

Stock Market Tracking App



Dipali Singh - RA1911030010138

Project Description

In the present times when investors need to be two steps ahead at all time, we need a web-means that can be as fast and efficient. Investors hold back from dabbling in the stock market due to lack of information. Our aim while building this website is to facilitate and assist investors to keep track of the stock market. We want to make a user-friendly website that will allow investors to favourite the stocks they have invested in and track their stocks by analysing the percentage increase or percentage decrease alongside with graphs over a given timeline. The app will be complete with blogs and helpline that will give more insights into the current market status.



Roles and Models

Stakeholder Name	Activity / Area / Phase	Interest	Influence	Priority (High / Medium/Low)
Regional Head of Sales & Marketing	Subscription using mobile App	High	High	1
Finance Account Receivable consultant	Multiple Currency Payment	High	Low	3
Developer	Development of the app	High	High	1
Stock broker	User	High	Low	3



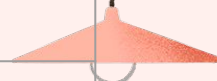
Roles and Models

Stakeholder Name	Activity / Area / Phase	Interest	Influence	Priority (High / Medium/Low)
Customer	User	High	High	2
Supplier	Make resources available	High	High	1
Investors	Provide financial assistance	High	High	1
Parent companies	Monitor the functioning	Low	High	3
Stock broker	User	High	Low	3

An illustration of a man and a woman in a modern office setting. The man, with a beard and wearing a red t-shirt and dark trousers, stands with his hands in his pockets, looking towards the woman. The woman, with dark curly hair and wearing a black and white striped long-sleeved shirt and blue jeans, is seated on a white stool, working on a laptop that sits on a wooden desk. A pink lamp hangs from the ceiling, and a pink vase with white flowers is on the desk. The background is a light gray wall with a large white circle.

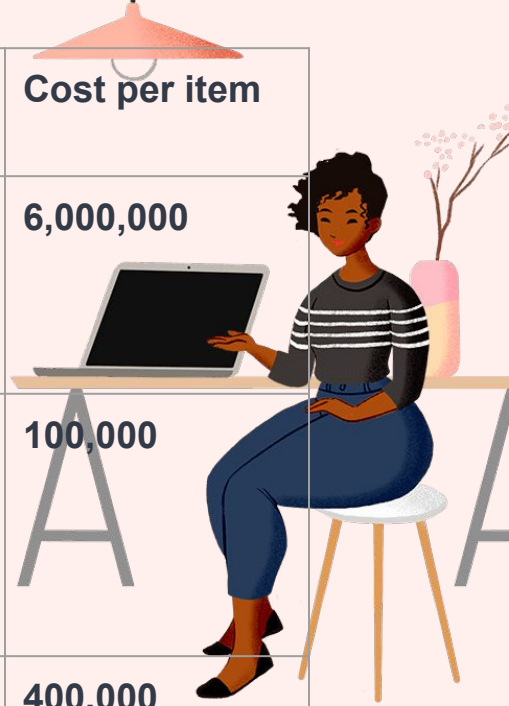
Estimation -

Activity Description	Sub - Task Desc.	Efforts in hours	Costs in INR
Design the web app	To create a webapp using python and django	6-12	Frontend- 30k Backend- 30k Integration- 5k
Research and development	Evaluating of software tech trends and incorporating them with updates and patches.	4-8	For Research - 15K For Rolling out patches/ updates - 5K
Data analytics	Administration Data, Database, Management, and evaluating stock market trends.	4-5	Data Analyst- 15k Data scientist- 20k
Identify Data Source for displaying units of Energy Consumption	Go through Interface contract (Application Data Exchange) documents	5	15K
Marketing	Advertisement	8-10	1L



Maintenance and Support Cost

Category	Details	Qty	Cost per annum	Cost per item
People	Network, System, Middleware and DB admin	5	2,000,000	6,000,000
License	Operating System, Database, Middleware, IDE	4	10000	100,000
Infrastructure	Server, Storage and Network	3	20000	400,000



Risk



Risk Identification

Risk identification and categorization:

- The predictions can be wrong due to inaccuracy in the app.
- Sometimes wrong stocks can be shown which may lead to misinformation.
- Out of date information can be shown.
- Sometimes personalised stocks may be incorrect due to inaccurate predictions.



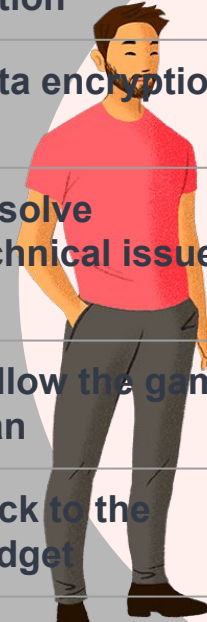
List (Describe) Register

Risk ID#	Risk Description	Impact Description
R01	Hacking of application	Has a high impact as it can leak information
R02	Technical difficulties like Data Security.	High impact technical risk
R03	Unable to meet deadlines.	Low impact risk
R04	Insufficiency of the budget	Medium impact risk as it slows down development
R05	Lack of stakeholder engagement	Medium impact risk



Managing Risk

Risk ID#	Status	Risk Appetite	Action
R01	Open	Avoid	Data encryption
R02	Open	Mitigate	Resolve technical issues
R03	Open	Mitigate	Follow the game plan
R04	Open	Avoid	Stick to the budget
R05	Open	Accept	Pitch the product

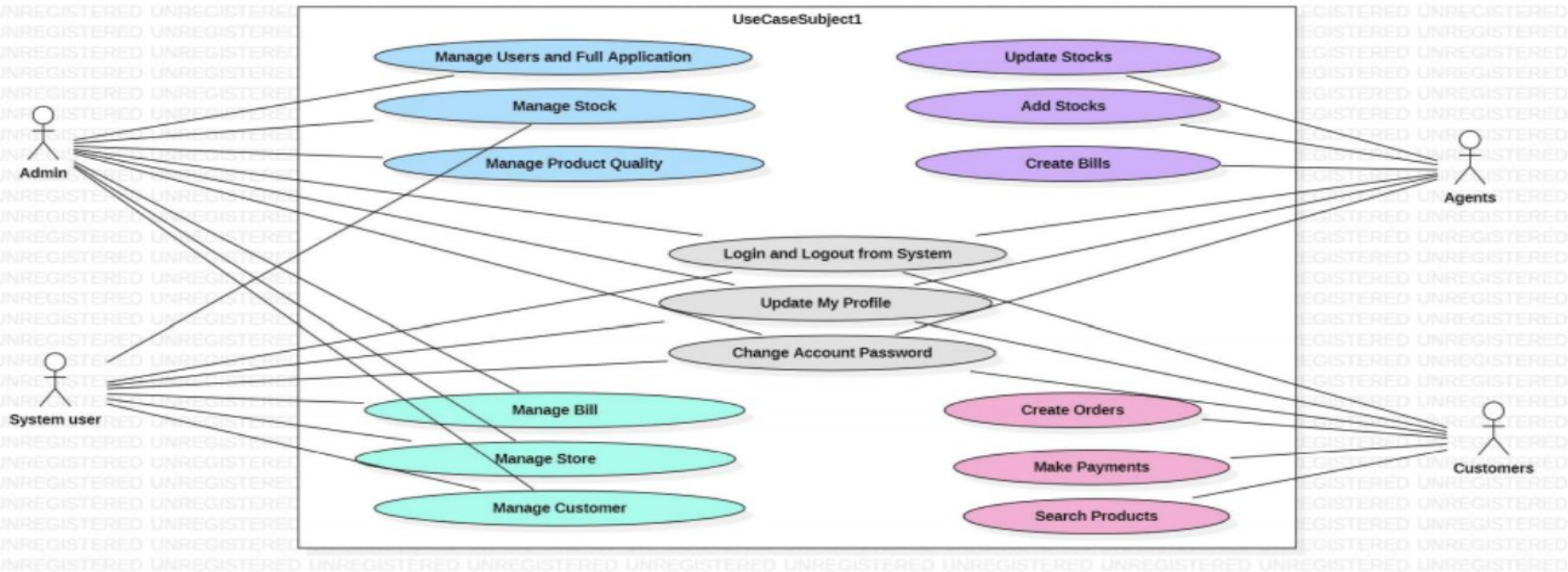
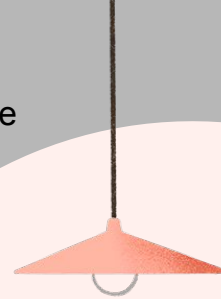


Architecture and Design of the System

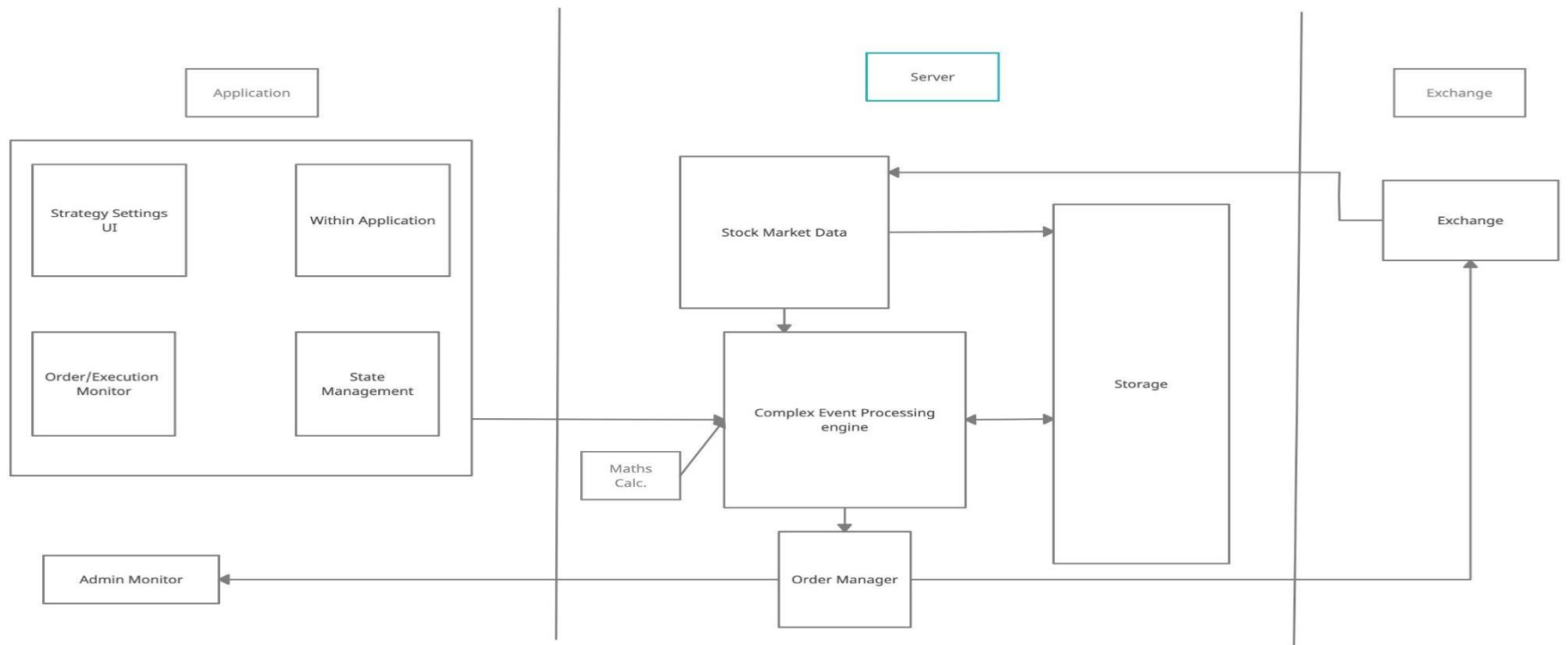


Use case Diagram

This use case diagram is a graphic depiction of the interactions among the elements of the Stock Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Stock Management System.



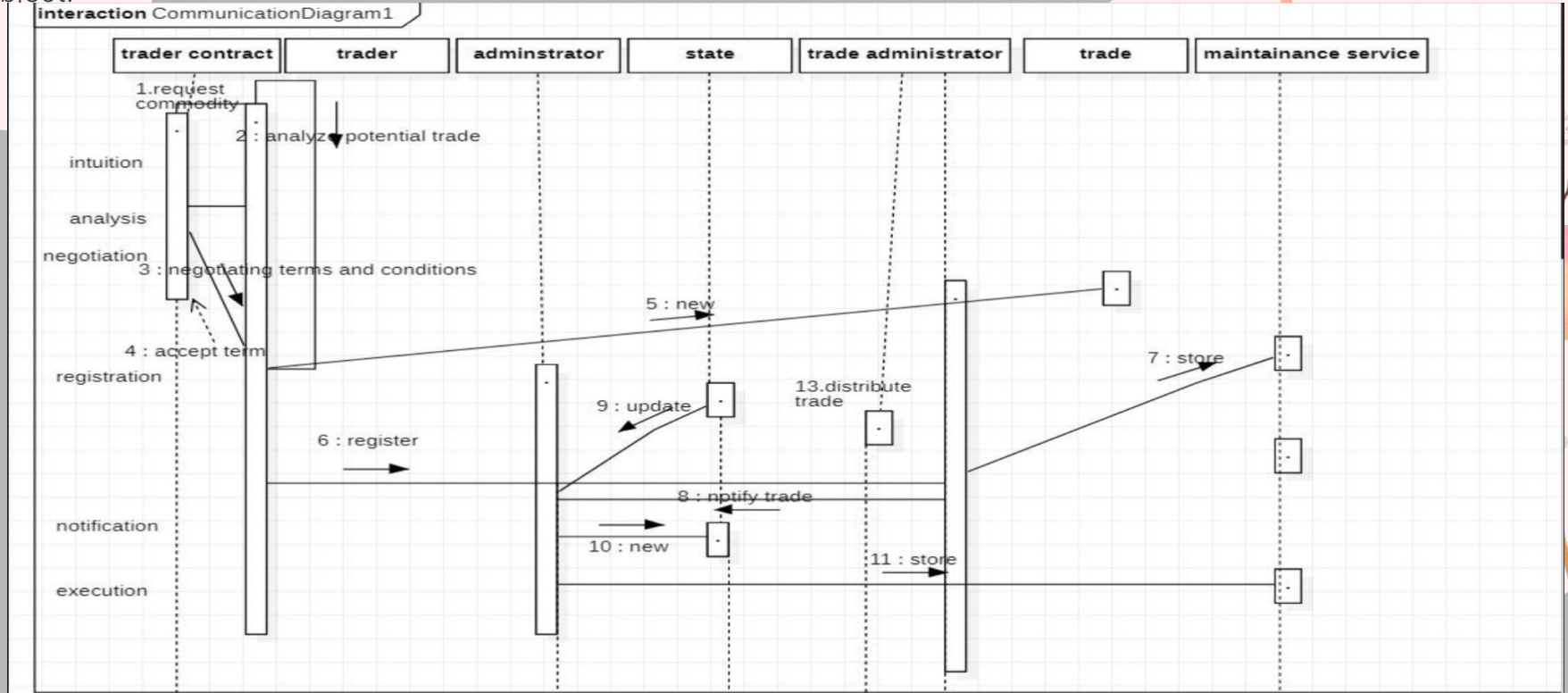
Architecture Diagram – An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.



Architecture Diagram Of App

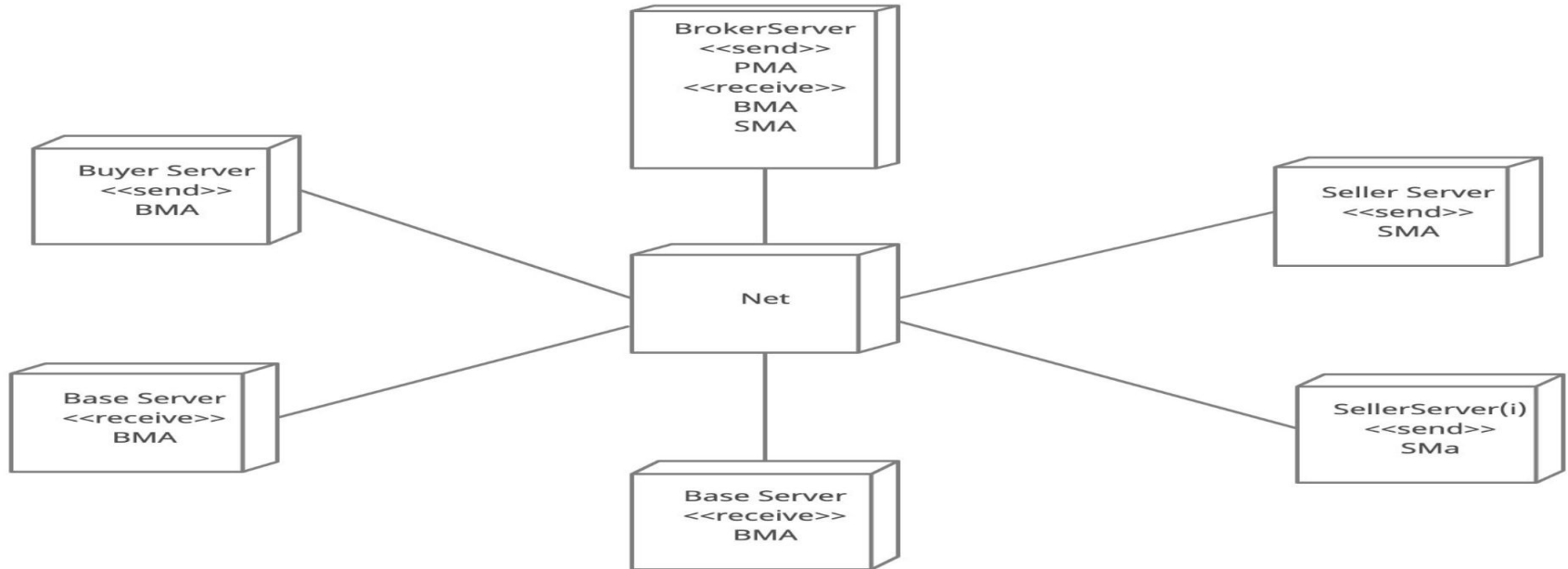
Collaboration Diagram

– A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



Deployment Diagram

– A UML deployment diagram is a diagram that shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams are a kind of structure diagram used in modeling the physical aspects of an object-oriented system. The deployment diagram of our application is shown below:

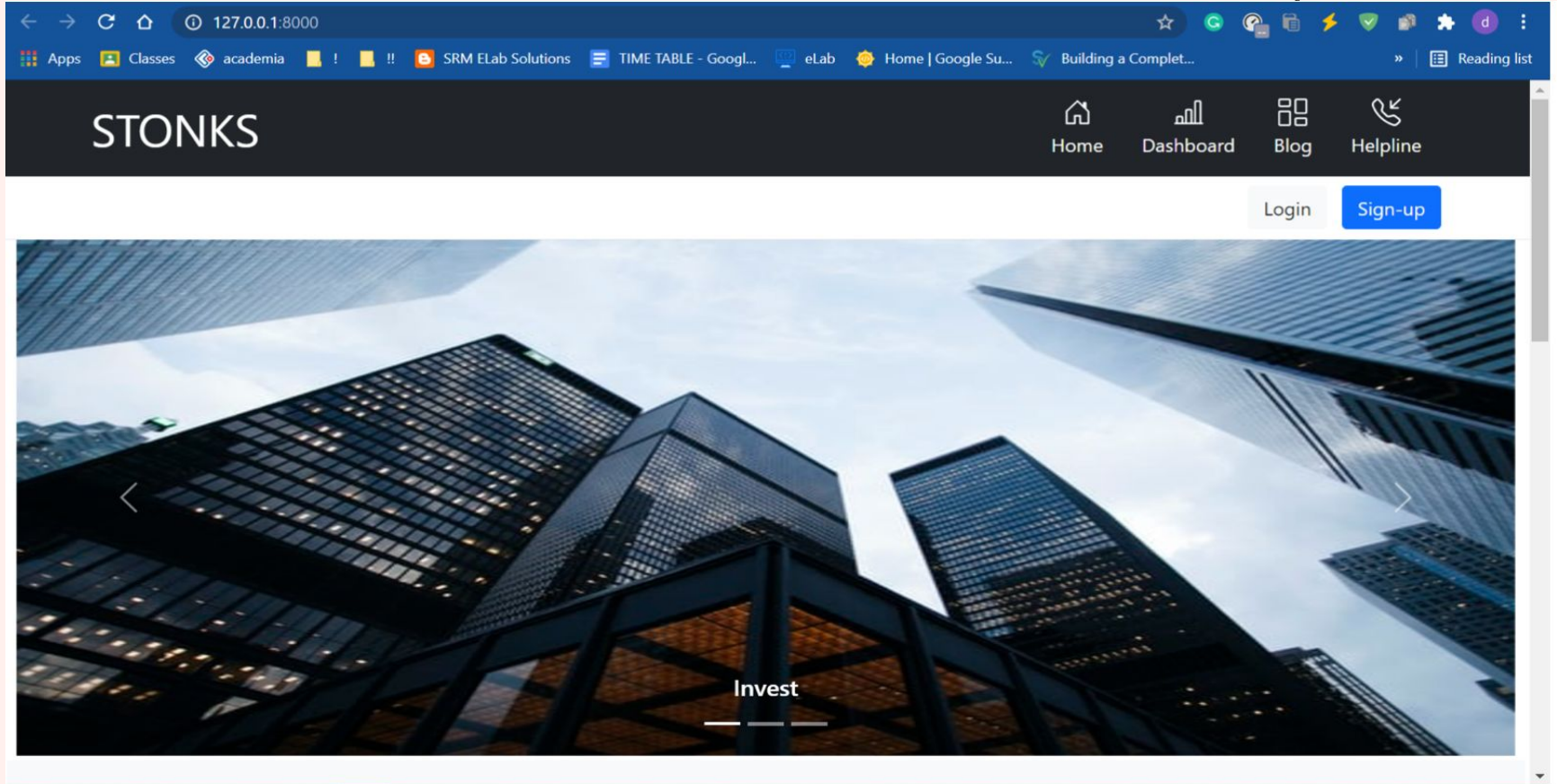


Deployment Diagram for Stock Marketing App

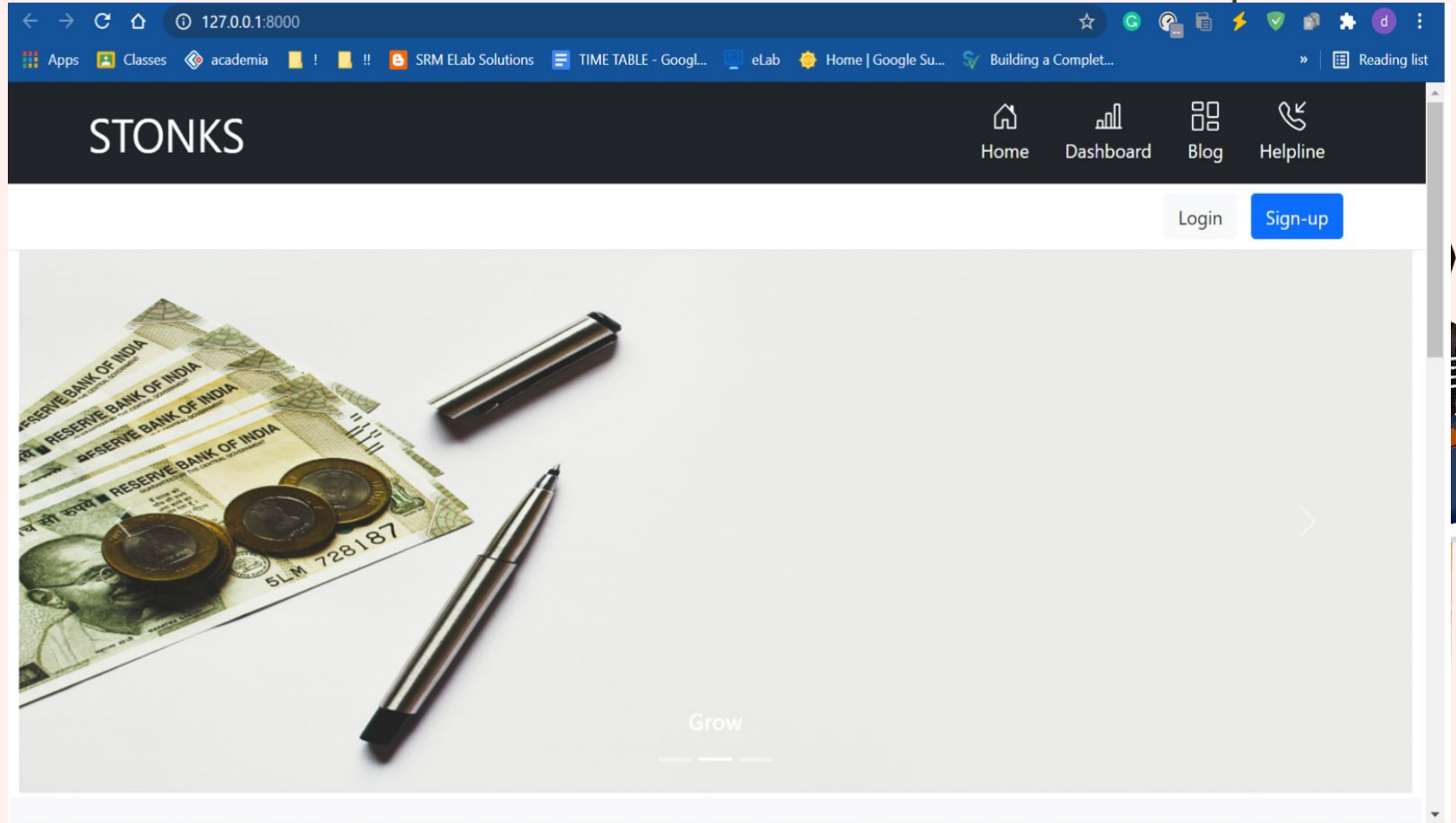
Front-end



Frontend



Frontend



Frontend

STONKS



Home



Dashboard



Blog



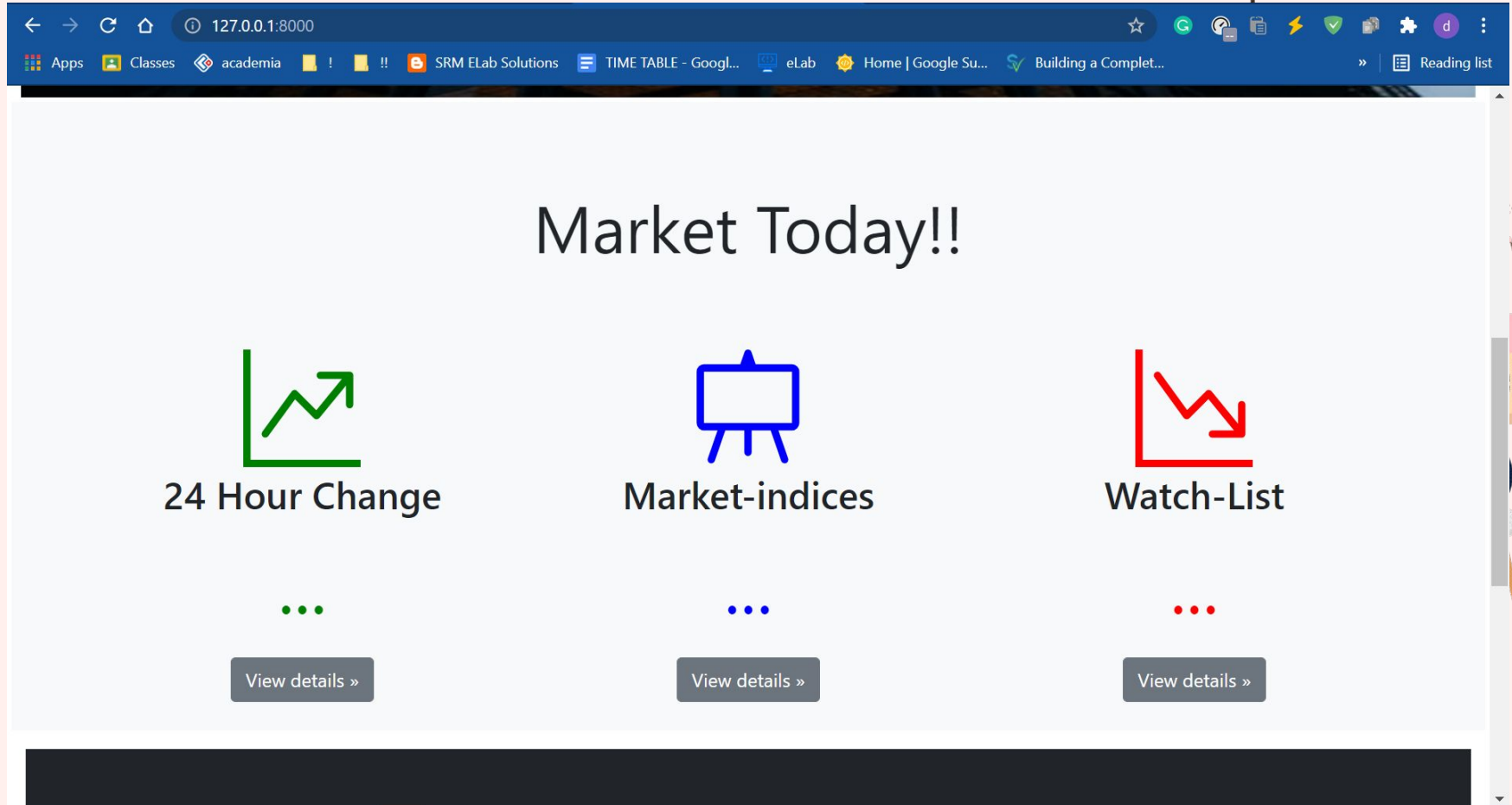
Helpline

Login

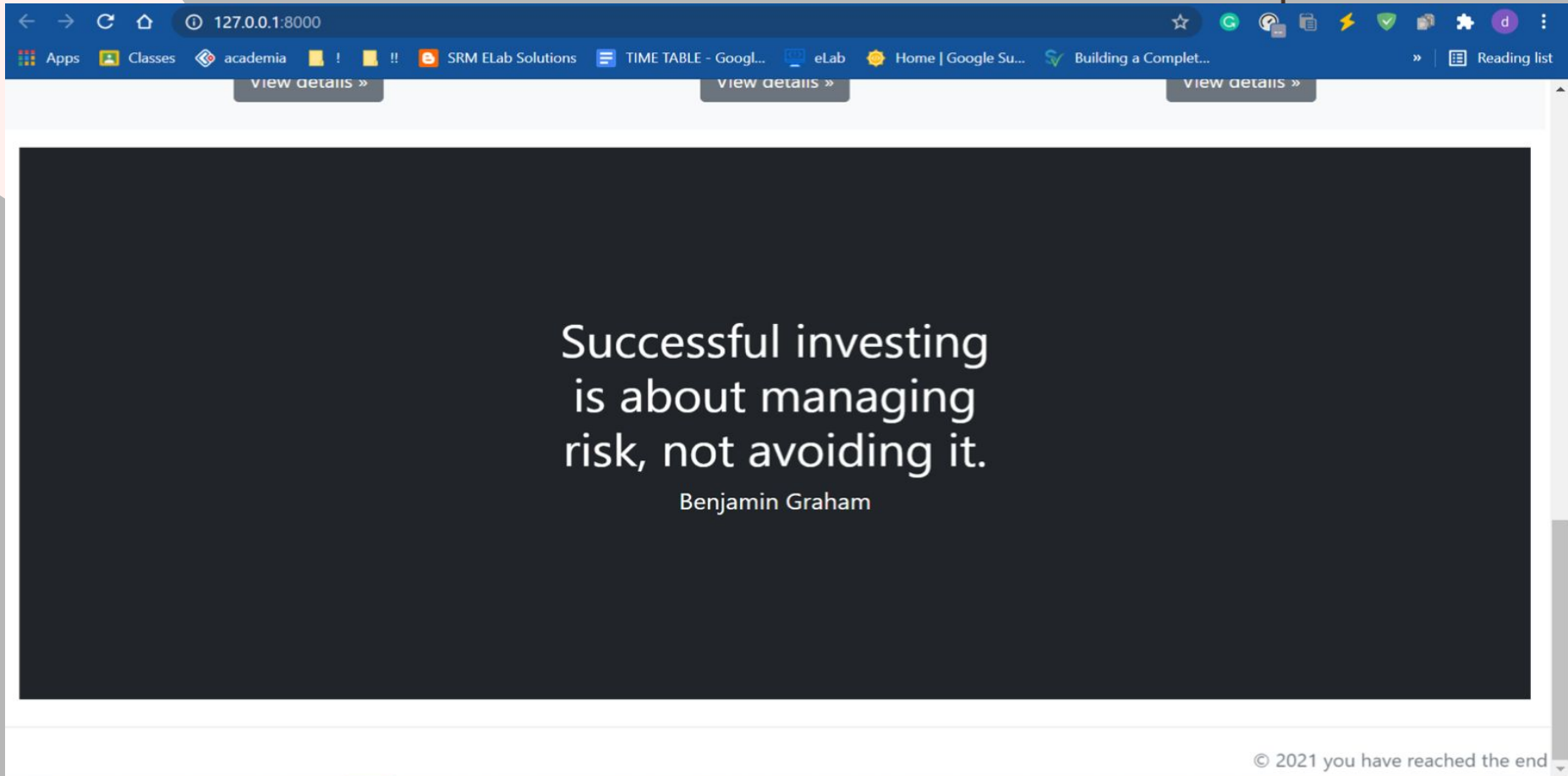
Sign-up



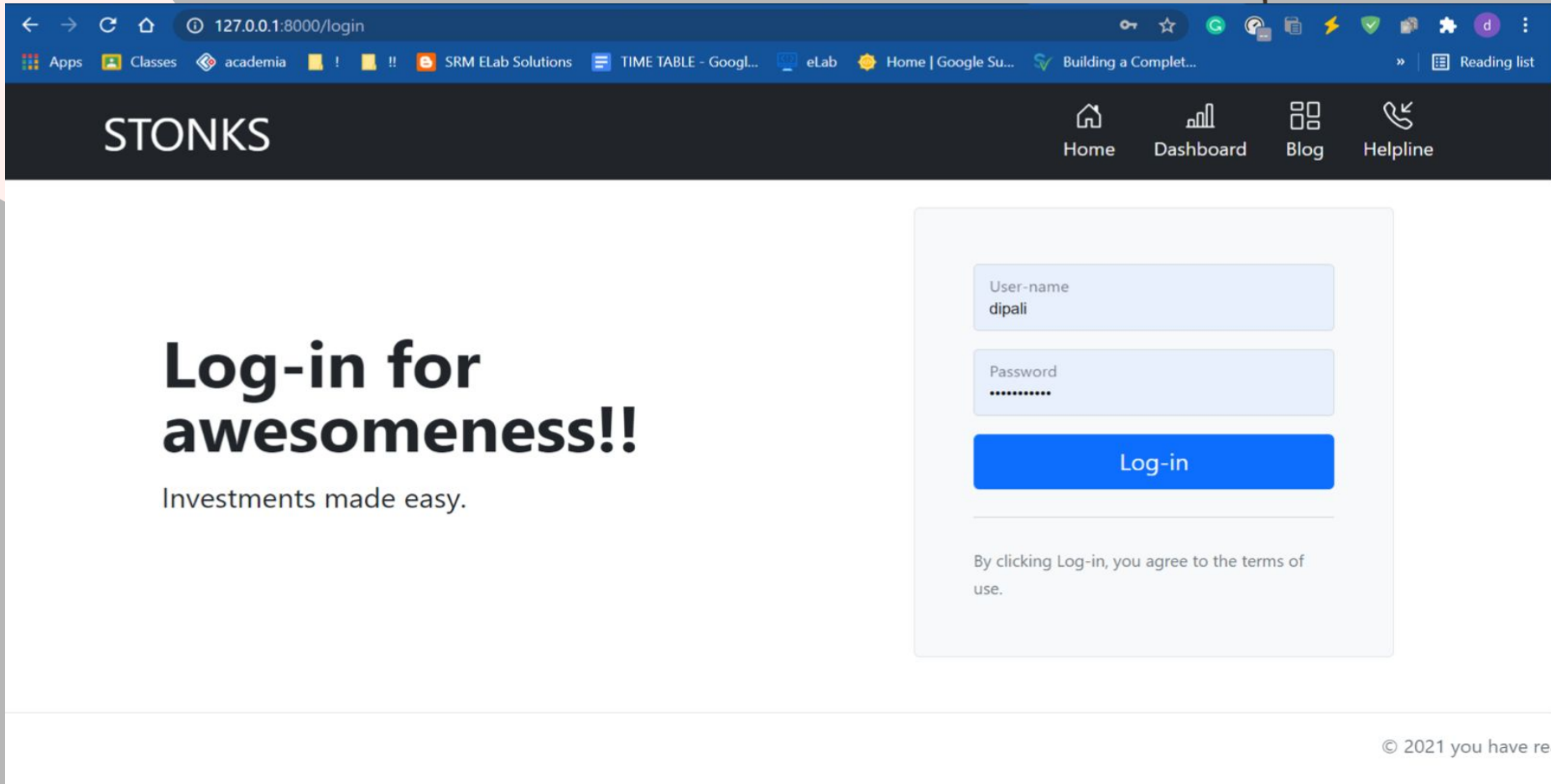
Frontend



Frontend



Frontend





Sign-up for awesomeness!!

Investments made easy.

Username

dipali

Password

.....

☐ Remember me

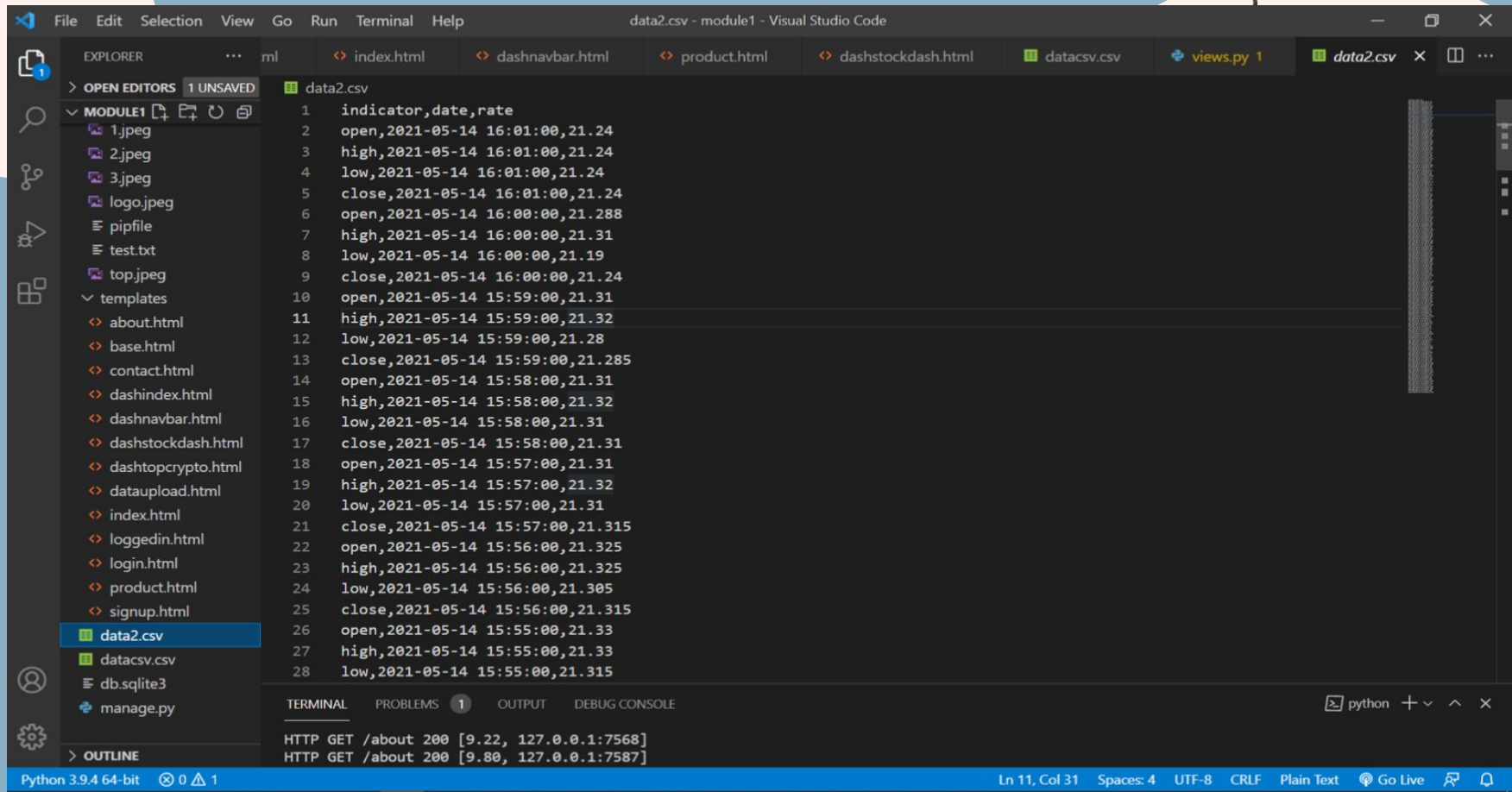
Sign-up

By clicking Sign-up, you agree to the terms of use.

Back-end



The Live data of highs and lows of each stock:



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'MODULE1' with various files including images, a pipfile, test files, and HTML templates. The code editor displays the contents of 'data2.csv', which contains stock market data with columns for indicator, date, and rate. The data includes open, high, low, and close prices for a specific date (2021-05-14) at different times. The terminal at the bottom shows HTTP GET requests to /about.

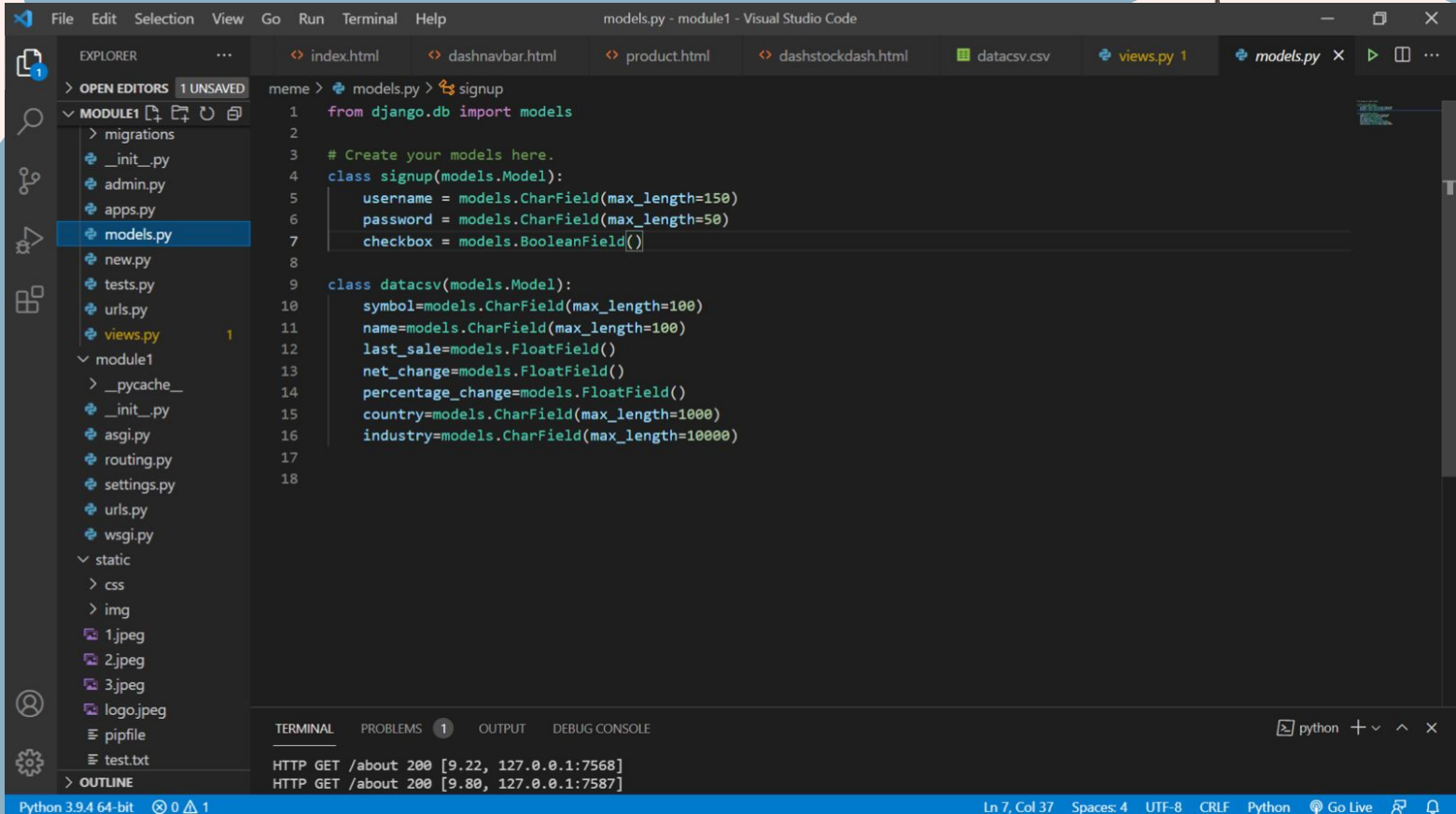
```
data2.csv
1 indicator,date,rate
2 open,2021-05-14 16:01:00,21.24
3 high,2021-05-14 16:01:00,21.24
4 low,2021-05-14 16:01:00,21.24
5 close,2021-05-14 16:01:00,21.24
6 open,2021-05-14 16:00:00,21.288
7 high,2021-05-14 16:00:00,21.31
8 low,2021-05-14 16:00:00,21.19
9 close,2021-05-14 16:00:00,21.24
10 open,2021-05-14 15:59:00,21.31
11 high,2021-05-14 15:59:00,21.32
12 low,2021-05-14 15:59:00,21.28
13 close,2021-05-14 15:59:00,21.285
14 open,2021-05-14 15:58:00,21.31
15 high,2021-05-14 15:58:00,21.32
16 low,2021-05-14 15:58:00,21.31
17 close,2021-05-14 15:58:00,21.31
18 open,2021-05-14 15:57:00,21.31
19 high,2021-05-14 15:57:00,21.32
20 low,2021-05-14 15:57:00,21.31
21 close,2021-05-14 15:57:00,21.315
22 open,2021-05-14 15:56:00,21.325
23 high,2021-05-14 15:56:00,21.325
24 low,2021-05-14 15:56:00,21.305
25 close,2021-05-14 15:56:00,21.315
26 open,2021-05-14 15:55:00,21.33
27 high,2021-05-14 15:55:00,21.33
28 low,2021-05-14 15:55:00,21.315
```

TERMINAL

```
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

Python 3.9.4 64-bit 0 1 Ln 11, Col 31 Spaces: 4 UTF-8 CRLF Plain Text Go Live

Code for backend -



The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Panel:** Displays the project structure. The 'models.py' file is selected under the 'module1' folder. Other files visible include migrations, __init__.py, admin.py, apps.py, new.py, tests.py, urls.py, views.py, and various static files like css, img, and images (1.jpeg, 2.jpeg, 3.jpeg, logo.jpeg).
- Editor Panel:** Shows the content of 'models.py'. The code defines two Django models: 'Signup' and 'Datacsv'.
- Terminal Panel:** Located at the bottom, it shows two HTTP GET requests to '/about' with status 200 and IP addresses 127.0.0.1:7568 and 127.0.0.1:7587.
- Status Bar:** At the very bottom, it indicates 'Python 3.9.4 64-bit' and 'Ln 7, Col 37'.

```
models.py - module1 - Visual Studio Code

EXPLORER
1 UNSAVED
> OPEN EDITORS
> MODULE1
  > migrations
  > __init__.py
  > admin.py
  > apps.py
  > models.py
  > new.py
  > tests.py
  > urls.py
  > views.py
  > module1
    > __pycache__
    > __init__.py
    > asgi.py
    > routing.py
    > settings.py
    > urls.py
    > wsgi.py
  > static
    > css
    > img
    > 1.jpeg
    > 2.jpeg
    > 3.jpeg
    > logo.jpeg
    > pipfile
    > test.txt
  > OUTLINE

models.py
1  from django.db import models
2
3  # Create your models here.
4  class Signup(models.Model):
5      username = models.CharField(max_length=150)
6      password = models.CharField(max_length=50)
7      checkbox = models.BooleanField()
8
9  class Datacsv(models.Model):
10     symbol=models.CharField(max_length=100)
11     name=models.CharField(max_length=100)
12     last_sale=models.FloatField()
13     net_change=models.FloatField()
14     percentage_change=models.FloatField()
15     country=models.CharField(max_length=1000)
16     industry=models.CharField(max_length=10000)
17
18

TERMINAL
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Python 3.9.4 64-bit 0 1 Ln 7, Col 37 Spaces: 4 UTF-8 CRLF Python Go Live
```

Code for backend -

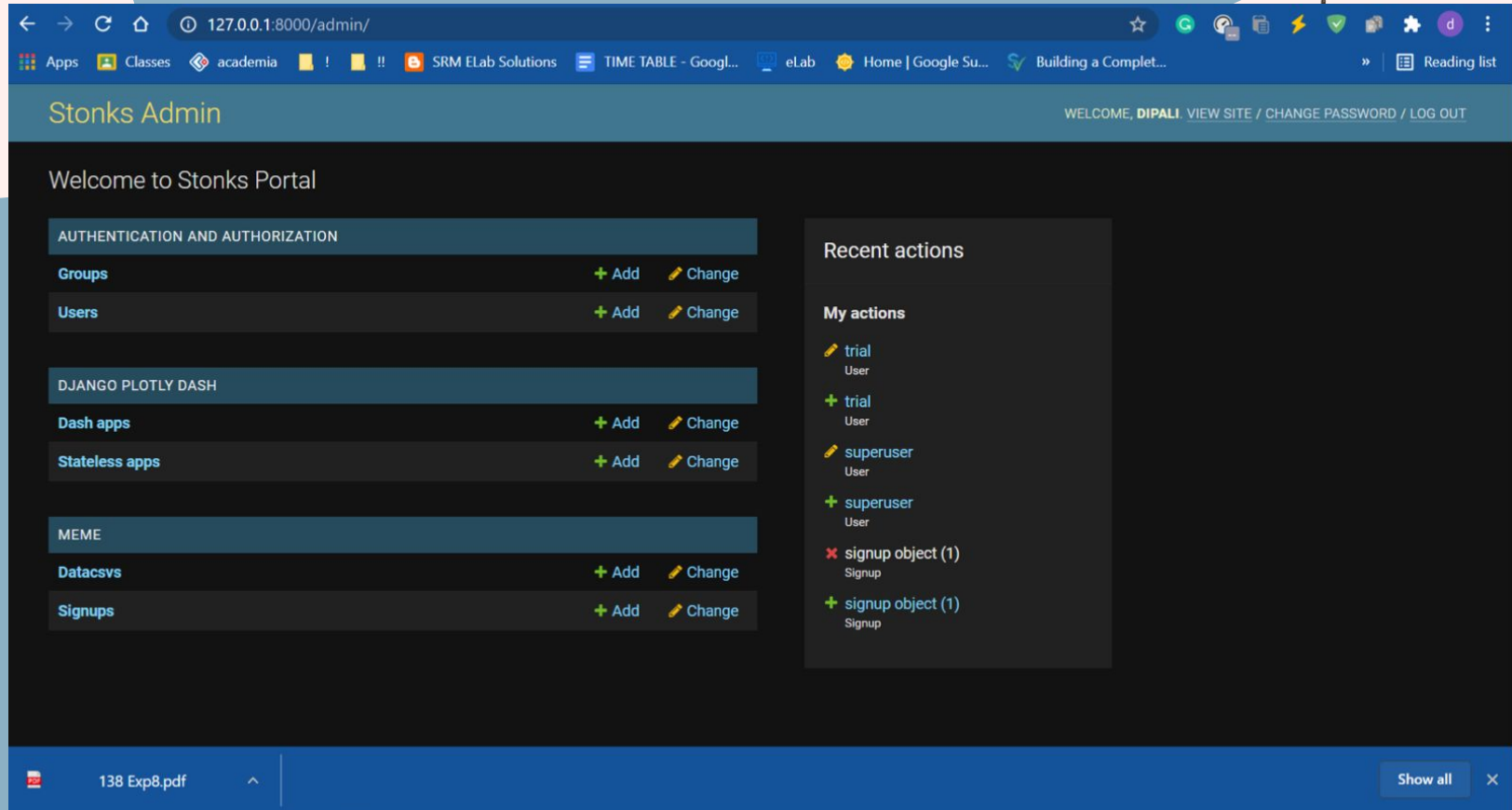
```
34 moredata = pricechange(symbol)
35
36 #get a fricken df
37
38 ts_df = candles(symbol)
39
40 #PlotlyGraph
41 def candlestick():
42     figure = go.Figure(
43         data = [
44             go.Candlestick(
45                 x = ts_df.index,
46                 high = ts_df['high'],
47                 low = ts_df['low'],
48                 open = ts_df['open'],
49                 close = ts_df['close'],
50             )
51         ]
52     )
53
54     candlestick_div = plot.figure, output_type='div')
55     return candlestick_div
56
57 #endPlotlyGraph
58 percentchange = pricedata['priceChangePercent']
59 buyers = pricedata['askQty']
60 sellers = pricedata['bidQty']
61
62 eth = pricechange(symbol='ETHUSD')
```

TERMINAL

```
HTTP GET /about 200 [9.22, 127.0.0.1:7568]
HTTP GET /about 200 [9.80, 127.0.0.1:7587]
```

python 3.9.4 64-bit 1 Ln 151, Col 30 (7 selected) Spaces: 4 UTF-8 CRLF Python Go Live

Result of Module 2



The screenshot shows a web browser window displaying the 'Stonks Admin' dashboard. The browser's address bar shows the URL '127.0.0.1:8000/admin/'. The dashboard has a dark blue header with the title 'Stonks Admin' and a welcome message 'WELCOME, DIPALI' with links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. The main content area is divided into two columns. The left column contains three sections: 'AUTHENTICATION AND AUTHORIZATION' with links for 'Groups' and 'Users' (each with '+ Add' and 'Change' buttons); 'DJANGO PLOTLY DASH' with links for 'Dash apps' and 'Stateless apps' (each with '+ Add' and 'Change' buttons); and 'MEME' with links for 'Datacsys' and 'Signups' (each with '+ Add' and 'Change' buttons). The right column contains a 'Recent actions' section with a list of actions: 'trial User' (with a pencil icon), '+ trial User' (with a plus icon), 'superuser User' (with a pencil icon), '+ superuser User' (with a plus icon), 'signup object (1) Signup' (with a red 'x' icon), and '+ signup object (1) Signup' (with a plus icon). At the bottom of the browser window, a taskbar shows a PDF file named '138 Exp8.pdf' and a 'Show all' button.

127.0.0.1:8000/admin/

Apps Classes academia ! !! SRM Elab Solutions TIME TABLE - Googl... eLab Home | Google Su... Building a Complet... Reading list

Stonks Admin

WELCOME, DIPALI [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Welcome to Stonks Portal

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

Users [+ Add](#) [Change](#)

DJANGO PLOTLY DASH

Dash apps [+ Add](#) [Change](#)

Stateless apps [+ Add](#) [Change](#)

MEME

Datacsys [+ Add](#) [Change](#)

Signups [+ Add](#) [Change](#)

Recent actions

My actions

- [trial](#)
User
- [+ trial](#)
User
- [superuser](#)
User
- [+ superuser](#)
User
- [✗ signup object \(1\)](#)
Signup
- [+ signup object \(1\)](#)
Signup

138 Exp8.pdf ^ Show all ×

Result of Module 2

The screenshot displays the Stonks Admin web application. The browser address bar shows the URL `127.0.0.1:8000/admin/auth/user/`. The page header includes the title "Stonks Admin" and a welcome message for "DIPALI" with links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT". The breadcrumb trail indicates the current location: "Home > Authentication and Authorization > Users".

The left sidebar contains a navigation menu with the following sections:

- AUTHENTICATION AND AUTHORIZATION**
 - Groups (+ Add)
 - Users (+ Add)
- DJANGO PLOTLY DASH**
 - Dash apps (+ Add)
 - Stateless apps (+ Add)
- MEME**
 - Datacsvs (+ Add)
 - Signups (+ Add)

The main content area is titled "Select user to change" and features a search bar and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. Three users are listed: "dipali" (staff status Yes), "superuser" (staff status No), and "trial" (staff status No). The "Action" dropdown is set to "Go", and the text "0 of 3 selected" is displayed. A "3 users" summary is shown at the bottom of the table.

On the right side, there is a "FILTER" panel with the following options:

- By staff status**
 - All
 - Yes
 - No
- By superuser status**
 - All
 - Yes
 - No
- By active**
 - All
 - Yes
 - No

The bottom of the interface shows a file explorer with "138 Exp8.pdf" and a "Show all" button.

Prediction



LSTM Model

- We will use the Long Short-Term Memory(LSTM) method to create a Machine Learning model to forecast stock values. They are used to make minor changes to the information by multiplying and adding. Long-term memory (LSTM) is a deep learning artificial recurrent neural network (RNN) architecture.
- However, with the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Market Prediction advancements have begun to include such approaches in analyzing stock market data. The Opening Value of the stock, the Highest and Lowest values of that stock on the same days, as well as the Closing Value at the end of the day, are all indicated for each date.



Dataset

The dataset includes the data of the stocks infosys, facebook and apple from 29 Jan 2015 to 29 Jan 2022. This dataset was acquired from the yahoo finance website.



Result of Module 3:

jupyter stock-price-prediction-using-1stm Last Checkpoint: 8 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3

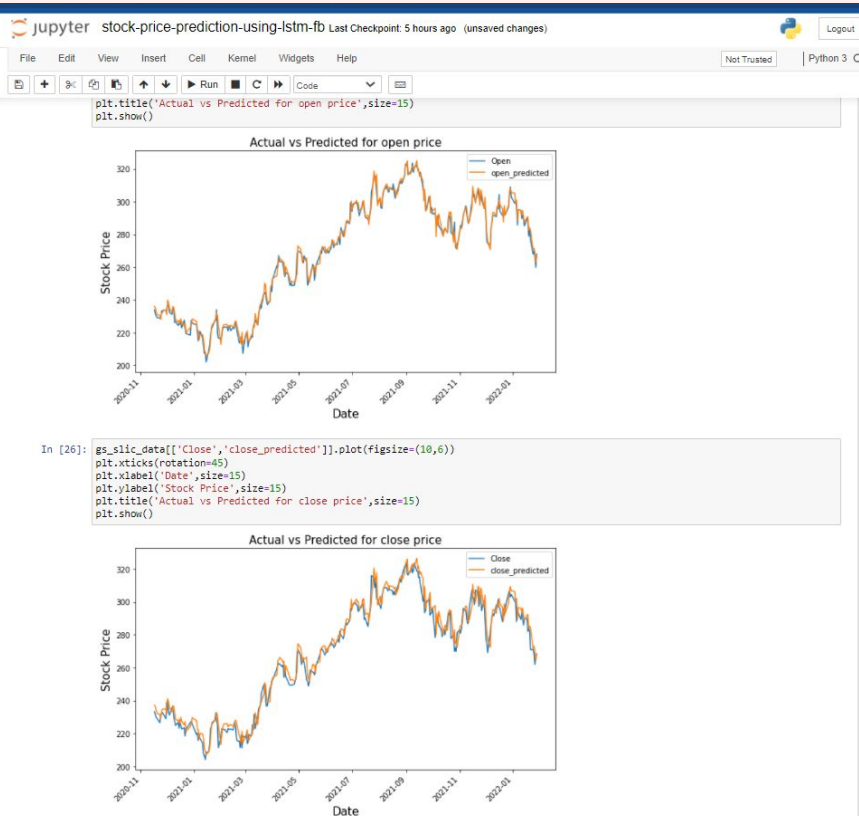
```
plt.xlabel('Date',size=15)  
plt.ylabel('Stock Price',size=15)  
plt.title('Actual vs Predicted for open price',size=15)  
plt.show()
```



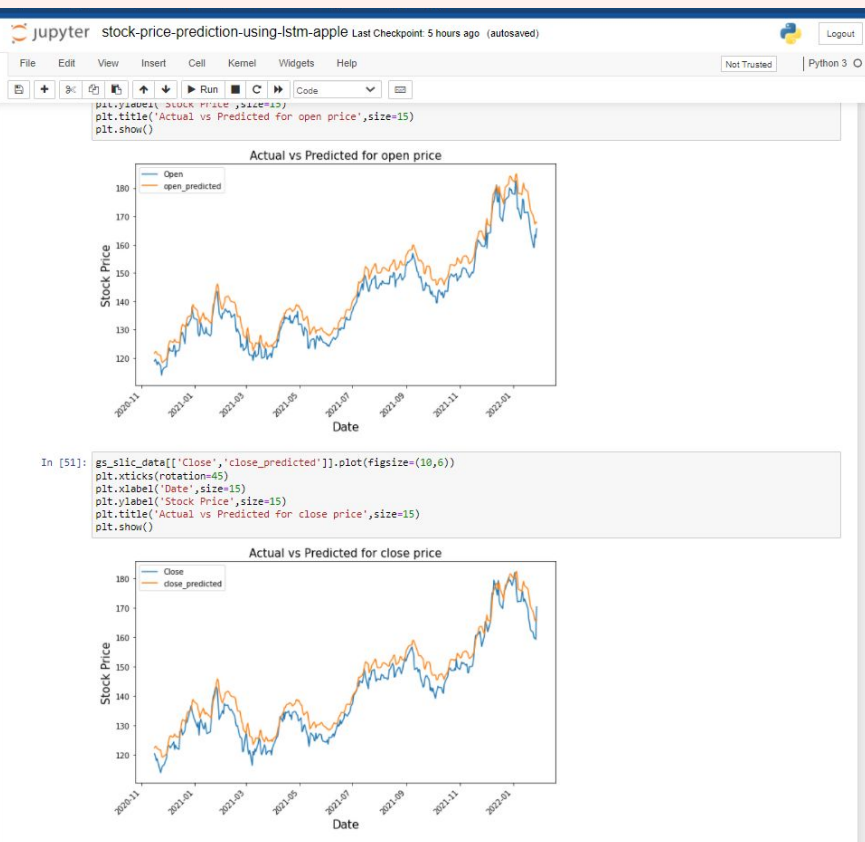
```
In [26]: gs_slic_data[['Close','close_predicted']].plot(figsize=(10,6))  
plt.xticks(rotation=45)  
plt.xlabel('Date',size=15)  
plt.ylabel('Stock Price',size=15)  
plt.title('Actual vs Predicted for close price',size=15)  
plt.show()
```



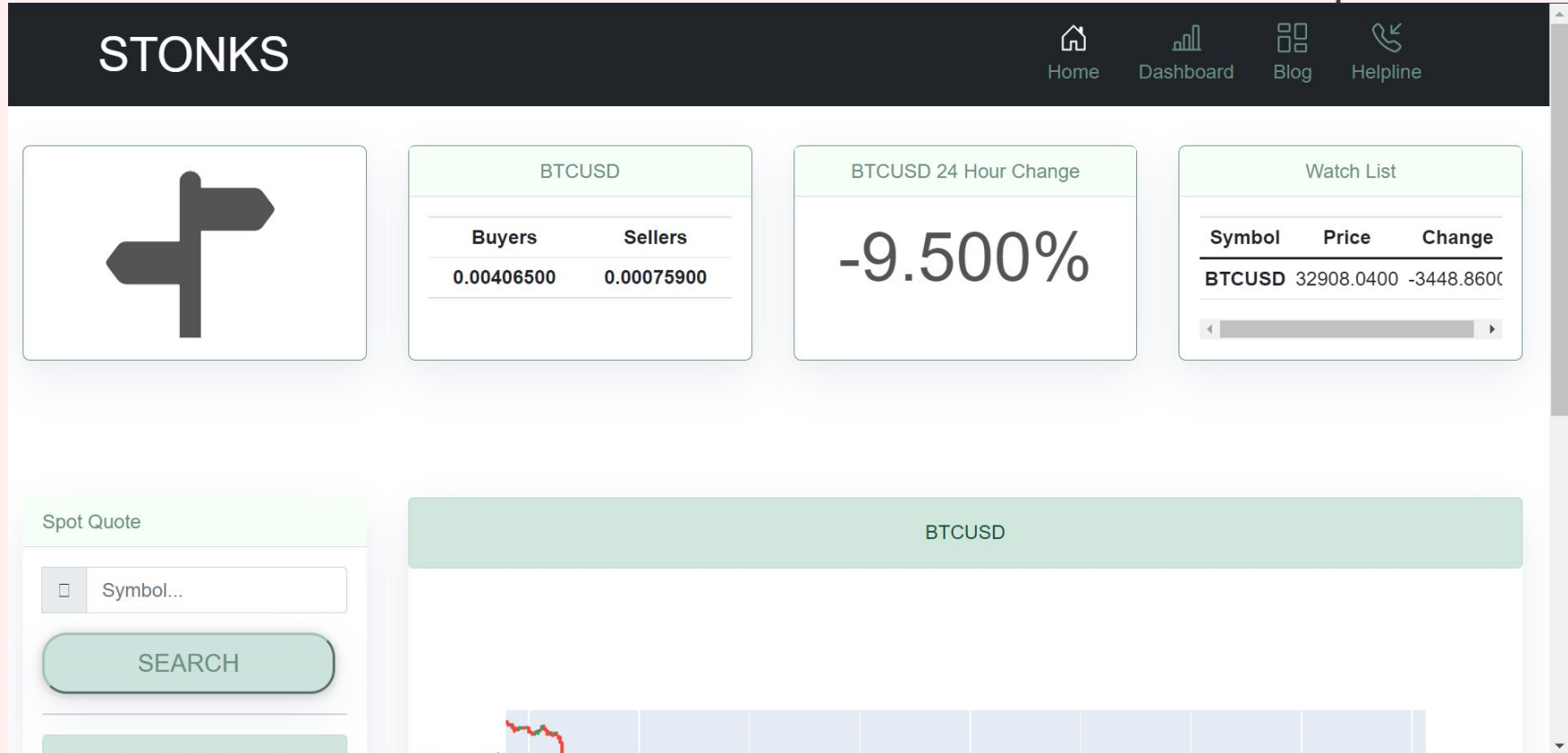
Result of Module 3:



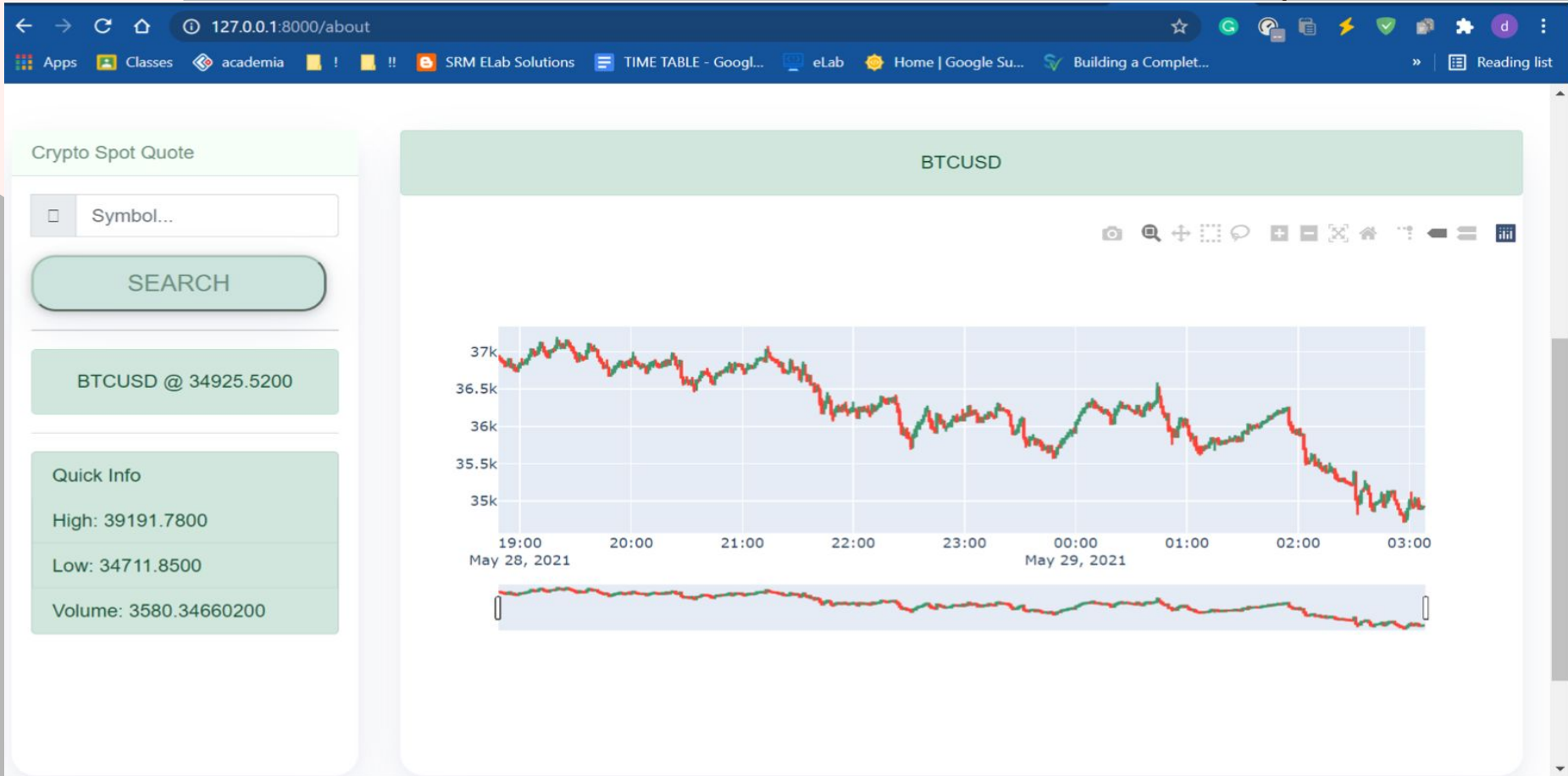
Result of Module 3:



Result of Module 3:



Result of Module 3:



Thankyou

