Centurion University

School: SOET Campus: VZM

Academic Year: 2021/25 Subject Name: DAVP Subject Code: CUTM1018.

Semester: 7th Program: B.TECH Branch: ECE Specialization: ECE

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

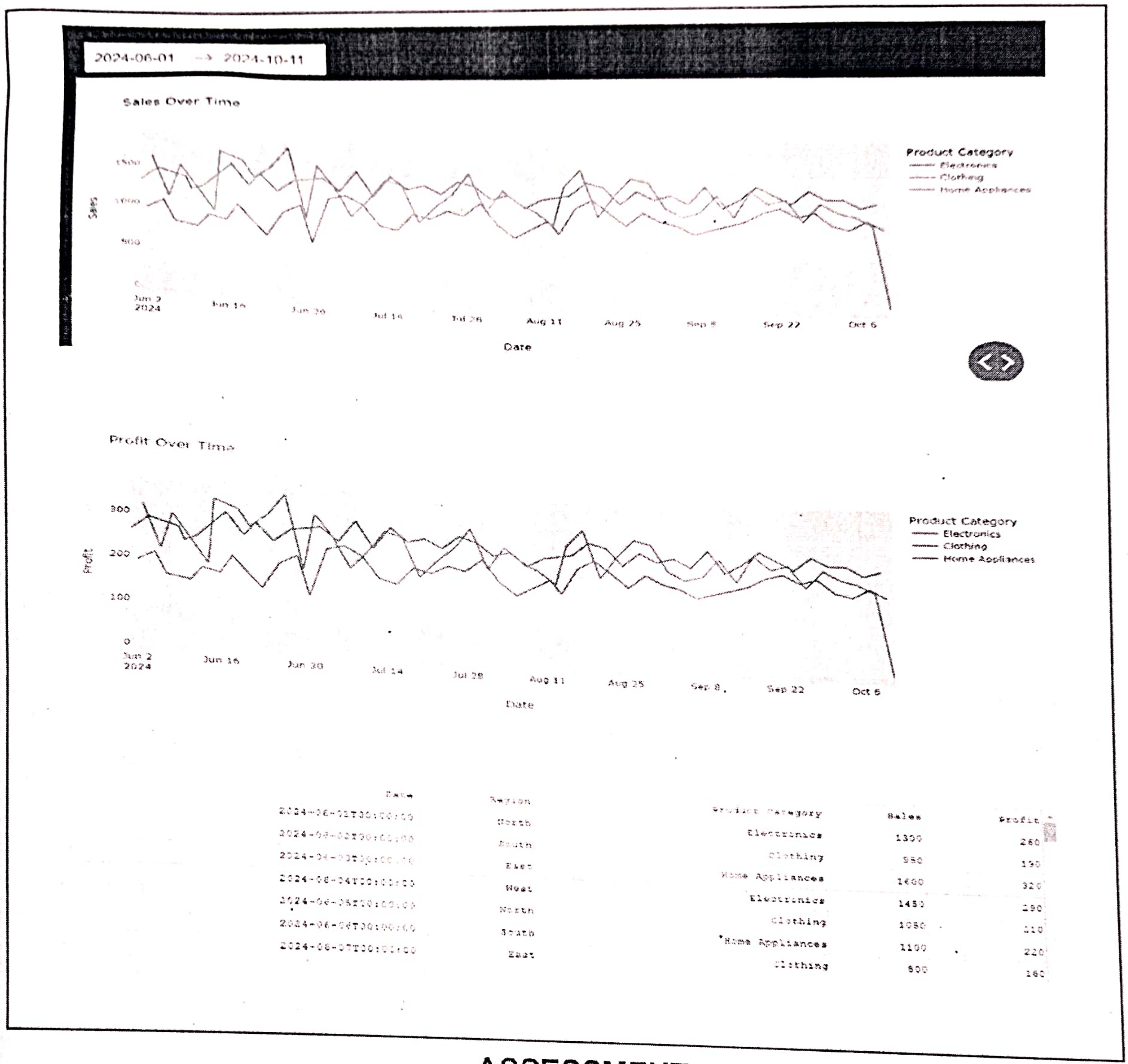
Name of the Experiement : Design a dashboard layout that includes an interactive Component to is display sales.

Coding Phase: Pseudo Code / Flow Chart / Algorithm

```
→ Import necessary libraries
→ load and preprocess the dataset
→ group data by relevant categories
→ create a Dash app layout with components
→ set up callback functions to filter data and update graphs
  and table based on selected date range.
→ Run the Dash app for interactivity and dynamic updates.
```

Testing Phase: Compilation of Code (error detection)

Implementation Phase: Final Output (no error)



Date	Region	Product Category	Sales	Profit
2024-06-01T00:00:00	North	Electronics	1300	260
2024-06-02T00:00:00	South	Clothing	950	190
2024-06-03T00:00:00	East	Home Appliances	1600	320
2024-06-04T00:00:00	West	Electronics	1450	290
2024-06-05T00:00:00	North	Clothing	1050	210
2024-06-06T00:00:00	South	Home Appliances	1100	220
2024-06-07T00:00:00	East	Clothing	800	160

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. S

Name : G. Sumanth

Regn. No. : 211801131001

Signature of the Faculty:

```
Import dash
from dash import dcc, html
Import plotly. express as px
import pandas as pd
from dash. dependencies import Input, Output
Import dash-table
df = pd. read_csv ('scales_data.csv')
df ['Date'] = pd.to_datetime (df ['Date'])
df ['Profit'] = df ['Sales'] * 0.2
grouped_df = df.groupby ([ 'Date'; 'Region', 'Product category'])
        'Sales' : 'Sum',
        'Profit' : 'Sum'
}).reset_index()
app = dash.Dash (_name_)
app. layout = html.DIV ([
    html.H1 ("Sales Dashboard"),
 dcc.DatepickerRange(
        id = 'date-picker-range',
    Start_date = df ['Date'].min().date(),
    end-date = df ['Date'].max().date(),
    display_format = 'YYYY-mm-DD',
    Style = {'padding': '10px'}
  ),
table_data = filtered_df.to_dict ('records')
return Sales_fig, profit_fig, table_data
if _name_ == '_main_':
    app.run_server(debug=True, port=8052)
```