**Fellow's Step-by-Step Guide to the Capstone Project**

### **Overview**

This document is your step-by-step guide to completing the Capstone Project: a fully functional weather dashboard built in Python. You will work individually on your app, and later contribute to a team-based collaborative feature. Follow the weekly instructions to stay on track and submit a project you can be proud of.

### **Week 11: Project Launch & Planning**

**Goal:** Understand the project scope, select features, and plan your development path.

* Attend capstone kickoff
* Review the full feature menu
* Choose 3 features from the list (⭐ to ⭐⭐⭐)
* Choose 1 enhancement (personality, visual flair, etc.)
* Sketch your app structure (Tkinter windows, file layout, etc.)
* Write a personal development timeline

#### **🧩 Feature Menu – Choose 3**

##### **⭐ Basic Data Features**

* **Weather History Tracker** – Save daily weather to a CSV file and display the last 7 days
* **Simple Statistics** – Track min/max temperatures and count weather types
* **City Comparison** – Compare current weather in two cities side-by-side

##### **⭐⭐ Visual Features**

* **Temperature Graph** – Show a line graph of temperature over time using matplotlib
* **Weather Icons** – Use canvas graphics to visually represent weather (e.g., sun, clouds)
* **Theme Switcher** – Toggle between day/night modes or custom color themes

##### **⭐⭐ Interactive Features**

* **Weather Journal** – Add daily mood notes and save them to a text file
* **Favorite Cities** – Let users save and quickly access preferred cities
* **Weather Alerts** – Notify the user when temperature crosses a certain threshold

##### **⭐⭐⭐ Smart Features**

* **Tomorrow’s Guess** – Use simple logic to predict tomorrow’s temperature
* **Trend Detection** – Show arrows or indicators for rising/falling weather trends
* **Activity Suggester** – Suggest activities based on current weather (e.g., go biking, stay in)

#### **✨ Enhancement Options – Choose 1**

* **Mascot or Weather Personality** – Add a character that reacts to weather changes
* **Achievement System** – Give rewards for actions (e.g., using app 5 days in a row)
* **Custom Descriptions** – Write your own descriptions instead of using API defaults
* **Sound Effects** – Add sounds for different weather types or events
* **Weather Poetry** – Generate a fun phrase or haiku based on weather
* **Creative Visuals** – Use canvas animations, emojis, or styled graphics for flair

✅ **Deliverables:** Feature selections, architecture plan, test cases draft

### **Week 12: Build the Core Application**

**Goal:** Create a basic but working weather app with GUI and API connection.

* Set up Tkinter window and layout
* Connect to OpenWeatherMap API
* Fetch weather data for a city
* Display data (temperature, city name, description)
* Add error handling and file saving (text or CSV)

✅ **Milestone:** Working core app with basic GUI and API integration

### **Week 13: Build Feature #1**

**Goal:** Implement your first feature and integrate it with the core app.

* Choose a feature from the menu (Data, Visual, or Interactive)
* Use templates if needed
* Connect this feature to your main GUI

✅ **Deliverables:** Feature #1 functional and visible in app

**Team Formation:**

* Teams of 3–4 are announced
* Choose a team name and a shared feature idea (e.g., Weather Battle)
* Elect a Project Owner

### **Week 14: Build Feature #2 + Start Team Repo**

**Goal:** Expand your individual app and prepare the team structure.

* Implement Feature #2
* Polish UI and integrate it
* Project Owner creates GitHub repo for team
* Set up team folders: team\_feature/, team\_data/, docs/
* Export your own weather data as JSON
* Submit pull request to team repo

✅ **Deliverables:** Feature #2 working, data exported to team repo

### **Week 15: Build Feature #3 + Begin Team Feature**

**Goal:** Complete your last solo feature and begin contributing to the team feature.

* Implement Feature #3
* Refactor and organize your app
* Begin work on your assigned team task
* Collaborate via GitHub branches and pull requests

✅ **Milestone:** All 3 solo features complete, team feature under development

### **Week 16: Enhancement + Documentation + Integration**

**Goal:** Finalize your app, add personality, and ensure a smooth demo experience.

* Add your enhancement (mascot, theme switcher, sounds, etc.)
* Write README.md and user\_guide.md
* Test integration with team feature (add a button to your app)
* Practice opening the team feature from your GUI
* Submit your full app folder + confirm your role in team repo

✅ **Deliverables:** Final code, documentation, and working integration

### **Week 17: Showcase Week**

**Goal:** Present your work clearly and confidently.

* Demo your individual app (3–4 minutes)
* Demo your team feature with your group (5–7 minutes)
* Submit peer evaluations
* Celebrate and take screenshots for your portfolio!

✅ **Final Submission Checklist:**

* Individual App
* 3 Features + Enhancement
* Linked Team Feature
* README + Docs
* Peer Evaluation
* Optional: Record Demo Video

### **📘 Notes**

* If you get stuck, ask for help early
* You can always opt out of team work with instructor approval
* The app is yours — make it personal, useful, and fun

**You’ve got this! Build something real.**