Weather Dashboard Project: WeatherVista (天气 - Tiānqì jǐngguān)

ISYS2001 Modules adjusted for online delivery

Michael Borck

Goals of the WeatherVista Project

The WeatherVista project offers students a hands-on learning experience in Python programming, data management, and visualisation. Through interactive sessions, students will learn to fetch, process, and visualise weather data using the OpenWeatherMap API. They will also work with pandas for data manipulation, Matplotlib for plotting, Jupyter Widgets for interactive dashboards, and basic testing to ensure code reliability. This project strengthens procedural programming skills and demonstrates practical uses of Python in real-world scenarios.

Partnership Between Curtin University and South China Normal University

The WeatherVista project is a joint initiative by Curtin University and South China Normal University, offering a rich learning environment for students in China. Instructors from Curtin University conduct live sessions tailored to the students' needs, addressing language barriers and ensuring clear communication of concepts. This collaboration enhances education, promotes international cooperation, cultural exchange, and a shared commitment to advancing technical education.

Simplified Approach

The simplified WeatherVista project introduces essential programming and data management concepts clearly and accessibly. By focusing on simple implementations and practical examples, we make learning easier. Instead of advanced testing frameworks or comprehensive documentation, we use basic tests and docstrings, allowing students to access documentation directly in their code with Python's `help()` function. Each session builds on the previous one, reinforcing key concepts through hands-on activities, live coding, and discussions. This approach ensures students effectively learn and apply their skills, laying a solid foundation for

further development. Brainstorming sessions for project extensions encourage creativity and practical application of learned skills.

Sessions Overview

1. **Introduction to the Project and Setting Up the Environment**
2. **Fetching Data from OpenWeatherMap API**
3. **Data Management with CSV**
4. **Basic Data Visualisation with Matplotlib**
5. **Advanced Data Visualisation and Dashboard Layout**
6. **Web Scraping and Using APIs (Optional)**
7. **Building the Dashboard with Jupyter Widgets**
8. **Basic Testing**
9. **Finalising the Project and Basic Distribution**
10. **Final Project Review and Brainstorming Extensions**

Session Breakdown

Session 1: Introduction to the Project and Setting Up the Environment

- **Overview:**
  - Introduce the course and project goals.
  - Set up the development environment.

- **Activities:**
  - Overview of the weather dashboard project.
  - Install Python, Jupyter, and necessary libraries.
  - Basic Git commands: clone, commit, push.

- **Outputs**
  - Working environment

- **Homework:**
  - Research the OpenWeatherMap API and sign up for an API key.

Timings

1. **Introduction to the Course (10 minutes)**
   - Welcome and course overview
   - Introduction to the WeatherVista project

2. **Demo of Completed WeatherVista Dashboard (20 minutes)**

   - Show the completed dashboard
   - Highlight key features and functionalities

3. **Setting Up Development Environment (30 minutes)**

   - Walkthrough installation of Python, Jupyter, and necessary libraries
   - Set up GitHub/GitLab repository

4. **Introduction to Git (15 minutes)**

   - Basic Git commands: clone, commit, push
   - Demonstrate cloning the project repository

5. **Creating Project Directory Structure (20 minutes)**

   - Create directories for data, scripts, notebooks, and tests
   - Brief explanation of the purpose of each directory

6. **Breakout Room Activity (15 minutes)**

   - Students work in small groups to set up their project structure
   - Instructors available for assistance

7. **Q&A and Wrap-Up (15 minutes)**

   - Address any questions about setup and project structure
   - Homework: Research the OpenWeatherMap API and sign up for an API key

---

Session 2: Fetching Data from OpenWeatherMap API

- **Overview:**

  - Learn to fetch weather data using the OpenWeatherMap API.

- **Activities:**

  - Fetch current weather data.
  - Parse JSON data and handle errors.
  - Save fetched data to a CSV file.

- **Outputs**

  - 01_fetch_data.ipynb
  - fetch_data.py

- **Homework:**

– Experiment with different endpoints and data parameters.

Timings

1. **Recap and Introduction (10 minutes)**

   - Brief recap of Session 1
   - Introduction to APIs and web requests

2. **Live Coding: Fetching Data (20 minutes)**

   - Demonstrate fetching current weather data using the OpenWeatherMap API

3. **Hands-On Activity: Parsing JSON Data (20 minutes)**

   - Students parse the JSON response and extract relevant data

4. **Error Handling (15 minutes)**

   - Discuss common errors and how to handle them

5. **Breakout Room Activity (20 minutes)**

   - Students work in groups to fetch and parse data for different cities

6. **Demo and Discussion (15 minutes)**

   - Groups present their findings and discuss challenges

7. **Q&A and Wrap-Up (10 minutes)**

   - Address any questions and provide additional tips
   - Homework: Experiment with different endpoints and data parameters

---

Session 3: Data Management with CSV

- **Overview:**

  – Manage and store data using CSV.

- **Activities:**

  – Read and write weather data to/from CSV files using pandas.

- **Output**

  – 02_data_management.ipynb
  – process_data.py

- **Homework:**

    - Practice reading and writing CSV files with different data sets.

Timings 1. **Recap and Introduction (10 minutes)** - Brief recap of Session 2 - Introduction to data management with CSV files 2. **Live Coding: Reading and Writing CSV Files (20 minutes)** - Demonstrate how to read and write weather data to/from CSV files using pandas 3. **Hands-On Activity: Storing Data (20 minutes)** - Students store fetched weather data in a CSV file 4. **Breakout Room Activity (20 minutes)** - Students practice reading and writing CSV files with different data sets 5. **Demo and Discussion (15 minutes)** - Groups present their data management solutions 6. **Q&A and Wrap-Up (10 minutes)** - Address any questions and provide additional tips - Homework: Practice reading and writing CSV files

---

Session 4: Basic Data Visualisation with Matplotlib

- **Overview:**

    - Visualise weather data using Matplotlib.

- **Activities:**

    - Plot current weather data (temperature, humidity, etc.).
    - Customise plots with titles, labels, and legends.

- **Output**

    - 03_visualisation.ipynb

- **Homework:**

    - Create additional visualisations with historical weather data.

Timings

1. **Recap and Introduction (10 minutes)**

    - Brief recap of Session 3
    - Introduction to data visualisation with Matplotlib

2. **Live Coding: Basic Plots (20 minutes)**

    - Demonstrate creating basic plots for weather data

3. **Hands-On Activity: Customising Plots (20 minutes)**

    - Students customise plots with titles, labels, and legends

4. **Breakout Room Activity (20 minutes)**

   - Students create additional visualisations with historical weather data

5. **Demo and Discussion (15 minutes)**

   - Groups present their visualisations

6. **Q&A and Wrap-Up (10 minutes)**

   - Address any questions and provide additional tips
   - Homework: Create additional visualisations

Session 5: Advanced Data Visualisation and Dashboard Layout

- **Overview:**

  - Create advanced visualisations and layout the dashboard.

- **Activities:**

  - Create daily forecast cards with high/low temperatures.
  - Design a layout for the dashboard using subplots and grids.
  - Use icons to represent weather conditions.

- **Output**

  - 04_advanced_vix.ipynb
  - visualise_data.py

- **Homework:**

  - Refine and enhance visualisations for better presentation.

Timings

1. **Recap and Introduction (10 minutes)**

   - Brief recap of Session 4
   - Introduction to advanced data visualisation and dashboard layout

2. **Live Coding: Subplots and Grids (20 minutes)**

   - Demonstrate creating subplots and grids

3. **Hands-On Activity: Daily Forecast Cards (20 minutes)**

   - Students create daily forecast cards with high/low temperatures

4. **Breakout Room Activity (20 minutes)**

   - Students design a layout for the dashboard using subplots and grids

5. **Demo and Discussion (15 minutes)**

   - Groups present their dashboard layouts

6. **Q&A and Wrap-Up (10 minutes)**

   - Address any questions and provide additional tips
   - Homework: Refine and enhance visualisations

Session 6: Web Scraping and Using APIs (Optional)

- **Overview:**

  - Learn web scraping basics and explore additional APIs.

- **Activities:**

  - Introduction to web scraping with BeautifulSoup.
  - Scrape weather-related data from a website.
  - Discuss ethics and legal aspects of web scraping.

- **Output**

  - 05_webscraping.ipynb

- **Homework:**

  - Find and document additional data sources or APIs.

Timings

1. **Recap and Introduction (10 minutes)**

   - Brief recap of Session 5
   - Introduction to web scraping and additional APIs

2. **Live Coding: Web Scraping Basics (20 minutes)**

   - Demonstrate web scraping with BeautifulSoup

3. **Hands-On Activity: Scraping Weather Data (20 minutes)**

   - Students scrape weather-related data from a website

4. **Breakout Room Activity (20 minutes)**

- Students document additional data sources or APIs

5. **Demo and Discussion (15 minutes)**

   - Groups present their scraping and API findings

6. **Q&A and Wrap-Up (10 minutes)**

   - Address any questions and provide additional tips
   - Homework: Explore additional data sources

Session 7: Building the Dashboard with Jupyter Widgets

- **Overview:**

  - Create an interactive dashboard using Jupyter Widgets.

- **Activities:**

  - Use `ipywidgets` to create interactive elements such as dropdowns and sliders.
  - Link widgets to data visualisation code for dynamic updates.

- **Homework:**

  - Add interactivity to the dashboard (e.g., user input for city or sip code).

Timings

1. **Recap and Introduction (10 minutes)**

   - Brief recap of Session 6
   - Introduction to creating interactive dashboards with Jupyter Widgets

2. **Live Coding: Jupyter Widgets (20 minutes)**

   - Demonstrate using `ipywidgets` to create interactive elements

3. **Hands-On Activity: Interactive Elements (20 minutes)**

   - Students create interactive elements such as dropdowns and sliders

4. **Breakout Room Activity (20 minutes)**

   - Students link widgets to data visualisation code for dynamic updates

5. **Demo and Discussion (15 minutes)**

   - Groups present their interactive dashboards

6. **Q&A and Wrap-Up (10 minutes)**

   - Address any questions and provide additional tips
   - Homework: Add interactivity to the dashboard

Session 8: Basic Testing

- **Overview:**

  – Learn the importance of testing and how to write basic tests.

- **Activities:**

  – Write unit tests for data fetching and processing functions using `pytest`.
  – Run tests and interpret the results.

- **Output**

  – 07_testing.ipynb

- **Homework:**

  – Write additional tests to improve code coverage.

Timings

1. **Recap and Introduction (10 minutes)**

   - Brief recap of Session 7
   - Introduction to testing and its importance

2. **Live Coding: Writing Tests (20 minutes)**

   - Demonstrate writing basic tests using `pytest`

3. **Hands-On Activity: Unit Tests (20 minutes)**

   - Students write unit tests for data fetching and processing functions

4. **Breakout Room Activity (20 minutes)**

   - Students run tests and interpret results

5. **Demo and Discussion (15 minutes)**

   - Groups present their testing results

6. **Q&A and Wrap-Up (10 minutes)**

- Address any questions and provide additional tips
- Homework: Write additional tests

Session 9: Finalising the Project and Basic Distribution

- **Overview:**
  - Finalise the project and discuss basic distribution methods.

- **Activities:**
  - Review and refine the code.
  - Document the project using README and comments.
  - Discuss distribution methods: GitHub repository, exporting notebooks as HTML/PDF.

- **Output**
  - 08_final_steps.ipynb

- **Homework:**
  - Prepare the project for presentation in the next session.

Timings

1. **Recap and Introduction (10 minutes)**
   - Brief recap of Session 8
   - Introduction to finalising and distributing the project

2. **Live Coding: Final Touches (20 minutes)**
   - Review and refine the code

3. **Hands-On Activity: Documentation (20 minutes)**
   - Students document the project using README and comments

4. **Breakout Room Activity (20 minutes)**
   - Discuss basic distribution methods: GitHub repository, exporting notebooks as HTML/PDF

5. **Demo and Discussion (15 minutes)**
   - Groups present their final projects and documentation

6. **Q&A and Wrap-Up (10 minutes)**

- Address any questions and provide additional tips
- Homework: Prepare the project for presentation

Session 10: Final Project Review and Brainstorming Extensions

- **Overview:**
  - Review the project and brainstorm extension ideas.

- **Activities:**
  - Group review and feedback on projects.
  - Brainstorm and share ideas for extending the project.
  - Discuss potential new projects and applications of the skills learned.

- **Output**
  - Completed project

- **Homework:**
  - Finalise the project and submit it to the GitHub repository.

Timings

1. **Introduction and Instructions (10 minutes)**
   - Overview of the session's objectives

2. **Group Review (30 minutes)**
   - Divide students into smaller groups to review each other's projects

3. **Instructor-Led Review (20 minutes)**
   - Randomly select a few projects to review in the main session

4. **Brainstorming Extensions (30 minutes)**
   - Groups brainstorm ways to extend the WeatherVista project

5. **Sharing Ideas (20 minutes)**
   - Groups share their best ideas with the class

6. **Discussion of New Project Ideas (20 minutes)**
   - Discuss potential new projects and applications of the skills learned

7. **Q&A and Course Wrap-Up (10 minutes)**

- Address any remaining questions
- Summarise key takeaways from the course
- Collect feedback for future improvements

Project Structure

Here's a simplified version of the project structure:

```
weather-dashboard/

    data/
        raw/
        processed/

    notebooks/
        01_fetch_data.ipynb
        02_data_management.ipynb
        03_visualisation.ipynb
        04_advanced_vis.ipynb
        05_web_scraping.ipynb
        06_dashboard_gui.ipynb
        07_testing.ipynb
        08_final_steps.ipynb

    scripts/
        fetch_data.py
        process_data.py
        visualise_data.py
        create_dashboard.py
        utils.py

    .gitignore
    README.md
    requirements.txt
    main.py
```