0. Record the (using the google map, screen recording on cellphone) journeyfor at least 10 kms (or 30mins of travel) which includes the start of the journey and the end of the journey both.Tasks:

a. Plot the speed versus time graph (using a python code with image processing or manually)

b. plot the instantaneous value of the-time-you-would-take-to-travel-10km versus time graph

c. c1. plot distance covered every two minutes as below time-stamp time-taken

0min (to be plotted at 0min mark on the graph) x0min=ZERO
0min-2min (to be plotted at 1min mark on the graph) x1min
1min-3min (to be plotted at 2min mark on the graph) x2min
2min-4min (to be plotted at 3min mark on the graph) x3min
3min-5min (to be plotted at 4min mark on the graph) x4min
4min-6min (to be plotted at 5min mark on the graph) x5min
5min-7min (to be plotted at 6min mark on the graph) x6min
6min-8min (to be plotted at 7min mark on the graph) x7min
7min-9min (to be plotted at 8min mark on the graph) x8min
8min-10min (to be plotted at 9min mark on the graph) x9min
9min-11min (to be plotted at 10min mark on the graph) x10min
10min-12min (to be plotted at 11min mark on the graph) x11min 11min-13min (to be plotted at 12min mark on the graph) x12min and similar till the end of the journey

c2. using the above graph, calculate the total distance covered by the vehicle. Calculate the percentage error.

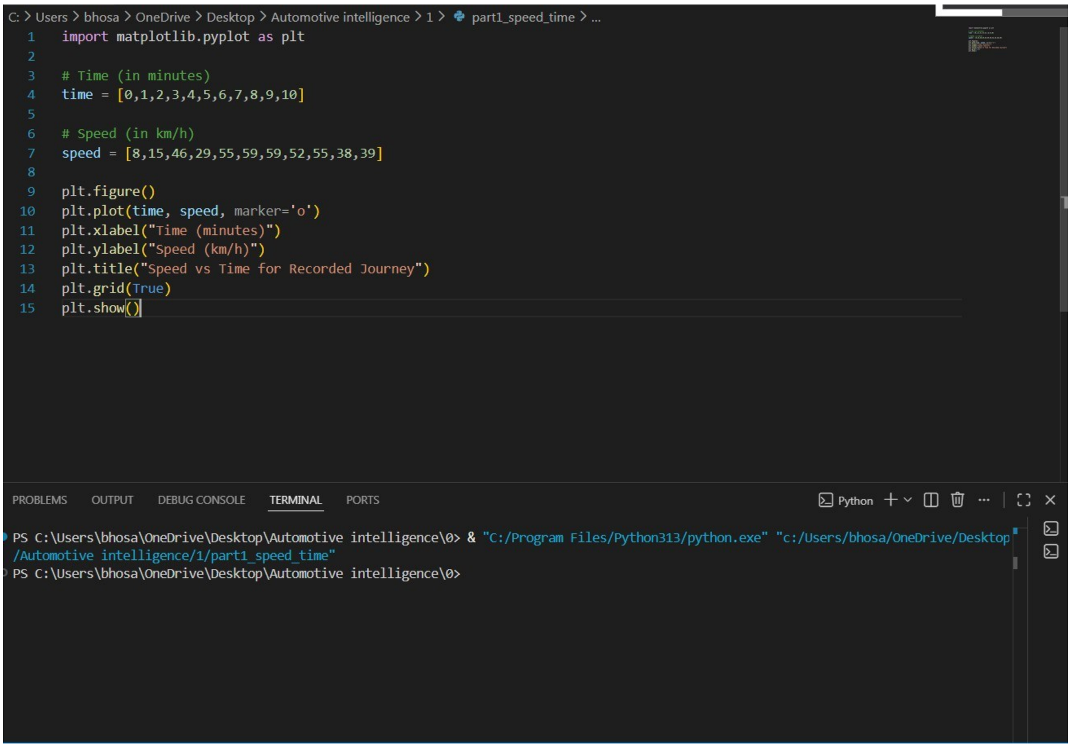
c3. Calculate the instantaneous speed value and compare with the instantaneous speed from the screen recording, and calculate the absolute error, mean error, RMSE (root mean square error)

d. plot the instantaneous acceleration versus time graph

e. plot the instantaneous jerk versus time graph

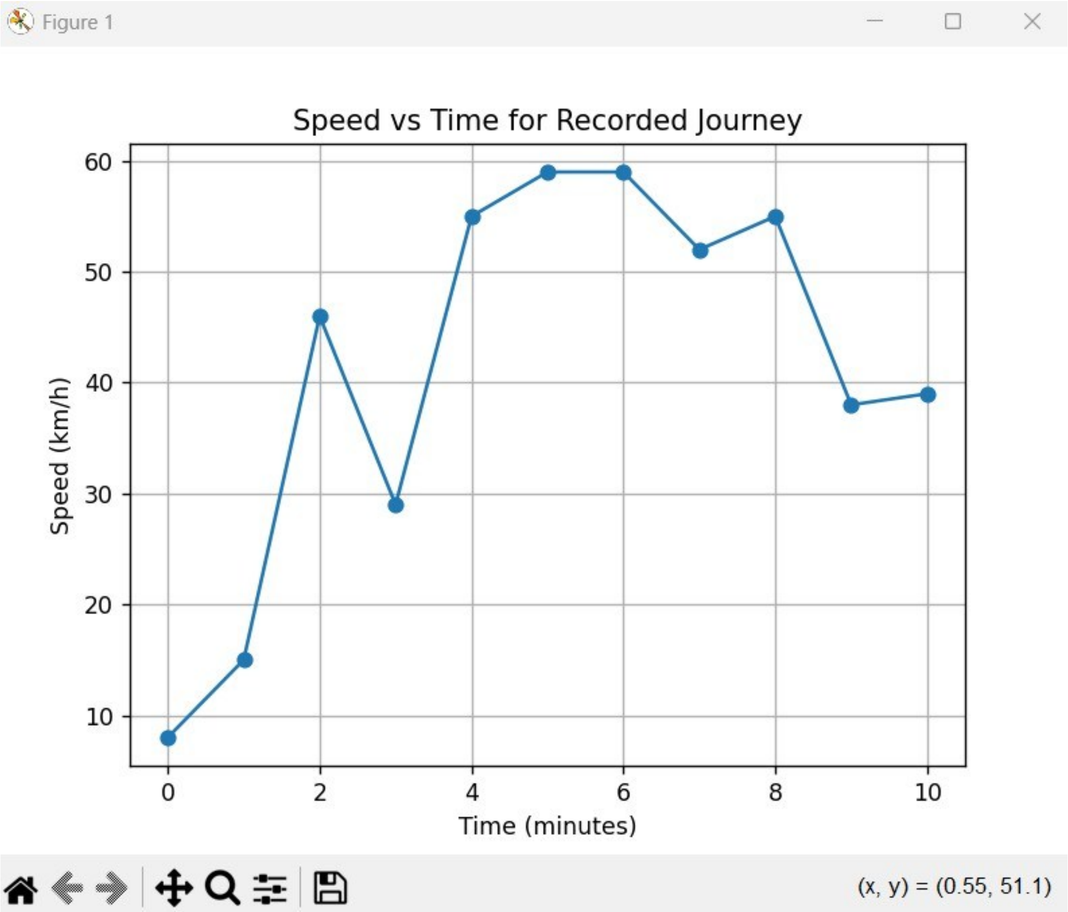
Solution:

a. speed versus time graph:

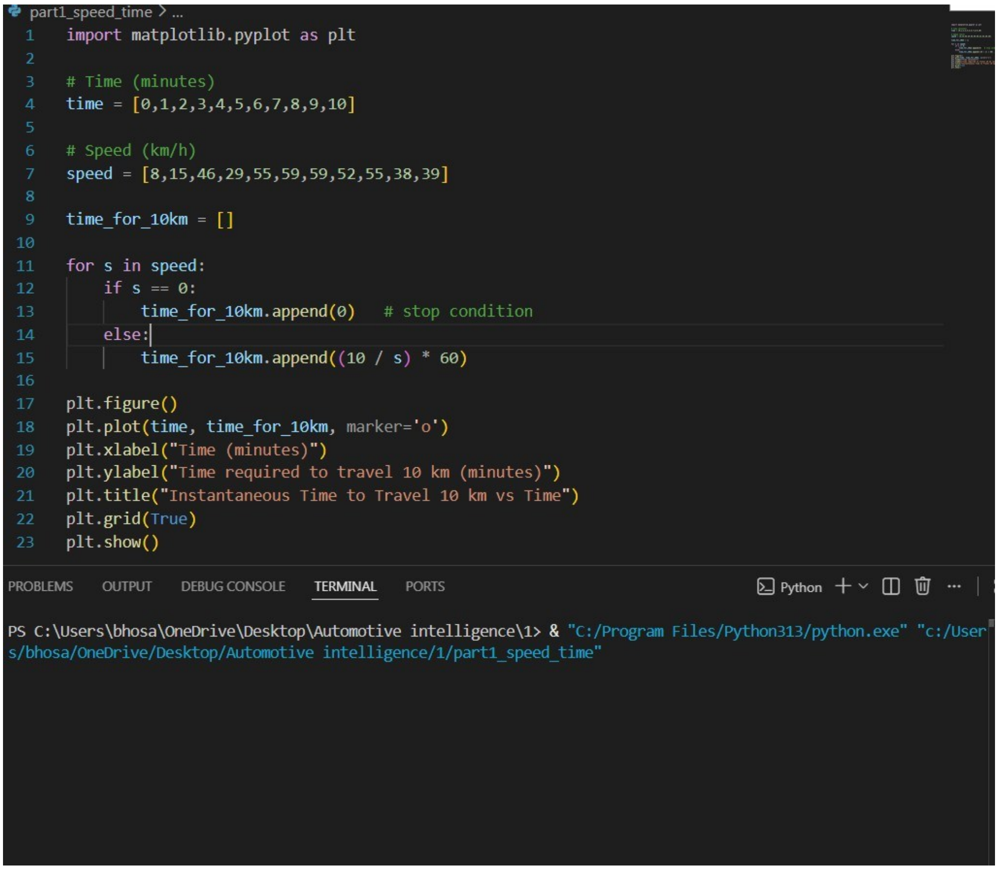

C: > Users > bhosa > OneDrive > Desktop > Automotive intelligence > 1 > part1_speed_time > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

```python
import matplotlib.pyplot as plt

# Time (in minutes)
time = [0,1,2,3,4,5,6,7,8,9,10]

# Speed (in km/h)
speed = [8,15,46,29,55,59,59,52,55,38,39]

plt.figure()
plt.plot(time, speed, marker='o')
plt.xlabel("Time (minutes)")
plt.ylabel("Speed (km/h)")
plt.title("Speed vs Time for Recorded Journey")
plt.grid(True)
plt.show()
```

```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed_time"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0>
```

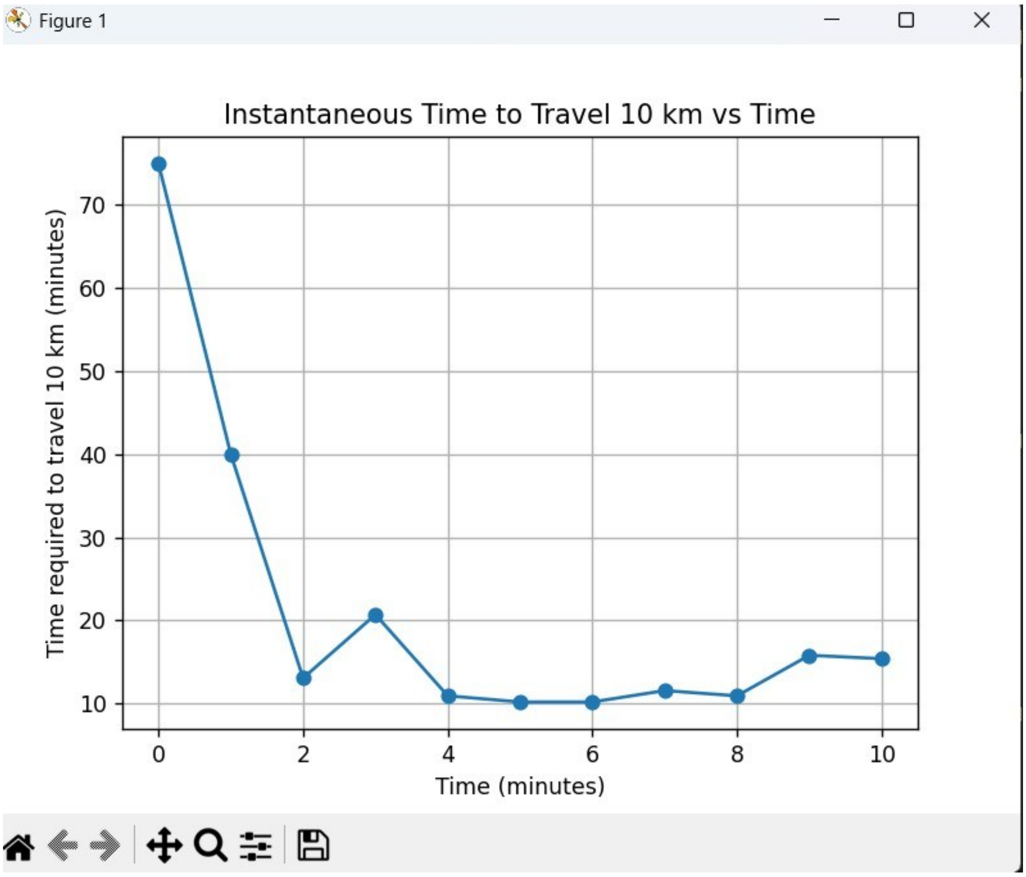


b. plot the instantaneous value of the-time-you-would-take-to-travel-10km versus time graph



```python
import matplotlib.pyplot as plt

# Time (minutes)
time = [0,1,2,3,4,5,6,7,8,9,10]

# Speed (km/h)
speed = [8,15,46,29,55,59,59,52,55,38,39]

time_for_10km = []

for s in speed:
    if s == 0:
        time_for_10km.append(0)    # stop condition
    else:
        time_for_10km.append((10 / s) * 60)

plt.figure()
plt.plot(time, time_for_10km, marker='o')
plt.xlabel("Time (minutes)")
plt.ylabel("Time required to travel 10 km (minutes)")
plt.title("Instantaneous Time to Travel 10 km vs Time")
plt.grid(True)
plt.show()
```

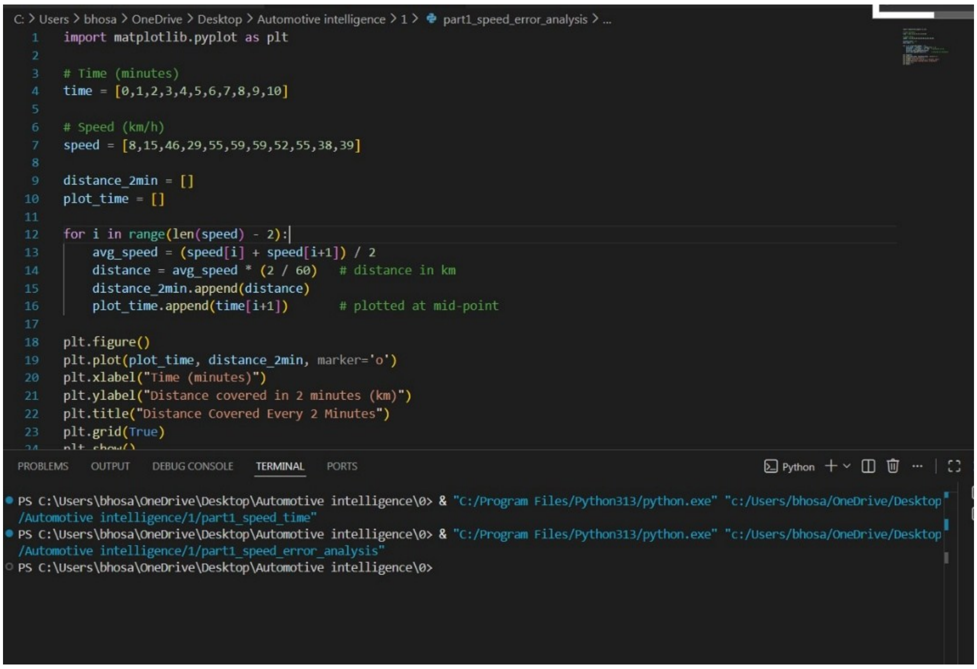
```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\1> & "C:/Program Files/Python313/python.exe" "c:/User
s/bhosa/OneDrive/Desktop/Automotive intelligence/1/part1_speed_time"
```



C.

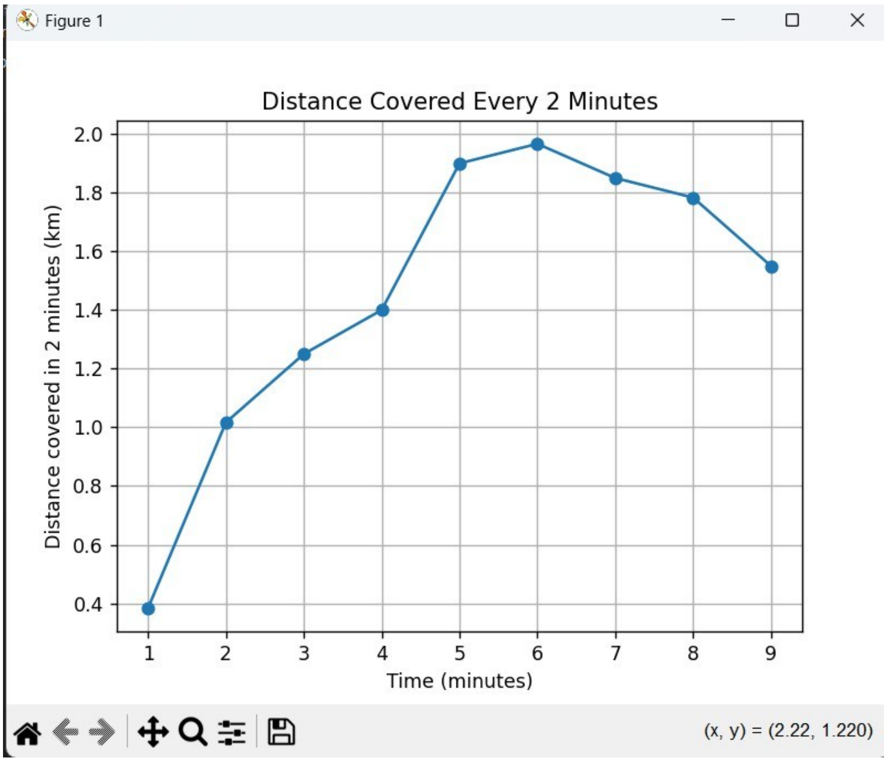
c1. plot distance covered every two minutes as below time-stamp time-taken

Time (minutes)	Speed (km/h)
0	8
1	15
2	46
3	29
4	55
5	59
6	59
7	52
8	55
9	38
10	39


C: > Users > bhosa > OneDrive > Desktop > Automotive intelligence > 1 > part1_speed_error_analysis > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

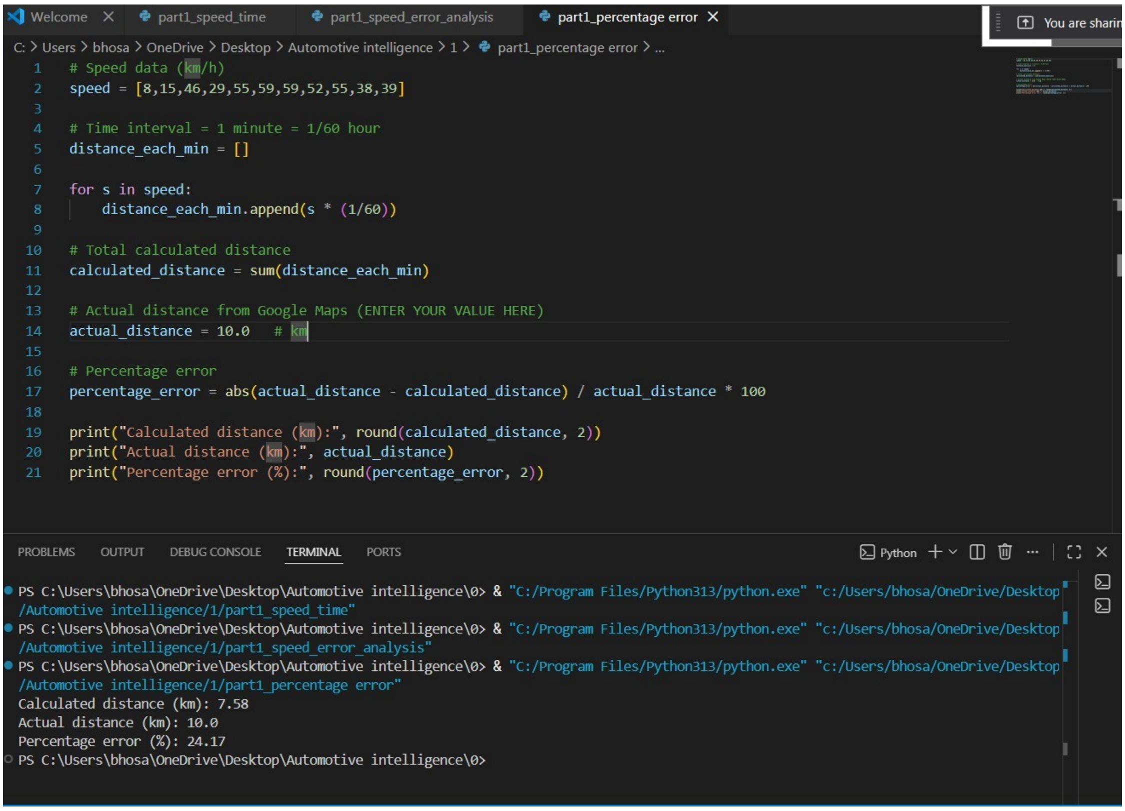
```python
import matplotlib.pyplot as plt

# Time (minutes)
time = [0,1,2,3,4,5,6,7,8,9,10]

# Speed (km/h)
speed = [8,15,46,29,55,59,59,52,55,38,39]

distance_2min = []
plot_time = []

for i in range(len(speed) - 2):
    avg_speed = (speed[i] + speed[i+1]) / 2
    distance = avg_speed * (2 / 60)   # distance in km
    distance_2min.append(distance)
    plot_time.append(time[i+1])       # plotted at mid-point

plt.figure()
plt.plot(plot_time, distance_2min, marker='o')
plt.xlabel("Time (minutes)")
plt.ylabel("Distance covered in 2 minutes (km)")
plt.title("Distance Covered Every 2 Minutes")
plt.grid(True)
plt.show()
```

```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed_time"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed_error_analysis"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0>
```



c2. Using the above graph, calculate the total distance covered by the vehicle. Calculate the percentage error.

Solution: Percentage error- 24.17 %


Welcome
part1_speed_time
part1_speed_error_analysis
part1_percentage error
You are sharin
C: > Users > bhosa > OneDrive > Desktop > Automotive intelligence > 1 > part1_percentage error > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

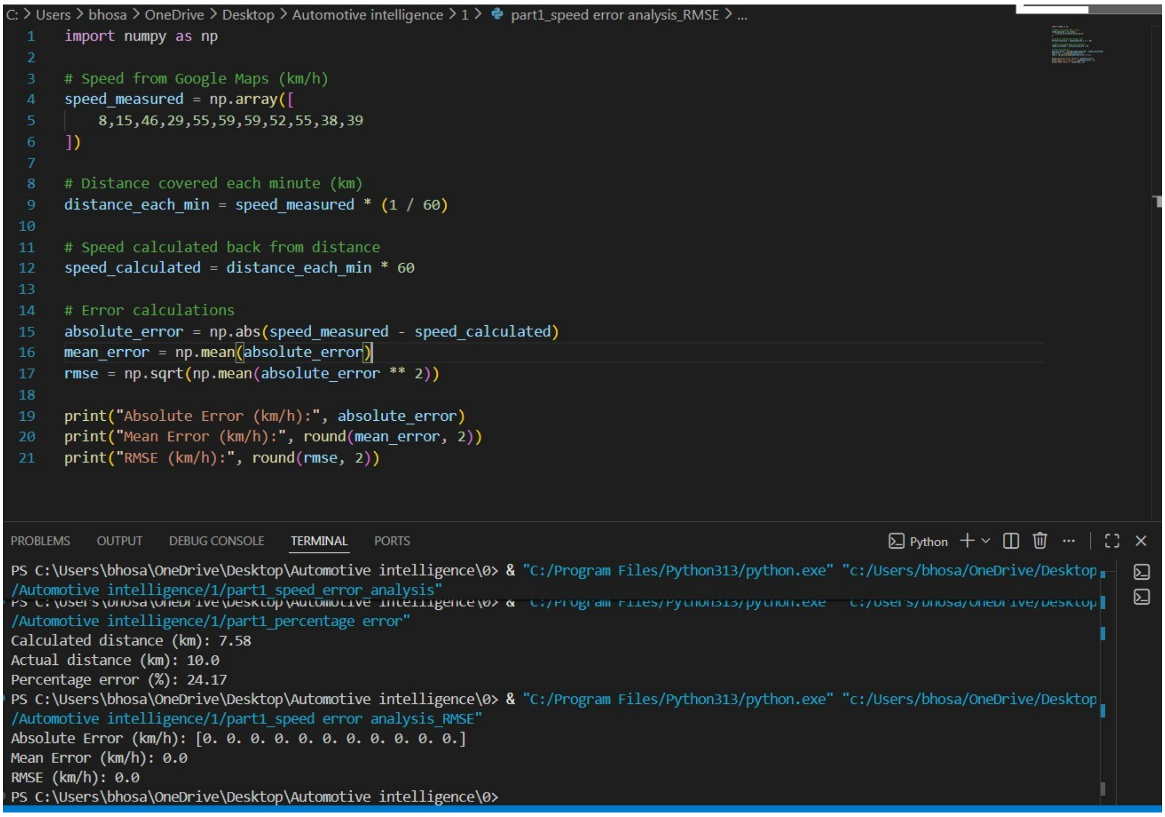
```python
# Speed data (km/h)
speed = [8,15,46,29,55,59,59,52,55,38,39]

# Time interval = 1 minute = 1/60 hour
distance_each_min = []

for s in speed:
    distance_each_min.append(s * (1/60))

# Total calculated distance
calculated_distance = sum(distance_each_min)

# Actual distance from Google Maps (ENTER YOUR VALUE HERE)
actual_distance = 10.0   # km

# Percentage error
percentage_error = abs(actual_distance - calculated_distance) / actual_distance * 100

print("Calculated distance (km):", round(calculated_distance, 2))
print("Actual distance (km):", actual_distance)
print("Percentage error (%):", round(percentage_error, 2))
```

```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed_time"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed_error_analysis"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_percentage error"
Calculated distance (km): 7.58
Actual distance (km): 10.0
Percentage error (%): 24.17
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0>
```

c3. Calculate the instantaneous speed value and compare with the instantaneous speed from the screen recording, and calculate the absolute error, mean error, RMSE (root mean square error).

Solution:

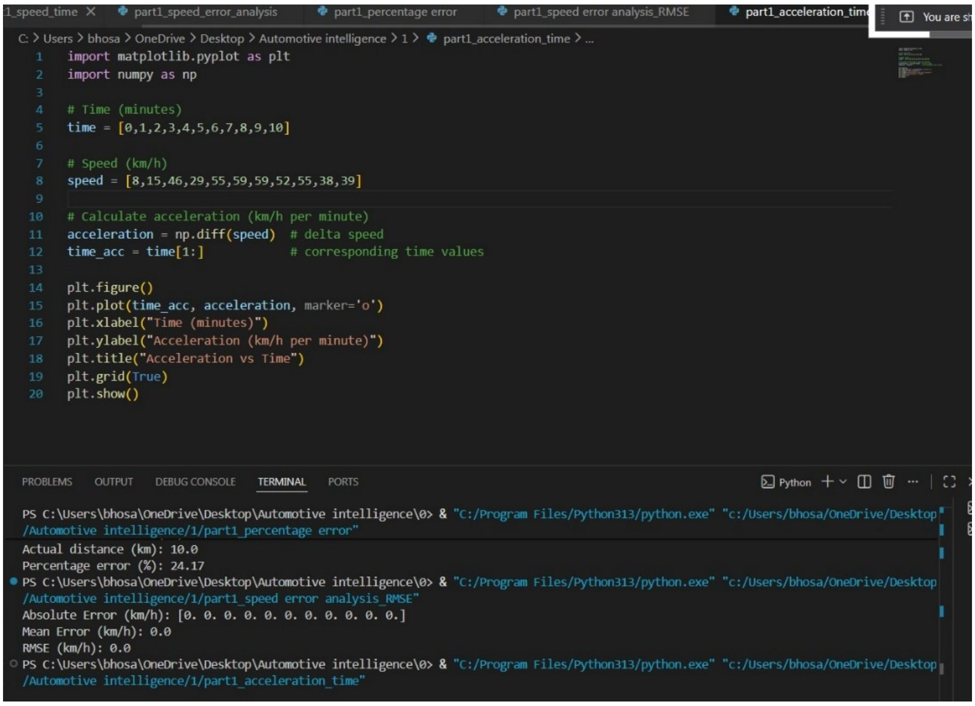
a. Absolute error: 0.0
b. Mean error: 0.0
c. RMSE :0.0


C: > Users > bhosa > OneDrive > Desktop > Automotive intelligence > 1 > part1_speed error analysis_RMSE > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

```python
import numpy as np

# Speed from Google Maps (km/h)
speed_measured = np.array([
    8,15,46,29,55,59,59,52,55,38,39
])

# Distance covered each minute (km)
distance_each_min = speed_measured * (1 / 60)

# Speed calculated back from distance
speed_calculated = distance_each_min * 60

# Error calculations
absolute_error = np.abs(speed_measured - speed_calculated)
mean_error = np.mean(absolute_error)
rmse = np.sqrt(np.mean(absolute_error ** 2))

print("Absolute Error (km/h):", absolute_error)
print("Mean Error (km/h):", round(mean_error, 2))
print("RMSE (km/h):", round(rmse, 2))
```

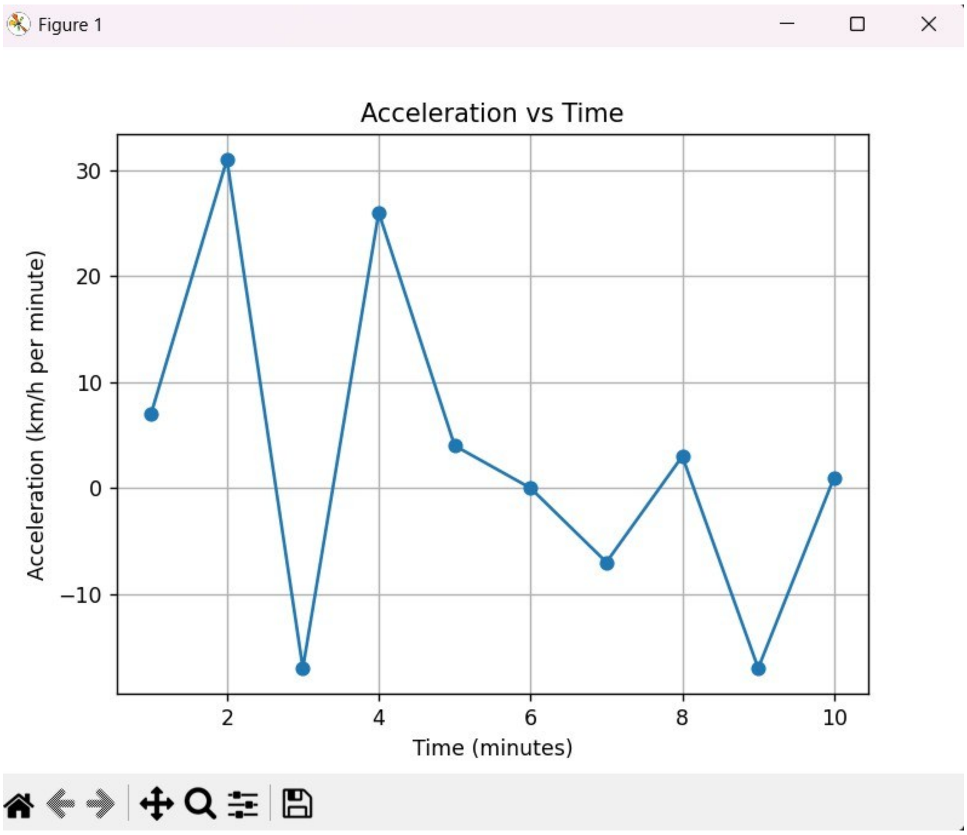
```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed_error_analysis"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_percentage error"
Calculated distance (km): 7.58
Actual distance (km): 10.0
Percentage error (%): 24.17
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed error analysis_RMSE"
Absolute Error (km/h): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Mean Error (km/h): 0.0
RMSE (km/h): 0.0
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0>
```

d. plot the instantaneous acceleration versus time graph

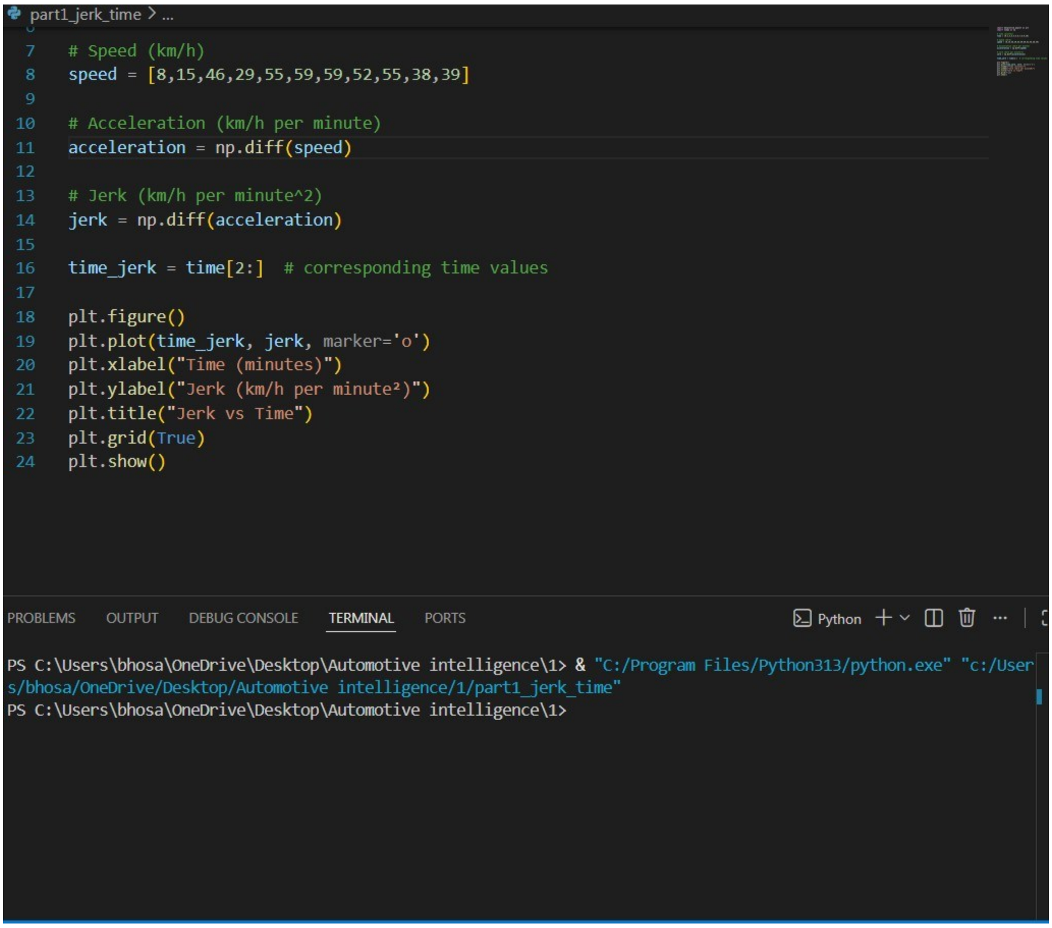

1_speed_time
part1_speed_error_analysis
part1_percentage error
part1_speed error analysis_RMSE
part1_acceleration_time
You are sh
C: > Users > bhosa > OneDrive > Desktop > Automotive intelligence > 1 > part1_acceleration_time > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

```python
import matplotlib.pyplot as plt
import numpy as np

# Time (minutes)
time = [0,1,2,3,4,5,6,7,8,9,10]

# Speed (km/h)
speed = [8,15,46,29,55,59,59,52,55,38,39]

# Calculate acceleration (km/h per minute)
acceleration = np.diff(speed)  # delta speed
time_acc = time[1:]            # corresponding time values

plt.figure()
plt.plot(time_acc, acceleration, marker='o')
plt.xlabel("Time (minutes)")
plt.ylabel("Acceleration (km/h per minute)")
plt.title("Acceleration vs Time")
plt.grid(True)
plt.show()
```

```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_percentage error"
Actual distance (km): 10.0
Percentage error (%): 24.17
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_speed error analysis_RMSE"
Absolute Error (km/h): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Mean Error (km/h): 0.0
RMSE (km/h): 0.0
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\0> & "C:/Program Files/Python313/python.exe" "c:/Users/bhosa/OneDrive/Desktop
/Automotive intelligence/1/part1_acceleration_time"
```



e. plot the instantaneous jerk versus time graph



```python
# Speed (km/h)
speed = [8,15,46,29,55,59,59,52,55,38,39]

# Acceleration (km/h per minute)
acceleration = np.diff(speed)

# Jerk (km/h per minute^2)
jerk = np.diff(acceleration)

time_jerk = time[2:]  # corresponding time values

plt.figure()
plt.plot(time_jerk, jerk, marker='o')
plt.xlabel("Time (minutes)")
plt.ylabel("Jerk (km/h per minute²)")
plt.title("Jerk vs Time")
plt.grid(True)
plt.show()
```

```
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\1> & "C:/Program Files/Python313/python.exe" "c:/User
s/bhosa/OneDrive/Desktop/Automotive intelligence/1/part1_jerk_time"
PS C:\Users\bhosa\OneDrive\Desktop\Automotive intelligence\1>
```

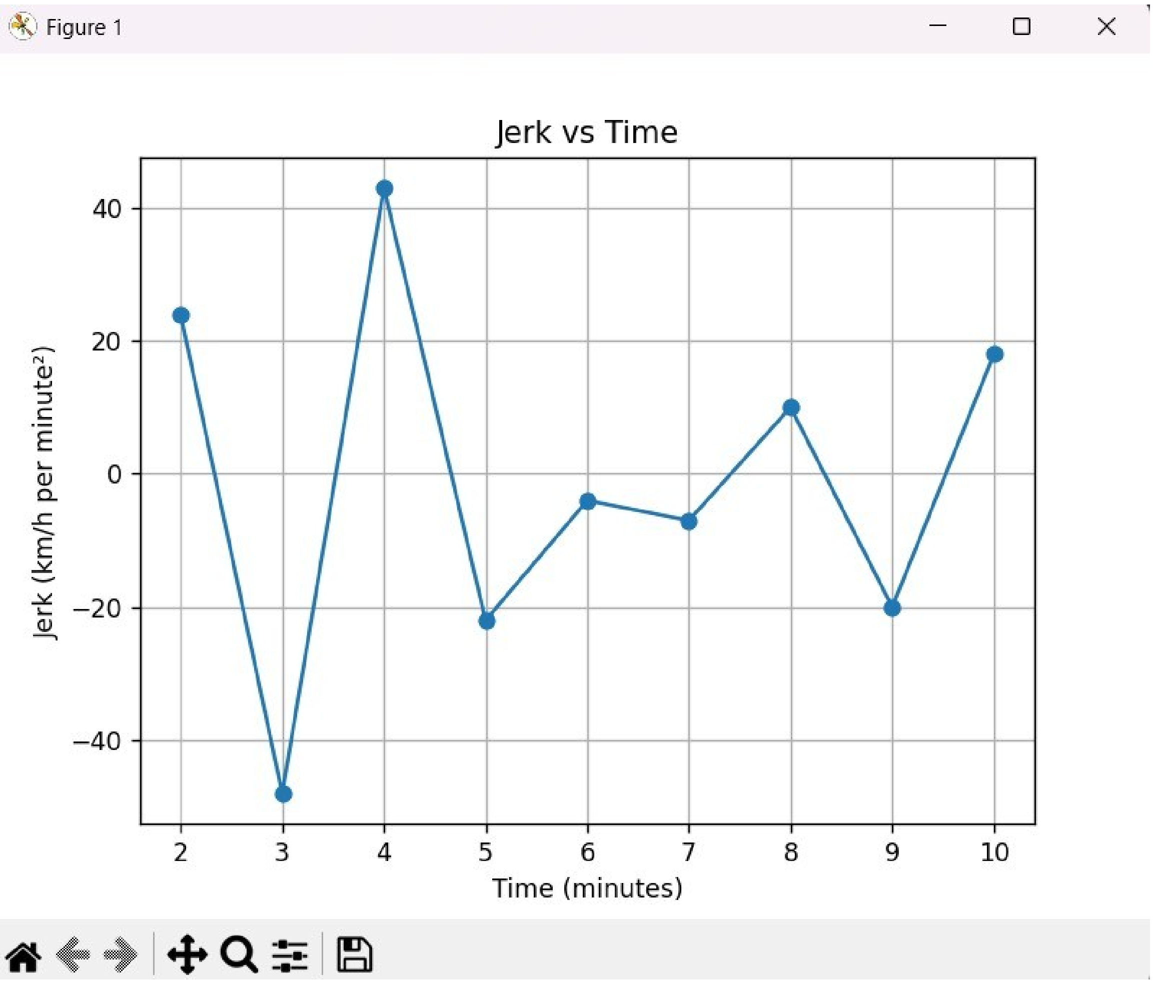


Figure 1
Jerk vs Time
Jerk (km/h per minute²)
Time (minutes)