**Python Mini Project : Stock Price Alerter**

**Aim:**

To create application to monitor the stock prices in real-time.

**Target Audience :**

Working professionals who keep track of the stocks.

**Requirements:**

* + Tkinter
  + Pystray
  + Internet connection
  + Twilio account
  + Yahoo\_fin
  + CustomTkinter
  + Winotify

**GUI:**

Graphical user interface, application

Description automatically generated

***Class App:***

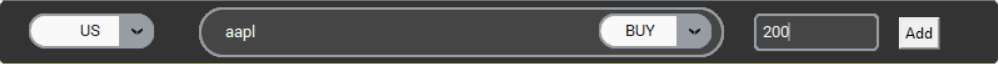
The interface to edit the database seamlessly and track the stock prices is created as an object.

**Nav\_bar :**



Here you can change the phone number in which you want to get the message from.

**Add\_stock\_frame:**



Here you can choose the exchange, stock , price and why you are setting the alert for.

**Menu\_frame:**



Here are the set of labels for users convenience.

**Main\_frame:**

Graphical user interface, application, table

Description automatically generated

Here once the stock data exceeds the main\_frame you can use scrollbar to navigate.

**Class data:**

****

* Each stock frame is created under this class.
* You can edit, delete the details here.



**Functions:**

**‘\_\_init\_\_’** creates the frame and puts the frame in the mainframe.

**‘createframe’** creates the frame once the add button is clicked.

**‘editframe’** and **‘update’** edits the created frame buy changing the price , reasoning and status of the stock.

**‘reset’** resets the status of the sock (i.e) *Alerted* or *Yet to alert.*

**‘delframe’** deletes the stock frame.

**‘refresh’** refreshes the current price in the gui.

**Storage.py**

**‘add’** adds the data of a new stock in the database.

**‘delete’** removes the data from the database.

**‘read’** reads the existing stock data from the database and returns the data.

**‘edit’** edits the stock data.

**‘off\_status’** off’s the data status.

**‘store\_phone’** stores the phone number in a txt file.

**‘read\_phone’** reads the phone number from the txt file.

**Tray.py**

As we know we need to constantly monitor the data it is not viable to run the whole program with gui all the time.

Thus pytray is used to interact with the gui and the code running in the background.

Here threading is used so that we can run the tray icon and the background code to monitor the data at the same time.



**‘on\_click’** when clicking the menu items what should happen.

**‘window\_icon’** creates the tray which will run in the background.

**Alert\_on.py**

**‘get\_stocks’** gets the stock data to notify it approaches the desired prices.

**‘alert’** this alerts when the stock reaches the desired price on both system and mobile.

**‘background’** it is the function which keeps on calling the alert function on a specified interval.