



School: SOET Campus: V2m
Academic Year: 2021/25 Subject Name: DAIP Subject Code: CUTM1018
Semester: 7th Program: B.TECH Branch: ECE Specialization: ECE
Date:

Applied and Action Learning

(Learning by Doing and Discovery)

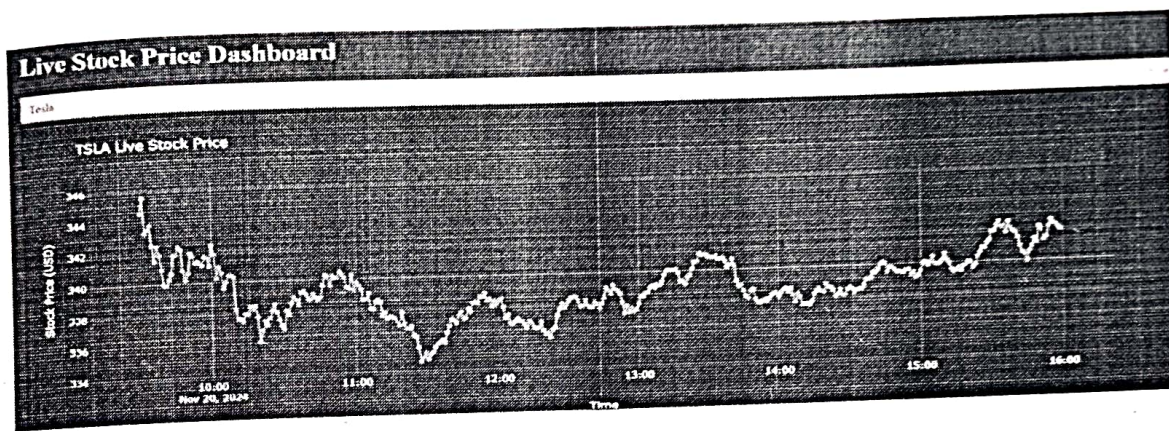
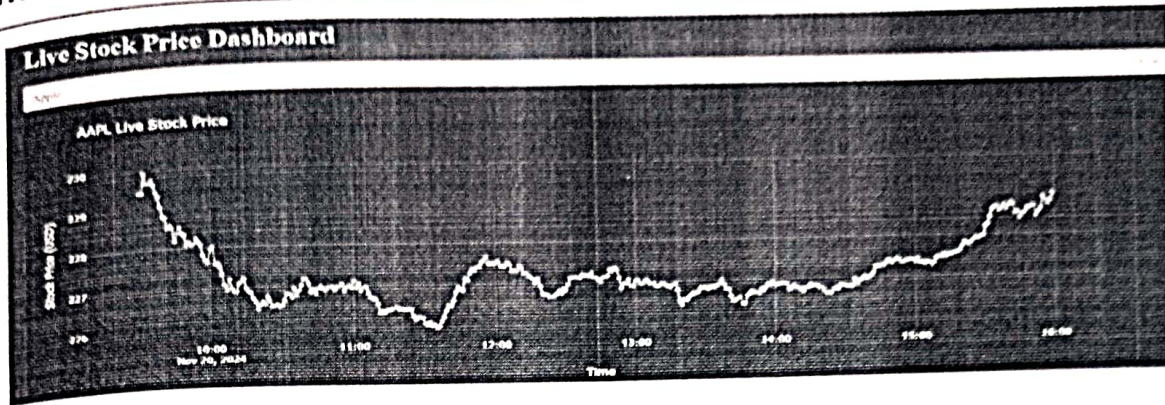
Name of the Experiment:

Coding Phase: Pseudo Code / Flow Chart / Algorithm

- Import necessary libraries like 'dash', 'Plotly', 'yfinance', & 'pandas' for dashboard creation and stock data fetching
- Initialize the Dash app and define the stock symbols for Apple, Microsoft, and Tesla.
- create a function to fetch stock data using 'yfinance' for the selected ticker, retrieving minute-level data.
- set up the dashboard layout with a header, dropdown for stock selection, graph for displaying stock prices, and interval for refreshing data every minute.

Testing Phase: Compilation of Code (error detection)

Implementation Phase: Final Output (no error)



ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	50		

Signature of the Student: *G. Sumanth*

Name : G. Sumanth

Regn. No. : 21801131001

Signature of the Faculty:

Page No.


```
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import plotly.graph_objs as go
import yfinance as yf
import pandas as pd
import time

app = dash.Dash(_name_)

companies = ['AAPL', 'MSFT', 'TSLA']
company_names = ['Apple', 'Microsoft', 'Tesla']

def get_stock_data(ticker):
    stock = yf.Ticker(ticker)
    df = stock.history(period="1d", interval="1m")
    df.reset_index(inplace=True)
    return df

initial_data = {}
for ticker in companies:
    initial_data[ticker] = get_stock_data(ticker)

app.layout = html.Div([
    html.H1('Live Stock Price Dashboard', style={'color':
        'Yellow'}),
```

```
dcc.Dropdown(
```

```
    id='stock-dropdown',
```

```
    options=[{'label': 'Company', 'value': ticker} for company, ticker in  
              zip(company_names, companies)],
```

```
    value="AAPL",
```

```
    multi=False
```

```
),
```

```
dcc.Graph(id='stock graph'),
```

```
dcc.Interval(
```

```
    id='interval-component',
```

```
    interval=60*1000,
```

```
    n_intervals=0
```

```
)
```

```
@app.callback(
```

```
    Output('stock-graph', 'figure'),
```

```
    [Input('interval-component', 'n-intervals'),
```

```
     Input('stock-dropdown', 'value')]
```

```
)
```

```
def update_stock_graph(n, selected_stock):
```

```
    stock_data = get_stock_data(selected_stock)
```

```
    figure = {
```

```
        'data': [
```

```
            go.Scatter(
```

```
                x=stock_data['datetime'],
```

```
                y=stock_data['close'],
```

```
            )
```

```
if __name__ == '__main__':
```

```
    app.run_server(debug=True)
```