**Dashboard Plotting Temperature and Humidity Project**

**(for Redspectra)**

**Project Requirements Document :**

Create a Dashboard for live stream plotting the data of Temperature and Humidity with Gauge meter , data gather from server (https-url-github)

**Layout :**

1. Temperature graph :

X axis **Time** and Y axis **Values**

2. Humidity graph :

X axis **Time** and Y axis **Values**

3. Plot type : **Area**

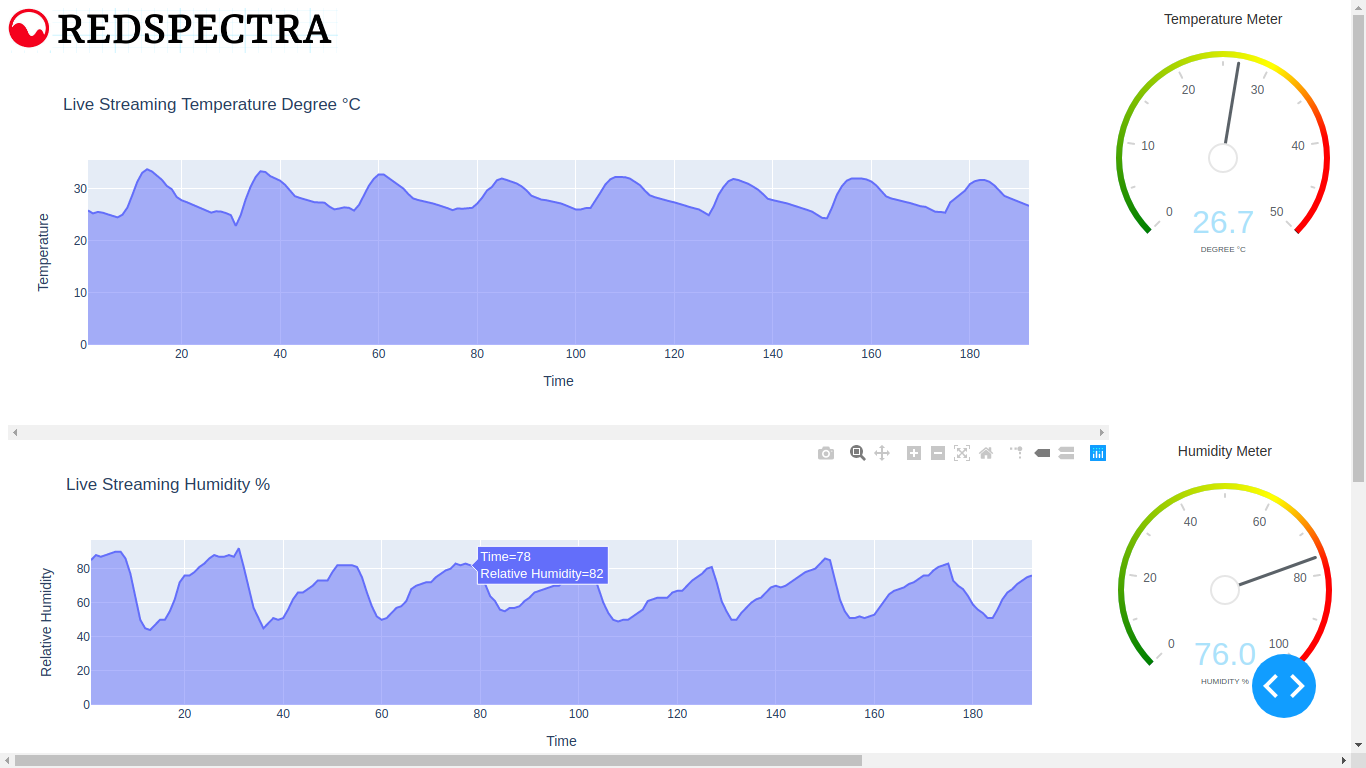
4. Each graph have Gauge meter side by

These meter showing **recently updated value**

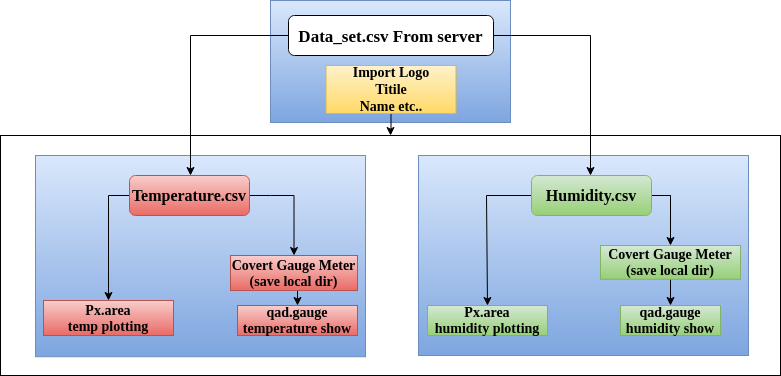
**SYSTEM REQUIREMENT :**

1. Python version ( 2.7 above)
2. Dash App (1.9.1)
3. Daq (0.4.0)
4. Core components(1.8.1)
5. Html components(1.0.2)
6. Plotly (4.5.4)
7. Web Browser
8. Pandas (0.24.2)
9. Base64
10. # coding=utf-8

**Output Structure :**



**Dashboard App Flow Chart:**



**Coding:**

# coding=utf-8

import base64

import dash\_daq as daq

import dash

import dash\_core\_components as dcc

import dash\_html\_components as html

import pandas as pd

import plotly.express as px

app = dash.Dash(\_\_name\_\_, external\_stylesheets=['https://codepen.io/amyoshino/pen/jzXypZ.css'])

image\_filename = 'logo.png'

encoded\_image = base64.b64encode(open(image\_filename, 'rb').read())

app.title = 'Redspectra'

df = pd.read\_csv('https://raw.githubusercontent.com/vimalv-AI/Vimal/master/dashboard/data/Temperature.csv')

# If you know the name of the column skip this

first\_column = df.columns[0]

# Delete first

d1 = df.drop([first\_column], axis=1)

d1.to\_csv('1\_temperature.csv', index=False)

with open('1\_temperature.csv', 'r') as f:

for row in f:

pass

df1 = pd.read\_csv('https://raw.githubusercontent.com/vimalv-AI/Vimal/master/dashboard/data/Humidity.csv')

# If you know the name of the column skip this

first\_column = df1.columns[0]

# Delete first

df2 = df1.drop([first\_column], axis=1)

df2.to\_csv('2\_humidity.csv', index=False)

with open('2\_humidity.csv', 'r') as f1:

for row1 in f1:

pass

df7 = pd.read\_csv('https://raw.githubusercontent.com/vimalv-AI/Vimal/master/dashboard/data/tmp\_hum.csv')

app.layout = html.Div([

html.Div([

html.Img(src='data:image/png;base64,{}'.format(encoded\_image), height=45, width=330),

dcc.Graph(id='example1', style={'width': 1101, 'height': 380, 'overflowX': 'scroll'},

figure=px.area(df, x='Time', y='Temperature', title='Live Streaming Temperature Degree °C ', )

)

], className='six columns'

),

html.Div([ # F

daq.Gauge(

id='my-gauge',

style={'width': 2100, 'height': 0},

color={"gradient": True,

"ranges": {"green": [0, 25], "yellow": [25, 35], "red": [35, 50]}},

showCurrentValue=True,

units="Degree °C ",

value=float(row),

label='Temperature Meter',

max=50,

min=0,

)

], className="row"

),

html.Div([

dcc.Graph(id='example5', style={'width': 1101, 'height': 360, 'overflowX': 'scroll'},

figure=px.area(df1, x='Time', y='Relative Humidity', title='Live Streaming Humidity % ', )

)

], className='six columns'

),

html.Div([

daq.Gauge(

id='my-gauge1',

style={'width': 2100, 'height': 0},

color={"gradient": True,

"ranges": {"green": [0, 50], "yellow": [50, 70], "red": [70, 100]}},

showCurrentValue=True,

units=" Humidity % ",

value=float(row1),

label='Humidity Meter',

max=100,

min=0,

)

]

),

dcc.Graph(id='example7', style={'width': 1400, 'height': 365.5, 'overflowX': 'scroll'},

figure=px.area(df7, x='Temperature', y='Relative Humidity',

title='COMPARING TEMPERATURE °C AND HUMIDITY % ', )

)

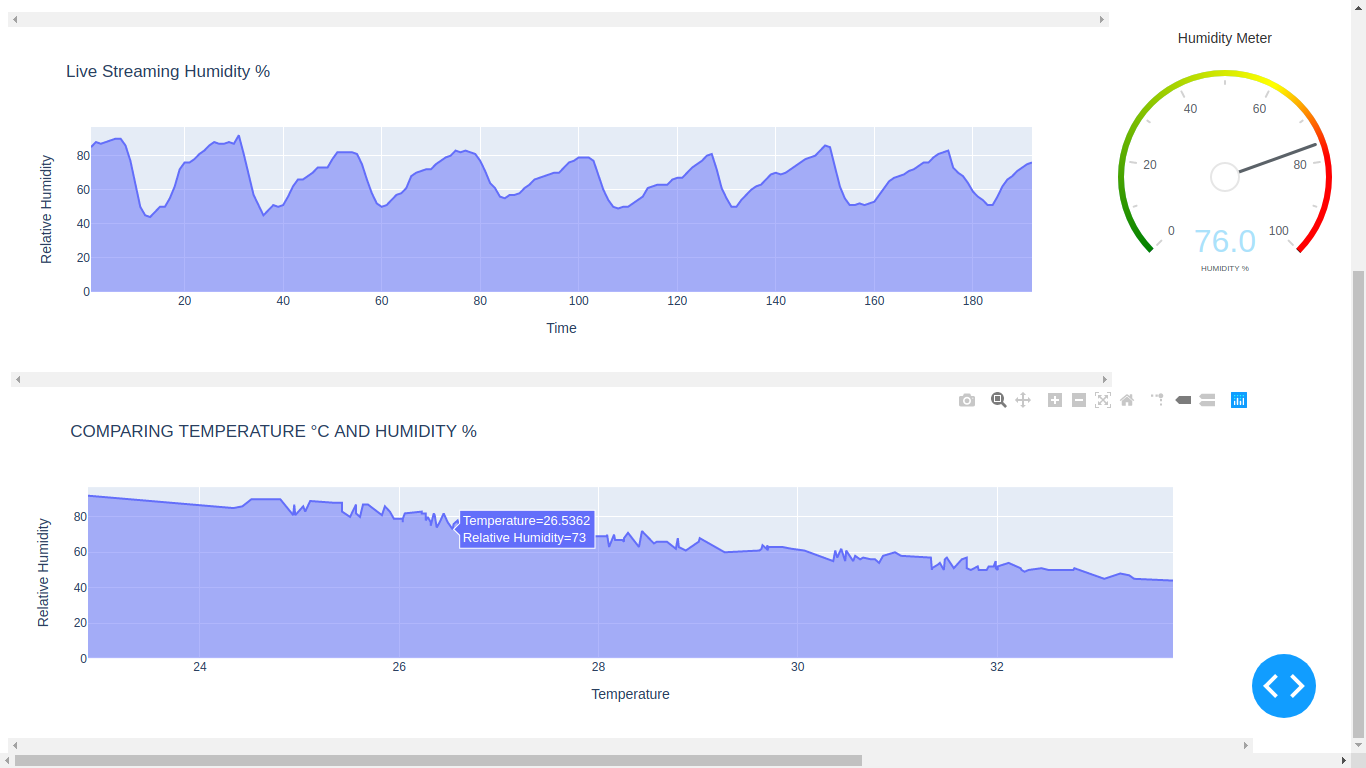
], className='six columns'

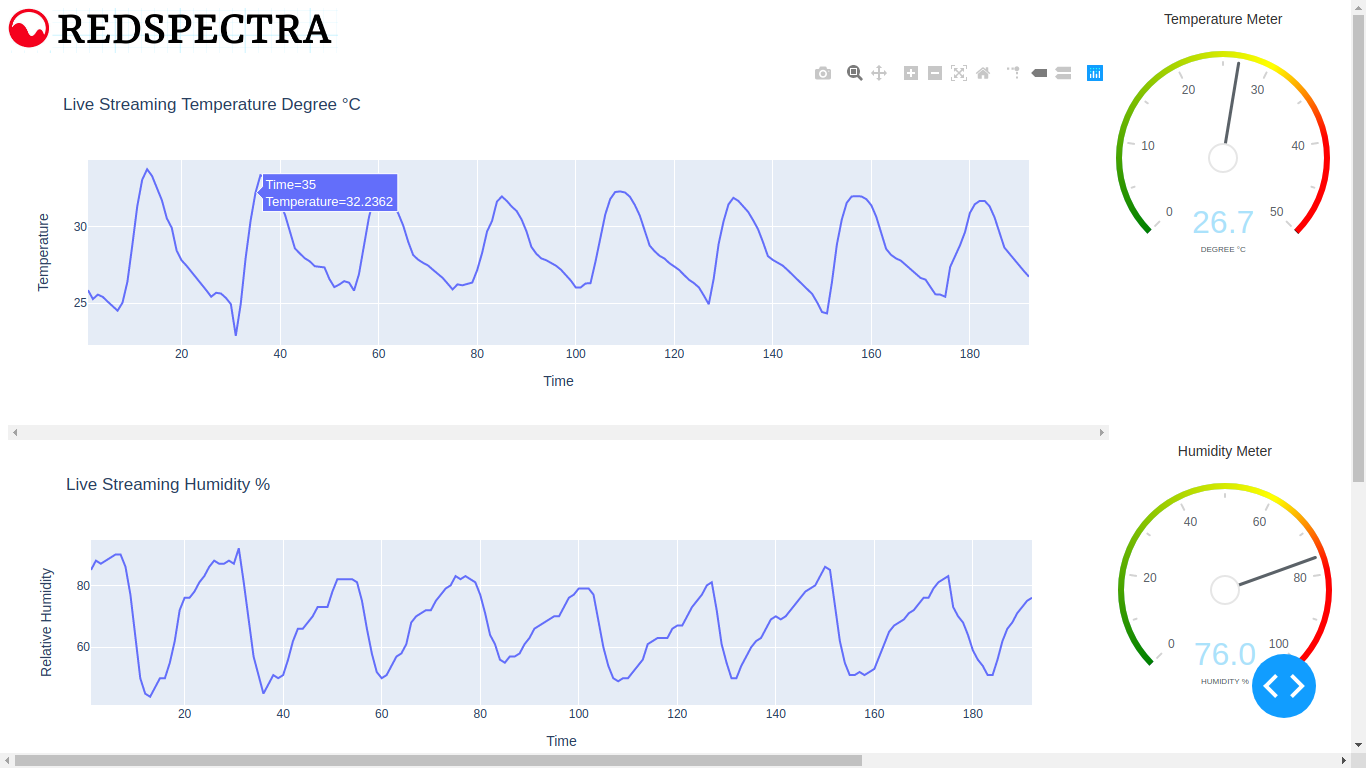
)

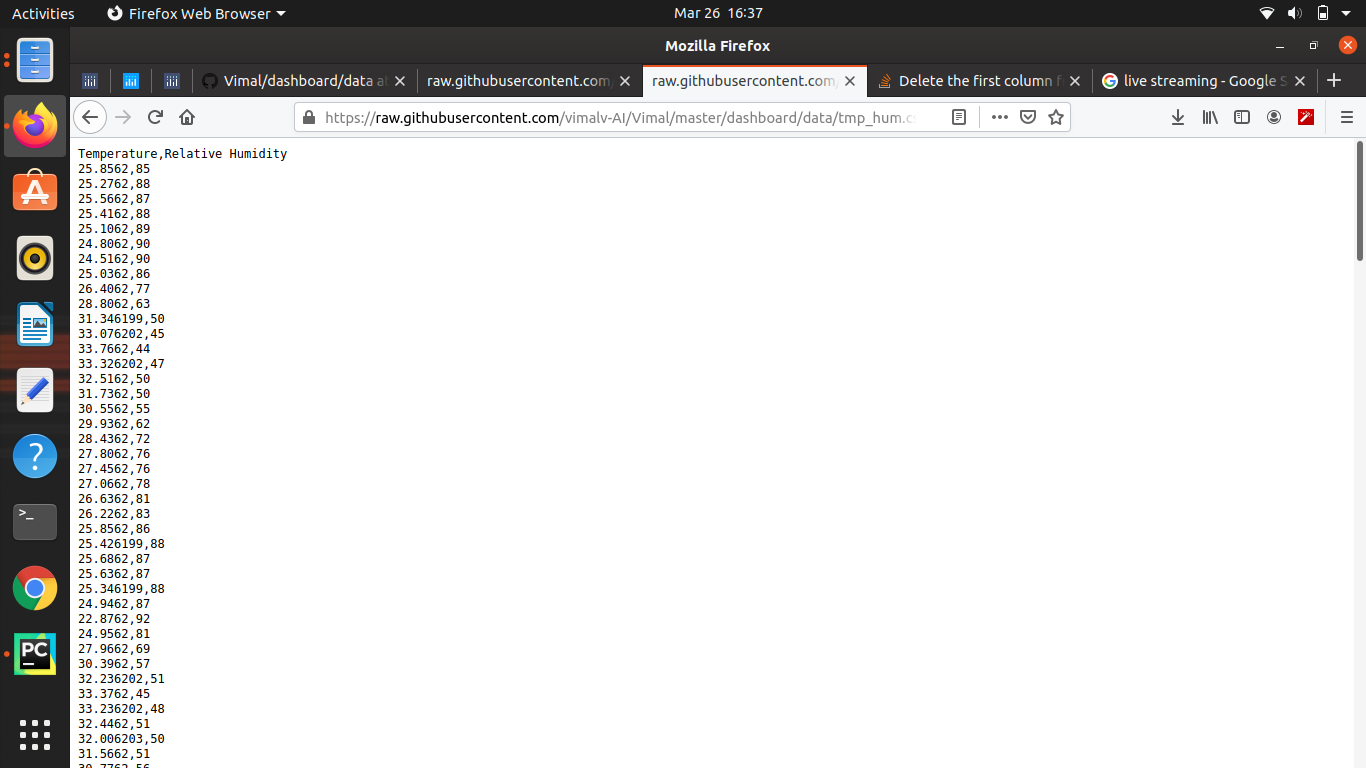
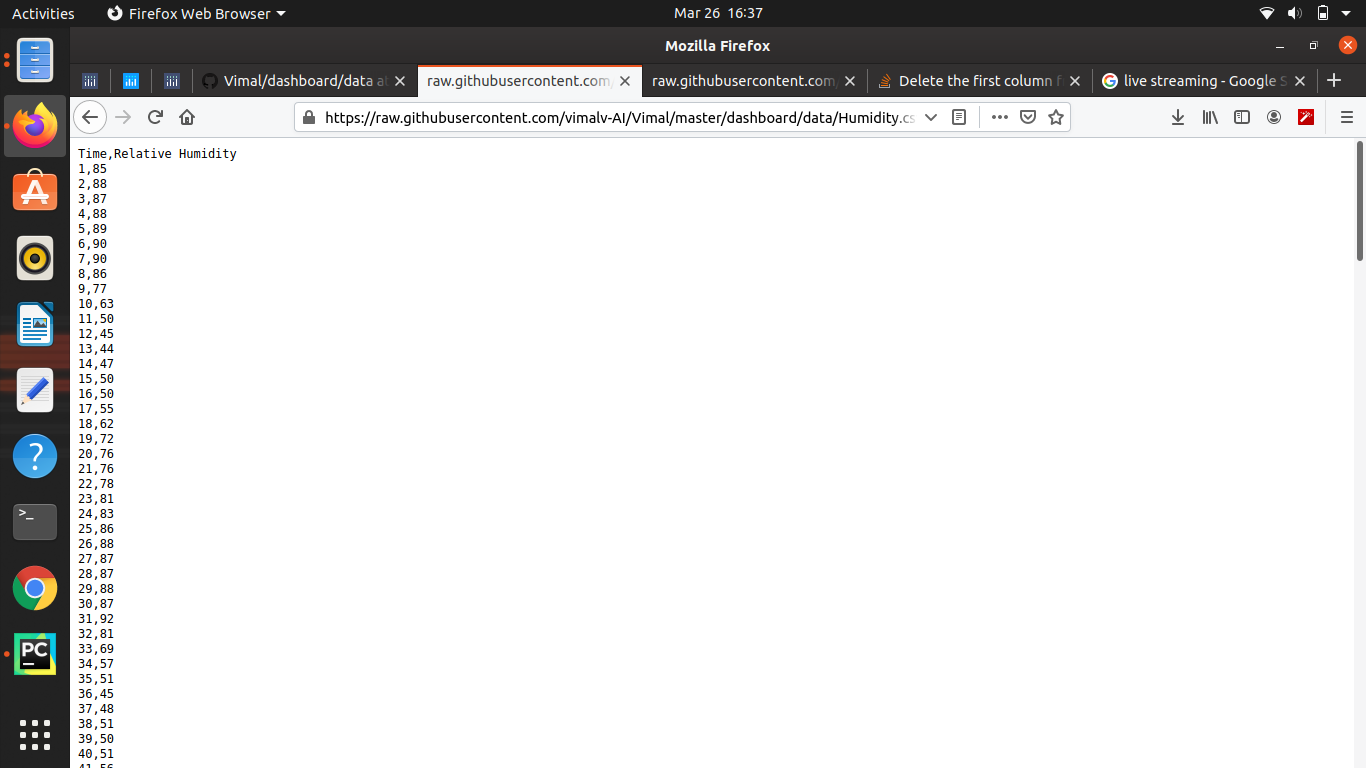
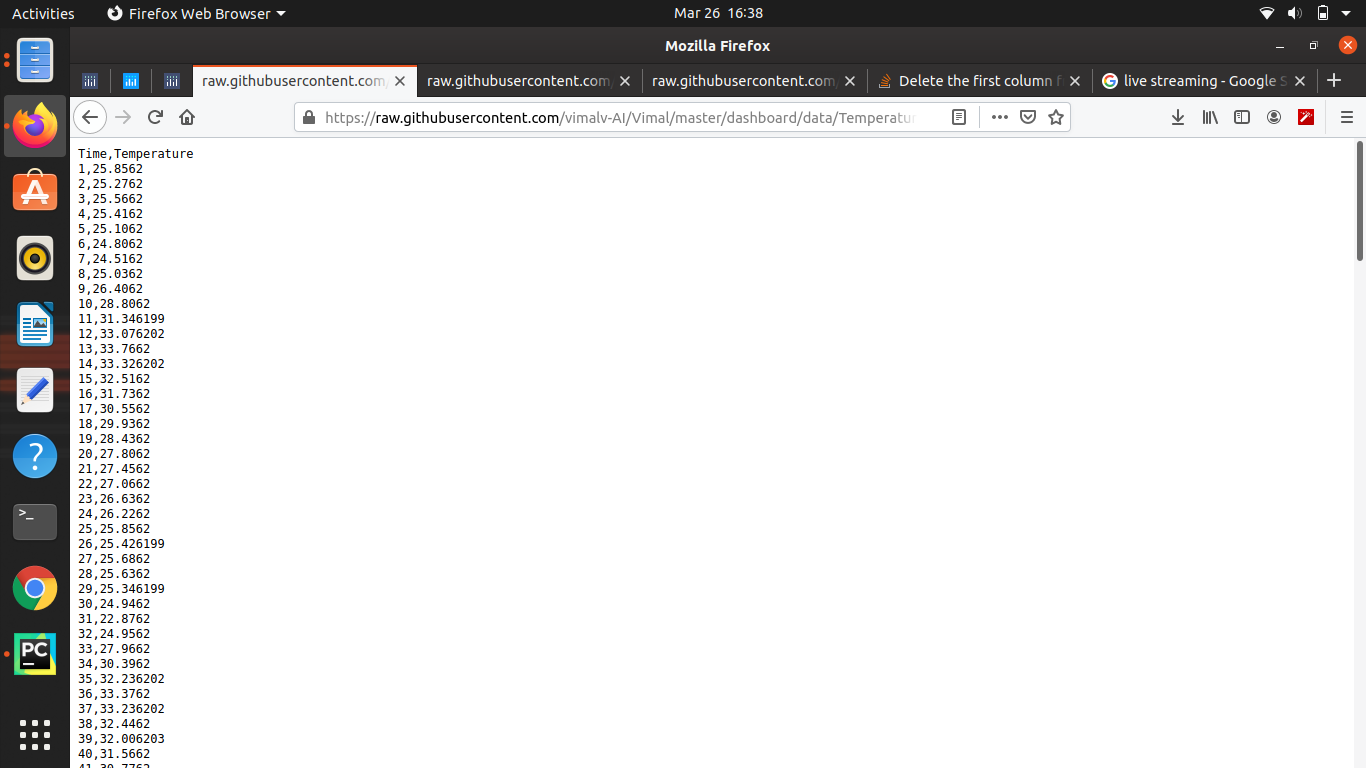
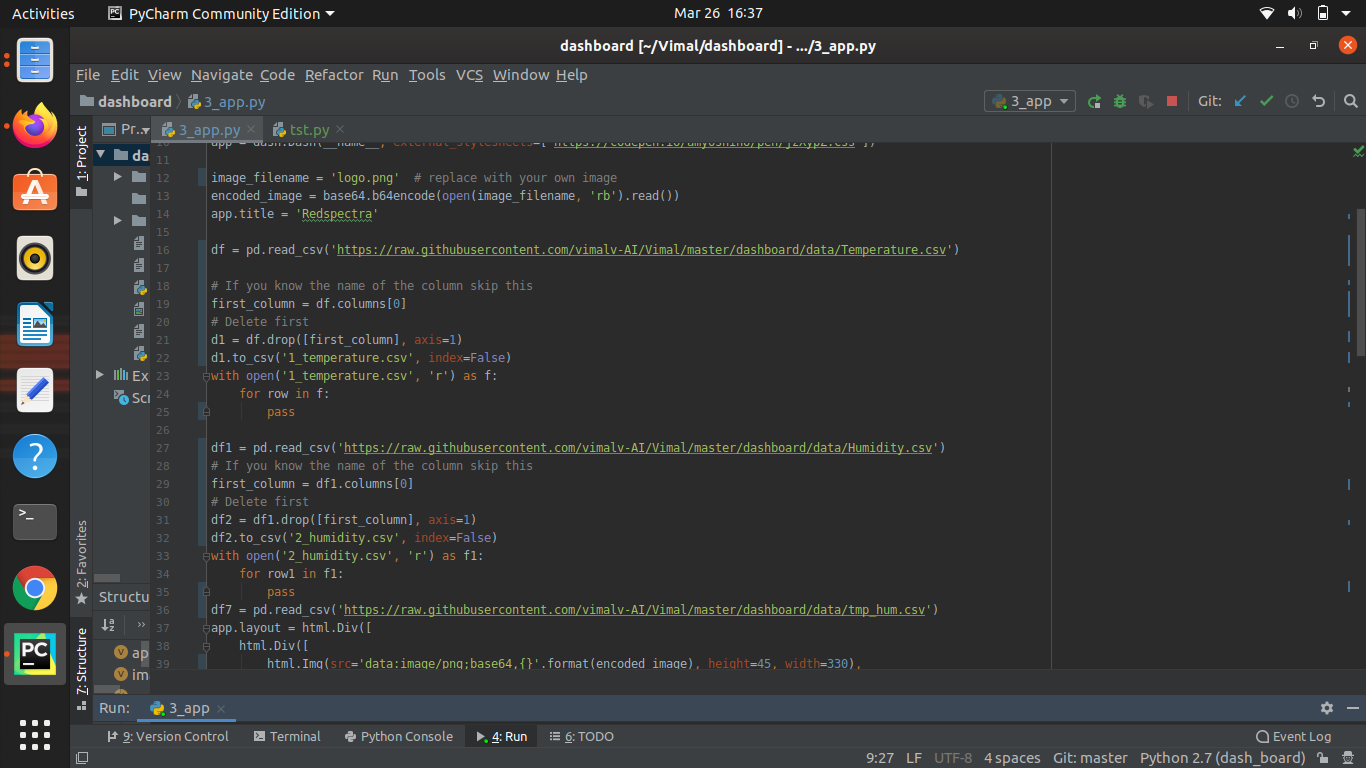
if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True)

**Others :**





**DATA SET FROM SERVER :**