

Feihuan Peng

Email: pengfeihuan@gmail.com | Phone: (725) 400-5628

[GitHub](#) | [LinkedIn](#) | [Portfolio](#)

Summary:

I'm a passionate self-taught developer with over two years of coding experience across Python, JavaScript, TypeScript, Swift, and GDScript. Experienced in building and optimizing projects with React, Next.js, Flask, Pygame, and React Native. Currently expanding skills in AI development with Python. Proven ability to solve complex problems and contribute effectively to team projects.

Skills:

Languages: Python, JavaScript, TypeScript, Swift, GDScript

Libraries/Frameworks: React, Next.js, Flask, Pygame, React Native

Tools/Platforms: Git, GitHub, VSCode, Xcode, Godot

Other: Redux, Firebase, Expo, HTML/CSS, API integration

projects:

Portfolio Website :

Code: [GitHub](#)

- Built a personal portfolio website using **React** to showcase projects and skills.
- Implemented 3D interactive elements using **Three.js** to create an engaging user experience.
- Integrated **EmailJS** for contact form functionality, allowing direct email from the website.
- Developed a responsive design with **Tailwind CSS** to ensure optimal viewing on various devices.
- Set up routing with **React Router DOM** to navigate between different sections.
- Applied code optimization techniques to enhance performance and reduce load times.
- Managed state effectively using **React's hooks**, ensuring a smooth user experience.
- Passed and managed props efficiently to maintain component modularity and readability.
- Incorporated a vertical timeline component to visually represent career progression.

Technologies: React, Tailwind, Three.JS , Emailjs.

Mobile App (Team project):

Code: [GitHub](#)

- Developed a comprehensive trip planning app that enables users to efficiently organize and manage travel itineraries, Leveraged React Native's **cross-platform capabilities** to ensure seamless performance on both iOS and Android devices.
- Contributed significantly to the development of the trip detail page, allowing users to view and edit trip specifics such as dates, destinations, activities, and accommodations.
- Applied the **MVC (Model-View-Controller)** design pattern to segregate concerns and enhance code maintainability, ensuring clear separation of data, presentation, and application logic.
- Utilized **Redux** for effective state management across the app, ensuring consistent data flow and smooth performance throughout user interactions.
- Integrated **Firebase** for cloud storage and real-time data synchronization, enabling seamless access to trip details across devices.
- Implemented **Async Storage** to facilitate local data storage, providing offline functionality and improving user experience in areas with limited connectivity.
- Leveraged **Expo Router** for intuitive navigation and seamless transitions between app screens, enhancing user experience and interaction flow.
- Collaborated effectively within a team environment, utilizing **Git version control** for codebase management, branching, and merging, ensuring streamlined development workflows.

Technologies: React Native, Expo, Redux, Firebase.

Weather App (python):

Code: [GitHub](#)

- Developed a weather application that allows users to get current weather conditions by entering a city name.
- Utilized **Flask** to build the backend server and handle API requests.
- Integrated the **OpenWeatherMap API** to fetch real-time weather data based on the user's input.
- Designed responsive HTML templates with **Jinja2** for dynamic content rendering.
- Implemented **error handling** for invalid city names, displaying a user-friendly message when the city is not found.
- Applied Python's **dotenv** package for environment variable management, ensuring secure API key usage.
- Used **requests** library to handle API requests and JSON responses.
- Employed **Waitress** as the WSGI server to deploy the Flask application.
- Developed a modular architecture by separating weather fetching logic into a standalone module.

Technologies: Python, Flask, OpenWeatherMap API, HTML, CSS, Waitress, dotenv, requests