Team :- Decimal Binary

Problem Statement :-

11. Weather Data Analysis and Prediction

Description: Use weather datasets to predict temperature, rainfall, or other conditions for specific regions. This can help in planning for agricultural or travel needs.

Reference from where the data scrapped :- https://www.visualcrossing.com/

Approach :-

1) **Data source :- visualcrossing .com**
2) **Methods of collection:- Web Scrapping using Jupiter Notebook and python Libraries.**
3) **Tools :- Python Libraries .**

Steps :

Step 1: Identify Target Cities

Create a list of the top 17 cities in Maharashtra. This list can include:

1. **Mumbai**
2. **Pune**
3. **Nagpur**
4. **Nashik**
5. **Aurangabad**
6. **Solapur**
7. **Kolhapur**

8. Thane

9. Amravati

10. Sangli

11. Akola

12. Jalgaon

13. Latur

14. Ahmednagar

15. Nanded

16. Chandrapur

17. Dhule

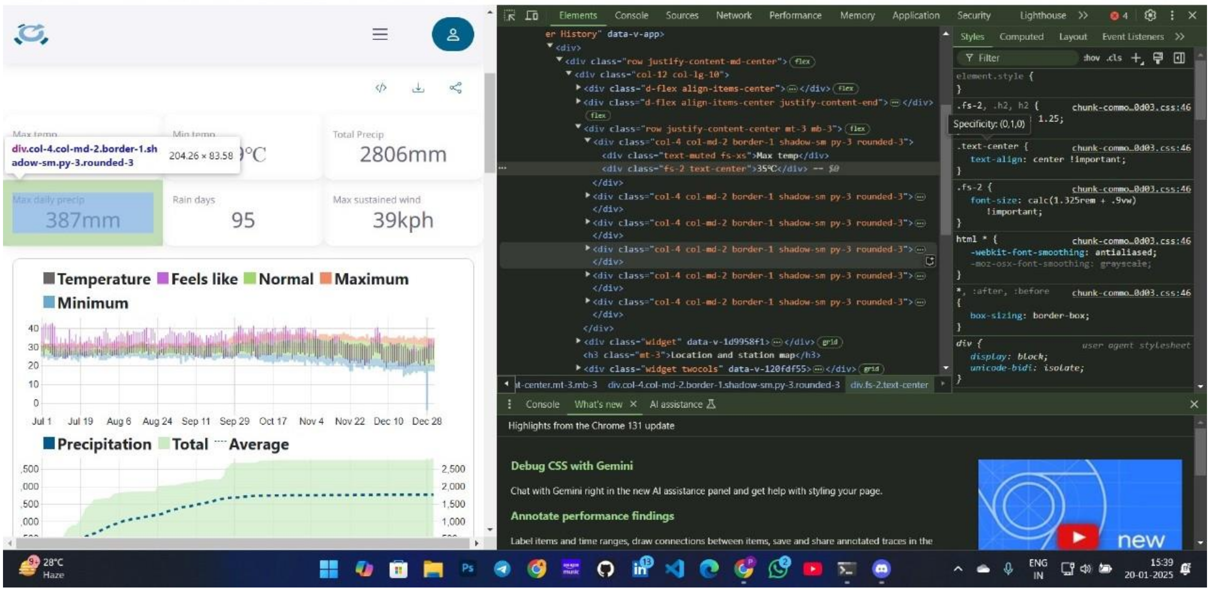
Step 2: Understand Website Structure

1. Visit Visual Crossing.

2. Explore the sections providing weather data. This may include temperature, humidity, wind speed, precipitation, etc.

3. Familiarize yourself with the URL patterns used to fetch weather data for different locations and dates.

1. Scrap the data of multiple cites



div.col-4.col-md-2.border-1.shadow-sm.py-3.rounded-3 204.26 × 83.58

Max temp

Min temp

Total Precip 2806mm

Max daily precip 387mm

Rain days 95

Max sustained wind 39kph

```
er History" data-v-app>
▼<div>
  ▼<div class="row justify-content-md-center"> flex
    ▼<div class="col-12 col-lg-10">
      ▶<div class="d-flex align-items-center">…</div> flex
      ▶<div class="d-flex align-items-center justify-content-end">…</div>
       flex
      ▼<div class="row justify-content-center mt-3 mb-3"> flex
        ▼<div class="col-4 col-md-2 border-1 shadow-sm py-3 rounded-3">
           <div class="text-muted fs-xs">Max temp</div>
           <div class="fs-2 text-center">35°C</div> == $0
         </div>
        ▶<div class="col-4 col-md-2 border-1 shadow-sm py-3 rounded-3">…
         </div>
        ▶<div class="col-4 col-md-2 border-1 shadow-sm py-3 rounded-3">…
         </div>
        ▶<div class="col-4 col-md-2 border-1 shadow-sm py-3 rounded-3">…
         </div>
        ▶<div class="col-4 col-md-2 border-1 shadow-sm py-3 rounded-3">…
         </div>
        ▶<div class="col-4 col-md-2 border-1 shadow-sm py-3 rounded-3">…
         </div>
       </div>
      ▶<div class="widget" data-v-1d9958f1>…</div> grid
       <h3 class="mt-3">Location and station map</h3>
      ▶<div class="widget twocols" data-v-120fdf55>…</div> grid
```

```
element.style {
}
.fs-2, .h2, h2 {          chunk-commo…0d03.css:46
                 1.25;
}
.text-center {            chunk-commo…0d03.css:46
  text-align: center !important;
}
.fs-2 {                   chunk-commo…0d03.css:46
  font-size: calc(1.325rem + .9vw)
      !important;
}
html * {                  chunk-commo…0d03.css:46
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
*, :after, :before        chunk-commo…0d03.css:46
{
  box-sizing: border-box;
}
div {                     user agent stylesheet
  display: block;
  unicode-bidi: isolate;
}
```

Specificity: (0,1,0)

t-center.mt-3.mb-3 div.col-4.col-md-2.border-1.shadow-sm.py-3.rounded-3 div.fs-2.text-center

Highlights from the Chrome 131 update

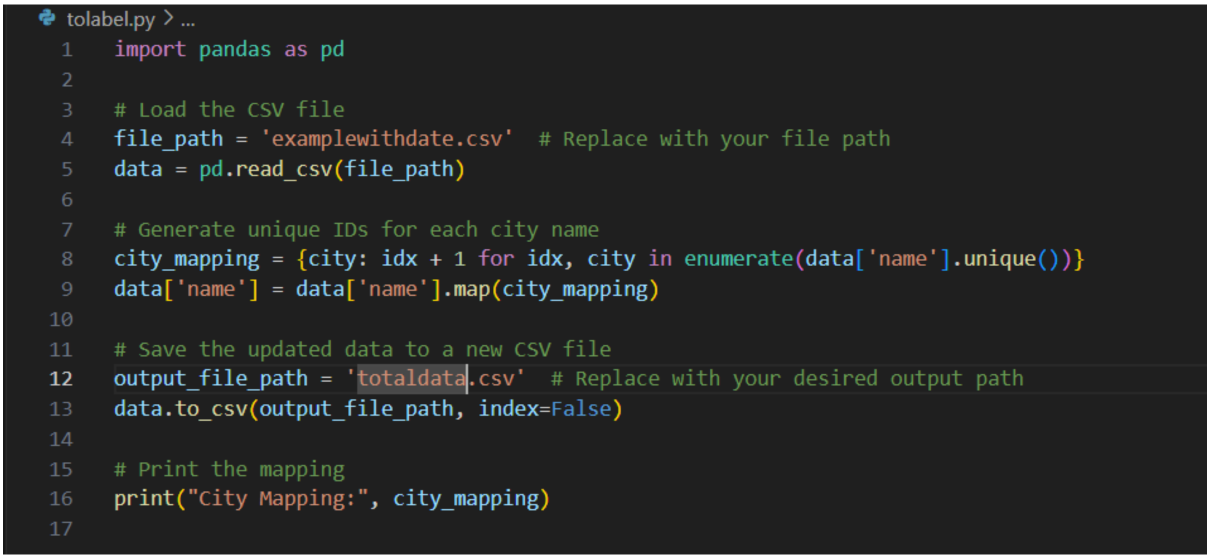
Debug CSS with Gemini

Chat with Gemini right in the new AI assistance panel and get help with styling your page.

Annotate performance findings

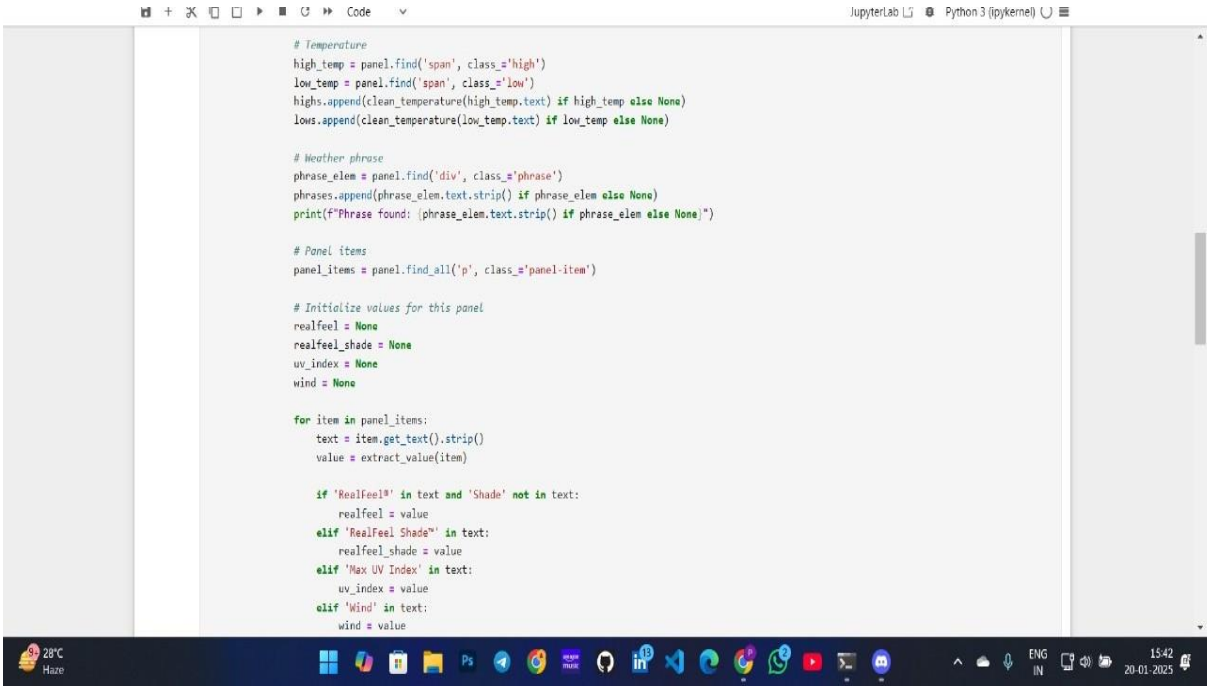
Label items and time ranges, draw connections between items, save and share annotated traces in the

2. Code for labelling ; the cites to numeric value



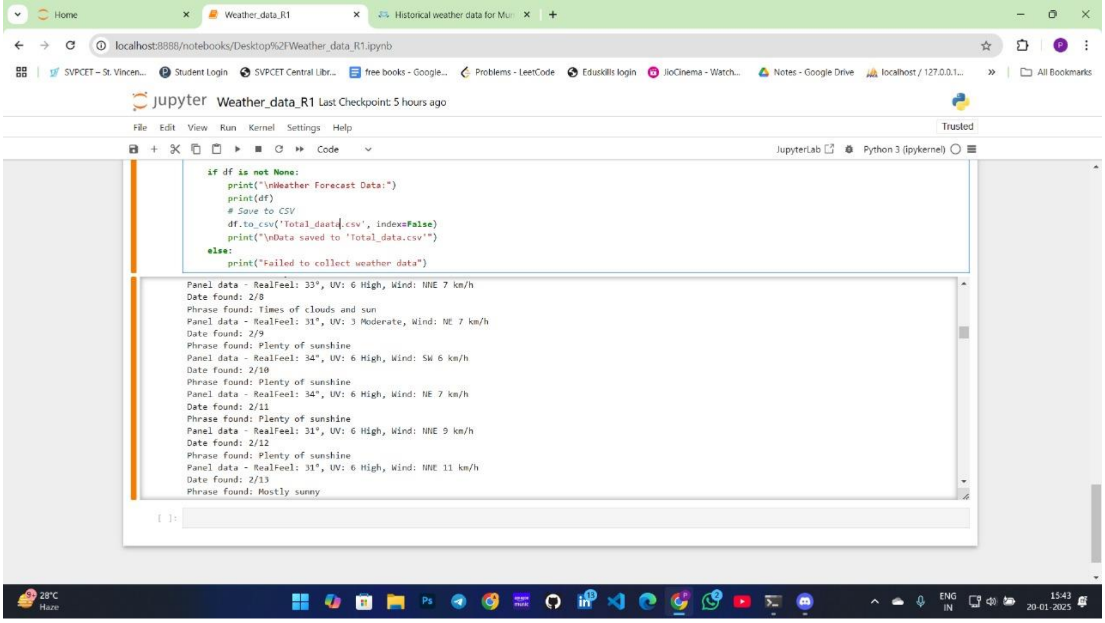
```python
import pandas as pd

# Load the CSV file
file_path = 'examplewithdate.csv'  # Replace with your file path
data = pd.read_csv(file_path)

# Generate unique IDs for each city name
city_mapping = {city: idx + 1 for idx, city in enumerate(data['name'].unique())}
data['name'] = data['name'].map(city_mapping)

# Save the updated data to a new CSV file
output_file_path = 'totaldata.csv'  # Replace with your desired output path
data.to_csv(output_file_path, index=False)

# Print the mapping
print("City Mapping:", city_mapping)

```

3. This is the Jupiter file for scraping data of more than 500 entries



```python
# Temperature
high_temp = panel.find('span', class_='high')
low_temp = panel.find('span', class_='low')
highs.append(clean_temperature(high_temp.text) if high_temp else None)
lows.append(clean_temperature(low_temp.text) if low_temp else None)

# Weather phrase
phrase_elem = panel.find('div', class_='phrase')
phrases.append(phrase_elem.text.strip() if phrase_elem else None)
print(f"Phrase found: {phrase_elem.text.strip() if phrase_elem else None}")

# Panel items
panel_items = panel.find_all('p', class_='panel-item')

# Initialize values for this panel
realfeel = None
realfeel_shade = None
uv_index = None
wind = None

for item in panel_items:
    text = item.get_text().strip()
    value = extract_value(item)

    if 'RealFeel®' in text and 'Shade' not in text:
        realfeel = value
    elif 'RealFeel Shade™' in text:
        realfeel_shade = value
    elif 'Max UV Index' in text:
        uv_index = value
    elif 'Wind' in text:
        wind = value
```

4. This is the scraped data we got using BeautifulSoup library

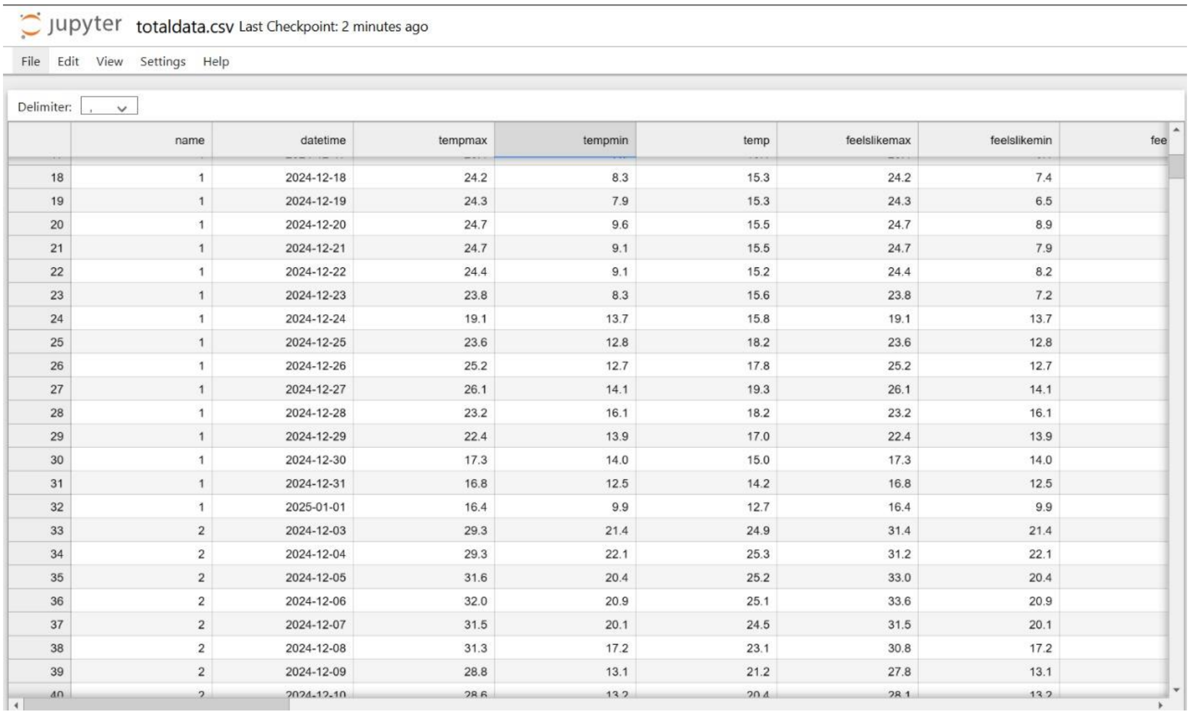


```
    if df is not None:
        print("\nWeather Forecast Data:")
        print(df)
        # Save to CSV
        df.to_csv('Total_daata.csv', index=False)
        print("\nData saved to 'Total_data.csv'")
    else:
        print("Failed to collect weather data")
```

```
Panel data - RealFeel: 33°, UV: 6 High, Wind: NNE 7 km/h
Date found: 2/8
Phrase found: Times of clouds and sun
Panel data - RealFeel: 31°, UV: 3 Moderate, Wind: NE 7 km/h
Date found: 2/9
Phrase found: Plenty of sunshine
Panel data - RealFeel: 34°, UV: 6 High, Wind: SW 6 km/h
Date found: 2/10
Phrase found: Plenty of sunshine
Panel data - RealFeel: 34°, UV: 6 High, Wind: NE 7 km/h
Date found: 2/11
Phrase found: Plenty of sunshine
Panel data - RealFeel: 31°, UV: 6 High, Wind: NNE 9 km/h
Date found: 2/12
Phrase found: Plenty of sunshine
Panel data - RealFeel: 31°, UV: 6 High, Wind: NNE 11 km/h
Date found: 2/13
Phrase found: Mostly sunny
```

```
[ ]:
```

5. This is the final dataset scrap from website



	name	datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	fee
18	1	2024-12-18	24.2	8.3	15.3	24.2	7.4	
19	1	2024-12-19	24.3	7.9	15.3	24.3	6.5	
20	1	2024-12-20	24.7	9.6	15.5	24.7	8.9	
21	1	2024-12-21	24.7	9.1	15.5	24.7	7.9	
22	1	2024-12-22	24.4	9.1	15.2	24.4	8.2	
23	1	2024-12-23	23.8	8.3	15.6	23.8	7.2	
24	1	2024-12-24	19.1	13.7	15.8	19.1	13.7	
25	1	2024-12-25	23.6	12.8	18.2	23.6	12.8	
26	1	2024-12-26	25.2	12.7	17.8	25.2	12.7	
27	1	2024-12-27	26.1	14.1	19.3	26.1	14.1	
28	1	2024-12-28	23.2	16.1	18.2	23.2	16.1	
29	1	2024-12-29	22.4	13.9	17.0	22.4	13.9	
30	1	2024-12-30	17.3	14.0	15.0	17.3	14.0	
31	1	2024-12-31	16.8	12.5	14.2	16.8	12.5	
32	1	2025-01-01	16.4	9.9	12.7	16.4	9.9	
33	2	2024-12-03	29.3	21.4	24.9	31.4	21.4	
34	2	2024-12-04	29.3	22.1	25.3	31.2	22.1	
35	2	2024-12-05	31.6	20.4	25.2	33.0	20.4	
36	2	2024-12-06	32.0	20.9	25.1	33.6	20.9	
37	2	2024-12-07	31.5	20.1	24.5	31.5	20.1	
38	2	2024-12-08	31.3	17.2	23.1	30.8	17.2	
39	2	2024-12-09	28.8	13.1	21.2	27.8	13.1	
40	2	2024-12-10	28.6	13.2	20.4	28.1	13.2	