

Weather app

This weather web app built with Flask and Open Weather API displays the weather forecast for any city. I designed a simplistic UI to make it easier for the user to find the relevant information. All HTML, CSS, and Python code was written from scratch and the CSS is optimised for mobile and desktop through the use of CSS grid, flexbox, and media queries.

1. Analysed user feedback and data from the app's initial release:

- Reviewed user reviews, ratings, and support inquiries to identify common pain points and feature requests.
- Examined usage data and analytics to understand user behaviours, engagement levels, and areas for improvement.

2. Prioritized the development roadmap:

- Categorized and ranked the identified improvement areas and new feature requests based on user impact, technical feasibility, and strategic alignment.
- Created a detailed roadmap outlining the key initiatives and timeline for the upcoming development cycles.

3. Implemented a location-based weather alert system:

- Integrated more granular weather data sources to enable personalized alerts for users' specific locations.
- Developed a robust notification system to deliver timely alerts about severe weather events, such as storms, floods, or extreme temperatures.
- Provided customization options for users to adjust the alert preferences and thresholds.

4. Improved the app's weather data visualization:

- Redesigned the weather forecast screens to present information in a more intuitive and engaging manner.
- Introduced interactive charts and graphs to display detailed weather data, including temperature, precipitation, wind, and humidity trends.
- Enabled users to switch between different data visualizations and customize the display preferences.

5. Conducted user testing and iterative refinements:

- Recruited a diverse group of beta testers to provide feedback on the new features and user experience enhancements.

- Analysed the test results and user feedback to identify areas for further improvement and refinement.
- Implemented the necessary changes and optimizations based on the user testing feedback.

Building a Python project from scratch without relying on a tutorial taught me a lot but I was also able to implement the app's key functionality (getting and displaying weather data) due to the work I did with APIs in Angela Yu's Python bootcamp. While deploying this web app, I learned about git and version control as well as storing API keys as environment variables with .env and the purpose of .gitignore. This project turned out to be frustrating and complicated at times but I learned and grew a lot as a developer by tackling each problem. For instance, I struggled to make this website responsive because I discovered that the Chrome browser tools are not entirely accurate for the mobile view. Hence, when the app was deployed, the website didn't look the way I expected or desired on mobile. So, I switched to a free desktop application called Responsively and it provided views for multiple devices which allowed me to improve my CSS. In addition, I had difficulty positioning my footer at the bottom of my page and had to refer to [this resource] (<https://stackoverflow.com/questions/51683107/making-a-footer-stay-at-the-bottom-of-the-page-both-in-mobile-view-and-desktop-v>) to adjust my CSS accordingly.

If I were to do this project again, there are a few changes I would make. I would make the website render beautifully on iPads as well by eliminating the space between elements. I also wasn't sure how to add more functionality to this app while maintaining a seamless UI/UX design this time and I would love to do so in the future. E.g. I could add an option for users to specify the country of the city they enter (if multiple countries share the same city name). I could also display the low and high temperatures for each day in the five-day forecast instead of only displaying the temperature at noon (which I understand is not an accurate estimate of the overall temperature). It would also be useful to enable location detection to allow the user to get more accurate weather data for their current location by using their precise coordinates instead of relying on the geocoding API provided by Open Weather.