Vidya Vikas Education Trust's Technical Campus

Universal College Of Engineering, Vasai (E).

Department Of Computer Engineering

Skill Based Lab Course (Python Programming)

MiniProject

Title:

Stock Portfolio Website Using Django

Name	:	**VIVEK. SHIVAKUMAR. HOTTI**
Sr No	:	**31**
Class	:	**SE-COMPS-A**
Semester	:	**IV**
Group No	:	**23**
Under the guidance of	:	**Professor John Kenny**

I N D E X

Name: Vivek S Hotti **Roll:** 31 **Class:** SE-Sem4 **Div.:** A

Group No: 23

1. Title, Aim & Theory

a) ***Title:*** Stock Portfolio Website Using Django

b) ***Aim:*** To create a website using Django that helps user take track of his stocks and keep him updated about the market.

c) ***Theory:***

DJANGO INTRODUCTION

What is Django?

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so the user can focus on writing the userr app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

What does Django code look like?

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or another client). When a request is received the application works out what is needed based on the URL and possibly 3 information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template. Django web applications typically group the code that handles each of these steps into separate files:

- **URLs:** While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

- **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
- **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- **Templates:** A template is a text file defining the structure or layer of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML

Rendering data (HTML templates)

Template systems allow the user to specify the structure of an output document, using placeholders for data that will be filled in when a page is generated. Templates are often used to create HTML, but can also create other types of documents. Django supports both its native templating system and another popular Python library called Jinja2 out of the box (it can also be made to support other systems if needed).

What is the Django development environment?

The development environment is an installation of Django on the userr local computer that the user can use for developing and testing Django apps prior to deploying them to a production environment. The main tools that Django itself provides are a set of Python scripts for creating and working with Django projects, along with a simple development web server that the user can use to test local (i.e., on the userr computer, not on an external web server) Django web applications on the userr computer's web browser.

2. Our Project's Abstract

- Stock Portfolio Website is a web page created using Django.
- The idea behind this project is to have the ability to monitor the stocks the user is interested in at one place.
- This makes it easy to keep up with the rise or drop of a stock the user is interested in or stocks the user are holding.
- For a new user interested in investing in the stock market having no prior knowledge, we have a news section which gives the users some knowledge about the stocks in the market.
- Users can add stocks and delete stocks to their portfolio by entering the stock ticker symbol in the given input box.
- Users can instantaneously check Realtime value and other details of a stock by entering the ticker symbol on homepage.

3. Different URL's

(1) *Homepage*:

The first webpage to be displayed after our website is opened by the user is the homepage. User is greeted along with giving what is the use of other webpages / modules. The homepage basically contains instructions that help the user get familiarize with the console.

(2) *DevTeam*:

The devteam section helps the user to get familiarized with the developers and contains the developers' profiles in case he/she needs to contact them.

(3) *Add Stock*:

This section helps in addition of a 'new' stock to the user's portfolio. All the data is fetched live from the market and the values are either form the live market or previous close, whichever is earlier.

(4) *Delete Stock*:

This section lists all the user's active stocks in the portfolio, and gives the user the choice to delete a stock from the user's portfolio. Although this can also be done through the active stocks section.

(5) *Stock News*:

This section keeps the user updated of the all the highs and the lows of the stock market along with tips that gives the user an edge over other, so that the user is kept up to date with the stock market.

(6) *Get Quotes Section*:

The user maybe able to see a search bar at the top right corner of the nav bar, with a Search button in green besides. This will help the user in searching for a stock using the "ticker symbol". It will fetch details like the company name, stock price, previous close, market cap, year-to-date change, 52week high and 52 weeks low

4. Initiating the Project

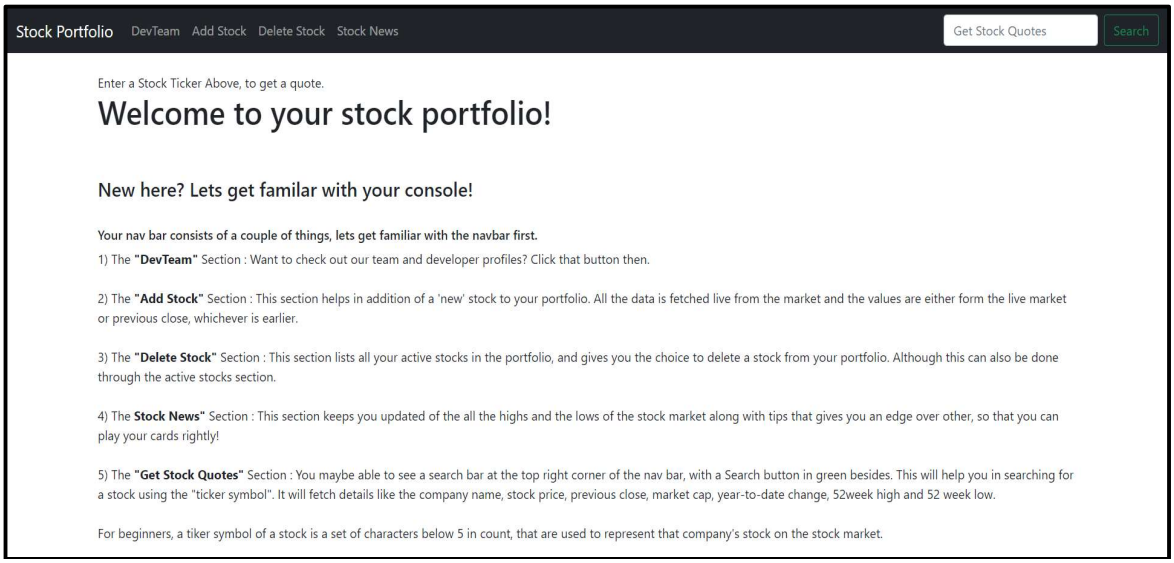
```
vivek hotti@LAPTOP-QQV9BLER MINGW64 ~
$ pwd
/c/Users/Vivek hotti

vivek hotti@LAPTOP-QQV9BLER MINGW64 ~
$ cd . .

vivek hotti@LAPTOP-QQV9BLER MINGW64 /c/Users
$ cd . .

vivek hotti@LAPTOP-QQV9BLER MINGW64 /c
$ cd . .

vivek hotti@LAPTOP-QQV9BLER MINGW64 /c
$ cd djangostock

vivek hotti@LAPTOP-QQV9BLER MINGW64 /c/djangostock
$ source venv/Scripts/activate

(venv)
vivek hotti@LAPTOP-QQV9BLER MINGW64 /c/djangostock
$ cd stocks

(venv)
vivek hotti@LAPTOP-QQV9BLER MINGW64 /c/djangostock
$ ls
db.sqlite3     Procfile        requirements.txt     stocks/
manage.py*     quotes/         staticfiles/

(venv)
vivek hotti@LAPTOP-QQV9BLER MINGW64 /c/djangostock
$ python manage.py runserver
Watching for file changes with State Reloader
[29/Apr/2021 16:00:15] "GET / HTTP/1.1" 200 4438
```

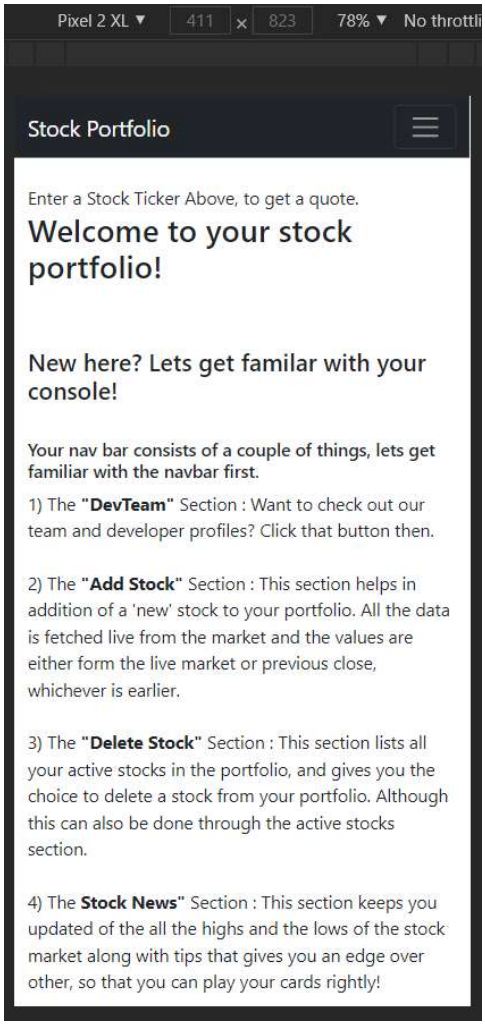
then proceed to:

http://localhost:8000/

5. Project Screenshots

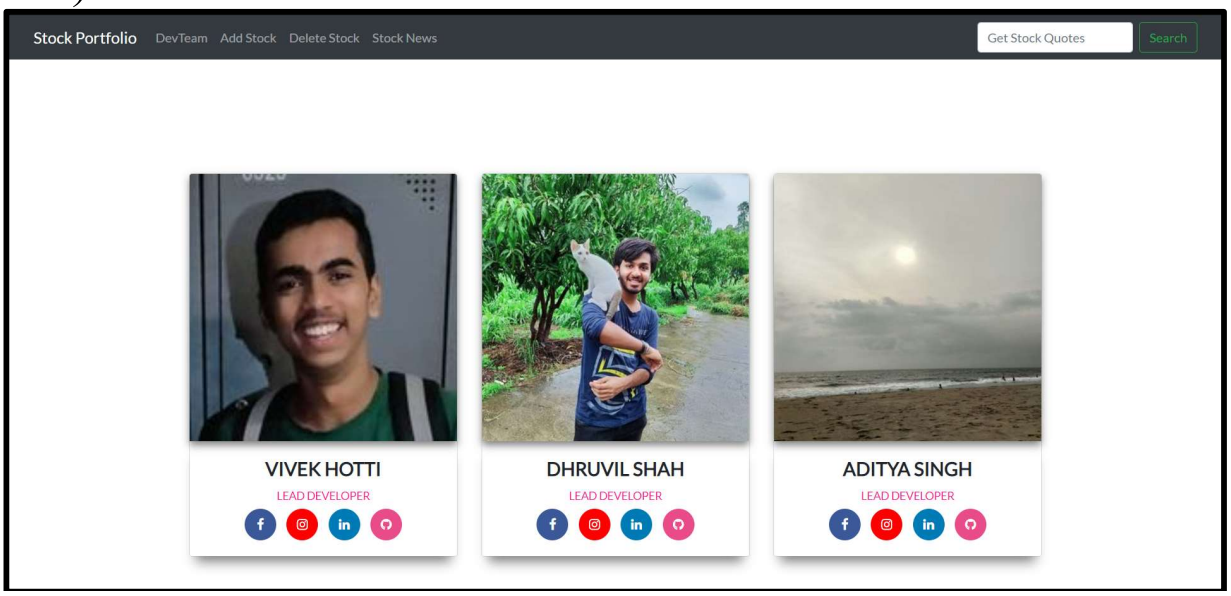
1) Homepage:

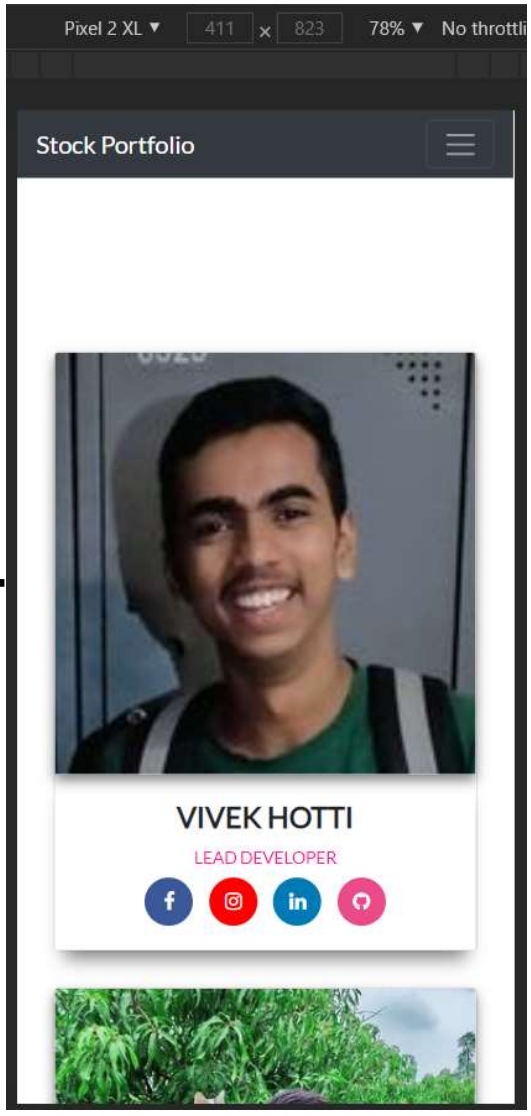




Responsive Layout of the Homepage

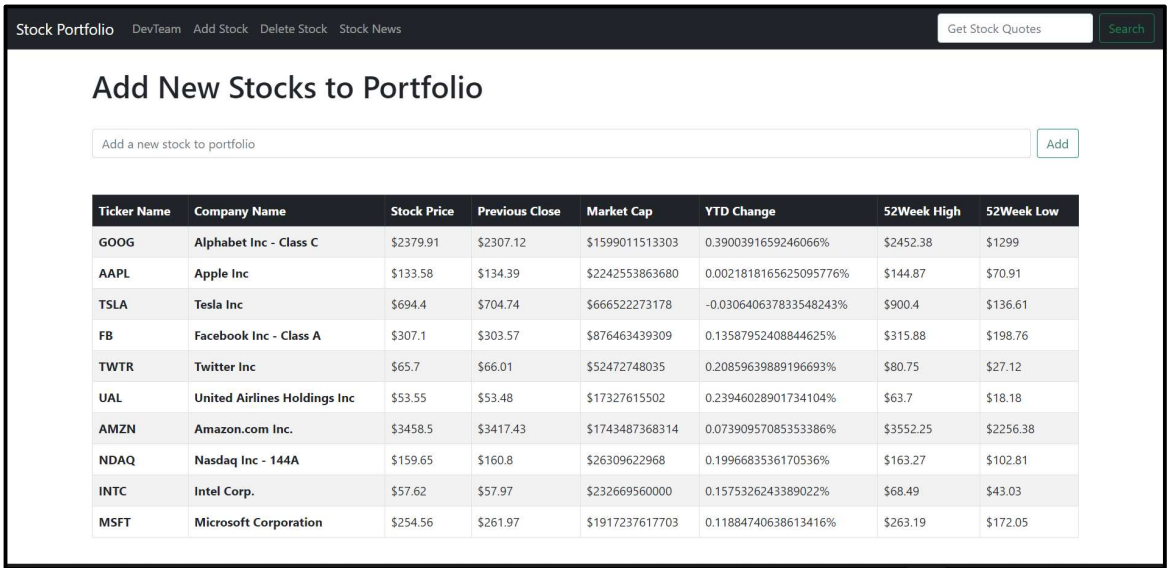
2) Dev Team:





Responsive Layout of the DevTeam Page

3) Add Stock:

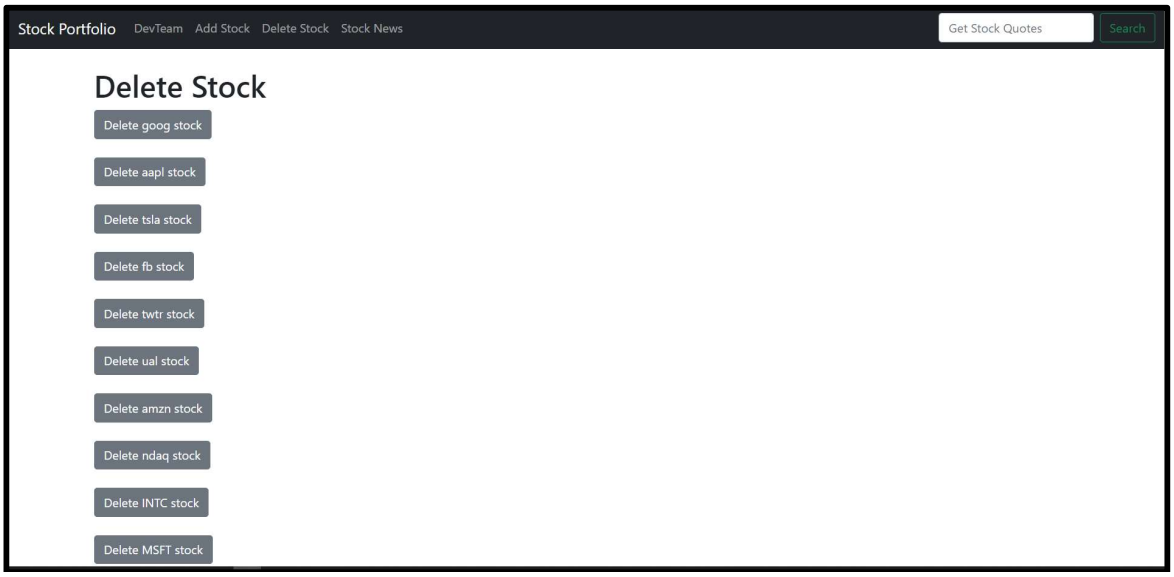


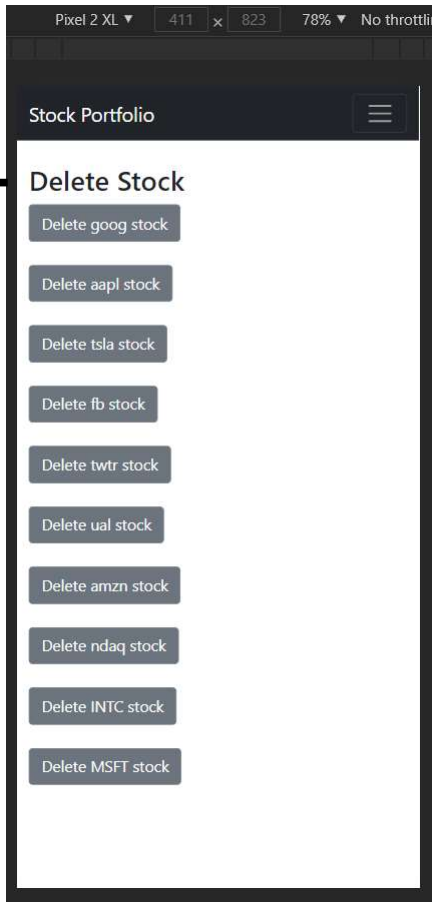
Ticker Name	Company Name	Stock Price	Previous Close	Market Cap	YTD Change	52Week High	52Week Low
GOOG	Alphabet Inc - Class C	$2379.91	$2307.12	$1599011513303	0.3900391659246066%	$2452.38	$1299
AAPL	Apple Inc	$133.58	$134.39	$2242553863680	0.0021818165625095776%	$144.87	$70.91
TSLA	Tesla Inc	$694.4	$704.74	$666522273178	-0.030640637833548243%	$900.4	$136.61
FB	Facebook Inc - Class A	$307.1	$303.57	$876463439309	0.13587952408844625%	$315.88	$198.76
TWTR	Twitter Inc	$65.7	$66.01	$52472748035	0.20859639889196693%	$80.75	$27.12
UAL	United Airlines Holdings Inc	$53.55	$53.48	$17327615502	0.23946028901734104%	$63.7	$18.18
AMZN	Amazon.com Inc.	$3458.5	$3417.43	$1743487368314	0.07390957085353386%	$3552.25	$2256.38
NDAQ	Nasdaq Inc - 144A	$159.65	$160.8	$26309622968	0.1996683536170536%	$163.27	$102.81
INTC	Intel Corp.	$57.62	$57.97	$232669560000	0.1575326243389022%	$68.49	$43.03
MSFT	Microsoft Corporation	$254.56	$261.97	$1917237617703	0.1188474063861 3416%	$263.19	$172.05

The user enters a new stock ticker inside this input box and hits add.

Details regarding that stock are immediately added to the below table and are directly fetched from the stocks API.

4) Delete Stock:



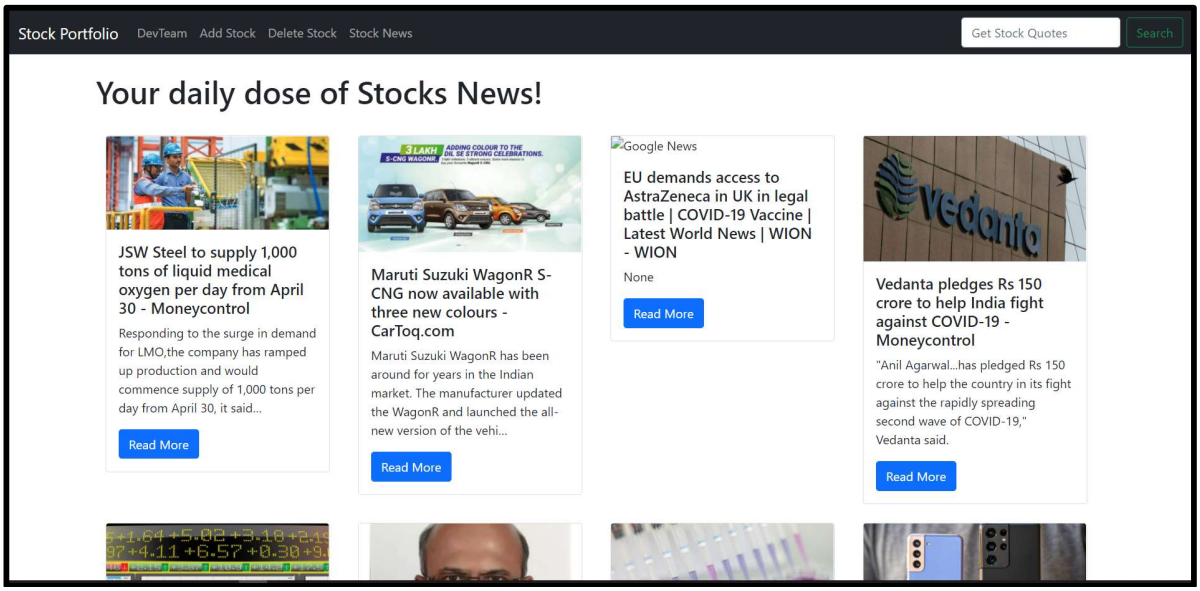


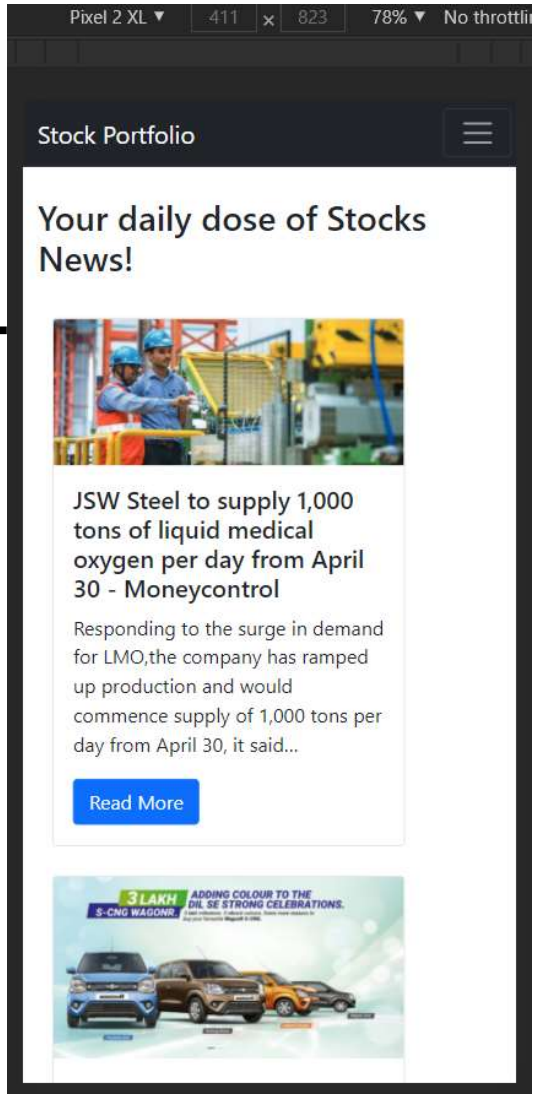
Responsive Layout of the Delete Stocks Page

Each and every added stock in the add stock page has a respective delete stock button in the Delete Stocks page.

When the user clicks this button, the respective stock gets automatically deleted from the user's stock portfolio in the Add Stocks Page.

5) Stock News:





Responsive Layout of the Stock News Page

6. Raw Code

admin.py

```
from django.contrib import admin
from .models import Stock

admin.site.register(Stock)
```

apps.py

```
from django.apps import AppConfig


class QuotesConfig(AppConfig):
    name = 'quotes'
```

forms.py

```
from django import forms
from .models import Stock

class StockForm(forms.ModelForm):
	class Meta:
		model = Stock
		fields = ["ticker"]
```

models.py

```
from django.db import models

class Stock(models.Model):
	ticker = models.CharField(max_length=10)

	def __str__(self):
		return self.ticker
```

tests.py

```
from django.test import TestCase


# Create your tests here.
```

urls.py

```
from django.urls import path
from . import views

urlpatterns = [
	path('', views.home, name="home"),
	path('about.html', views.about, name="about"),
	path('add_stock.html', views.add_stock, name="add_stock"),
	path('delete/<stock_id>', views.delete, name="delete"),
	path('delete_stock.html', views.delete_stock,
name="delete_stock"),
    path('news.html', views.news, name="news")
]
```

views.py

```
from django.shortcuts import render, redirect
from .models import Stock
from .forms import StockForm
from django.contrib import messages
from django.shortcuts import render


# pk_e4c317db75c8461fb6c92a1ea3c0e496

def home(request):
	import requests
	import json

	if request.method == 'POST':
		ticker = request.POST['ticker']
		api_request =
requests.get("https://cloud.iexapis.com/stable/stock/" + ticker +
"/quote?token=pk_e4c317db75c8461fb6c92a1ea3c0e496")
		try:
			api = json.loads(api_request.content)
		except Exception as e:
			api = "Error..."
		return render(request, 'home.html', {'api' : api})

	else:
		return render(request, 'home.html', {'ticker' : "Enter a
Stock Ticker Above, to get a quote."})


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


def add_stock(request):
	import requests
	import json

	if request.method == 'POST':
```

```
                form = StockForm(request.POST or None)

                if form.is_valid():
                        form.save()
                        messages.success(request, ('Stock Has Been Added
Successfully!'))
                        return redirect('add_stock')



        else:
                ticker = Stock.objects.all()
                output = []
                for ticker_item in ticker:
                        api_request =
requests.get("https://cloud.iexapis.com/stable/stock/" + str(ticker_item)
+ "/quote?token=pk_e4c317db75c8461fb6c92a1ea3c0e496")

                        try:
                                api = json.loads(api_request.content)
                                output.append(api)
                        except Exception as e:
                                api = "Error..."

                return render(request, 'add_stock.html', {'ticker':
ticker, 'output': output})

def delete(request, stock_id):
        item = Stock.objects.get(pk=stock_id)
        item.delete()
        messages.success(request, ("Stock Has Been Deleted
Successfully"))
        return redirect(delete_stock)

def delete_stock(request):
        ticker = Stock.objects.all()
        return render(request, 'delete_stock.html', {'ticker': ticker})

def news(request):
    import requests
    import json
    news_api_request=requests.get("https://newsapi.org/v2/top-
headlines?country=in&category=business&apiKey=7fbe44ccbd564351af262ff843f
db6d5")
    api=json.loads(news_api_request.content)
    return render(request,'news.html',{'api':api})
```

requirements.txt

```
asgiref==3.3.1
certifi==2020.12.5
chardet==4.0.0
dj-database-url==0.5.0
Django==3.1.6
django-heroku==0.3.1
gunicorn==20.0.4
```

```
idna==2.10
psycopg2==2.8.6
python-decouple==3.4
pytz==2021.1
requests==2.25.1
sqlparse==0.4.1
urllib3==1.26.3
whitenoise==5.2.0
```

about.html

```
{% extends 'base.html' %}
{% block content %}
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,400;1,700
&display=swap" rel="stylesheet">

<div class="wrapper">
        <div class="container">
        <div class="row">
                <div class="col-md-6 col-lg-4">
                        <div class="card mx-30">
                          <img src="https://instagram.fbom16-
1.fna.fbcdn.net/v/t51.2885-
19/s150x150/131302578_858380791651445_7003223779733779034_n.jpg?tp=1&_nc_
ht=instagram.fbom16-
1.fna.fbcdn.net&_nc_ohc=InENJIYiPtEAX_5thj0&edm=ABfd0MgAAAAA&ccb=7-
4&oh=012329bdfdbefa0e8333fde7f050ac19&oe=609D0637&_nc_sid=7bff83"
class="card-img-top" alt="...">
                        <div class="card-body">
                                <h5 class="card-title">Vivek Hotti</h5>
                                <h6>Lead Developer</h6>
                                        <div class="socials">

        <a href="https://www.facebook.com/vivek.hotti.9/"><i class="fa fa-
facebook"></i></a>

        <a href="https://www.instagram.com/vivek_hotti/"><i class="fa fa-
instagram"></i></a>

        <a href="https://www.linkedin.com/in/vivekhotti/"><i class="fa
fa-linkedin"></i></a>

        <a href="https://github.com/Vivek-Hotti"><i class="fa fa-
github"></i></a>
                                                                </div>
                                        </div>
                                </div>
                        </div>
                        <div class="col-md-6 col-lg-4">
```

```
                                    <div class="card mx-30">
                                      <img src="https://instagram.fbom16-
1.fna.fbcdn.net/v/t51.2885-
19/s320x320/123970665_1000771497013961_3687167906004879556_n.jpg?tp=1&_nc
_ht=instagram.fbom16-
1.fna.fbcdn.net&_nc_ohc=45jlXArbns0AX_OWnFh&edm=ABfd0MgAAAAA&ccb=7-
4&oh=aa2d80ddd8abed80f37a00c6b9bc69b2&oe=609C7847&_nc_sid=7bff83"
class="card-img-top" alt="...">
                                      <div class="card-body">
                                            <h5 class="card-
title">Dhruvil Shah</h5>
                                            <h6>Lead Developer</h6>
                                                  <div class="socials">









                                                  </div>
                                            </div>
                                    </div>
                                    <div class="col-md-6 col-lg-4">
                                          <div class="card mx-30">
                                            <img
src="https://instagram.fbom16-1.fna.fbcdn.net/v/t51.2885-
19/s320x320/35574404_211240169516285_3644553245583474688_n.jpg?tp=1&_nc_h
t=instagram.fbom16-
1.fna.fbcdn.net&_nc_ohc=sTDaaVUryc8AX_tpsG5&edm=ABfd0MgAAAAA&ccb=7-
4&oh=b267a763127e04b4294916617c36c9c0&oe=609D7B8E&_nc_sid=7bff83"
class="card-img-top" alt="...">
                                            <div class="card-body">
                                                  <h5 class="card-
title">Aditya Singh</h5>
                                                  <h6>Lead
Developer</h6>
                                                        <div
class="socials">









                                                        </div>
                                                  </div>
                                          </div>


</div>
</div>
</div>

<style>
      body {
 font-family: 'Lato', sans-serif;
 align-content: center;
 align-items: center;
 align-self: center;
}
```

```
.wrapper {
padding-top: 120px;
align-items: center;
}
.row{
       align-self: center;
}
.card-img-top {
box-shadow: 0 5px 10px rgba(0,0,0,0.5);
}
.card-body{
    text-align: center;
    box-shadow: 0px 15px 15px -8px rgba(0,0,0,0.5)
}
.card-body h6 {
font-size: 14px;
text-transform: uppercase;
color: deeppink;
}
.card-title {
text-transform: uppercase;
font-weight: bold;
font-size: 24px;
}

.socials a {
width: 40px;
height: 40px;
display: inline-block;
border-radius: 50%;
margin: 0 5px;
}
.socials a i {
color: #fff;
padding: 12px 0;
}

.socials a:nth-child(1) {
background: #3b5998;
}.socials a:nth-child(2) {
background: #ff0000;
}.socials a:nth-child(3) {
background: #007bb5;
}.socials a:nth-child(4) {
background: #ea4c89;
}

@media (max-width: 800px){
    .mx-30{
        margin-bottom: 30px;
    }
}
</style>
{% endblock %}
```

add_stock.html

```
{% extends 'base.html' %}
{% block content %}

<h1>Add New Stocks to Portfolio</h1>
<br/>

      <form action="{% url 'add_stock' %}" class="d-flex" method="POST">
                {% csrf_token %}
        <input class="form-control me-2" type="search" placeholder="Add a new
stock to portfolio" aria-label="Search" name="ticker">
        <button class="btn btn-outline-success" type="submit">Add</button>
      </form>

<br/>
<br/>
<table class="table table-striped table-bordered table-hover">
  <thead class="table-dark">
    <tr>

      <th scope="col">Ticker Name</th>
      <th scope="col">Company Name</th>
      <th scope="col">Stock Price</th>
      <th scope="col">Previous Close</th>
      <th scope="col">Market Cap</th>
      <th scope="col">YTD Change</th>
      <th scope="col">52Week High</th>
      <th scope="col">52Week Low</th>


    </tr>
  </thead>
  <tbody>
  {% if ticker %}



                    {% for list_item in output %}
                            <tr>
                                    <th scope="row">{{ list_item.symbol
}}</th>
                                    <th scope="row">{{
list_item.companyName }}</th>
                                    <td>${{ list_item.latestPrice   }}</td>
                                    <td>${{ list_item.previousClose }}</td>
                                    <td>${{ list_item.marketCap }}</td>
                                    <td>{{ list_item.ytdChange   }}%</td>
                                    <td>${{ list_item.week52High }}</td>
                                    <td>${{ list_item.week52Low   }}</td>

                            </tr>
                <style>
```

```
                        a{
                                text-decoration: none;
                        }
                </style>
                        {% endfor %}




                </tbody>
                </table>
        {% endif %}
<br/>
        {% for item in ticker %}
                <a href="{% url 'delete' item.id %}" class="btn btn-
danger">
                Delete {{ item }} stock</a><br/><br/>
        {% endfor %}
<br/><br/>
{% endblock %}
```

base.html

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"
crossorigin="anonymous">

    <title>Stock Portfolio</title>
  </head>
  <body>
       <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="{% url 'home' %}">Stock Portfolio</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link" href="{% url 'about' %}">DevTeam</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{% url 'add_stock' %}">Add Stock</a>
        </li>
```

```
          <li class="nav-item">
            <a class="nav-link" href="{% url 'delete_stock' %}">Delete Stock</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{% url 'news' %}">Stock News</a>
          </li>
        </ul>
        <form action="{% url 'home' %}" class="d-flex" method="POST">
                {% csrf_token %}
          <input class="form-control me-2" type="search" placeholder="Get Stock
Quotes" aria-label="Search" name="ticker">
          <button class="btn btn-outline-success" type="submit">Search</button>
        </form>
      </div>
    </div>
</nav>


        <div class="container">
                <br/>

                {% if messages %}
                        {% for message in messages %}
                                <div class="alert alert-warning alert-dismissable"
role="alert">
                                        <button class="close" data-
dismiss="alert"><small><sup>x</sup></small></button>
                                        {{ message }}
                                </div><br/>
                        {% endfor %}
                {% endif %}

        {% block content %}

                {% endblock %}

        </div>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"
crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.6.0/dist/umd/popper.mi
n.js" integrity="sha384-
KsvD1yqQ1/1+IA7gi3P0tyJcT3vR+NdBTt13hSJ2lnve8agRGXTTyNaBYmCR/Nwi"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/js/bootstrap.min.js" integrity="sha384-
```

```
nsg8ua9HAw1y0W1btsyWgBklPnCUAFLuTMS2G72MMONqmOymq585AcH49TLBQObG"
crossorigin="anonymous"></script>
    -->

  </body>
</html>
```

delete.html

```
{% extends 'base.html' %}
{% block content %}

<h1>Delete Stock</h1>


{% if ticker %}
       {% for item in ticker %}
              <a href="{% url 'delete' item.id %}" class="btn btn-
secondary">
              Delete {{ item }} stock</a><br/><br/>
       {% endfor %}

{% else %}
       You don't have any stocks to delete.
{% endif %}

{% endblock %}
```

home.html

```
{% extends 'base.html' %}
{% block content %}


{% if ticker %}
       {{ ticker }}
{% endif %}

{% if api %}
       {% if api == "Error..." %}
              There was a problem with your ticker symbol. Please try
again.
       {% else %}
              <h1>{{ api.companyName }}</h1><br/>
              Price: ${{ api.latestPrice  }}<br/>
              Previous Close: ${{ api.previousClose }}<br/>
              Market Cap: ${{ api.marketCap  }}<br/>
              YTD Change: {{ api.ytdChange  }}<br/>
              52Week High: {{ api.week52High  }}<br/>
              52Week Low: {{ api.week52Low  }}<br/>

       {% endif %}
```

```
{% endif %}
<h1>Welcome to your stock portfolio!</h1><br/><br/>

<h4> New here? Lets get familar with your console! </h4><br/>

<h6>Your nav bar consists of a couple of things, lets get familiar with
the navbar first.</h6>
1) The <b>"DevTeam"</b> Section : Want to check out our team and
developer profiles? Click that button then.<br/><br/>

2) The <b>"Add Stock"</b> Section : This section helps in addition of a
'new' stock to your portfolio. All the data is fetched live from the
market and the values are either form the live market or previous close,
whichever is earlier.<br/><br/>

3) The <b>"Delete Stock"</b> Section : This section lists all your active
stocks in the portfolio, and gives you the choice to delete a stock from
your portfolio. Although this can also be done through the active stocks
section.<br/><br/>

4) The <b>Stock News"</b> Section : This section keeps you updated of the
all the highs and the lows of the stock market along with tips that gives
you an edge over other, so that you can play your cards rightly!
<br/><br/>

5) The <b>"Get Stock Quotes"</b> Section : You maybe able to see a search
bar at the top right corner of the nav bar, with a Search button in green
besides. This will help you in searching for a stock using the "ticker
symbol". It will fetch details like the company name, stock price,
previous close, market cap, year-to-date change, 52week high and 52 week
low.<br/><br/>
For beginners, a tiker symbol of a stock is a set of characters below 5
in count, that are used to represent that company's stock on the stock
market.<br/>

{% endblock %}
```

news.html

```
{% extends 'base.html' %}
{% block content %}

<h1>Your daily dose of Stocks News!</h1><br />
<div class="container">
    <div class="row">
    {% for x in api.articles %}
         <div class="col-sm">
          <div class="card" style="width: 18rem;">
            <img src="{{x.urlToImage}}" class="card-img-top"
alt="{{x.source.name}}">
            <div class="card-body">
              <h5 class="card-title">{{x.title}}</h5>
              <p class="card-
text">{{x.description|truncatechars:150}}</p>
```

```
                <a href="{{x.url}}" target="_blank" class="btn btn-
primary">Read More</a>
              </div>
            </div>
            <br>
          </div>
    {% endfor %}
    </div>
    </div>

{% endblock %}
```

7. Conclusion

Thus, as required, we have successfully created a responsive Django stock portfolio and news web application. The user is now provided with up-to-date content of stock values and stock news.

The code of this project has been uploaded on GitHub on the Url:

https://github.com/Vivek-Hotti/DjangoStockPortfolio

T H A N K – Y O U

***** *(you have reached the end of this project's document)* *****

(((((())))))