PROJECT REPORT

ON

STOCK MARKET-SIMULATOR

NATIONAL INSTITUTE OF TECHNOLOGY, KARNATAKA

MASTER OF COMPUTER APPLICATIONS

2021-2022

Submitted By:

ADARSH THAKUR (194CA002)

RAVI KUMAR GAUTAM (194CA036)

RAJAT KUMAR JAIN (194CA035)

HARENDRA SINGH (184CA024)

MUKESH KUMAR GUPTA (184CA040)

Submitted To:

Ms. TEJASWINI GONDHALE

DECLARATION

We hereby declare that the work, which has been presented in this project report entitled "**STOCK MARKET SIMULATOR**" in partial fulfilment for the award of degree of **MASTER OF COMPUTER APPLICATIONS** from "**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA**", in department of mathematical and computer sciences is record of our own investigations carried under the guidance OF **Ms. TEJASWINI GONDHALE**.

Name of the candidates:

ADARSH THAKUR (194CA002)

RAVI KUMAR GAUTAM (194CA036)

RAJAT KUMAR JAIN (194CA035)

HARENDRA SINGH (184CA024)

MUKESH KUMAR GUPTA (184CA040)

CERTIFICATE

This is to certify that the P.G. Project Work Report entitled **"STOCK MARKET SIMULATOR"** submitted by **'ADARSH THAKUR'**(Roll No:-194CA002),**' RAVI KUMAR GAUTAM'** (Roll No:- 194CA036), **'RAJAT KUMAR JAIN'**(Roll No:-194CA035), **'HARENDRA SINGH'**(Roll No:-184CA024) and, **'MUKESH KUMAR GUPTA'**(Roll No:-184CA040) as the record of the work carried out by them, is accepted as the P.G. Project Work Report submission in partial fulfilment of the requirements for mandatory learning course of **MASTER OF COMPUTER APPLICATIONS** in the Department of **MATHEMATICAL AND COMPUTATIONAL SCIENCES.**

ABSTRACT

A stock market simulator is computer software that reproduces behaviour and features of a stock market, so that a user may practice trading stocks without financial risk. The use of an open-source, virtual stock exchange simulators can be seen as a viable alternative for various learners to train trading practice without risking loses of real funds. At present, there is a shortage of suitable tools that could allow entry level users to engage in a real market trading after gaining a certain degree of confidence from practicing in a both very practical and safe environment. We have built a stock market simulator using Python programming language.

CONTENTS

TECHNOLOGIES USED IN THE PROJECT

yfinance python library: - It's an open-source tool that uses Yahoo's publicly available APIs, and is intended for research and educational purposes. Real-time data is available during an exchange's market hours, and in some cases during pre-market and post-market hours.

Streamlit library: - Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps. In just a few minutes you can build and deploy powerful data apps.

PIL library:- Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

SOFTWARE AND HARDWARE REQUIREMENTS: -

Table 1: - MINIMUM HARDWARE REQUIREMENTS

Hardware	**Minimum Requirements**	**Reason**
Processor speed	1.6GHZ	Accommodate most pc
Memory of user pc	512MB RAM	Relatively fast
Memory of server pc	8GB	Fast
Disk space of server	200GB	Adequate storage

Table 2: - MINIMUM SOFTWARE REQUIREMENTS

Software	**Minimum requirements**	**Reason**
Operating system for pc	Windows 8,10, Mac	Accessed all locations
Browser	Firefox, chrome	Standard browser

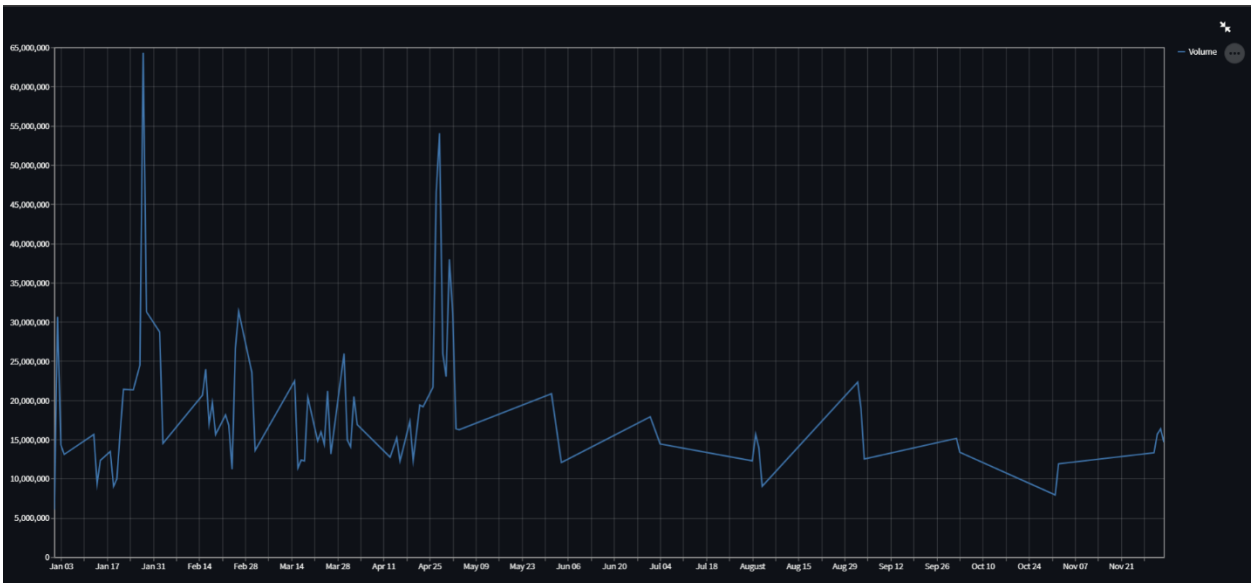
HOME UI: -

Code: -

```
#Import the libraries
import streamlit as st
from PIL import Image
import yfinance as yf



#Add a title and an image
st.write("""
#stock Market Web Application
***visually show data on a stock! Data range from Jan 1, 2021-April 30, 2021
""")

st.title('Stock Market Simulator')

image = Image.open("E:/learning/various technololgy learned/computer simulation project/Preview_Stock_Market_Data.jpg")
st.image(image, use_column_width=True)


#create a sidebar header
st.sidebar.header('User Input')


#create a function to get the users input
def get_input():
        start_date = st.sidebar.text_input("Start Date", "2021-01-01")
        end_date = st.sidebar.text_input("End Date", "2021-04-30")
```

```
        stock_symbol = st.sidebar.text_input("Stock Symbol", "tsla")
        return start_date, end_date, stock_symbol

#Get the users input
start, end, symbol = get_input()

tickerSymbol = symbol
tickerData = yf.Ticker(tickerSymbol)
tickerDf = tickerData.history(period='1d', start=start, end=end)
```

Line Chart visualization of any Company's Data: -



Code: -

```
st.header(tickerSymbol+" Volume\n")

st.line_chart(tickerDf['Volume'])
```

Statistics of Company's Data: -

	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
count	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000	80.0000
mean	702.7344	703.6038	715.8144	691.0956	703.9363	703.9150	703.7064	18,797,674.2250	1,322,450,000,000,000.0000	238,766.9875	6,286,828.9250	0.3312
std	40.9201	41.9850	41.1104	40.3933	39.9947	39.8649	40.1036	9,498,199.0145	652,076,156,536,988.0000	93,972.4318	3,458,399.7951	0.0688
min	620.4500	618.0000	625.4500	616.0000	623.4000	623.8000	622.1500	6,047,062.0000	376,000,000,000,000.0000	84,795.0000	1,473,702.0000	0.1977
25%	670.3250	669.6625	677.2875	659.6250	669.7875	671.0000	669.3425	13,262,247.7500	905,000,000,000,000.0000	177,038.0000	4,029,106.2500	0.2848
50%	703.4250	707.1000	715.1500	692.6250	707.5000	707.2750	701.0000	15,899,563.0000	1,160,000,000,000,000.0000	210,889.0000	5,213,754.0000	0.3293
75%	735.1375	735.5625	748.3125	722.0625	736.0625	735.1375	734.0175	21,342,064.5000	1,512,500,000,000,000.0000	281,198.5000	7,481,618.7500	0.3730
max	794.0000	799.0000	799.0000	767.0000	797.0000	794.0000	779.3300	64,303,975.0000	4,190,000,000,000,000.0000	573,501.0000	18,643,969.0000	0.5357

Code: -

```
st.header('Data Statistics')
st.write(tickerDf.describe())
```

Company's Calendar: -

Company Calander

	0	1
Earnings Date	2022-01-25T21:00:00	2022-01-31T21:00:00
Earnings Average	1970-01-01T00:00:00	1970-01-01T00:00:00
Earnings Low	1970-01-01T00:00:00	1970-01-01T00:00:00
Earnings High	1970-01-01T00:00:00	1970-01-01T00:00:00
Revenue Average	1970-01-01T04:10:42	1970-01-01T04:10:42
Revenue Low	1970-01-01T03:45:06	1970-01-01T03:45:06
Revenue High	1970-01-01T04:36:17	1970-01-01T04:36:17

Code: -

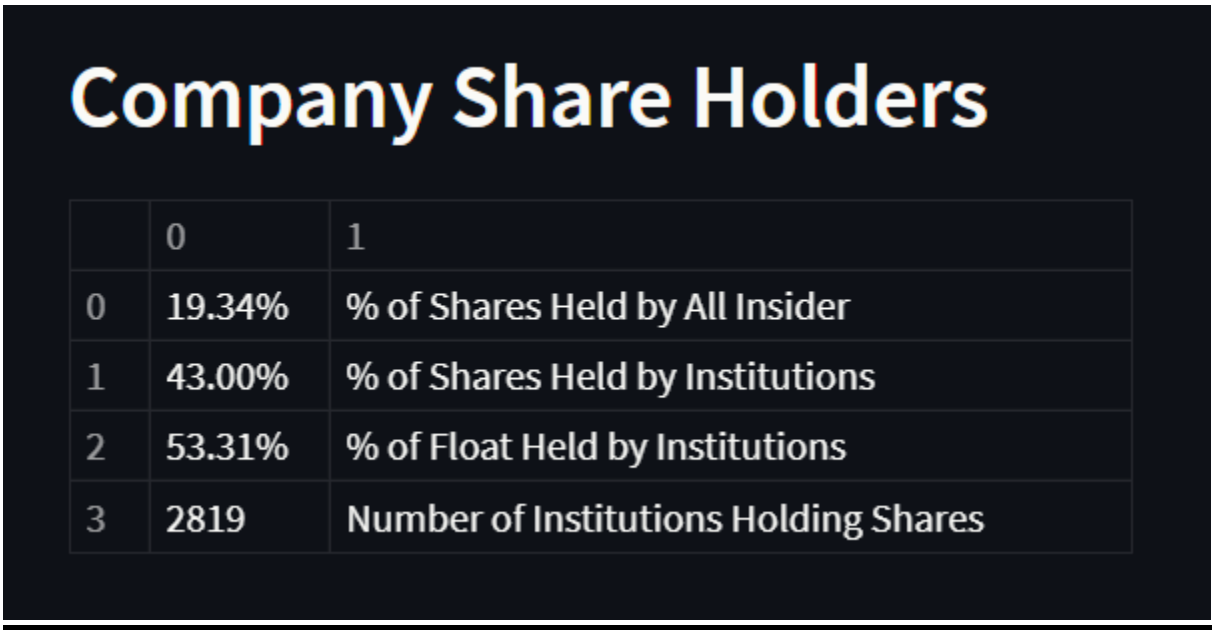
```
st.header('Company Calander')
st.write(tickerData.calendar)
```

Company Dividend Data: -

Code: -

```
st.header('Company Divident Data')

st.write(tickerData.dividends)
```

Company Share Holders: -



Code: -

```
st.header('Company Share Holders')
st.write(tickerData.major_holders)
```

Company Institutional Investors: -

	Holder	Shares	Date Reported	% Out	Value
0	Vanguard Group, Inc. (The)	61992012	2021-09-29T00:00:00	0.0617	48073565465
1	Blackrock Inc.	52829488	2021-09-29T00:00:00	0.0526	40968211354
2	Capital World Investors	37427314	2021-09-29T00:00:00	0.0373	29024133460
3	State Street Corporation	31230141	2021-09-29T00:00:00	0.0311	24218349742
4	Baillie Gifford and Company	13853124	2021-09-29T00:00:00	0.0138	10742820599
5	Geode Capital Management, LLC	13169939	2021-09-29T00:00:00	0.0131	10213024295
6	Jennison Associates LLC	10630339	2021-09-29T00:00:00	0.0106	8243615287
7	FMR, LLC	9921292	2021-09-29T00:00:00	0.0099	7693763520
8	Northern Trust Corporation	8634516	2021-09-29T00:00:00	0.0086	6695894467
9	Norges Bank Investment Management	7790070	2020-12-30T00:00:00	0.0078	5497218696

Code: -

```
st.header('Company Institutional Investors')
st.write(tickerData.institutional_holders)
```

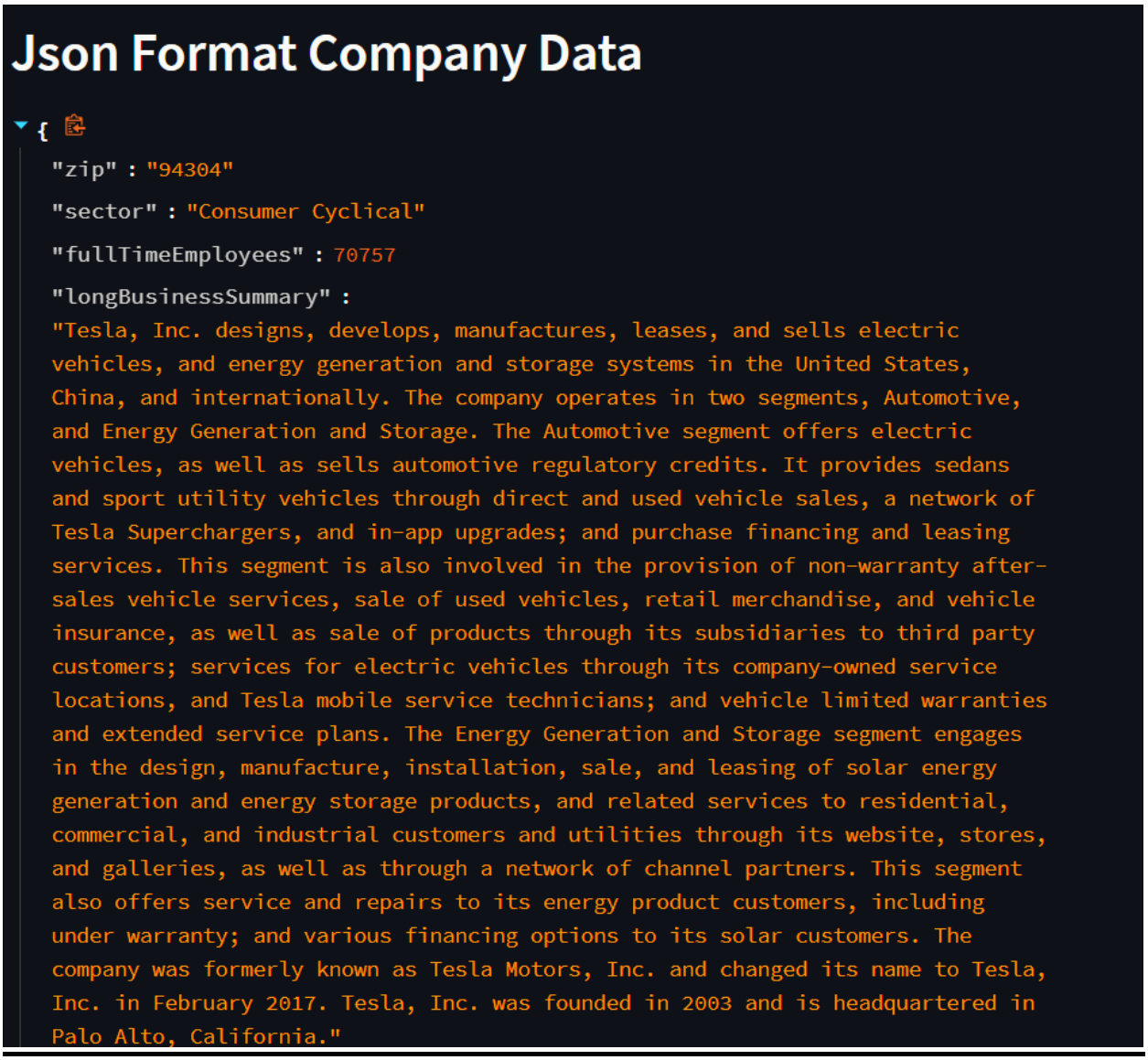
Company Recommendations: -

	Firm	To Grade	From Grade	Action
2012-02-16T07:42:00	JP Morgan	Overweight		main
2012-02-16T13:53:00	Wunderlich	Hold		down
2012-02-17T06:17:00	Oxen Group	Buy		init
2012-03-26T07:31:00	Wunderlich	Buy		up
2012-05-22T05:57:00	Maxim Group	Buy		init
2012-09-17T05:46:00	Morgan Stanley	Overweight	Neutral	up
2012-09-25T08:12:00	Goldman Sachs	Buy		main
2012-09-25T12:58:00	Bank of America	Neutral		main
2012-09-26T06:28:00	Jefferies	Buy		main
2012-10-08T07:56:00	Morgan Stanley	Overweight		main
2012-11-06T07:46:00	Jefferies	Buy		main
2012-12-18T06:10:31	JP Morgan	Neutral		init
2013-02-11T11:30:27	Jefferies	Buy		main
2013-02-21T06:53:02	Bank of America	Underperform	Neutral	down
2013-02-21T09:16:32	Deutsche Bank	Hold		main
2013-05-07T11:59:12	Jefferies	Buy		main
2013-05-09T06:21:48	JP Morgan	Neutral		main
2013-05-09T06:22:18	Goldman Sachs	Neutral	Buy	down
2013-05-14T09:27:31	Morgan Stanley	Overweight		main
2013-05-15T06:57:02	Deutsche Bank	Hold		main
2013-05-17T10:54:42	Morgan Stanley	Overweight		main
2013-05-20T09:41:45	Bank of America	Underperform		main

Code: -

```
st.header('Company Recommendations')
st.write(tickerData.recommendations)
```

Json Format Company Data: -



Code: -

st.header('Json Format Company Data')

st.write(tickerData.info)

Full Code: -

```
#Import the libraries
import streamlit as st
from PIL import Image
import yfinance as yf


#Add a title and an image
st.write("""
#stock Market Web Application
***visually show data on a stock! Data range from Jan 1, 2021-April 30, 2021
""")


st.title('Stock Market Simulator')


image = Image.open("E:/learning/various technololgy learned/computer simulation project/Preview_Stock_Market_Data.jpg")
st.image(image, use_column_width=True)



#create a sidebar header
st.sidebar.header('User Input')



#create a function to get the users input
def get_input():
        start_date = st.sidebar.text_input("Start Date", "2021-01-01")
        end_date = st.sidebar.text_input("End Date", "2021-04-30")
        stock_symbol = st.sidebar.text_input("Stock Symbol", "tsla")
```

```
    return start_date, end_date, stock_symbol

#Get the users input
start, end, symbol = get_input()



tickerSymbol = symbol
tickerData = yf.Ticker(tickerSymbol)
tickerDf = tickerData.history(period='1d', start=start, end=end)



#Display the close price
st.header(tickerSymbol+" Volume\n")
st.line_chart(tickerDf['Volume'])

# Get statistics ont he data
st.header('Data Statistics')
st.write(tickerDf.describe())

st.header('Company Calander')  #“calendar” function can be used to know about the earnings and revenue of the company.
st.write(tickerData.calendar)

st.header('Company Divident Data')  #A dividend can be described as a reward that publicly-listed companies extend to their shareholders. Dividends are sourced from company's net profits.
st.write(tickerData.dividends)

st.header('Company Share Holders')
```

```
st.write(tickerData.major_holders)

st.header('Company Institutional Investors')
st.write(tickerData.institutional_holders)

st.header('Company Recommendations')
st.write(tickerData.recommendations)

st.header('Json Format Company Data')
st.write(tickerData.info)



#stock symbol list
#https://stockanalysis.com/stocks/

#AAPL
#MSFT
#tsla
```

CONCLUSION: -

It helps to raise capital for business, mobilize savings for investment, facilitates the growth of companies, and enables profit sharing. It assists in creating investment opportunities for small investors, and raising capital for development projects taken up by the large organizations.