Weather Forecast Application

Andrii Kozlov

ČVUT–FIT

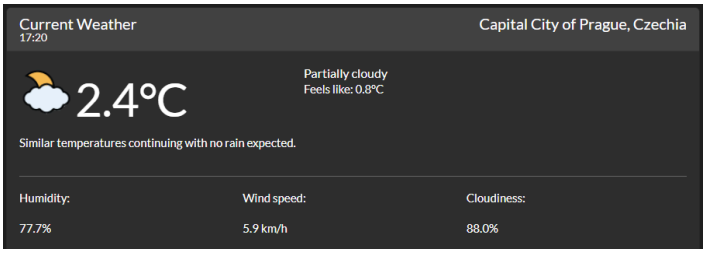
kozloan3@fit.cvut.cz

23.12.2024

Introduction

This project addresses the task of developing a weather forecast application as part of the BI–PYT semester assignment. The application allows users to view real-time weather data, including current conditions and a 7-day forecast, for locations worldwide.

The primary objective was to integrate real-world weather APIs and implement a user-friendly interface using Dash and Dash Bootstrap Components (DBC) to visualize data effectively.



Picture 1. Card showing real time weather data

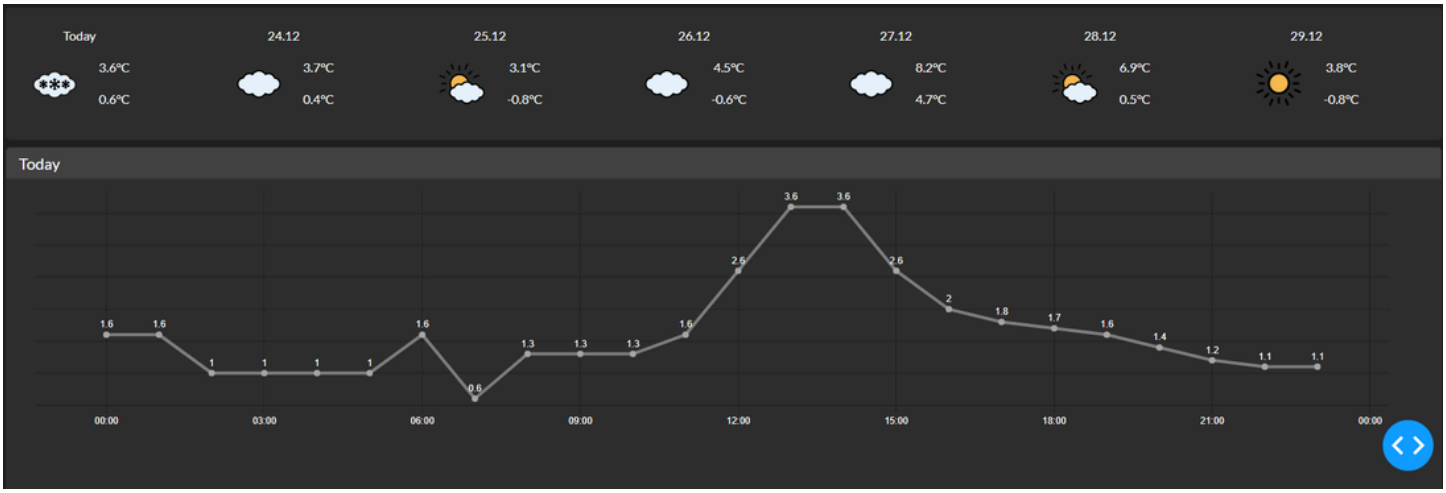
Methods and Tools

1. **Data Retrieval**:
   - Weather data is fetched using the Visual Crossing Weather API.
   - Data is processed using Pandas.
   - Geolocation data is obtained via GeoPy's Nominatim service.
2. **Technologies**:
   - *Dash*: Used for building the web-based interactive UI.
   - *Dash Bootstrap Components*: For styling and layout design.
3. **Functionalities**:
   - Real-time weather updates for user-selected locations via search bar or map interaction.
   - Visualization of a 7-day weather forecast in graphical format.
   - Collapsible cards for each forecasted day to display additional details upon expansion.

Results

- The application successfully integrates real-time weather data, providing accurate current conditions and forecasts.
- A graphical representation of temperature trends over seven days is displayed with grid support for better data interpretation.
- Challenges encountered:
  - Managing asynchronous API calls to ensure data consistency.
  - Resolving issues related to accessing assets, such as weather icons.

Below is an example output of the application showing a 7-day forecast with temperature trends.



Picture 2. 7-day forecast with temperature trends.

Future Development

- Add multilingual support to enhance accessibility.
- Include more advanced forecasting features, such as precipitation probability graphs.
- Add the clothing advice feature with location-specific cultural considerations (integrating such data with AI-generated advice would have required developing a custom pipeline, which exceeded the scope of the semester project).

References

- Visual Crossing Weather API Documentation: https://www.visualcrossing.com/documentation
- Dash Framework: https://dash.plotly.com/
- GeoPy Documentation: https://geopy.readthedocs.io/