Computer Science
Final Project

STOCKSONLINE

A revolutionary Stock Trading Firm

BY-
Vinay Karthik Kiran, Class 12-B
Shashank Pandey, Class 12-A

Acknowledgement

During the course of this project, we have amassed a great amount of knowledge and have been exposed to many new ideas and concepts . We would like to thank all our teachers and our peers, who have provided us with their helpful advice, continuous support and have been cooperative throughout this entire process.

This project would not have been possible without the supervision and support of the teachers of the Computer Science department, Mrs. Pavani.K and Mrs. Smitha .R . We would also like to thank our Principal, Mrs. Deepa Sridhar, for giving us this opportunity to expand our knowledge and learn valuable skills that will undoubtedly help us in the near future.

Table of Contents

Synopsis

What is the Stock Market?

The stock market is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks (also called shares), which represent ownership claims on businesses; these may include *securities* listed on a public stock exchange, as well as stock that is only traded privately.

A stock exchange is an exchange where stock brokers and traders can buy and sell shares of stock, bonds, and other securities. Many large companies have their stocks listed on a stock exchange. This makes the stock more liquid and thus more attractive to many investors.

Our Project

Our project is titled "Stocks Online". It is an online stock trading firm, i,e a brokerage firm. This firm helps in the sale and purchase of shares of publicly listed companies in the stock market on a daily basis. This system was developed to provide ease of access of data of various stock owners,including salient information about the price, volatility and change of each share.

This firm assists first-time clients who use the firm as a gateway into stock trading and into the wider market, offering them ways to buy and sell stocks at a fair industry-standard 30% transaction fee.

The firm database stores information about every transaction done by any certain consumer, containing details of the transaction, the stocks involved, bought or sold, date of transaction and other details. We use this to provide users with a look at

their portfolio every time they log in. The firm also offers facilities for a customer to add money into their account.

The intention behind StocksOnline is to provide people with little to no knowledge about the stock market to be able to invest money and get returns.

System Requirements

During the course of the project, we used the following applications-

- **MySQL Workbench -**
  - CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU
  -Cores: Single (Dual/Quad Core is recommended)
  -RAM: 4 GB (6 GB recommended)
  -Graphic Accelerators: nVidia or ATI with support of OpenGL 1.5 or higher
  -Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

- **The following operating systems are officially supported:**
  -Windows 7 (64-bit, Professional level or higher)
  -Mac OS X 10.6.1+
  -Ubuntu 9.10 (64bit) or higher
  -Ubuntu 8.04 (32bit/64bit)

Recommended System Requirements for Python and Django

- Processors:Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAMIntel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAMIntel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled)
- Disk space: 2 to 3 GB
- Operating systems: Windows® 10, macOS*, and Linux*

Minimum System Requirements for Django

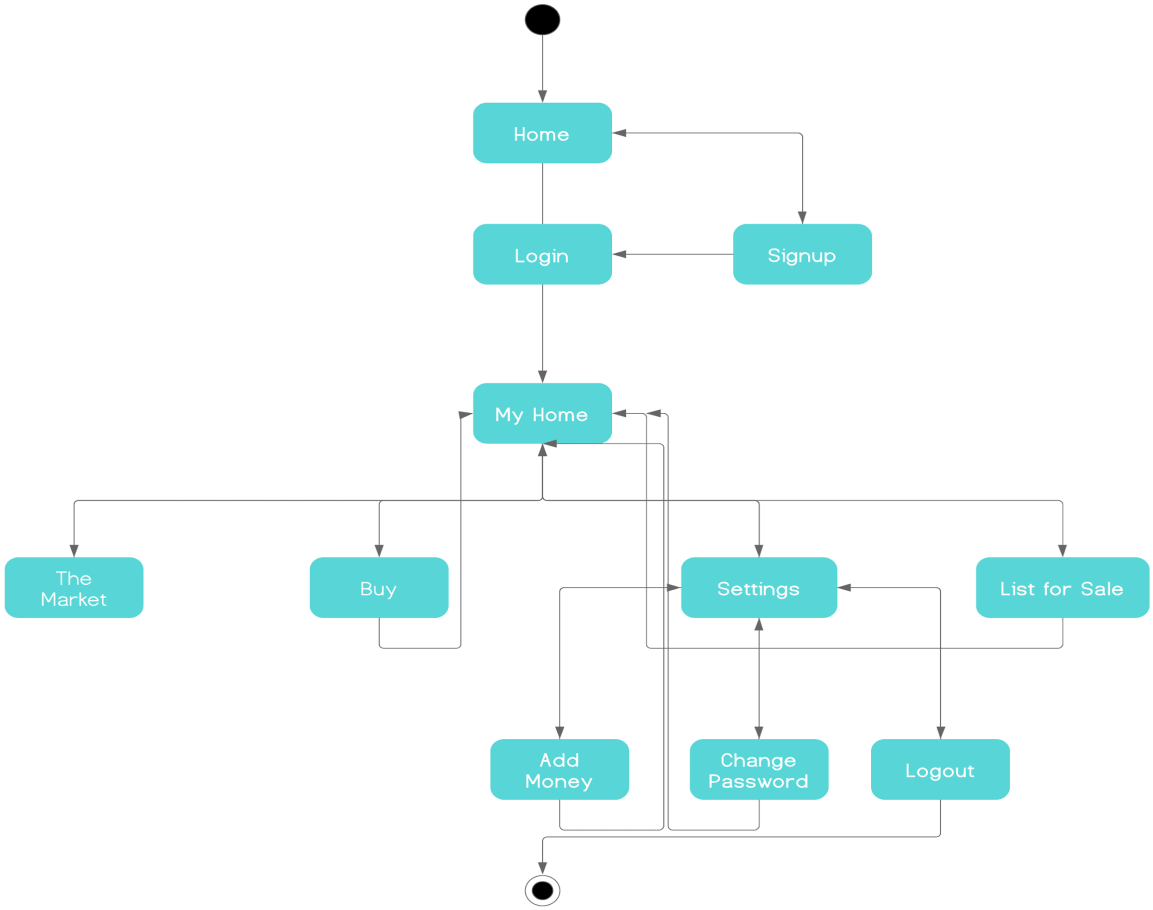
- Processors: Intel Atom® processor or Intel® Core™ i3 processor
- Disk space: 1 GB
- Operating systems: Windows* 7 or later, macOS, and Linux
- Python* versions: 2.7.X, 3.6.X

We also used the Google Drive suite of web applications, including Google Docs and Google Sheets, which are web-based development tools and can be accessed on the following web browsers-

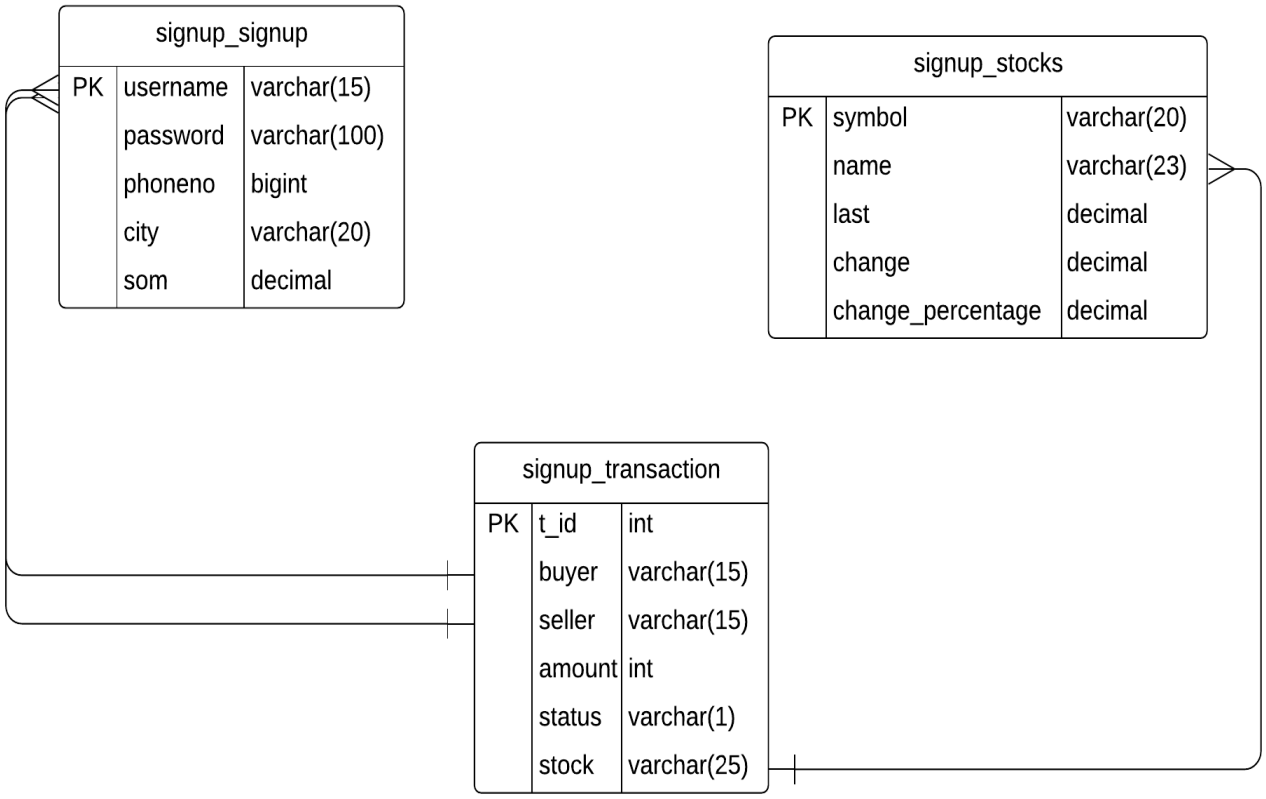
Chrome
Firefox
Microsoft Edge
Safari

Functional Specifications

Flow Diagram-



ER Diagram-


signup_signup
PK
username
varchar(15)
password
varchar(100)
phoneno
bigint
city
varchar(20)
som
decimal
signup_stocks
PK
symbol
varchar(20)
name
varchar(23)
last
decimal
change
decimal
change_percentage
decimal
signup_transaction
PK
t_id
int
buyer
varchar(15)
seller
varchar(15)
amount
int
status
varchar(1)
stock
varchar(25)

Python Code

urls.py

```
from django.contrib import admin
from django.urls import path,include
from login import views as logv
from signup import views as signv

urlpatterns = [
    path('admin/', admin.site.urls),
    path('home/', signv.home),
    path('signup/', signv.signup_l),
    path('login/',logv.login_l),
    path('myhome/',logv.myhome),
    path('themarket/',logv.themarket),
    path('buy/',logv.buy_direct),
    path('listing/',logv.for_sale),
    path('topup/',logv.change_details),
    path('changepasswd/',logv.change_passwd),
    path('about/',logv.about),
]
```

models.py

```
from django.db import models
```

```
#The signup model is required as it stores all details of customers
who have registered with StocksOnline. It contains all the personal
details of the user.

class signup(models.Model):
    username=models.CharField(max_length=15, primary_key=True)
    password=models.CharField(max_length=8)
    phoneno=models.BigIntegerField(max_length=10)
    city=models.CharField(max_length=20)
    som=models.IntegerField(10000000000)


#The stocks model is needed as it is the source from which the
program gets the different stocks available and contains information
about them.

class stocks(models.Model):
    company=models.CharField(max_length=6,primary_key=True)
    name=models.CharField(max_length=20)
    price=models.IntegerField(100000)
    change=models.DecimalField(max_digits=5,decimal_places=5)
    volatility=models.DecimalField()

#The transaction model stores transactions completed, the
buyer/seller (if any) and contains all the details about the
transaction.

class transaction(models.Model):
    t_id=models.AutoField(primary_key=True)
    buyer=models.CharField(max_length=15)
    seller=models.CharField(max_length=15)
    stock=models.CharField(max_length=20)
    amount=models.IntegerField(10000)
```

login/views.py

```
from django.shortcuts import render,redirect

# Create your views here.
from signup.models import stocks
import mysql.connector
from signup.models import transaction

login_check=False
con=mysql.connector.connect(host="localhost", user="root",
passwd="root",database="stocksonline")

mycursor=con.cursor()

# for logging in. compares entered username and password. if matches,
generates relevant information and goes to myhome. if
# not, reloads page with a message..
def login_1(request):
    global username
    global login_check
    con=mysql.connector.connect(host="localhost", user="root",
passwd="root",database="stocksonline")

    mycursor=con.cursor()
    sql="select username,password from signup_signup"
    mycursor.execute(sql)
    a=mycursor.fetchall()
    print(a)

    usr=[]
    pwd=[]
    #login_check=False
    for i in a:
        usr.append(i[0])
        pwd.append(i[1])
    l=len(usr)
```

```
    if request.method=="POST":
        global username
        d=request.POST["username"]
        password=request.POST["password"]
        for i in range(l):
            if usr[i]==d:
                if pwd[i]==password:
                    #global username
                    username=d
                    #global login_check
                    login_check=True
                    #request.session['lc']=True
                    print(login_check)
                    '''global l1
                    global l2
                    global dl'''
                    sql18="select som from signup_signup where
username='{}'".format(username)
                    mycursor.execute(sql18)
                    a=mycursor.fetchall()
                    balance=a[0][0]
                    text="Welcome, "+username+",You have
"+str(balance)+" in your account!"

l1=transaction.objects.filter(buyer=username).values('amount','stock'
,'shares')

l2=transaction.objects.filter(seller=username).filter(status='y').val
ues('amount','stock','shares')
                    dl=[]
                    if con.is_connected:
                        print('yes')
                    sql15="select name from signup_stocks"
                    mycursor.execute(sql15)
```

```
                    n=mycursor.fetchall()
                    l=len(n)
                    for i in range(l):
                        sql8="select sum(shares) from
signup_transaction where buyer='{}' and
stock='{}'".format(username,n[i][0])
                        sql9="select sum(shares) from
signup_transaction where seller='{}' and
stock='{}'".format(username,n[i][0])
                        mycursor.execute(sql8)
                        a=mycursor.fetchall()
                        mycursor.execute(sql9)
                        b=mycursor.fetchall()
                        if a==[(None,)]:
                            bought=0
                        else:
                            bought=float(a[0][0])
                        if b==[(None,)]:
                            sold=0
                        else:
                            sold=float(b[0][0])
                        net=bought-sold
                        if net>0:
                            sql16="select
signup_stocks.symbol,signup_stocks.name,signup_stocks.last,signup_sto
cks.change,signup_stocks.change_percentage from signup_stocks where
name='{}'".format(n[i][0])
                            mycursor.execute(sql16)
                            g=mycursor.fetchall()
                            de=list(g[0])
                            d=de+[net]
                            print(d)
                            dl.append(d)
                    return render(request,
"myhome.html",{'a':text,'l1':l1,'l2':l2,'l3':dl})
                    #return redirect("../myhome")
```

```
                '''else:
                    login_check=False
                    return render(request, "login.html")'''
        else:
            login_check=False
            return render(request,"login.html",{'a':'Incorrect-
please try again'})
    else:
        return render(request, "login.html",{'a':''})

        #usr = login.Objects.all()
        #print(usr)
print(login_check)


# displaying the home page
def myhome(request):

   if login_check==True:
        con=mysql.connector.connect(host="localhost", user="root",
passwd="root",database="stocksonline")
        mycursor=con.cursor()
        sql18="select som from signup_signup where
username='{}'".format(username)
        mycursor.execute(sql18)
        a=mycursor.fetchall()
        balance=a[0][0]

        text="Welcome, "+username+" !, You have "+str(balance)+" in
your account!"

l1=transaction.objects.filter(buyer=username).values('amount','stock'
,'shares')
```

```
l2=transaction.objects.filter(seller=username).values('amount','stock
','shares')
        dl=[]
        if con.is_connected:
            print('yes')
        sql15="select name from signup_stocks"
        mycursor.execute(sql15)
        n=mycursor.fetchall()
        l=len(n)
        for i in range(l):
            sql8="select sum(shares) from signup_transaction where
buyer='{}' and stock='{}'".format(username,n[i][0])
            sql9="select sum(shares) from signup_transaction where
seller='{}' and stock='{}'".format(username,n[i][0])
            mycursor.execute(sql8)
            a=mycursor.fetchall()
            mycursor.execute(sql9)
            b=mycursor.fetchall()
            if a==[(None,)]:
                bought=0
            else:
                bought=float(a[0][0])
            if b==[(None,)]:
                sold=0
            else:
                sold=float(b[0][0])
            net=bought-sold
            if net>0:
                sql16="select
signup_stocks.symbol,signup_stocks.name,signup_stocks.last,signup_sto
cks.change,signup_stocks.change_percentage,signup_transaction.shares
from signup_stocks inner join signup_transaction on
signup_stocks.name=signup_transaction.stock where
name='{}'".format(n[i][0])
                mycursor.execute(sql16)
```

```
                g=mycursor.fetchall()
                de=list(g[0])
                d=de+[net]
                print(d)
                dl.append(d)
        return render(request,
'myhome.html',{'a':text,'l1':l1,'l2':l2,'l3':dl})
   else:
       return render(request,'login.html',{'a':''})


#the market- static page solely displaying stock details
def themarket(request):
    print("here",login_check)
    if login_check==True:
        #con=mysql.connector.connect(host="localhost", user="root",
passwd="root",database="stocksonline")
        #mycursor=con.cursor()
        sql1="select * from signup_stocks"
        #mycursor.execute(sql1)
        #a=mycursor.fetchall()
        #name=[]
        #comp=[]
        #price=[]
        #change=[]
        #volatility=[]
        a=stocks.objects.all()

        #n=len(a)
        '''for i in a:
            name.append(i[0])
            comp.append(i[1])
            price.append(i[2])
            change.append(i[3])
            volatility.append(i[4])'''
        return render(request, 'themarket.html',{'a':a})
```

```
    else:
        return render(request,'login.html',{'a':''})

#settings
def settings(request):
   if login_check==True:
         return render(request, 'settings.html')
   else:
        return render(request,'login.html',{'a':''})


import mysql.connector
con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
mycursor=con.cursor()

# to buy stocks- checking whether the customer has sufficient funds
to buy the given no. of shares- if no, redirecting and if yes,'
# updating transaction and deducting from customer
def buy_direct(request):
con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
    mycursor=con.cursor()
    if login_check ==True:
        if request.method=="POST":

con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
            mycursor=con.cursor()
            stock=request.POST["option"]
            num_of=request.POST["num"]
            num_of=int(num_of)
            sql3="select * from signup_stocks"
            mycursor.execute(sql3)
            t=mycursor.fetchall()
```

```
            n=len(t)
            for i in range(n):
                if t[i][1]==stock:
                    vall=t[i][2]
                    val=float(vall)
            sql4="select * from signup_signup"
            mycursor.execute(sql4)
            w=mycursor.fetchall()

            p=len(w)
            for i in range(p):
                if w[i][0]==username:
                    baln=w[i][4]
                    bal=float(baln)
                #sql2="insert into signup_transaction
values("username,","None,","stock,","val*num_of*1.3,","True")"
            if val*num_of*1.3 > bal:

                disp="Insufficient Funds to complete transaction!"
                #sql2="insert into signup_transaction
values("username,","None,","stock,","val*num_of*1.3,","T")"
                return render(request,'buy.html',{'a':disp})
            else:
                new_bal=bal-(val*num_of*1.3)
                disp="You have "+str(new_bal) + " money left in your
StocksOnline wallet."
                sql2='insert into
signup_transaction(buyer,seller,amount,status,stock,shares) values
("{}","{}","{}","{}","{}","{}")'.format(username,None,val*num_of*1.3,
'y',stock,num_of)
                sql5='update signup_signup set som = {} where
username = "{}"'.format(new_bal,username)
                mycursor.execute(sql5)
                mycursor.execute(sql2)
                con.commit()
```

```
con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
                mycursor=con.cursor()

l1=transaction.objects.filter(buyer=username).values('amount','stock'
,'shares')

l2=transaction.objects.filter(seller=username).values('amount','stock
','shares')
                dl=[]
                if con.is_connected:
                    print('yes')
                sql15="select name from signup_stocks"
                mycursor.execute(sql15)
                n=mycursor.fetchall()
                l=len(n)
                for i in range(l):
                    sql8="select sum(shares) from signup_transaction
where buyer='{}' and stock='{}'".format(username,n[i][0])
                    sql9="select sum(shares) from signup_transaction
where seller='{}' and stock='{}'".format(username,n[i][0])
                    mycursor.execute(sql8)
                    a=mycursor.fetchall()
                    mycursor.execute(sql9)
                    b=mycursor.fetchall()
                    if a==[(None,)]:
                        bought=0
                    else:
                        bought=float(a[0][0])
                    if b==[(None,)]:
                        sold=0
                    else:
                        sold=float(b[0][0])
                    net=bought-sold
                    if net>0:
```

```
                        sql16="select
signup_stocks.symbol,signup_stocks.name,signup_stocks.last,signup_sto
cks.change,signup_stocks.change_percentage from signup_stocks where
name='{}'".format(n[i][0])
                        mycursor.execute(sql16)
                        g=mycursor.fetchall()
                        de=list(g[0])
                        d=de+[net]
                        print(d)
                        dl.append(d)

                return
render(request,'myhome.html',{'a':disp,'l1':l1,'l2':l2,'l3':dl})
        else:
            return render(request,'buy.html')
    else:
        return render(request,'login.html')

#to sell stock- calculating the amount of stock the person has by
going through their transaction history, and subsequently
#running an update statement to change balance and add transaction
def for_sale(request):
    if login_check==True:
        if request.method=="POST":

con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
            mycursor=con.cursor()
            stock=request.POST["option"]
            num_of=request.POST["num"]
            num=int(num_of)
            sql8="select sum(shares) from signup_transaction where
buyer='{}' and stock='{}'".format(username,stock)
            sql9="select sum(shares) from signup_transaction where
seller='{}' and stock='{}'".format(username,stock)
            mycursor.execute(sql8)
```

```
            a=mycursor.fetchall()
            mycursor.execute(sql9)
            b=mycursor.fetchall()
            if a==[(None,)]:
                bought=0
            else:
                bought=float(a[0][0])
            if b==[(None,)]:
                sold=0
            else:
                sold=float(b[0][0])
            net=bought-sold
            print(net)
            sql10="select last from signup_stocks where
name='{}'".format(stock)
            mycursor.execute(sql10)
            c=mycursor.fetchall()
            if c==[(None,)]:
                val=0
            else:
                val=float(c[0][0])

            if net >= num:
                add=val*num*1.3
                sql7='insert into
signup_transaction(buyer,seller,amount,status,stock,shares) values
("{}","{}","{}","{}","{}","{}")'.format(None,username,add,'y',stock,n
um)
                mycursor.execute(sql7)
                con.commit()
                sql4="select * from signup_signup"
                mycursor.execute(sql4)
                w=mycursor.fetchall()
                p=len(w)
                for i in range(p):
                    if w[i][0]==username:
```

```
                        bal_1=w[i][4]
                bal=float(bal_1)
                newbal=bal+add
                sql11="update signup_signup set som = {} where
username='{}'".format(newbal,username)
                mycursor.execute(sql11)
                con.commit()
                con.close()

con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
                mycursor=con.cursor()
                a="Sold!"

l1=transaction.objects.filter(buyer=username).values('amount','stock'
,'shares')

l2=transaction.objects.filter(seller=username).values('amount','stock
','shares')
                dl=[]
                if con.is_connected:
                    print('yes')
                sql15="select name from signup_stocks"
                mycursor.execute(sql15)
                n=mycursor.fetchall()
                l=len(n)
                for i in range(l):
                    sql8="select sum(shares) from signup_transaction
where buyer='{}' and stock='{}'".format(username,n[i][0])
                    sql9="select sum(shares) from signup_transaction
where seller='{}' and stock='{}'".format(username,n[i][0])
                    mycursor.execute(sql8)
                    a=mycursor.fetchall()
                    mycursor.execute(sql9)
                    b=mycursor.fetchall()
                    if a==[(None,)]:
```

```
                        bought=0
                    else:
                        bought=float(a[0][0])
                    if b==[(None,)]:
                        sold=0
                    else:
                        sold=float(b[0][0])
                    net=bought-sold
                    if net>0:
                        sql16="select
signup_stocks.symbol,signup_stocks.name,signup_stocks.last,signup_sto
cks.change,signup_stocks.change_percentage from signup_stocks where
name='{}'".format(n[i][0])
                        mycursor.execute(sql16)
                        g=mycursor.fetchall()
                        de=list(g[0])
                        d=de+[net]
                        print(d)
                        dl.append(d)
                sql19="select som from signup_signup where
username='{}'".format(username)
                mycursor.execute(sql19)
                d=mycursor.fetchall()
                num=d[0][0]
                a="Sold! You now have "+str(num)+" money in your
account."
                return
render(request,'myhome.html',{'a':a,'l1':l1,'l2':l2,'l3':dl})
            else:


                return render(request,'listing.html',{'a':'You do not
have sufficient amount of this stock to sell..'})
        else:
            return render(request,'listing.html')
    else:
```

```
        return render(request,'login.html')

#  to add money to account by running an update query, and loading
details to display on homepage
def change_details(request):
    if login_check==True:
        if request.method=="POST":

con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
            mycursor=con.cursor()
            top_up=request.POST['amount']
            add=int(top_up)
            sql4="select * from signup_signup"
            mycursor.execute(sql4)
            w=mycursor.fetchall()
            p=len(w)
            for i in range(p):
                if w[i][0]==username:
                    bal_1=w[i][4]
            bal=int(bal_1)
            newbal=bal+add
            sql11="update signup_signup set som = {} where
username='{}'".format(newbal,username)
            mycursor.execute(sql11)
            con.commit()
            con.close()

con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
            mycursor=con.cursor()

l1=transaction.objects.filter(buyer=username).values('amount','stock'
,'shares')
```

```
l2=transaction.objects.filter(seller=username).values('amount','stock
','shares')
            dl=[]
            if con.is_connected:
                print('yes')
            sql15="select name from signup_stocks"
            mycursor.execute(sql15)
            n=mycursor.fetchall()
            l=len(n)
            for i in range(l):
                sql8="select sum(shares) from signup_transaction
where buyer='{}' and stock='{}'".format(username,n[i][0])
                sql9="select sum(shares) from signup_transaction
where seller='{}' and stock='{}'".format(username,n[i][0])
                mycursor.execute(sql8)
                a=mycursor.fetchall()
                mycursor.execute(sql9)
                b=mycursor.fetchall()
                if a==[(None,)]:
                    bought=0
                else:
                    bought=float(a[0][0])
                if b==[(None,)]:
                    sold=0
                else:
                    sold=float(b[0][0])
                net=bought-sold
                if net>0:
                    sql16="select
signup_stocks.symbol,signup_stocks.name,signup_stocks.last,signup_sto
cks.change,signup_stocks.change_percentage from signup_stocks where
name='{}'".format(n[i][0])
                    mycursor.execute(sql16)
                    g=mycursor.fetchall()
                    de=list(g[0])
```

```
                    d=de+[net]
                    print(d)
                    dl.append(d)
            sql20="select som from signup_signup where
username='{}'".format(username)
            mycursor.execute(sql20)
            b=mycursor.fetchall()
            num=b[0][0]

            a='Money deposited! You now have '+str(num)+' money in
your account.'
            return
render(request,'myhome.html',{'a':a,'l1':l1,'l2':l2,'l3':dl})
        else:
            return render(request,'topup.html')
    else:
        return render(request,'login.html')

#  to change password by running an update query, and loading details
to display on homepage
def change_passwd(request):
    if login_check==True:
        if request.method=='POST':
            oldpd=request.POST['old']
            newpd=request.POST['password']
            sql13="select password from signup_signup where
username='{}'".format(username)
            mycursor.execute(sql13)
            print(newpd)
            e=mycursor.fetchall()
            older=e[0][0]
            if oldpd==older:
                sql12="update signup_signup set password = '{}' where
username = '{}'".format(newpd,username)
                mycursor.execute(sql12)
                con.commit()
```

```
                con.close()
            con.close()

con=mysql.connector.connect(host="localhost",user="root",passwd="root
",database="stocksonline")
            mycursor=con.cursor()

l1=transaction.objects.filter(buyer=username).values('amount','stock'
,'shares')

l2=transaction.objects.filter(seller=username).values('amount','stock
','shares')
            dl=[]
            if con.is_connected:
                print('yes')
            sql15="select name from signup_stocks"
            mycursor.execute(sql15)
            n=mycursor.fetchall()
            l=len(n)
            for i in range(l):
                sql8="select sum(shares) from signup_transaction
where buyer='{}' and stock='{}'".format(username,n[i][0])
                sql9="select sum(shares) from signup_transaction
where seller='{}' and stock='{}'".format(username,n[i][0])
                mycursor.execute(sql8)
                a=mycursor.fetchall()
                mycursor.execute(sql9)
                b=mycursor.fetchall()
                if a==[(None,)]:
                    bought=0
                else:
                    bought=float(a[0][0])
                if b==[(None,)]:
                    sold=0
                else:
                    sold=float(b[0][0])
```

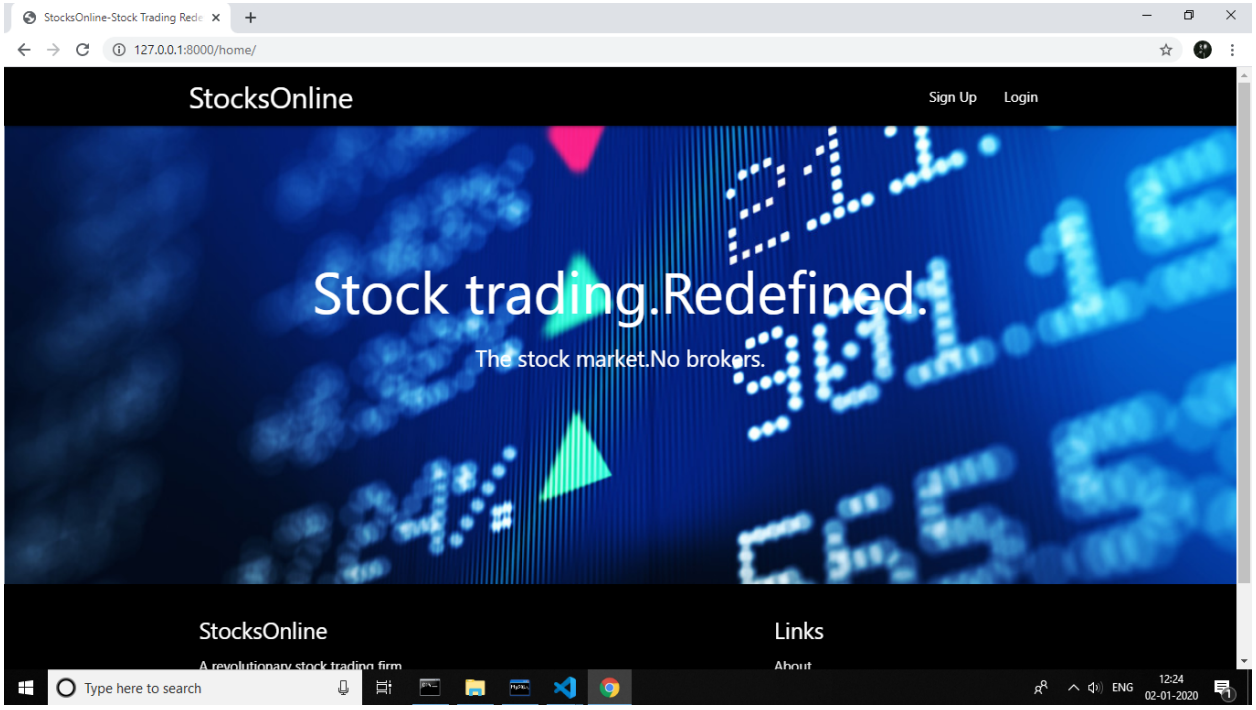
```
                net=bought-sold
                if net>0:
                    sql16="select
signup_stocks.symbol,signup_stocks.name,signup_stocks.last,signup_sto
cks.change,signup_stocks.change_percentage from signup_stocks where
name='{}'".format(n[i][0])
                    mycursor.execute(sql16)
                    g=mycursor.fetchall()
                    de=list(g[0])
                    d=de+[net]
                    print(d)
                    dl.append(d)

                return render(request,'myhome.html',{'a':'Password
changed!','l1':l1,'l2':l2,'l3':dl})
            else:
                return render(request,'changepasswd.html')
        else:
            return render(request,'changepasswd.html')
    else:
        return render(request,'login.html')

#for logging out
def logout(request):
    global login_check
    if login_check==True:

        login_check=False
        return render(request,'stocks.html')
    else:
        return render(request,'login.html')

def about(request):
    return render(request,'about.html')
```

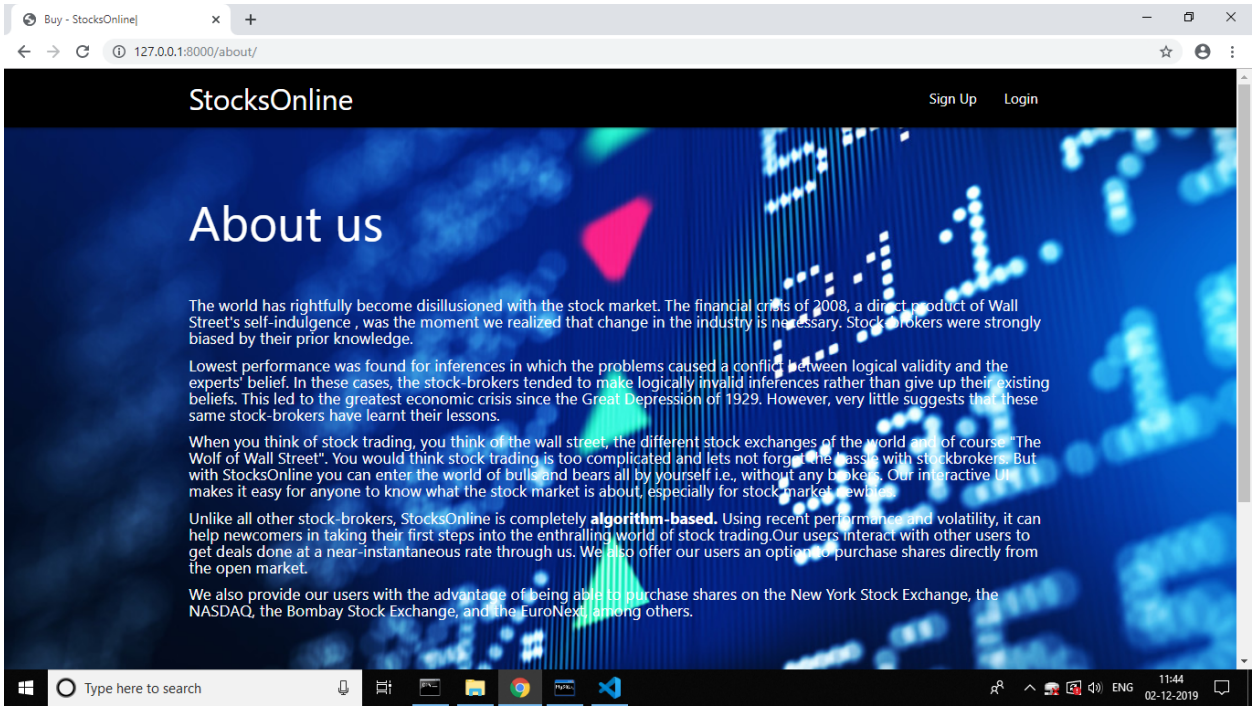
---

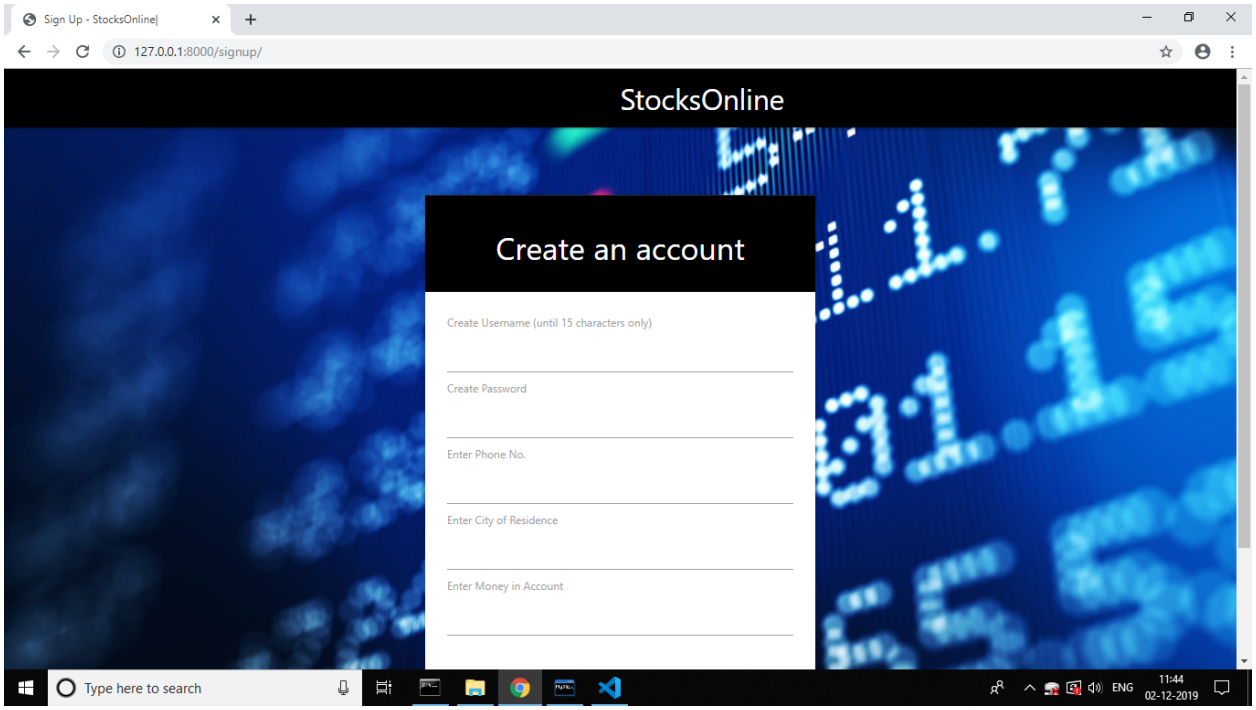
signup/views.py

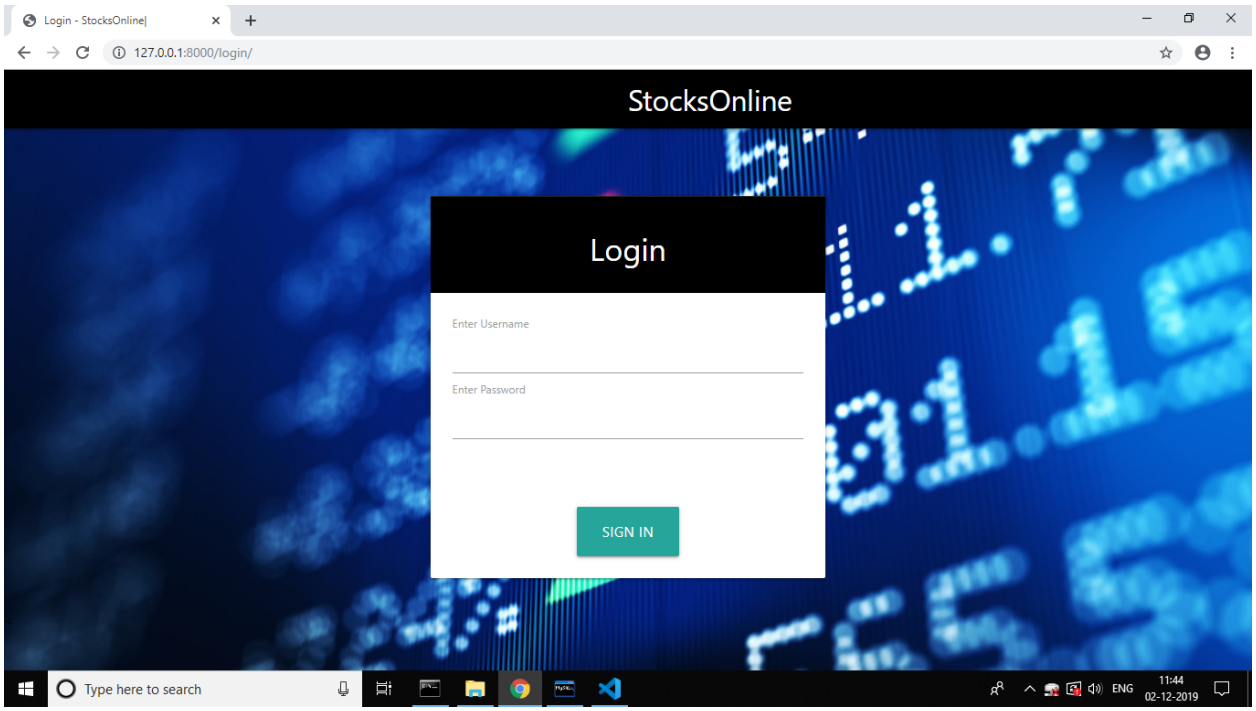
```
from django.shortcuts import render

# Create your views here.

#to sign up
from signup.models import signup
def signup_1(request):
    if request.method=="POST":
        username=request.POST["username"]
        password=request.POST["password"]
        phoneno=request.POST["phone"]
        city=request.POST["city"]
        som=request.POST["som"]

        lst=signup()
        lst.username=username
        lst.password=password
        lst.phoneno=phoneno
        lst.city=city
        lst.som=som
        lst.save()
        return render(request,'welcome.html')
    else:
        return render(request,'signup.html')

def home(request):
    return render(request, 'stocks.html')
```

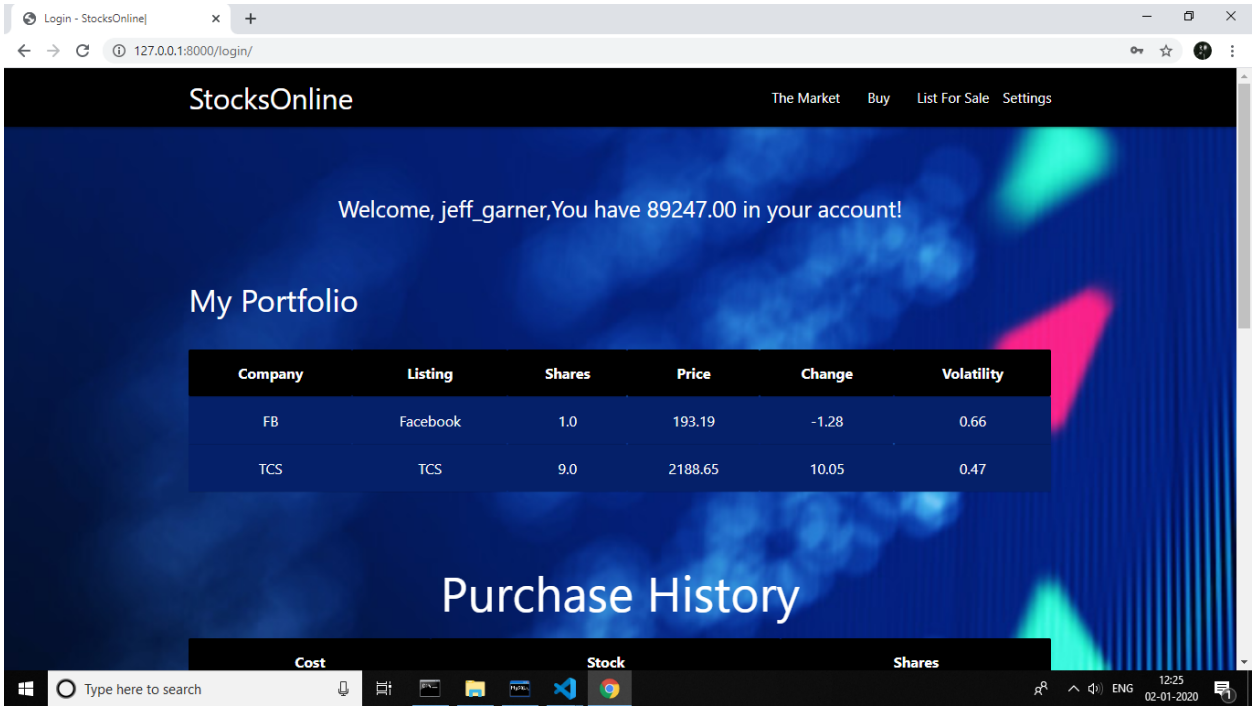
Screenshots

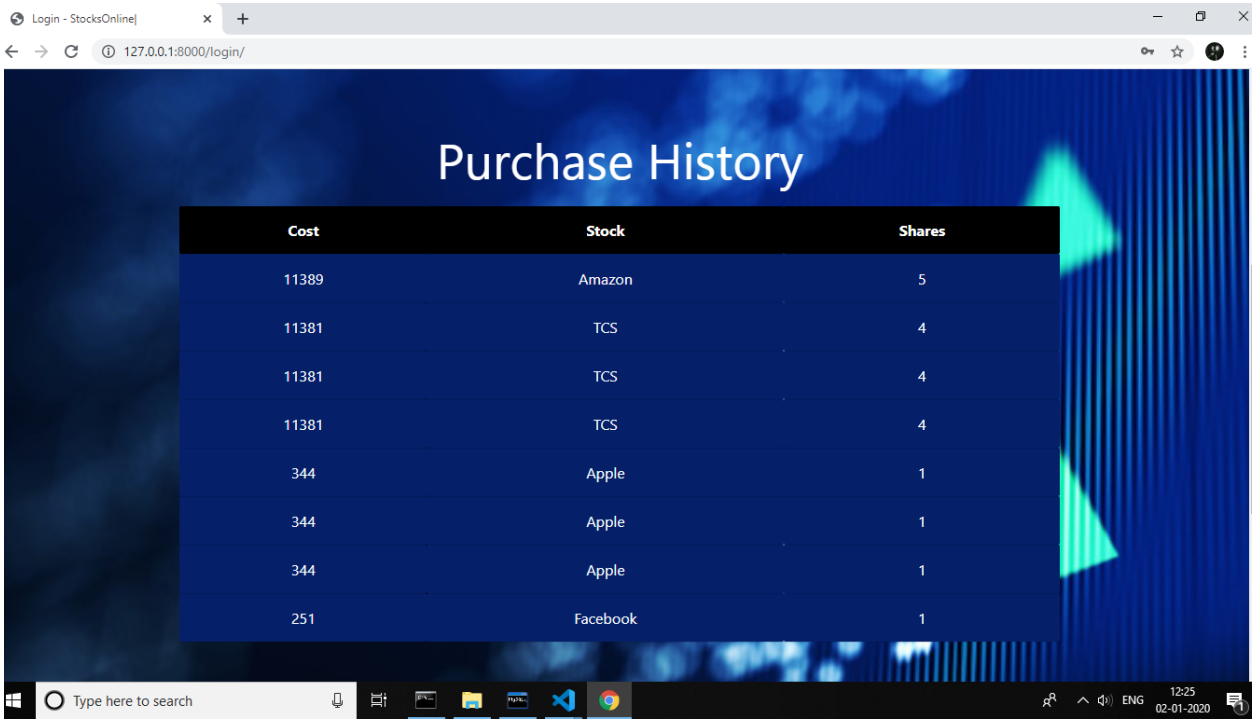


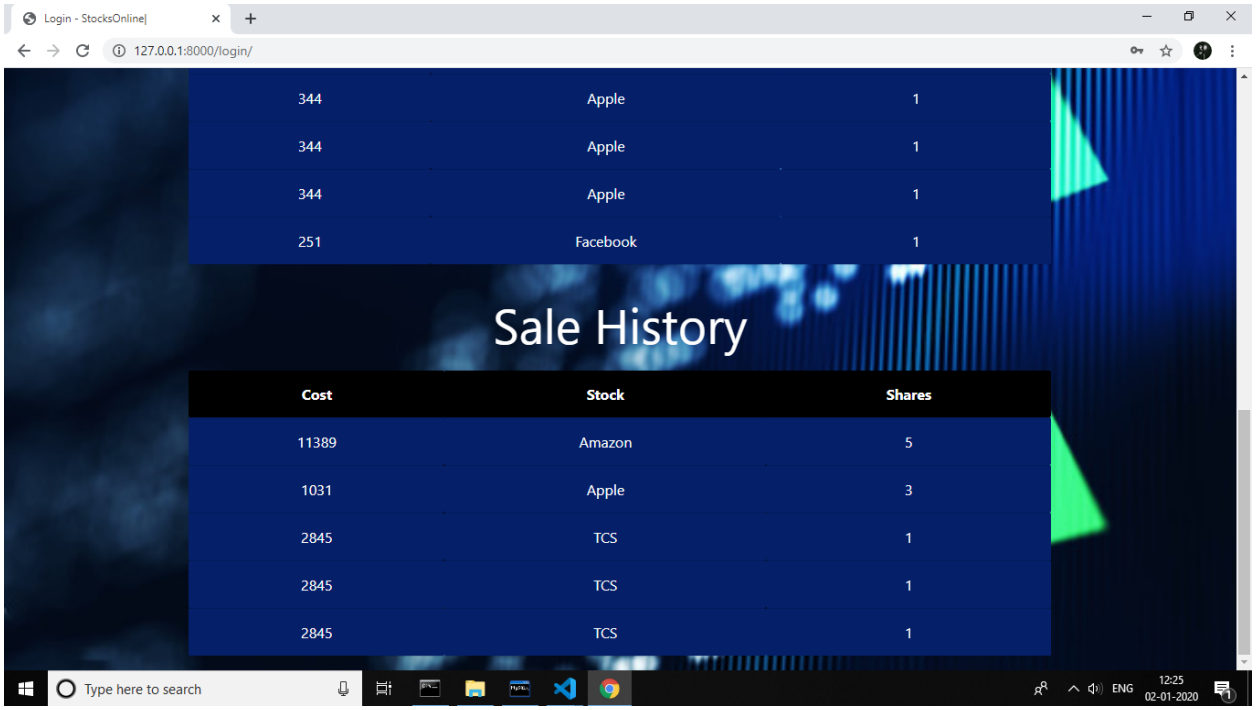


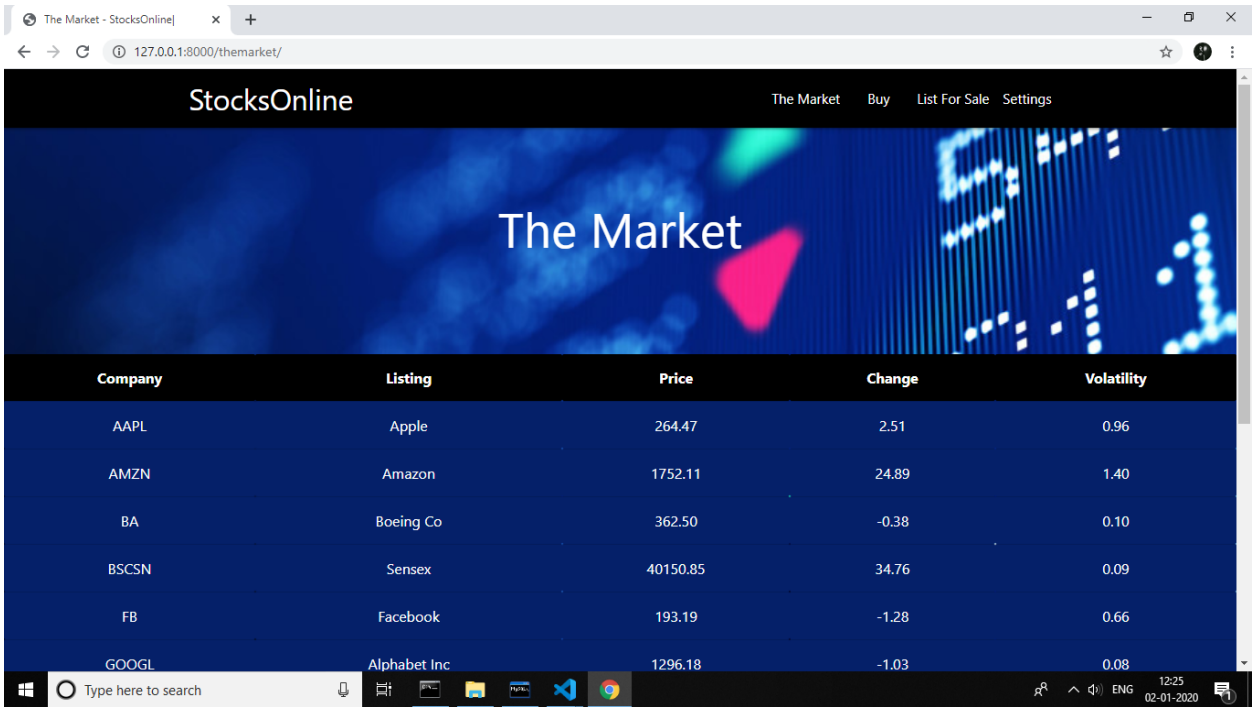

Sign Up - StocksOnline
127.0.0.1:8000/signup/
StocksOnline
Create an account
Create Username (until 15 characters only)
Create Password
Enter Phone No.
Enter City of Residence
Enter Money in Account
Type here to search
ENG
11:44
02-12-2019

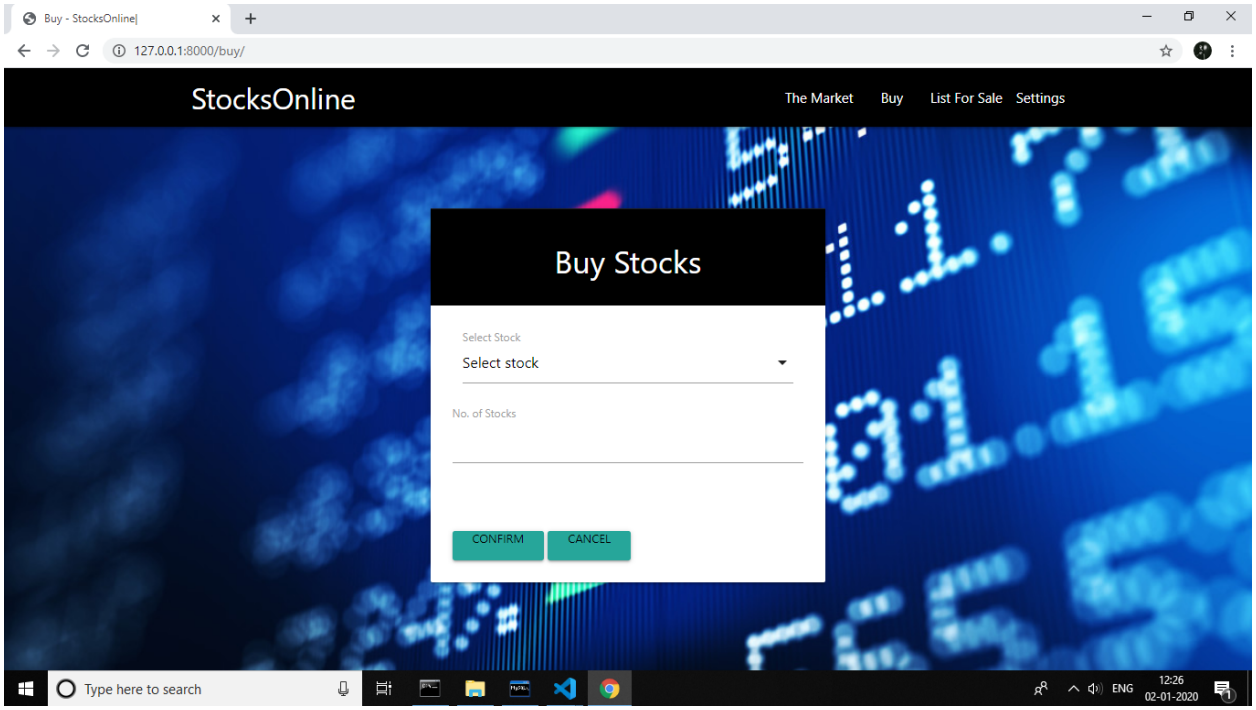

Login - StocksOnline
127.0.0.1:8000/login/
StocksOnline
Login
Enter Username
Enter Password
SIGN IN
Type here to search
ENG
11:44
02-12-2019

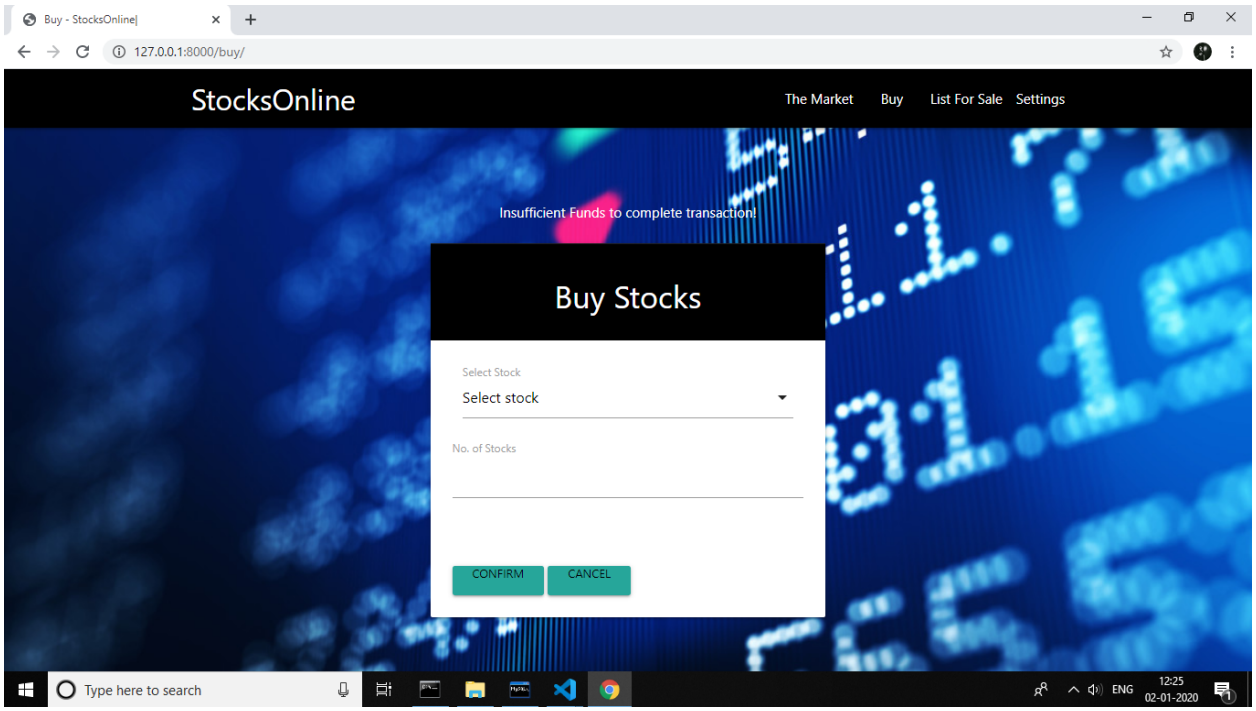

Login - StocksOnline
127.0.0.1:8000/login/
StocksOnline
The Market
Buy
List For Sale
Settings
Welcome, jeff_garner,You have 89247.00 in your account!
My Portfolio
Company
Listing
Shares
Price
Change
Volatility
FB
Facebook
1.0
193.19
-1.28
0.66
TCS
TCS
9.0
2188.65
10.05
0.47
Purchase History
Cost
Stock
Shares
Type here to search
ENG
12:25
02-01-2020

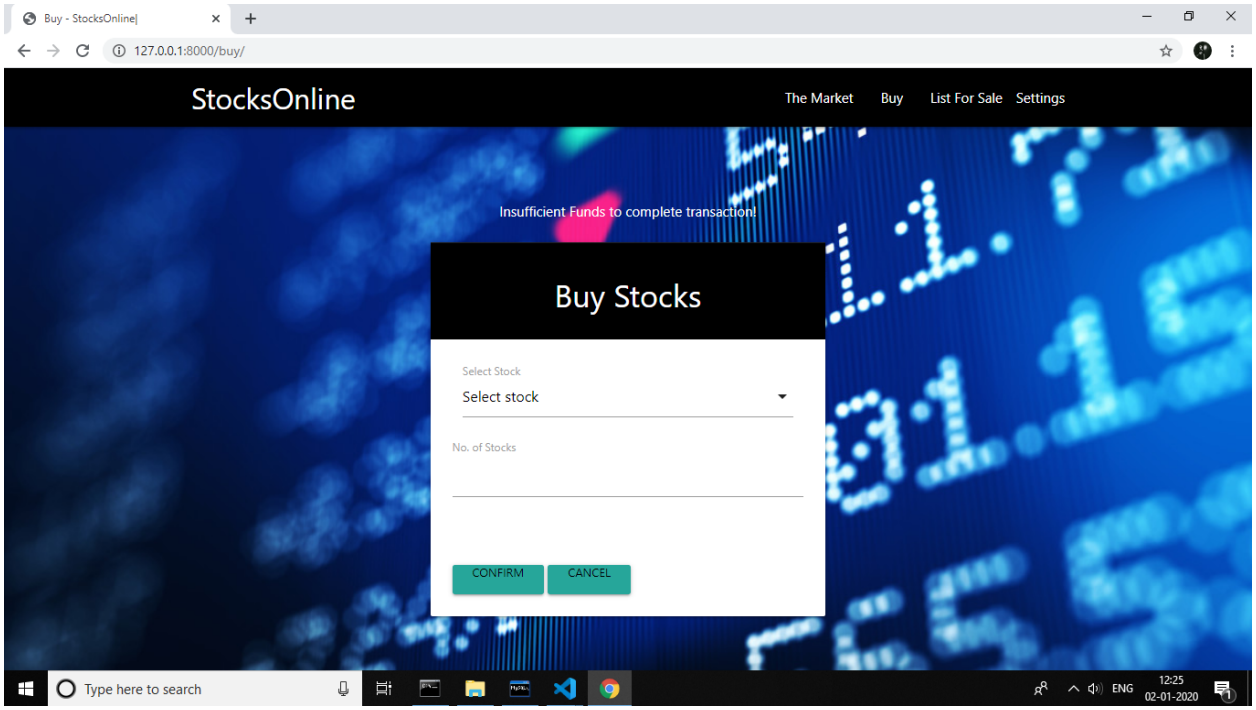

Login - StocksOnline
127.0.0.1:8000/login/
Purchase History
Cost
Stock
Shares
11389
Amazon
5
11381
TCS
4
11381
TCS
4
11381
TCS
4
344
Apple
1
344
Apple
1
344
Apple
1
251
Facebook
1
Type here to search
ENG
12:25
02-01-2020

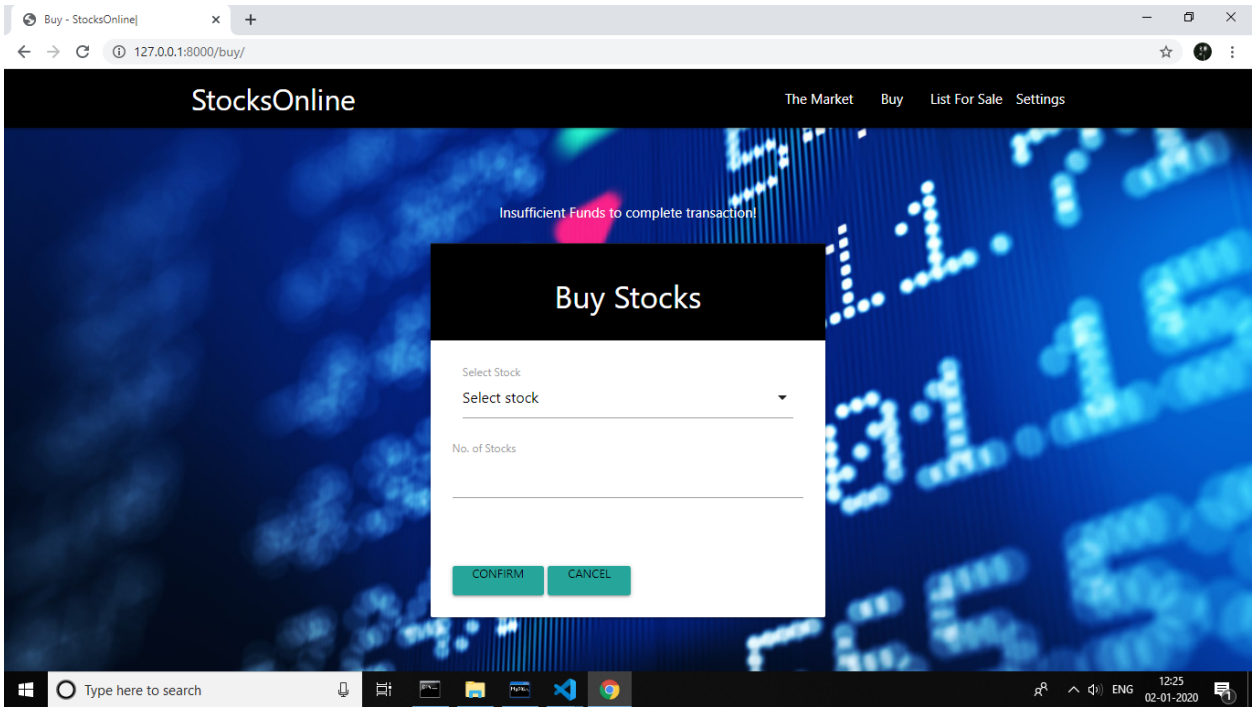

Login - StocksOnline
127.0.0.1:8000/login/
344 Apple 1
344 Apple 1
344 Apple 1
251 Facebook 1
Sale History
Cost Stock Shares
11389 Amazon 5
1031 Apple 3
2845 TCS 1
2845 TCS 1
2845 TCS 1
Type here to search
ENG
12:25
02-01-2020

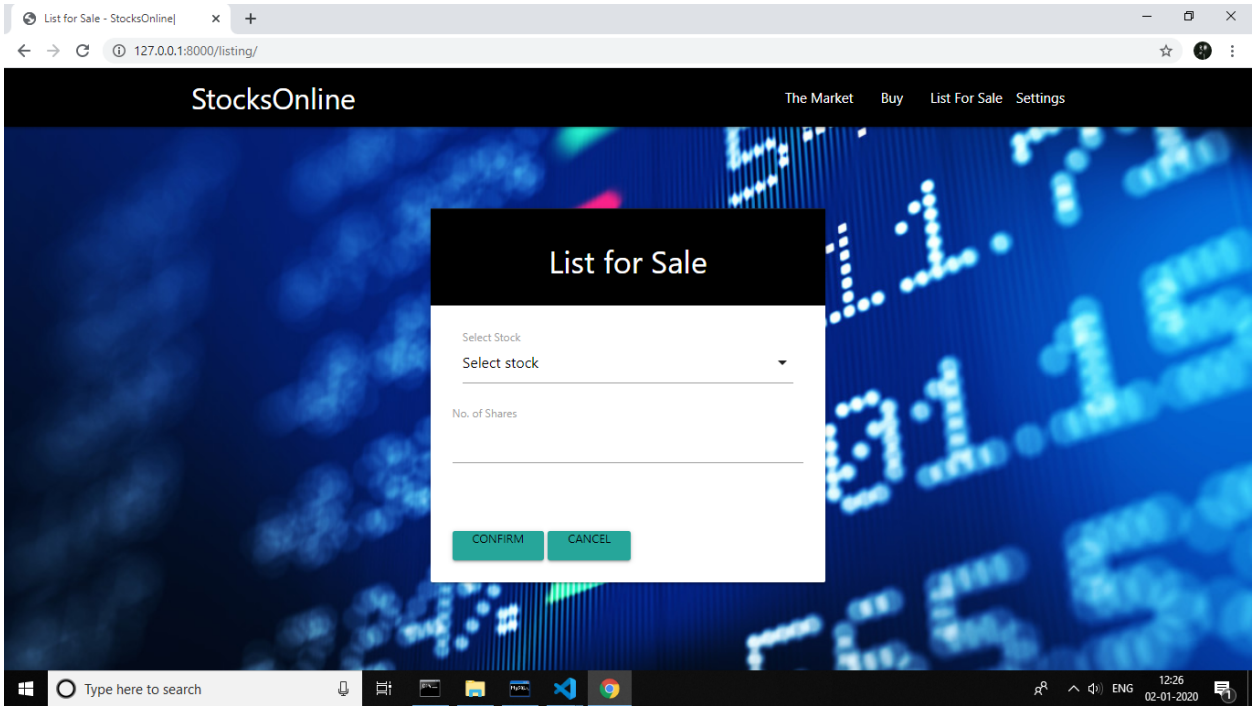

The Market - StocksOnline
127.0.0.1:8000/themarket/
StocksOnline
The Market Buy List For Sale Settings
The Market
Company Listing Price Change Volatility
AAPL Apple 264.47 2.51 0.96
AMZN Amazon 1752.11 24.89 1.40
BA Boeing Co 362.50 -0.38 0.10
BSCSN Sensex 40150.85 34.76 0.09
FB Facebook 193.19 -1.28 0.66
GOOGL Alphabet Inc 1296.18 -1.03 0.08
Type here to search
ENG
12:25
02-01-2020

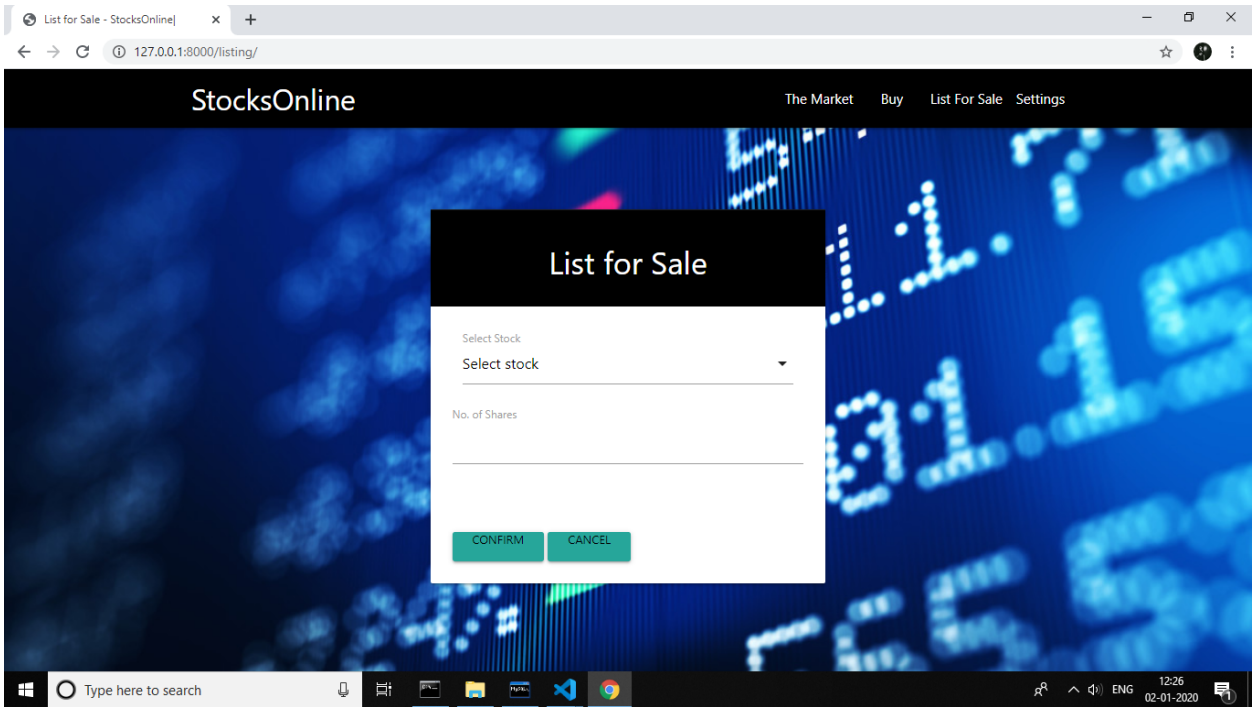

Buy - StocksOnline
127.0.0.1:8000/buy/
StocksOnline
The Market
Buy
List For Sale
Settings
Buy Stocks
Select Stock
Select stock
No. of Stocks
CONFIRM
CANCEL
Type here to search
ENG
12:26
02-01-2020

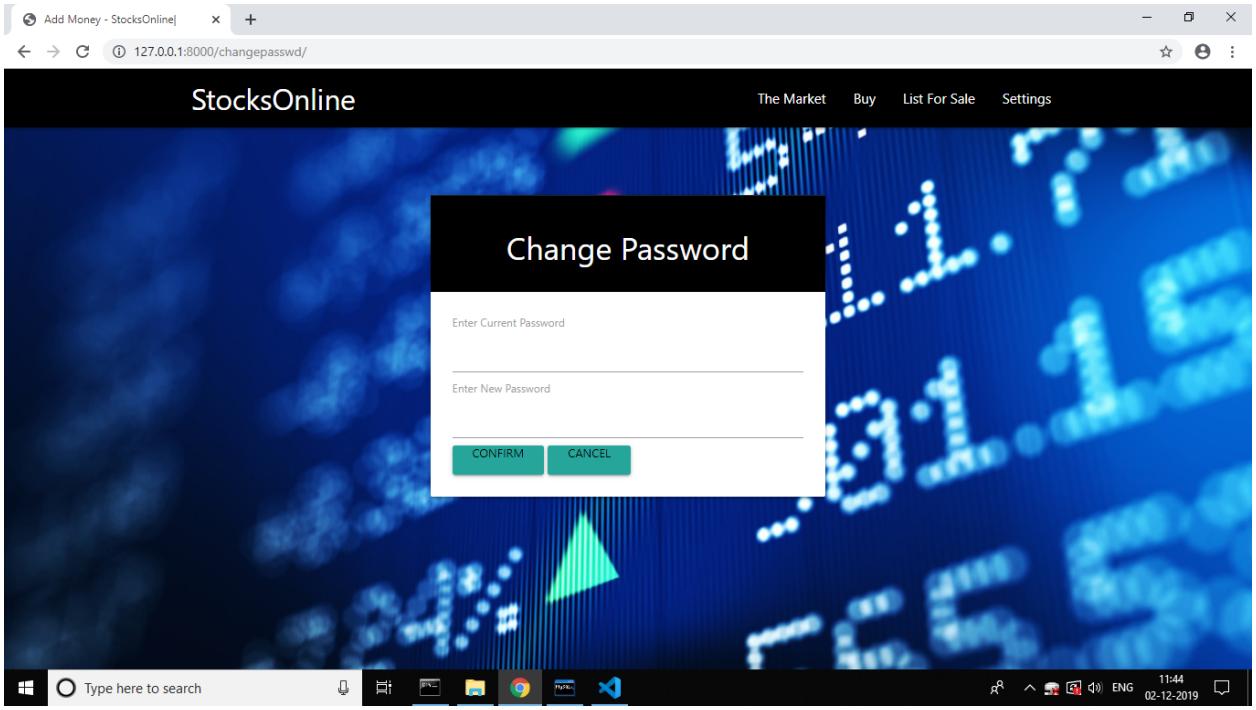

Buy - StocksOnline
127.0.0.1:8000/buy/
StocksOnline
The Market
Buy
List For Sale
Settings
Insufficient Funds to complete transaction!
Buy Stocks
Select Stock
Select stock
No. of Stocks
CONFIRM
CANCEL
Type here to search
ENG
12:25
02-01-2020

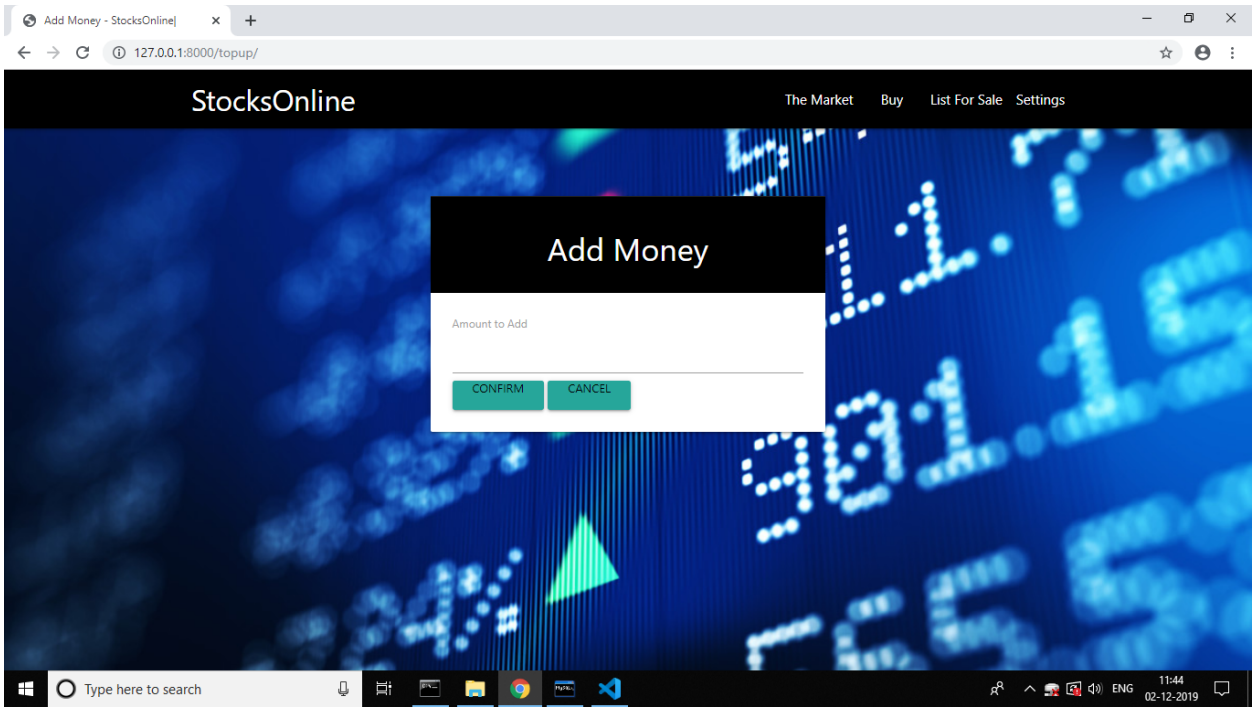

Buy - StocksOnline
127.0.0.1:8000/buy/
StocksOnline
The Market
Buy
List For Sale
Settings
Insufficient Funds to complete transaction!
Buy Stocks
Select Stock
Select stock
No. of Stocks
CONFIRM
CANCEL
Type here to search
ENG
12:25
02-01-2020


Buy - StocksOnline
127.0.0.1:8000/buy/
StocksOnline
The Market
Buy
List For Sale
Settings
Insufficient Funds to complete transaction!
Buy Stocks
Select Stock
Select stock
No. of Stocks
CONFIRM
CANCEL
Type here to search
ENG
12:25
02-01-2020


List for Sale - StocksOnline|
127.0.0.1:8000/listing/
StocksOnline
The Market
Buy
List For Sale
Settings
List for Sale
Select Stock
Select stock
No. of Shares
CONFIRM
CANCEL
Type here to search
ENG
12:26
02-01-2020


List for Sale - StocksOnline|
127.0.0.1:8000/listing/
StocksOnline
The Market
Buy
List For Sale
Settings
List for Sale
Select Stock
Select stock
No. of Shares
CONFIRM
CANCEL
Type here to search
ENG
12:26
02-01-2020


Add Money - StocksOnline
127.0.0.1:8000/changepasswd/
StocksOnline
The Market
Buy
List For Sale
Settings
Change Password
Enter Current Password
Enter New Password
CONFIRM
CANCEL
Type here to search
ENG
11:44
02-12-2019


Add Money - StocksOnline
127.0.0.1:8000/topup/
StocksOnline
The Market
Buy
List For Sale
Settings
Add Money
Amount to Add
CONFIRM
CANCEL
Type here to search
ENG
11:44
02-12-2019

Scope for Improvement

- Implementation of a inter-customer trading of stocks, i.e, allowing customers to directly sell to each other.
- Creation of an administrator account with special privileges like updating stock prices, adding new Publicly Listed Corporations to trade.
- Making a more efficient method to reload the dynamic values every time on entering the home page.

References

- https://www.geeksforgeeks.org/
- https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/stock-market/
- https://www.investopedia.com/university/stocks/stocks3.asp