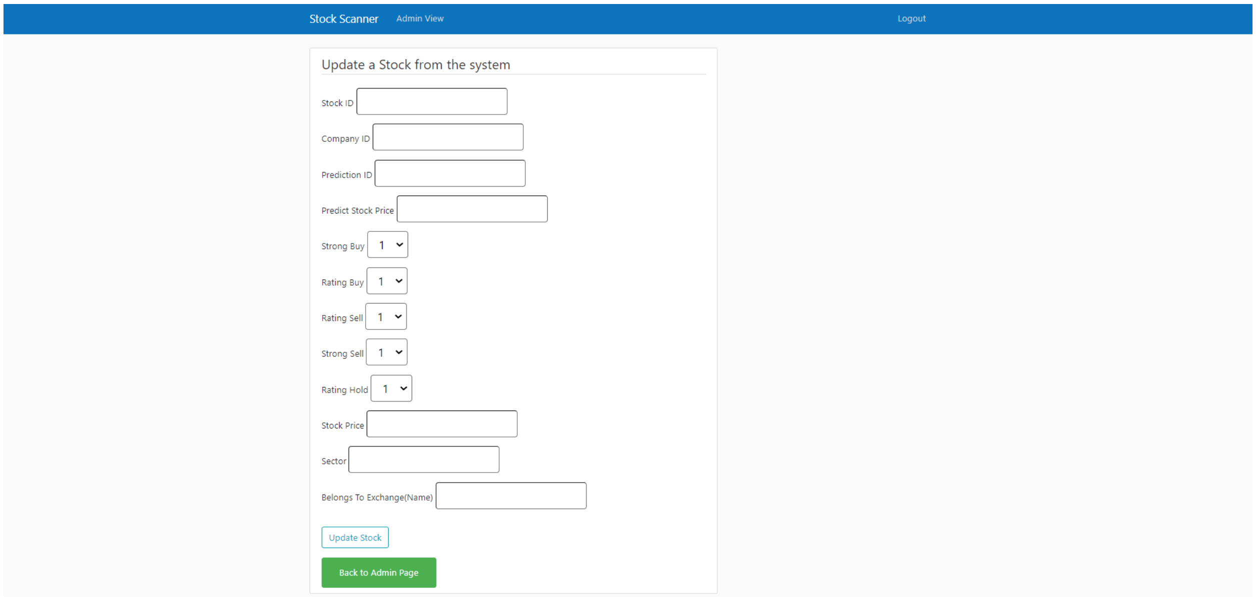


Delete a Stock from the system

Stock ID

Here we will need the stock id again if the admin wants to remove the stock from the database they will need to input the id in the format mentioned above.

Admin Update Stock Page(URL:http://127.0.0.1:5000/updateStockAdmin)



Update a Stock from the system

Stock ID

Company ID

Prediction ID

Predict Stock Price

Strong Buy 1

Rating Buy 1

Rating Sell 1

Strong Sell 1

Rating Hold 1

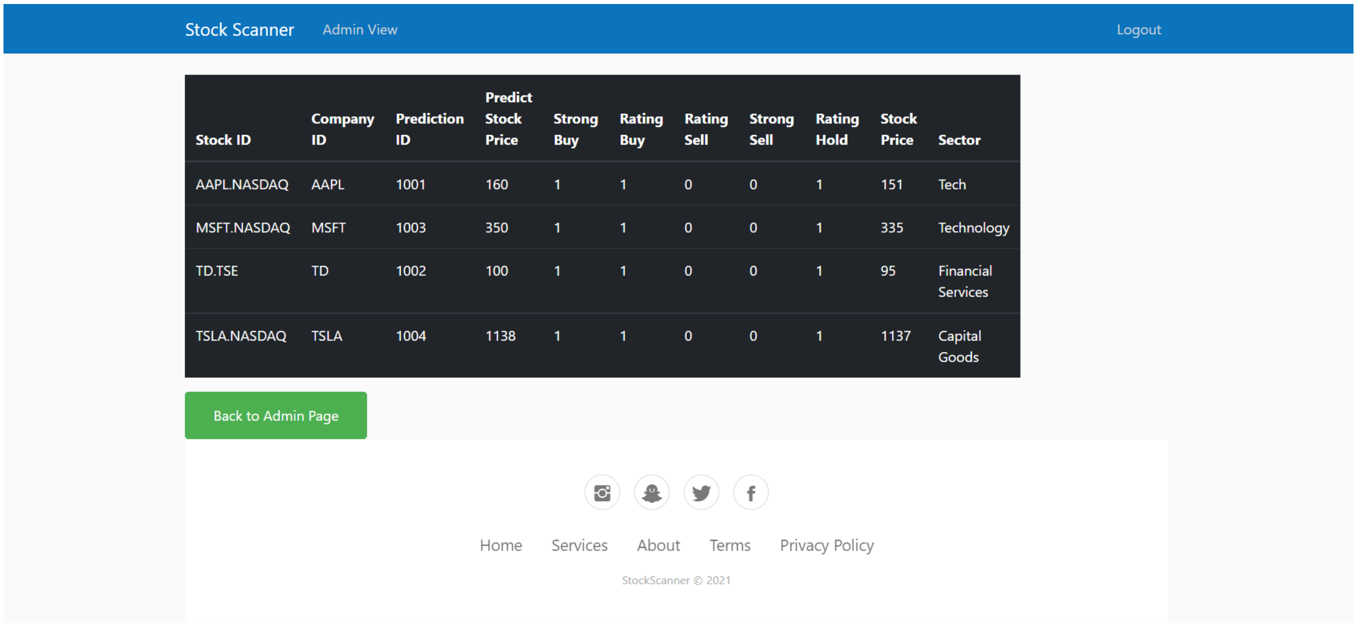
Stock Price

Sector

Belongs To Exchange(Name)

Same restrictions are applied as if you were adding a stock cannot change the stock id since that is being used to determine the other details, same with the company id and prediction id as they are being linked to other tables and the exchange name the belongs to field.

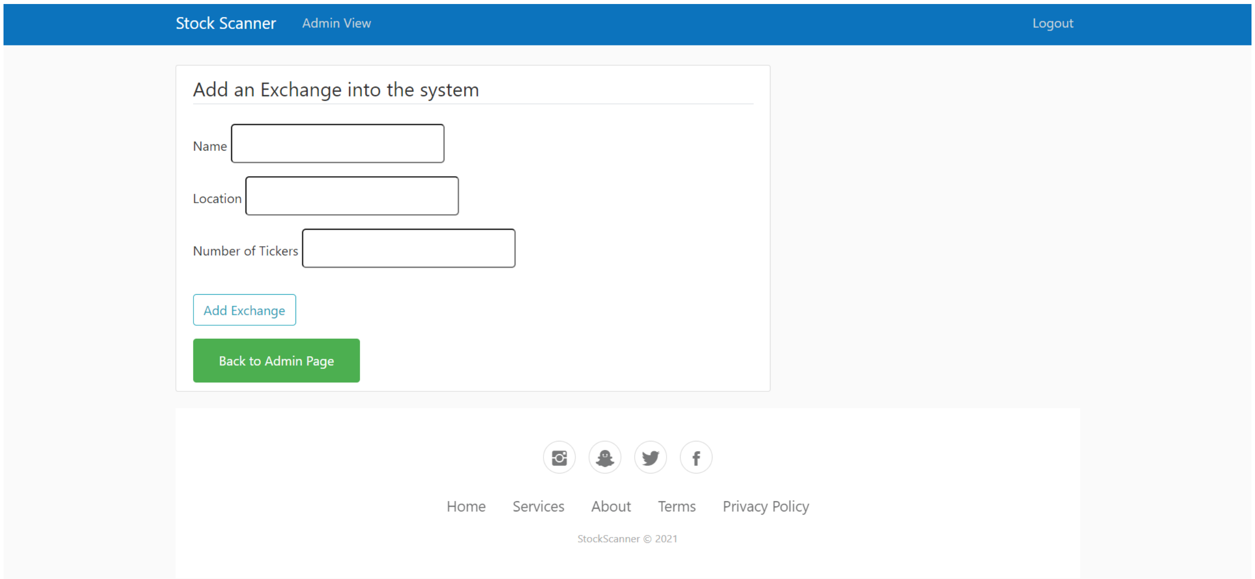
Admin Show All Stocks Page(URL:http://127.0.0.1:5000/showStockAdmin)



Stock ID	Company ID	Prediction ID	Predict Stock Price	Strong Buy	Rating Buy	Rating Sell	Strong Sell	Rating Hold	Stock Price	Sector
AAPL.NASDAQ	AAPL	1001	160	1	1	0	0	1	151	Tech
MSFT.NASDAQ	MSFT	1003	350	1	1	0	0	1	335	Technology
TD.TSE	TD	1002	100	1	1	0	0	1	95	Financial Services
TSLA.NASDAQ	TSLA	1004	1138	1	1	0	0	1	1137	Capital Goods

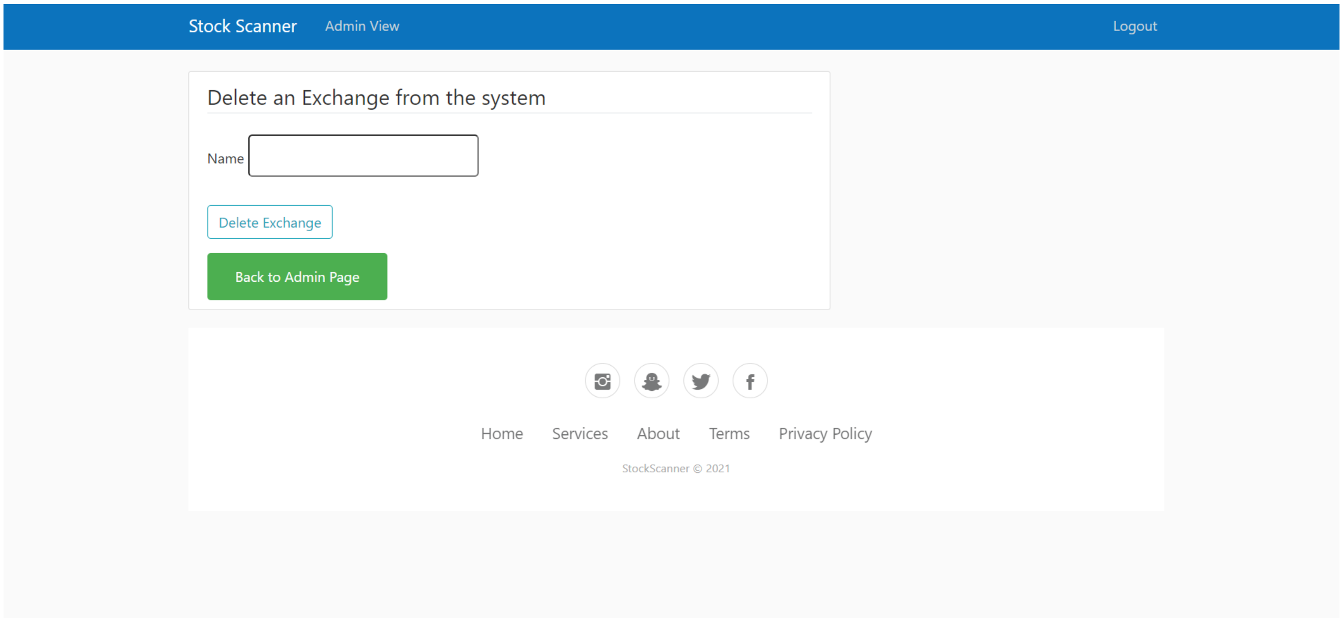
Here the admin gets an insight onto the database, with all of the stocks displayed in a table format similar to how it would look in MySQL command-line client.

Admin Create Exchange Page(URL:http://127.0.0.1:5000/addExchangeAdmin)



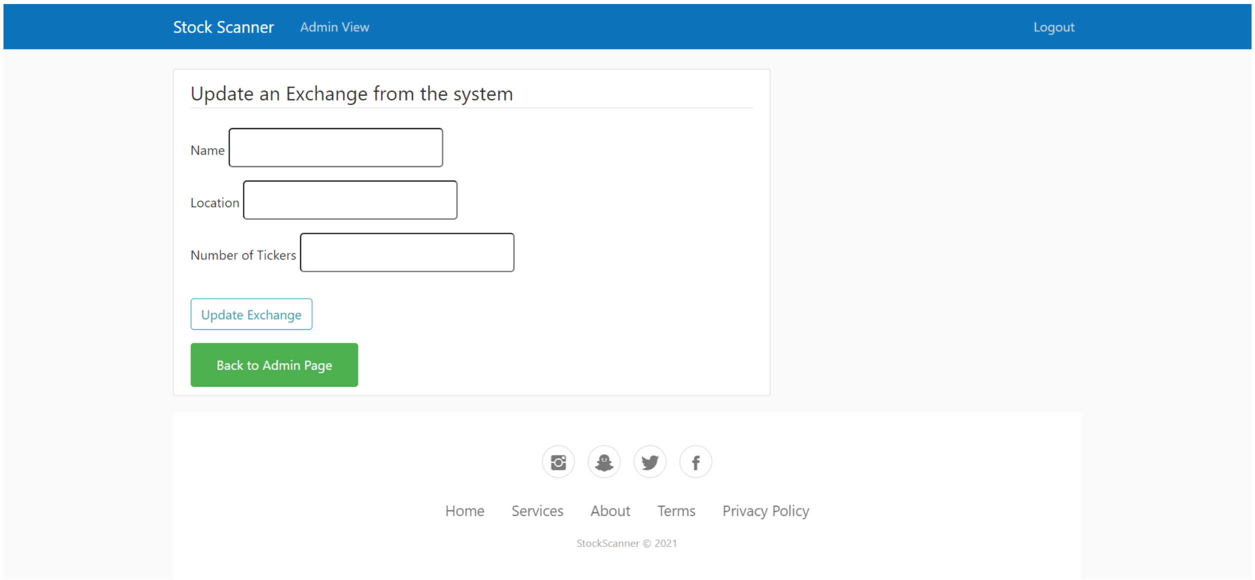
This is similar to other creating pages that the admin has access to, here the name of the exchange cannot be over 25 characters and the location as well for the exchange. The number of tickers has to be an integer and no doubles.

Admin Delete Exchange Page(URL:http://127.0.0.1:5000/deleteExchangeAdmin)



Using the name of the exchange can be deleted from the database as the admin has the control to do this.

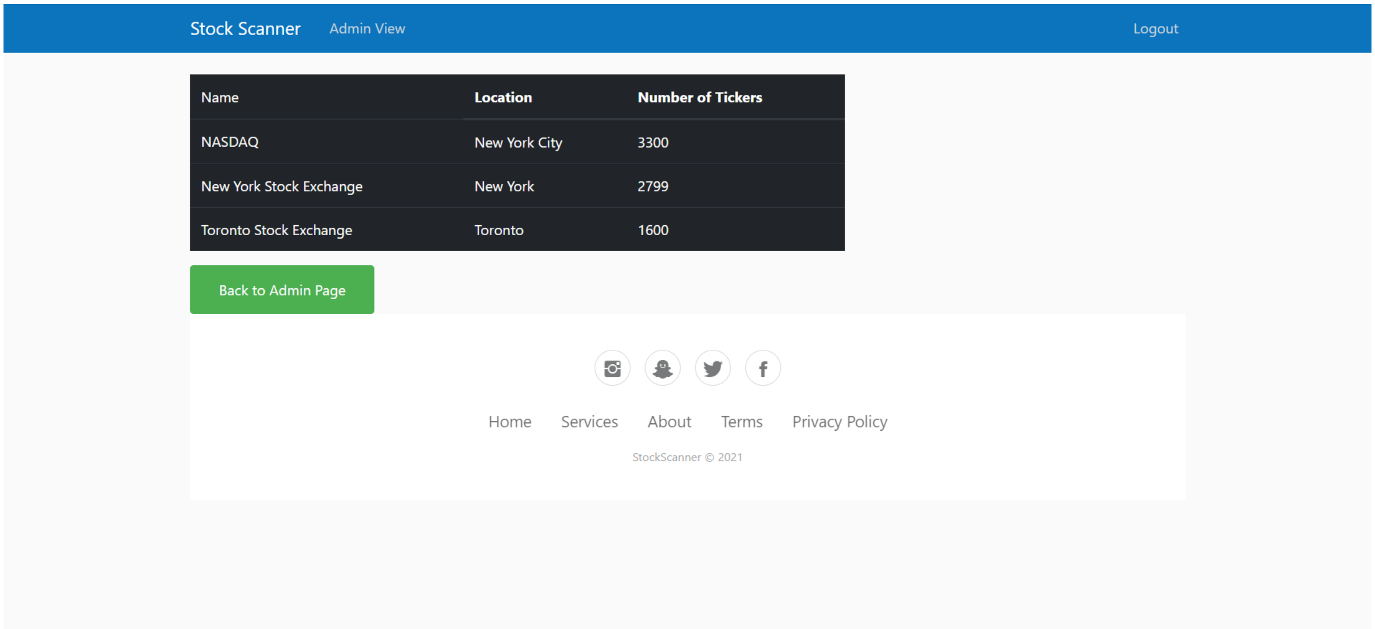
Admin Update Exchange Page(URL:http://127.0.0.1:5000/updateExchangeAdmin)



Update an Exchange from the system

The name cannot be updated for the exchange but other things can be on this page.

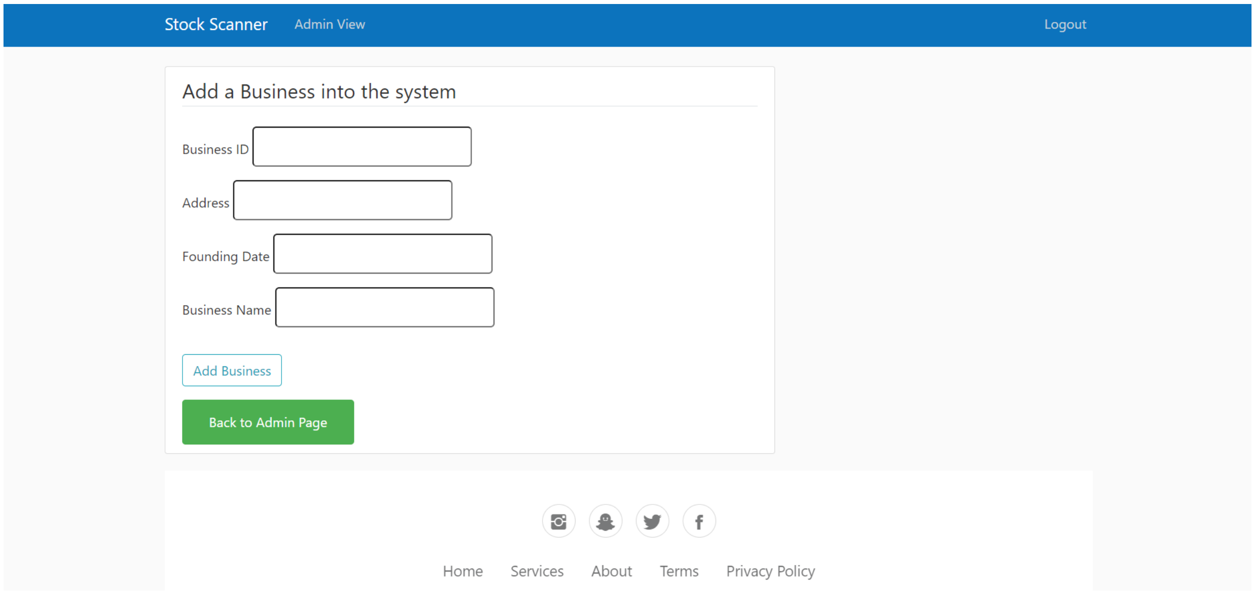
Show All Exchanges Page(URL:http://127.0.0.1:5000/showExchangeAdmin)



Name	Location	Number of Tickers
NASDAQ	New York City	3300
New York Stock Exchange	New York	2799
Toronto Stock Exchange	Toronto	1600

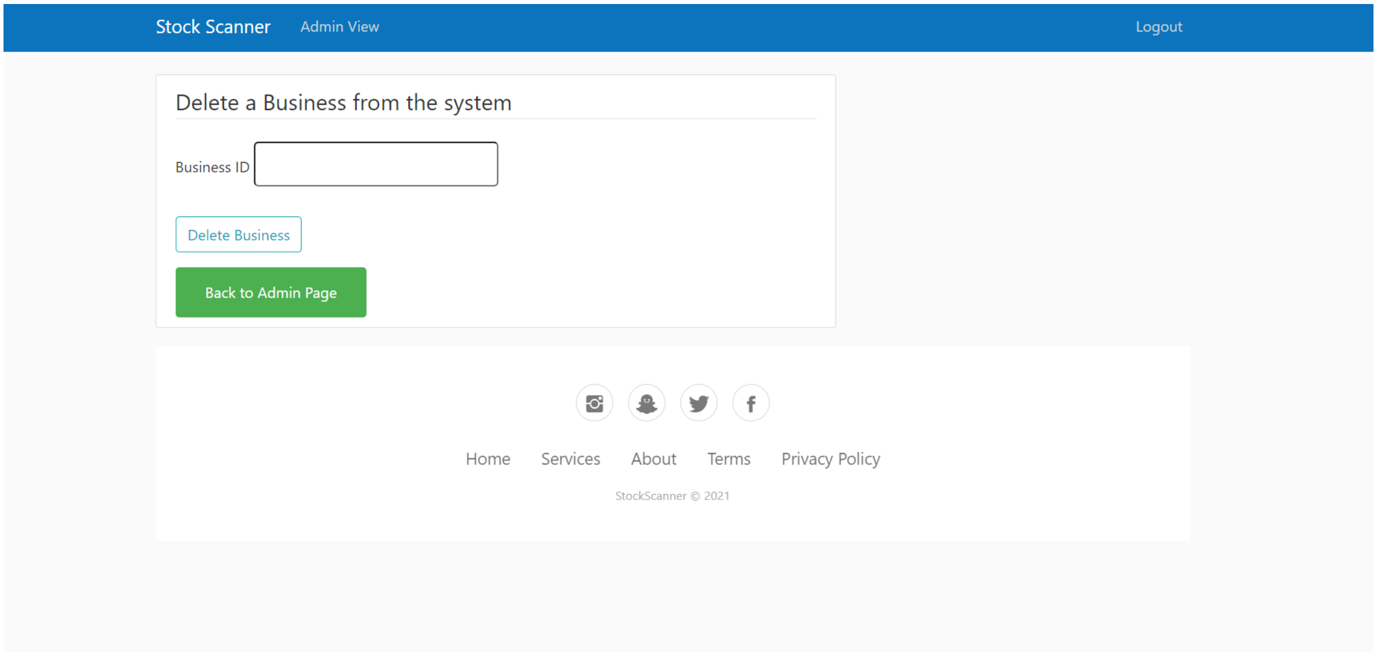
Here the admin gets an insight onto the database, with all of the exchanges displayed in a table format similar to how it would look in MySQL command-line client.

Admin Create Business Page(URL:http://127.0.0.1:5000/addBusinessAdmin)



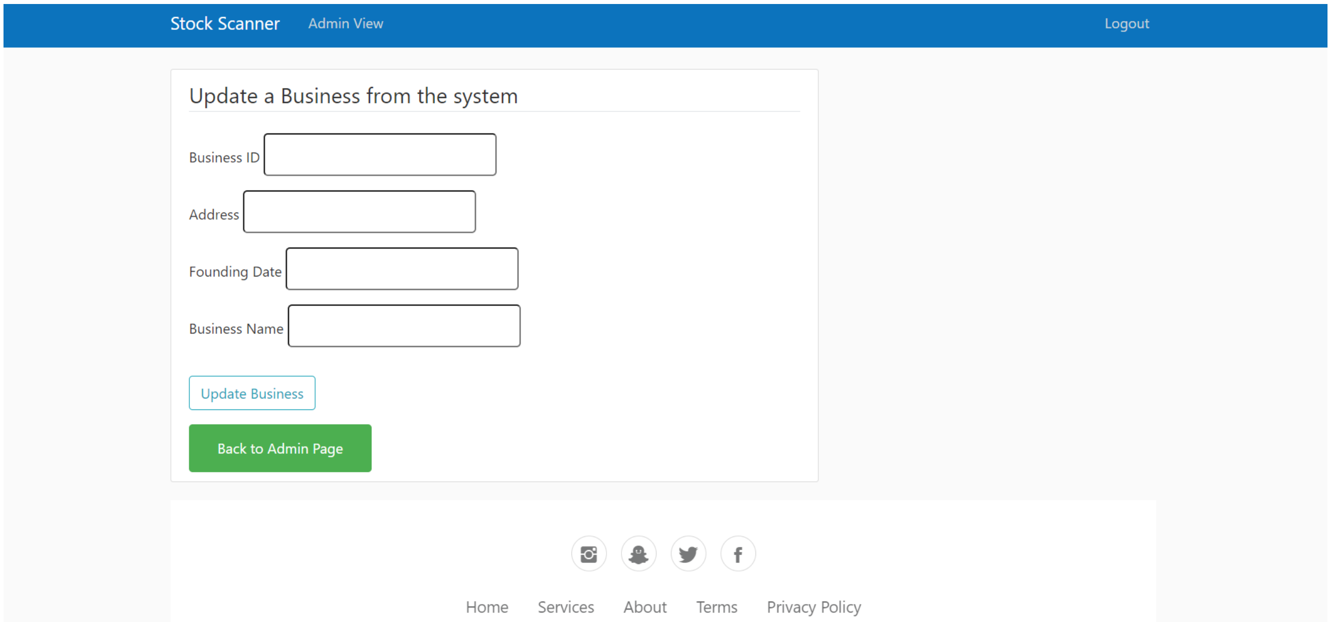
On this page, the admin can create a business with some restrictions on here as well, with the business id only being 4 characters long, the address cannot be longer then 45 characters and the founding date and business name all need to be 25 characters long and no more than that.

Admin Delete Business Page(URL:http://127.0.0.1:5000/deleteBusinessAdmin)



Here the admin can delete the business by its id and need to make sure it’s exactly the same which is in the database.

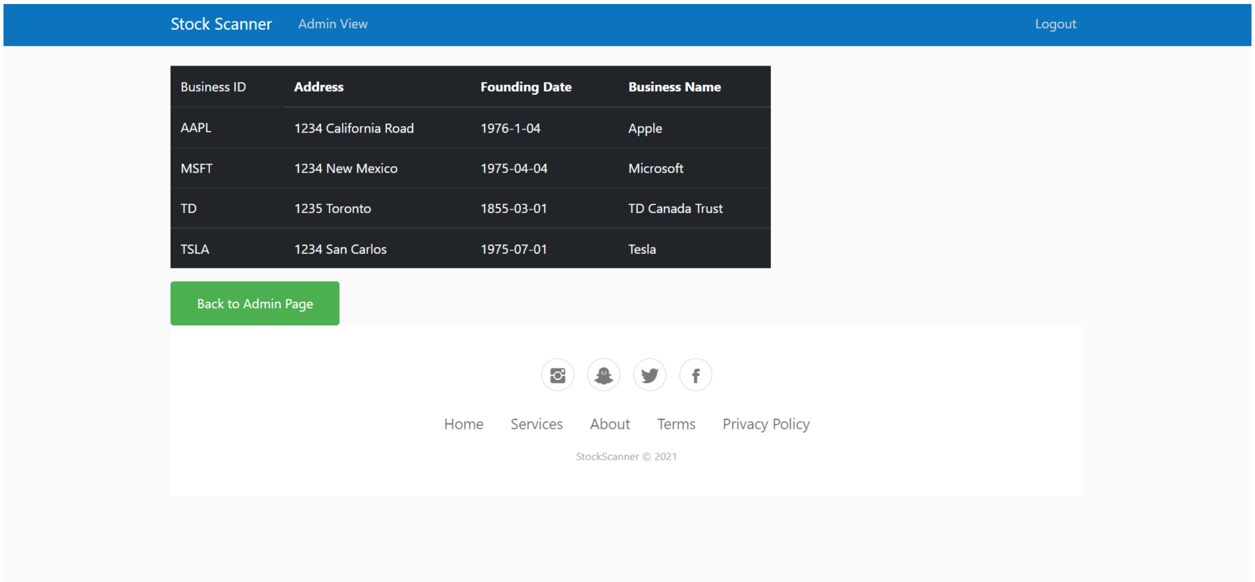
Admin Update Business Page(URL:http://127.0.0.1:5000/updateBusinessAdmin)



Update a Business from the system

This page is similar to adding a business into the database but here the admin cannot update the business id since it’s a primary key and it is being used in other tables by referencing it.

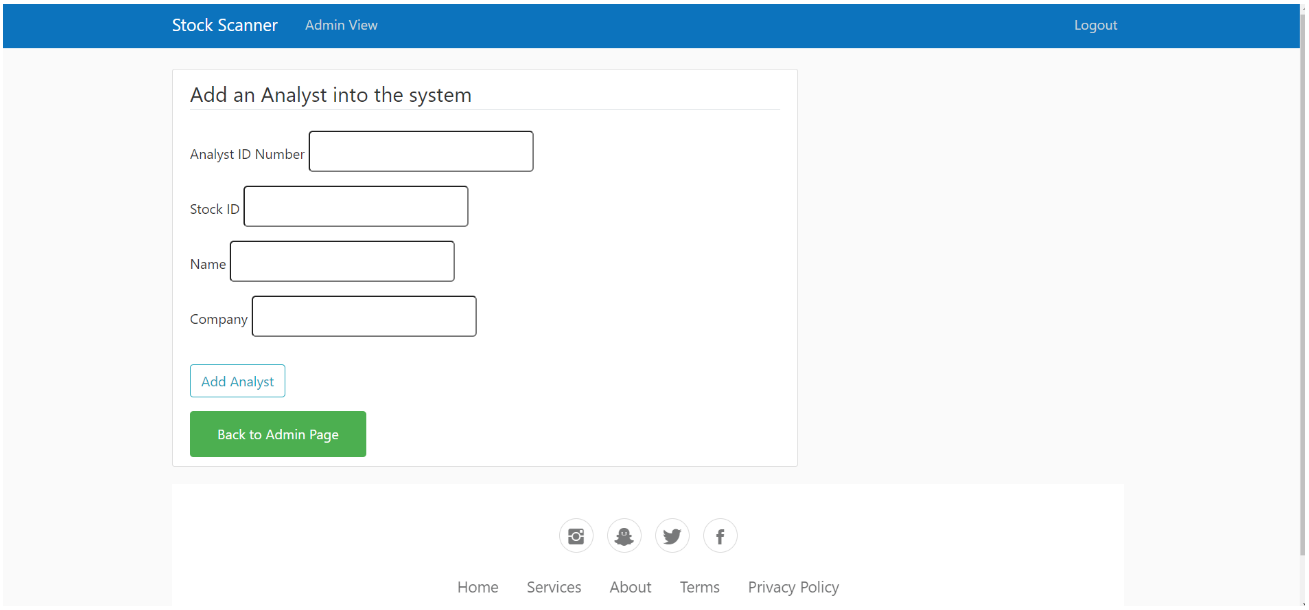
Show All Businesses Page(URL:http://127.0.0.1:5000/showBusinessAdmin)



Business ID	Address	Founding Date	Business Name
AAPL	1234 California Road	1976-1-04	Apple
MSFT	1234 New Mexico	1975-04-04	Microsoft
TD	1235 Toronto	1855-03-01	TD Canada Trust
TSLA	1234 San Carlos	1975-07-01	Tesla

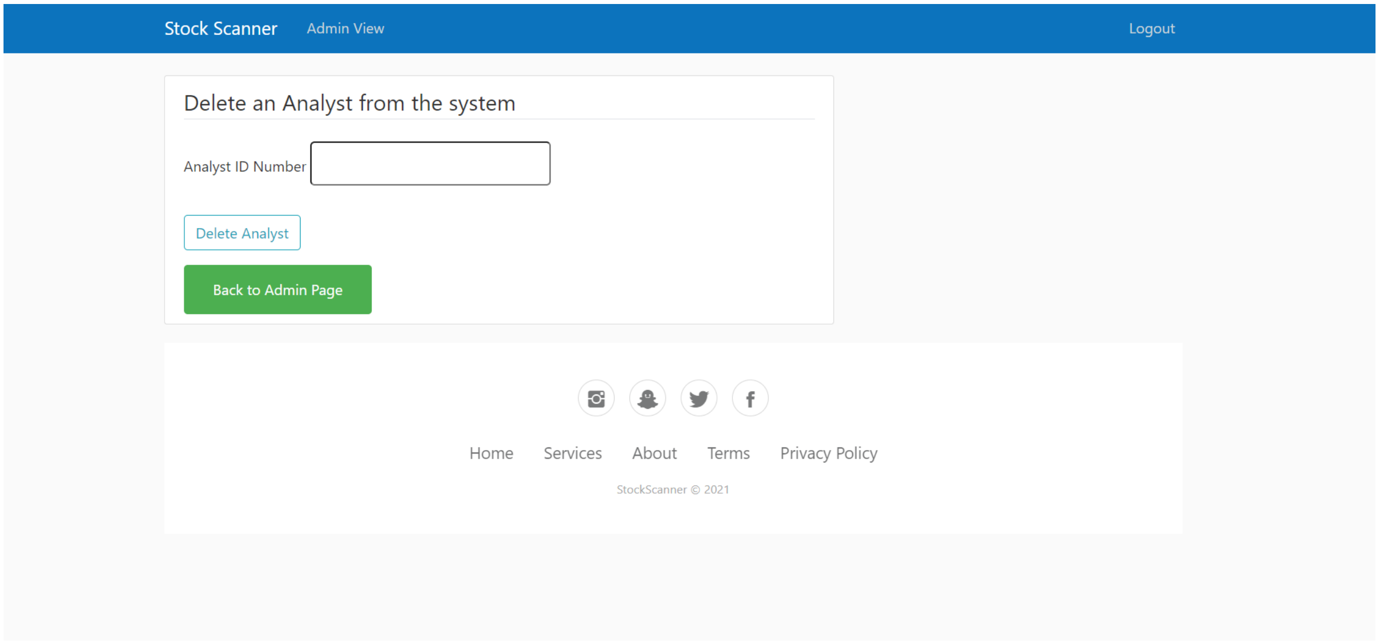
Here the admin gets an insight onto the database, with all of the businesses displayed in a table format similar to how it would look in MySQL command-line client.

Admin Create Analyst Page(URL:http://127.0.0.1:5000/addAnalystAdmin)


Add an Analyst into the system

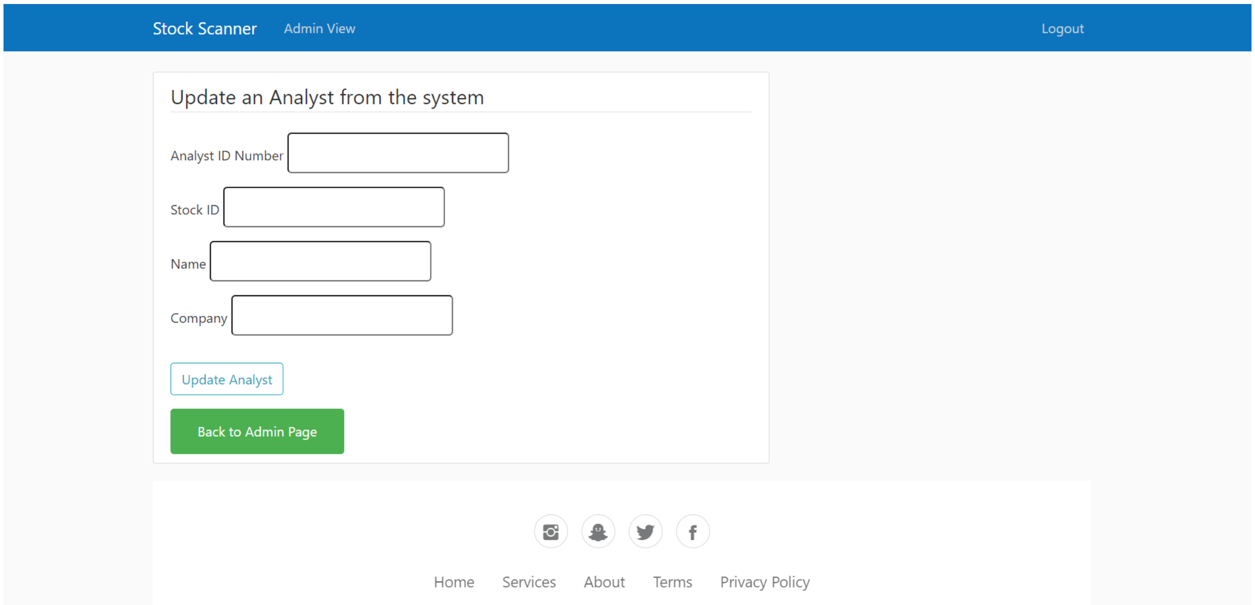
The admin can also create an analyst by doing a similar setup, there are some restrictions as well on this the stock id needs to exist in the database beforehand assigning them to analyze it. Not multiple analysts can be assigned to the same stock.

Admin Delete Analyst Page(URL:http://127.0.0.1:5000/deleteAnalystAdmin)


Delete an Analyst from the system

The analyst id number is used here to delete them from the database if needed to do be happening.

Admin Update Analyst Page(URL:http://127.0.0.1:5000/deleteAnalystAdmin)



Update an Analyst from the system

Analyst ID Number

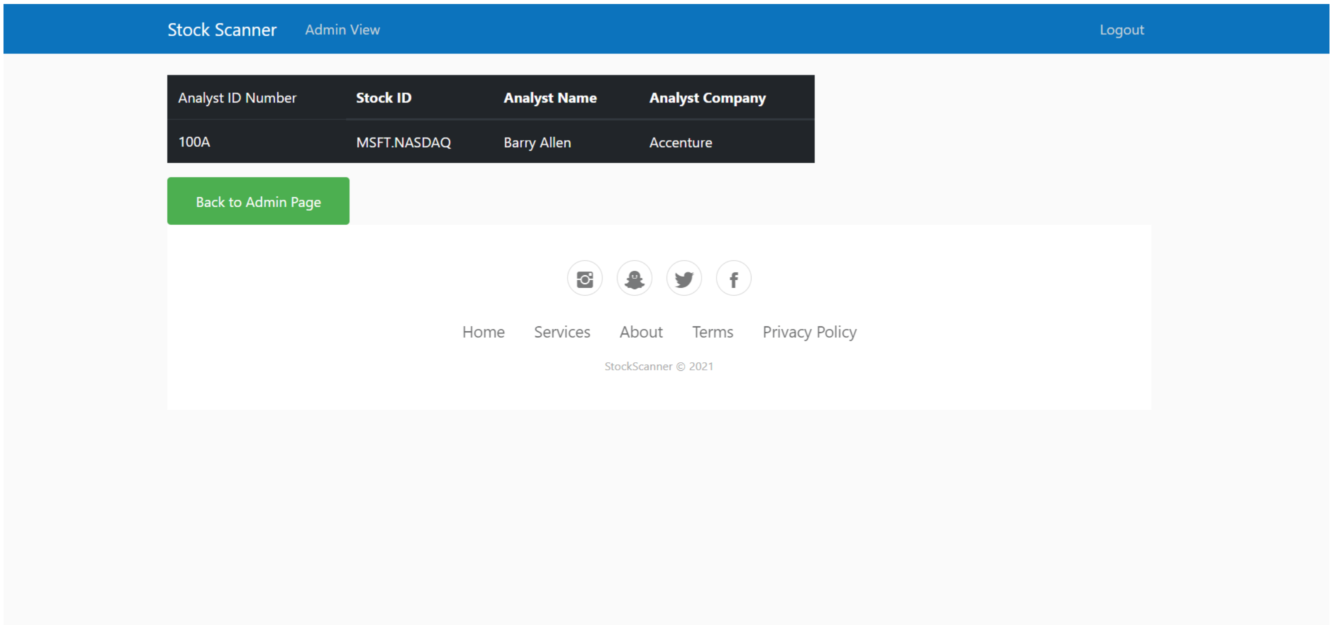
Stock ID

Name

Company

The only thing which the admin should not update is the Analyst ID number which is used by other tables the others are able to change without any hesitation.

Admin Show All Analysts Page(URL: http://127.0.0.1:5000/showAnalystAdmin)

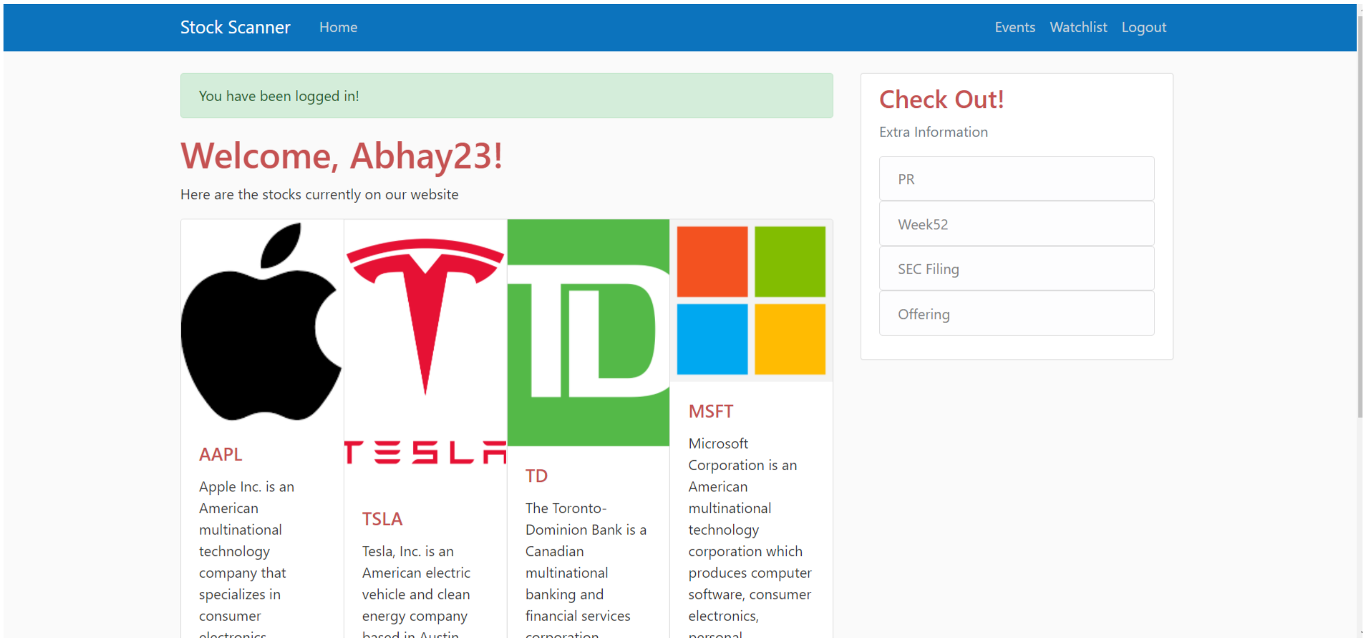


Analyst ID Number	Stock ID	Analyst Name	Analyst Company
100A	MSFT.NASDAQ	Barry Allen	Accenture

Here the admin gets an insight onto the database, with all of the analysts displayed in a table format similar to how it would look in MySQL command-line client.

As a professional user you can make an offering to buy the number of stocks from the company here you the professional user view looks a bit different:

Show Stocks Page For Professional User(URL:http://127.0.0.1:5000/showProView)



You have been logged in!

Welcome, Abhay23!

Here are the stocks currently on our website

AAPL

Apple Inc. is an American multinational technology company that specializes in consumer

TSLA

Tesla, Inc. is an American electric vehicle and clean energy company

TD

The Toronto-Dominion Bank is a Canadian multinational banking and financial services

MSFT

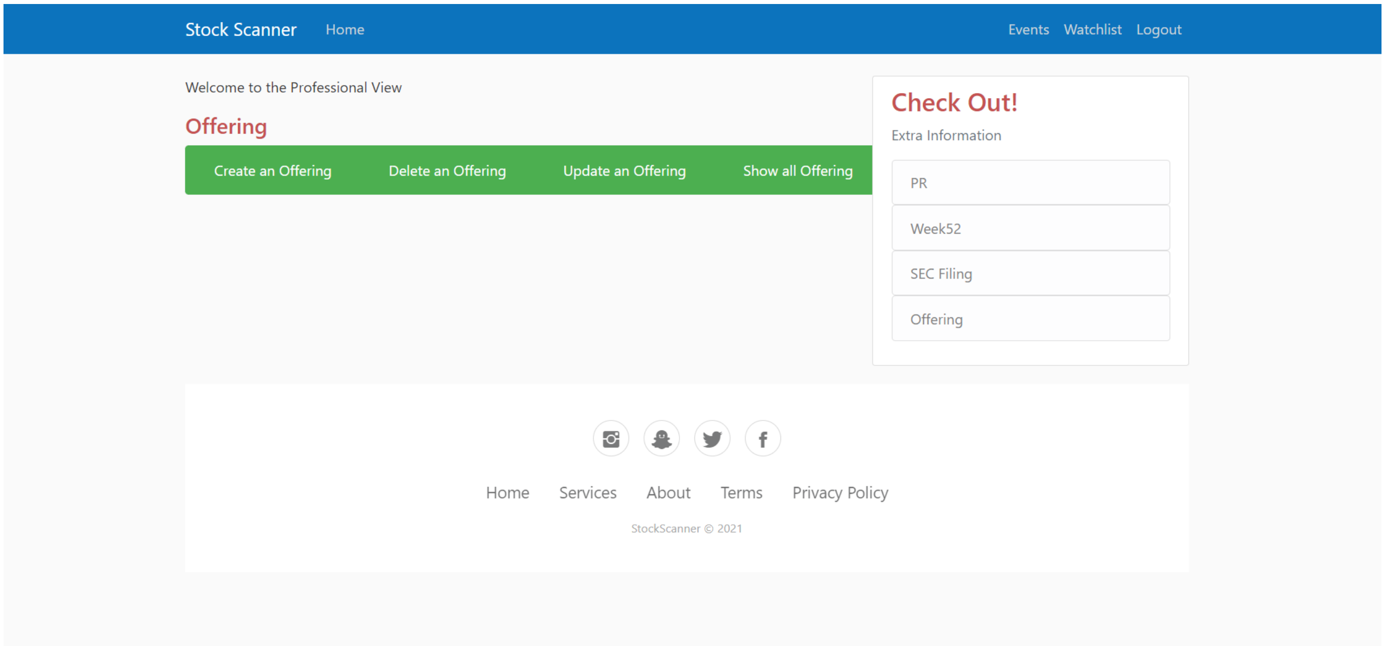
Microsoft Corporation is an American multinational technology corporation which produces computer software, consumer electronics,

Check Out!

Extra Information

This page is almost similar to as if you were a private user but as a professional, since you have more experience with the market you can make offerings.

Show Offering Page For Professional User(URL:http://127.0.0.1:5000/showOffering)



Welcome to the Professional View

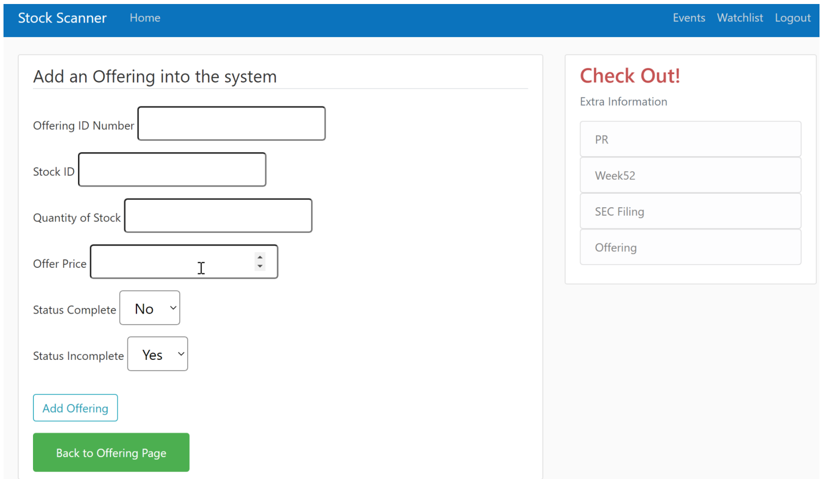
Offering

Check Out!

Extra Information

StockScanner © 2021

This page allows you to modify the offerings which a professional user can make to the business and can be seen by other companies using our website. The other options on this page resemble the admin page and follow a similar styling as well.



Add an Offering into the system

Offering ID Number

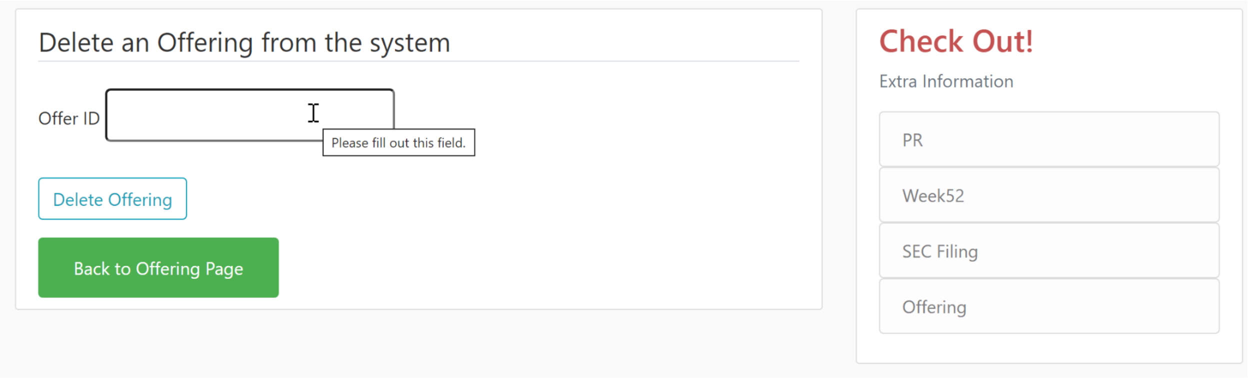
Stock ID

Quantity of Stock

Offer Price

Status Complete: No

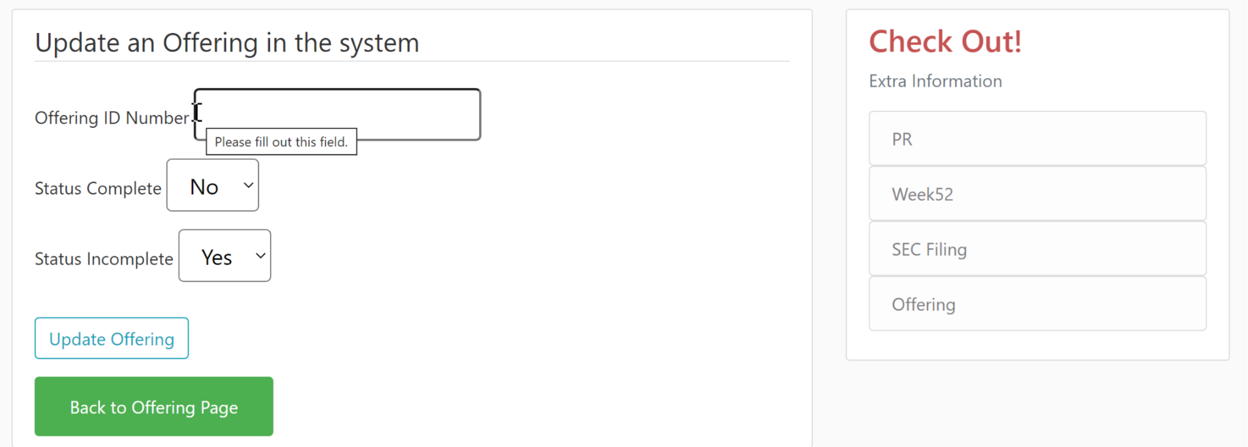
Status Incomplete: Yes



Delete an Offering from the system

Offer ID

Please fill out this field.



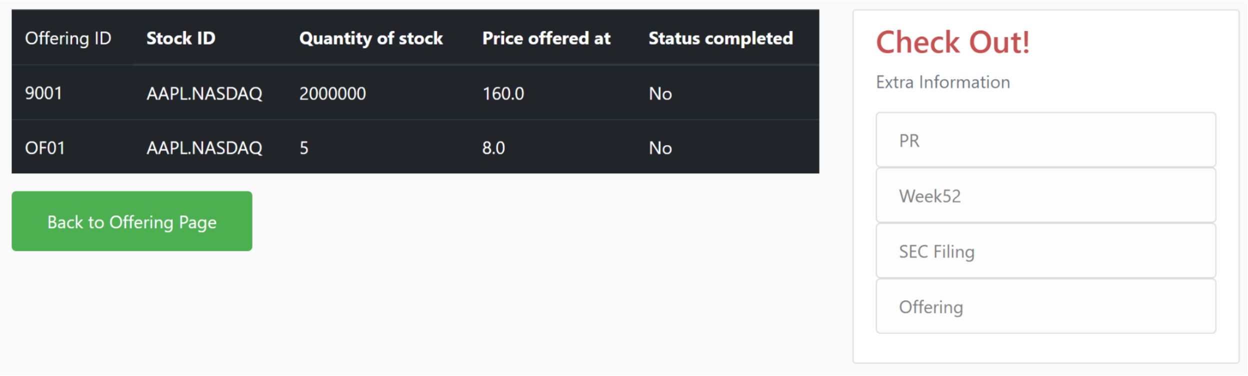
Update an Offering in the system

Offering ID Number

Please fill out this field.

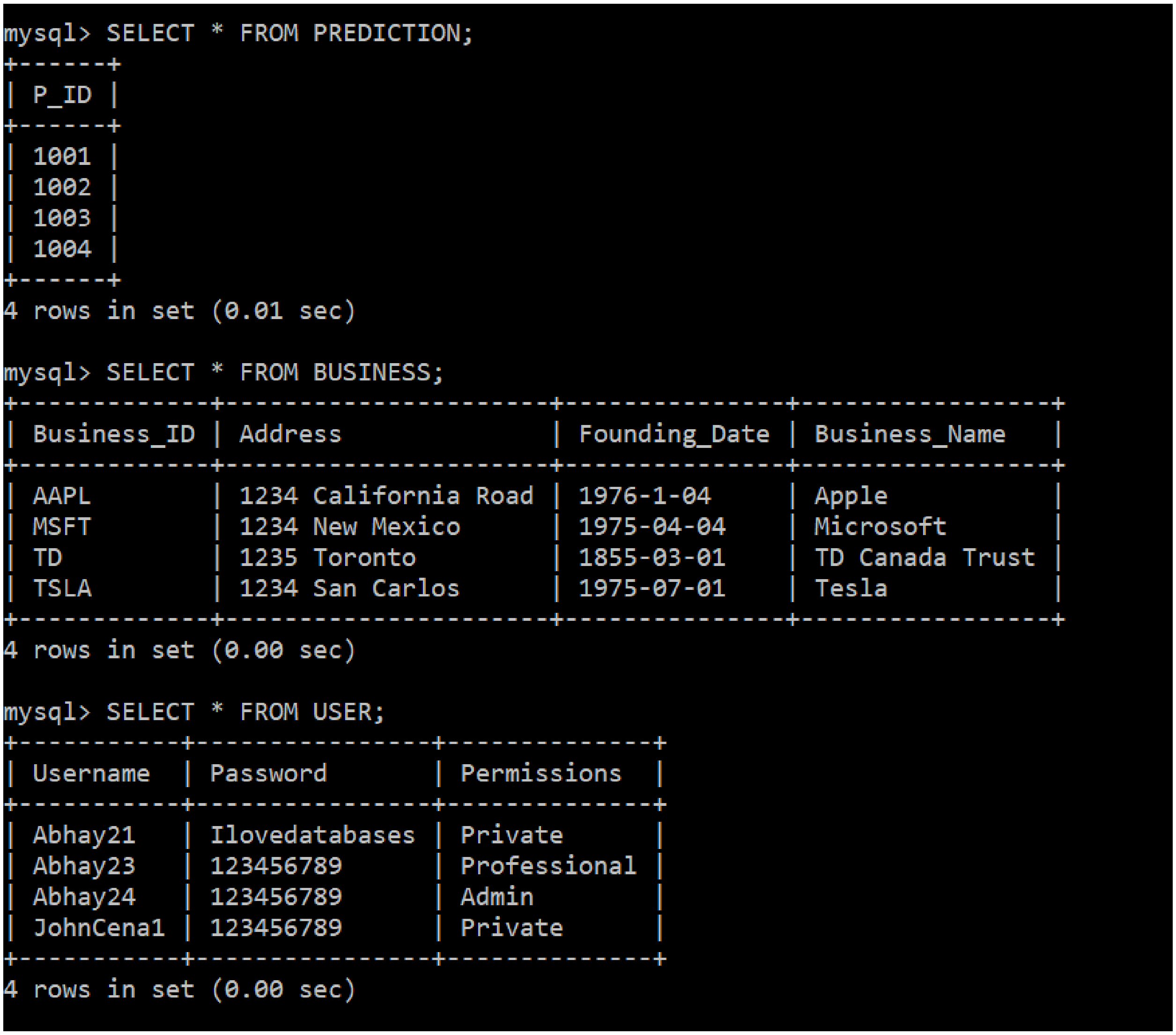
Status Complete: No

Status Incomplete: Yes



Offering ID	Stock ID	Quantity of stock	Price offered at	Status completed
9001	AAPL.NASDAQ	2000000	160.0	No
OF01	AAPL.NASDAQ	5	8.0	No

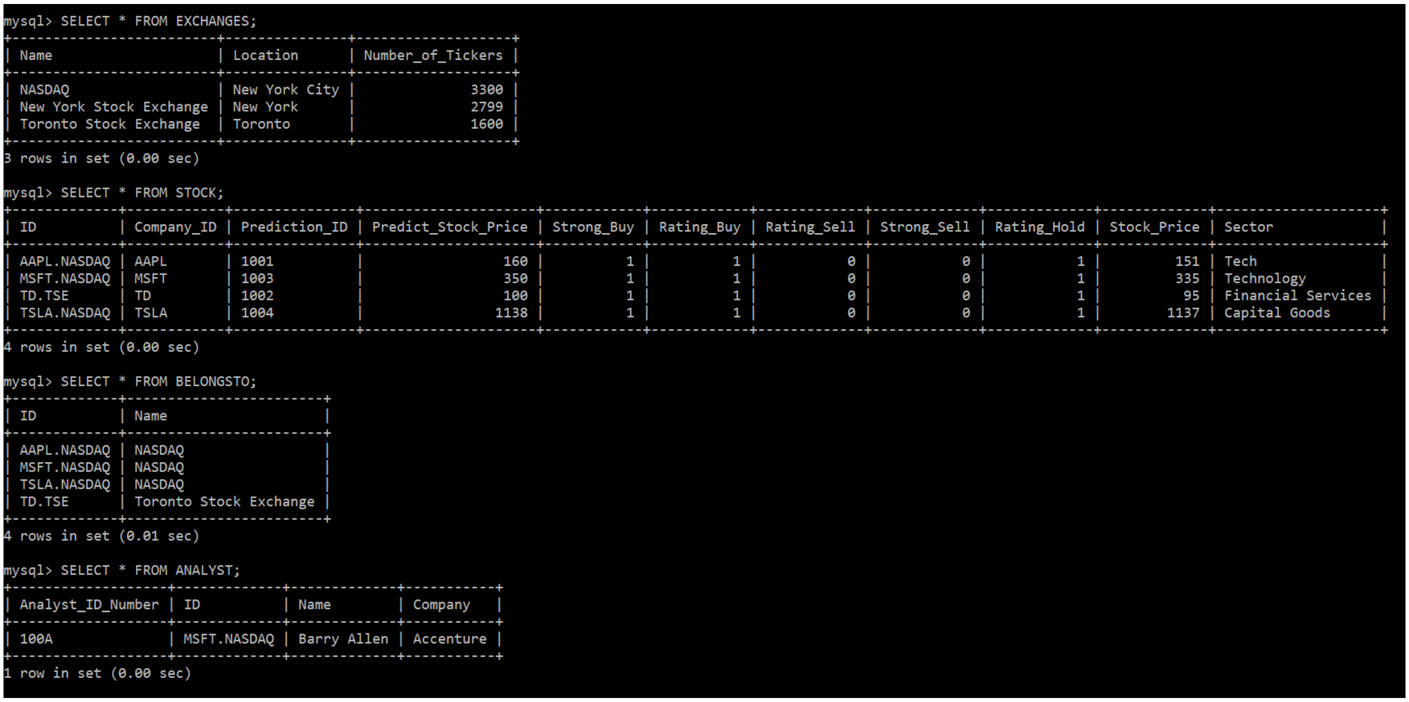
The Initial State Of Database



```
mysql> SELECT * FROM PREDICTION;
+------+
| P_ID |
+------+
| 1001 |
| 1002 |
| 1003 |
| 1004 |
+------+
4 rows in set (0.01 sec)

mysql> SELECT * FROM BUSINESS;
+-------------+----------------------+--------------+-----------------+
| Business_ID | Address              | Founding_Date | Business_Name  |
+-------------+----------------------+--------------+-----------------+
| AAPL        | 1234 California Road | 1976-1-04     | Apple          |
| MSFT        | 1234 New Mexico      | 1975-04-04    | Microsoft      |
| TD          | 1235 Toronto         | 1855-03-01    | TD Canada Trust |
| TSLA        | 1234 San Carlos      | 1975-07-01    | Tesla          |
+-------------+----------------------+--------------+-----------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM USER;
+-----------+----------------+--------------+
| Username  | Password       | Permissions  |
+-----------+----------------+--------------+
| Abhay21   | Ilovedatabases | Private      |
| Abhay23   | 123456789      | Professional |
| Abhay24   | 123456789      | Admin        |
| JohnCena1 | 123456789      | Private      |
+-----------+----------------+--------------+
4 rows in set (0.00 sec)
```

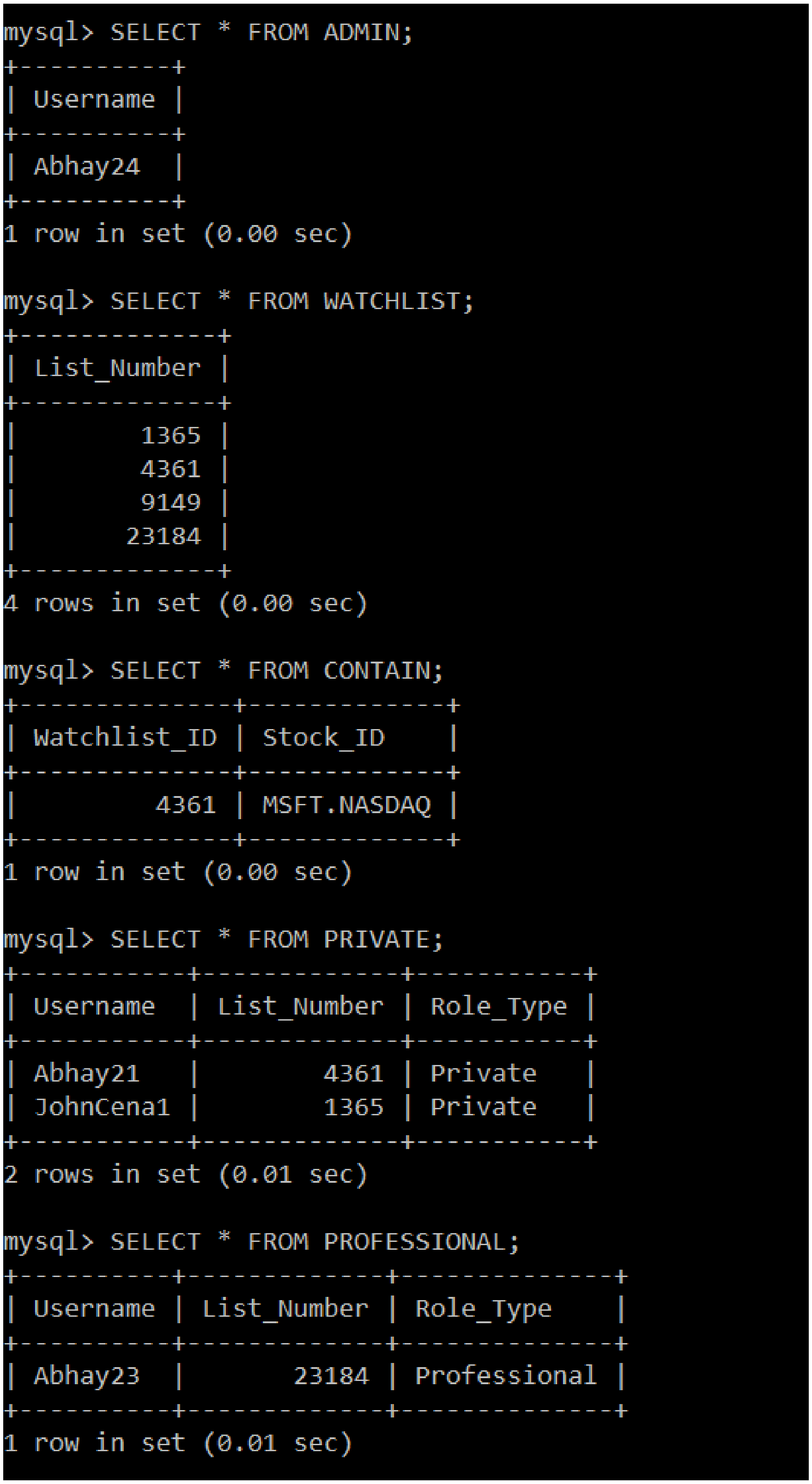


```
mysql> SELECT * FROM EXCHANGES;
+-------------------------+---------------+-------------------+
| Name                    | Location      | Number_of_Tickers |
+-------------------------+---------------+-------------------+
| NASDAQ                  | New York City |              3300 |
| New York Stock Exchange | New York      |              2799 |
| Toronto Stock Exchange  | Toronto       |              1600 |
+-------------------------+---------------+-------------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM STOCK;
+-------------+------------+---------------+---------------------+------------+------------+-------------+-------------+-------------+-------------+--------------------+
| ID          | Company_ID | Prediction_ID | Predict_Stock_Price | Strong_Buy | Rating_Buy | Rating_Sell | Strong_Sell | Rating_Hold | Stock_Price | Sector             |
+-------------+------------+---------------+---------------------+------------+------------+-------------+-------------+-------------+-------------+--------------------+
| AAPL.NASDAQ | AAPL       | 1001          |                 160 |          1 |          1 |           0 |           0 |           1 |         151 | Tech               |
| MSFT.NASDAQ | MSFT       | 1003          |                 350 |          1 |          1 |           0 |           0 |           1 |         335 | Technology         |
| TD.TSE      | TD         | 1002          |                 100 |          1 |          1 |           0 |           0 |           1 |          95 | Financial Services |
| TSLA.NASDAQ | TSLA       | 1004          |                1138 |          1 |          1 |           0 |           0 |           1 |        1137 | Capital Goods      |
+-------------+------------+---------------+---------------------+------------+------------+-------------+-------------+-------------+-------------+--------------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM BELONGSTO;
+-------------+------------------------+
| ID          | Name                   |
+-------------+------------------------+
| AAPL.NASDAQ | NASDAQ                 |
| MSFT.NASDAQ | NASDAQ                 |
| TSLA.NASDAQ | NASDAQ                 |
| TD.TSE      | Toronto Stock Exchange |
+-------------+------------------------+
4 rows in set (0.01 sec)

mysql> SELECT * FROM ANALYST;
+-------------------+-------------+-------------+-----------+
| Analyst_ID_Number | ID          | Name        | Company   |
+-------------------+-------------+-------------+-----------+
| 100A              | MSFT.NASDAQ | Barry Allen | Accenture |
+-------------------+-------------+-------------+-----------+
1 row in set (0.00 sec)
```



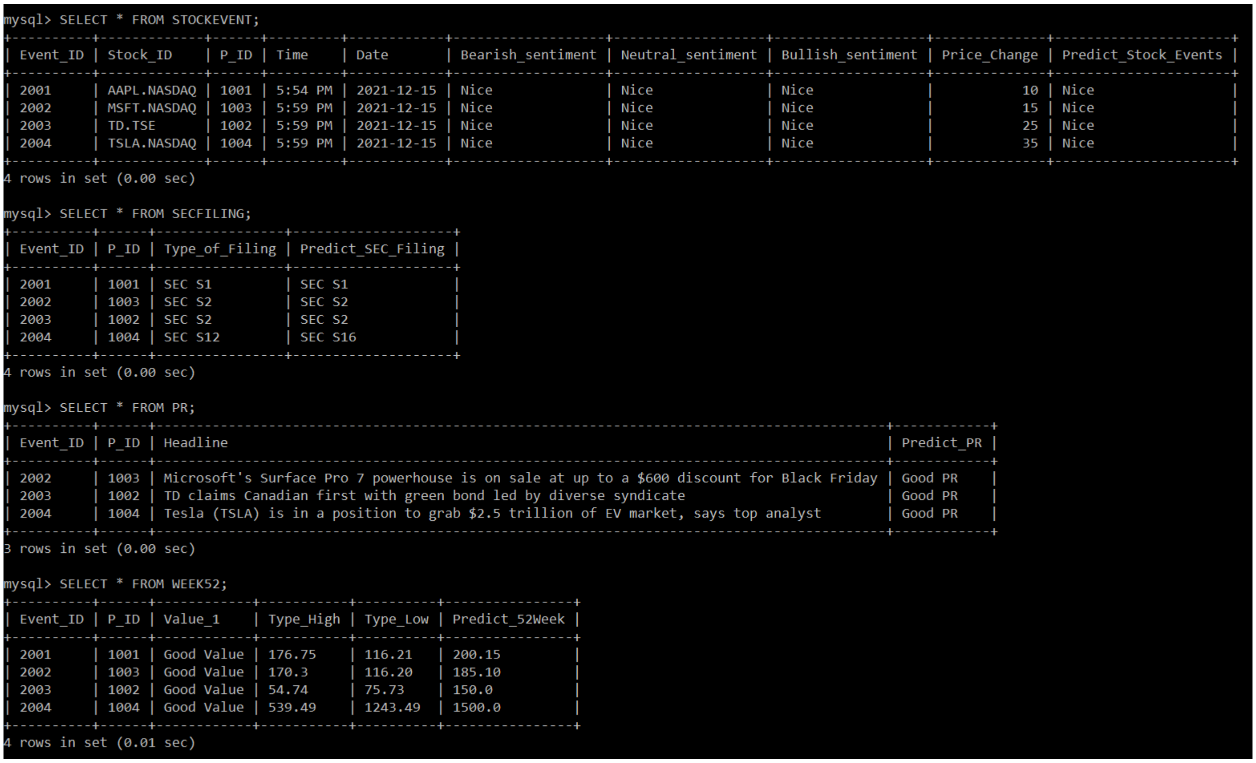
```
mysql> SELECT * FROM ADMIN;
+----------+
| Username |
+----------+
| Abhay24  |
+----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM WATCHLIST;
+-------------+
| List_Number |
+-------------+
|        1365 |
|        4361 |
|        9149 |
|       23184 |
+-------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM CONTAIN;
+--------------+-------------+
| Watchlist_ID | Stock_ID    |
+--------------+-------------+
|         4361 | MSFT.NASDAQ |
+--------------+-------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM PRIVATE;
+-----------+-------------+-----------+
| Username  | List_Number | Role_Type |
+-----------+-------------+-----------+
| Abhay21   |        4361 | Private   |
| JohnCena1 |        1365 | Private   |
+-----------+-------------+-----------+
2 rows in set (0.01 sec)

mysql> SELECT * FROM PROFESSIONAL;
+----------+-------------+--------------+
| Username | List_Number | Role_Type    |
+----------+-------------+--------------+
| Abhay23  |       23184 | Professional |
+----------+-------------+--------------+
1 row in set (0.01 sec)
```

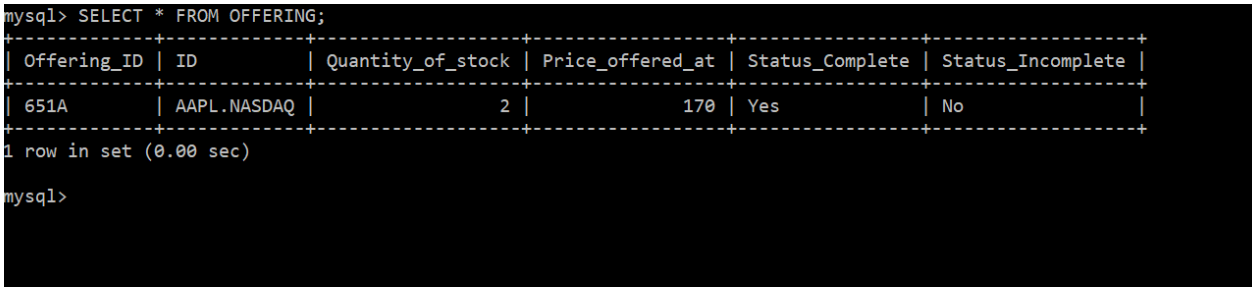


```
mysql> SELECT * FROM STOCKEVENT;
+----------+-------------+------+---------+------------+-------------------+-------------------+-------------------+--------------+----------------------+
| Event_ID | Stock_ID    | P_ID | Time    | Date       | Bearish_sentiment | Neutral_sentiment | Bullish_sentiment | Price_Change | Predict_Stock_Events |
+----------+-------------+------+---------+------------+-------------------+-------------------+-------------------+--------------+----------------------+
| 2001     | AAPL.NASDAQ | 1001 | 5:54 PM | 2021-12-15 | Nice              | Nice              | Nice              |           10 | Nice                 |
| 2002     | MSFT.NASDAQ | 1003 | 5:59 PM | 2021-12-15 | Nice              | Nice              | Nice              |           15 | Nice                 |
| 2003     | TD.TSE      | 1002 | 5:59 PM | 2021-12-15 | Nice              | Nice              | Nice              |           25 | Nice                 |
| 2004     | TSLA.NASDAQ | 1004 | 5:59 PM | 2021-12-15 | Nice              | Nice              | Nice              |           35 | Nice                 |
+----------+-------------+------+---------+------------+-------------------+-------------------+-------------------+--------------+----------------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM SECFILING;
+----------+------+----------------+--------------------+
| Event_ID | P_ID | Type_of_Filing | Predict_SEC_Filing |
+----------+------+----------------+--------------------+
| 2001     | 1001 | SEC S1         | SEC S1             |
| 2002     | 1003 | SEC S2         | SEC S2             |
| 2003     | 1002 | SEC S2         | SEC S2             |
| 2004     | 1004 | SEC S12        | SEC S16            |
+----------+------+----------------+--------------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM PR;
+----------+------+-------------------------------------------------------------------------------------------+------------+
| Event_ID | P_ID | Headline                                                                                  | Predict_PR |
+----------+------+-------------------------------------------------------------------------------------------+------------+
| 2002     | 1003 | Microsoft's Surface Pro 7 powerhouse is on sale at up to a $600 discount for Black Friday | Good PR    |
| 2003     | 1002 | TD claims Canadian first with green bond led by diverse syndicate                         | Good PR    |
| 2004     | 1004 | Tesla (TSLA) is in a position to grab $2.5 trillion of EV market, says top analyst        | Good PR    |
+----------+------+-------------------------------------------------------------------------------------------+------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM WEEK52;
+----------+------+------------+-----------+----------+----------------+
| Event_ID | P_ID | Value_1    | Type_High | Type_Low | Predict_52Week |
+----------+------+------------+-----------+----------+----------------+
| 2001     | 1001 | Good Value | 176.75    | 116.21   | 200.15         |
| 2002     | 1003 | Good Value | 170.3     | 116.20   | 185.10         |
| 2003     | 1002 | Good Value | 54.74     | 75.73    | 150.0          |
| 2004     | 1004 | Good Value | 539.49    | 1243.49  | 1500.0         |
+----------+------+------------+-----------+----------+----------------+
4 rows in set (0.01 sec)
```



```
mysql> SELECT * FROM OFFERING;
+-------------+-------------+-------------------+------------------+-----------------+-------------------+
| Offering_ID | ID          | Quantity_of_stock | Price_offered_at | Status_Complete | Status_Incomplete |
+-------------+-------------+-------------------+------------------+-----------------+-------------------+
| 651A        | AAPL.NASDAQ |                 2 |              170 | Yes             | No                |
+-------------+-------------+-------------------+------------------+-----------------+-------------------+
1 row in set (0.00 sec)

mysql>
```

References

Beck, C. (2020, November 9). *Predicting the stock market is hard: Creating a machine-learning model (probably) won't help*. Medium. Retrieved December 17, 2021, from https://towardsdatascience.com/predicting-the-stock-market-is-hard-creating-a-machine-learning-model-probably-wont-help-e449039c9fe3

CoreyMSchafer. (n.d.). *Code_snippets/python/flask_blog at master · coreymschafer/code_snippets*. GitHub. Retrieved December 17, 2021, from https://github.com/CoreyMSchafer/code_snippets/tree/master/Python/Flask_Blog

CoreyMSchafer. (2018, May 4). *Python flask tutorial: Full-featured web app part 4 - database with Flask-SQLAlchemy*. YouTube. Retrieved December 17, 2021, from https://www.youtube.com/watch?v=cYWiDiIUxQc&list=PL-osiE80TeTs4UjLw5MM6OjgkjFeUxCYH&index=5

David HerliantoDavid Herlianto 12311 gold badge11 silver badge44 bronze badges, bgsebgse 7, DobleLDobleL 3, & Eltac ShikhsaidovEltac Shikhsaidov 8366 bronze badges. (1965, September 1). *Flask - display database from python to HTML*. Stack Overflow. Retrieved December 17, 2021, from https://stackoverflow.com/questions/45558349/flask-display-database-from-python-to-html

Eric PruittEric Pruitt 1, Mike GrahamMike Graham 67.3k1313 gold badges9595 silver badges127127 bronze badges, Alex MartelliAlex Martelli 791k160160 gold badges11831183 silver badges13631363 bronze badges, & wescpywescpy 9. (1958, May 1). *Purpose of SQLALCHEMY over mysqldb*. Stack Overflow. Retrieved December 17, 2021, from https://stackoverflow.com/questions/2550292/purpose-of-sqlalchemy-over-mysqldb

Free themes for Bootstrap. Bootswatch. (n.d.). Retrieved December 17, 2021, from https://bootswatch.com/

Google. (n.d.). *JSON Viewer Pro*. Google. Retrieved December 17, 2021, from https://chrome.google.com/webstore/detail/json-viewer-pro/eifflpmocdbdmepbjaopkkhbfmdgijcc

MySQL community downloads. MySQL. (n.d.). Retrieved December 17, 2021, from https://dev.mysql.com/downloads/

nesoacademy. (2021, February 24). *Introduction to database management systems (DBMS)*. YouTube. Retrieved December 17, 2021, from https://www.youtube.com/watch?v=6Iu45VZGQDk&list=PLBlnK6fEyqRi_CUQ-FXxgzKQ1dwr_ZJWZ

Published by Statista Research Department, & 19, O. (2021, October 19). *U.S. Snapchat usage by age 2020*. Statista. Retrieved December 12, 2021, from https://www.statista.com/statistics/814300/snapchat-users-in-the-united-states-by-age/

Quick Code. (2021, August 23). *10+ best flask python tutorials for beginners - learn flask online*. Medium. Retrieved December 17, 2021, from https://medium.com/quick-code/top-online-tutorials-to-learn-flask-python-c2723df5326c

YouTube. (2018, January 22). *Python flask with mysql database*. YouTube. Retrieved December 17, 2021, from https://www.youtube.com/watch?v=6L3HNyXEais

YouTube. (2020, May 24). *Python Flask Tutorial API - how to get data from an API with FLASK | python 2020 | HD*. YouTube. Retrieved December 17, 2021, from https://www.youtube.com/watch?v=F_SBxcV335k