

Portfolio Generator and Enhancer

1. Introduction

This application is built using **Streamlit**, a Python framework for creating auto generation of flutter based web portfolio's. The goal of the application is to facilitate the creation , deployment and management of portfolio projects.

The main components of the application include:

- **Profile setup and management** for users to define their portfolio details.
- **AI enhancement** features to assist users in refining their project details.
- **GitHub and Vercel integration** to help users fork, modify, and deploy their projects.
- **Customizable themes** and layout options to personalise the user experience.
- **GCP Big Query/Big Query Api** for data management.
- **Smtplib Server** for mail authentication.

The application uses various backend integrations and UI features to allow users to manage their projects easily, ensuring a streamlined process from creation to deployment.

2. Tech Stack and Libraries Used

- **Streamlit**: A Python library used for building interactive and dynamic web applications with minimal code.
- **Streamlit Option Menu**: Used for creating a horizontal menu to navigate between different sections such as "Home", "Projects", and "Info-Hub".
- **BigQuery(GCP)**: Data Warehouse
- **BigQuery API**: To fetch and display project data.
- **GitHub API**: For interacting with GitHub repositories (forking, renaming repos, modifying files).
- **Vercel API**: For deploying projects to Vercel directly from the application.
- **Regex (re)**: Used for URL validation (GitHub, LinkedIn, Gmail, etc.).
- **Base64**: For encoding images uploaded by users.
- **Python pathlib**: To manage file paths and directories.
- **AI Integration (Llm_prompt)**: For enhancing user-provided content via AI(**Gemini-1.5-pro-latest**).
- **Other Libraries**:
 - **time**: For adding delays in UI updates.
 - **Smtplib** for mail authentication(for sending **OTP**).

3. Architecture

The architecture of the application is primarily a client-server model. The key components are:

- **Frontend (Streamlit):** The user interacts with the application through a web interface created using Streamlit. The frontend is responsible for rendering input forms, menus, and displaying real-time updates to the user.
- **Backend:**
 - **GitHub Integration:** Allows users to interact with their repositories. Through API calls, users can fork repositories, modify files, and push changes.
 - **Vercel Deployment:** Deploys the user's project using Vercel API.
 - **AI Processing:** The app uses an AI model to help users rewrite or refine project details.
- **Session State:** Streamlit's `session_state` is used to manage session data across user interactions, allowing persistence of user inputs and selections across different sections of the app.
- **External Services:**
 - **BigQuery:** For retrieving project details such as deployment links.
 - In terms of data warehousing created complex **user to user group map tables** leveraging **data modelling** skills which helped in the updation of data and validity of the data(data quality) quite easy.
 - **Mail ID Validation:** To ensure that users provide valid email addresses for notifications.

4. Functionality

The app includes several interactive sections, each performing specific tasks:

Home Section:

- **AI Enhancer:** Allows users to refine project details using AI by clicking a button to trigger the content improvement feature.
- **GitHub and Vercel Integration:** Users can fork GitHub repositories, make modifications, and deploy them via Vercel.
- **Session Management:** The application keeps track of user actions (e.g., whether a GitHub repo has been forked or a project has been deployed) using `session_state`.
- **Otp Based Mail Authentication:** The Custom OTP-Based Email Authentication Service is a secure and efficient solution designed to provide users with an additional layer of authentication via One-Time Passwords (OTPs) sent directly to their email addresses. This service is highly customizable and can be integrated into any application requiring email verification for user registration, login, or critical transactions.
- **Social Media Links:** Allows users to input and validate URLs for various social media profiles like GitHub, LinkedIn, DevPost, Gmail, and Medium.

- **Customizable Colors:** Users can pick and define custom colors for background, text, and UI elements, and the app generates the necessary Dart code to apply these customisations in a Flutter project.
- **Custom Navigation:** Users can define custom names for the navigation items in their project. **It also accommodates blog page redirection in nav bar.**
- **Footer Customization:** Users can input a custom footer string, which is then integrated into the project.
- **Skills And Technology:** Users can input custom skills and upload their choice of photo for the corresponding skills and platform section.
- **Projects:** The Project Upload Section allows users to showcase two distinct categories of projects: Work Projects and Hobby Projects. This feature provides an intuitive, professional way for users to present their work experience and personal projects. Each project category has its own specific fields for details, making the system flexible and tailored to different types of projects. Users can add important details such as a project title, description, content, images, and links that highlight their work or personal achievements.

Key Features:

1. Project Categories:

- **Work Projects:** These represent professional projects the user has worked on, including work done for employers, clients, or collaborative team efforts. Each work project allows users to provide additional verification links to validate the work, such as the manager's social media profile or other professional references.
- **Hobby Projects:** These are personal projects that the user has worked on outside of their professional job. Hobby projects often include open-source contributions, personal apps, experimental projects, etc. For hobby projects, users can provide various external links, including GitHub, PyPI, personal websites, App Store, and Google Play Store links.

2. Project Details:

- **Title:** A clear, concise title to describe the project.
- **Brief Description:** A short paragraph that summarizes the core functionality and purpose of the project.
- **Full Content:** An extended description that provides in-depth details about the project, technologies used, challenges faced, and outcomes.
- **Image Upload:** Users can upload a related image for the project, such as a screenshot, logo, or app preview.

3. Link Integration for Verification and External Resources:

- **Work Projects:** Users can add a **verifier link**, which could be a link to their manager's LinkedIn profile or other social media accounts that can verify the work they contributed to.
- **Hobby Projects:** Users can add multiple links to showcase their hobby projects. The links may include:

- **GitHub:** Link to the project repository.
- **PyPI:** Link to the Python package (if applicable).
- **Website:** A link to a demo or project website.
- **App Store:** A link to the project's mobile app on the Apple App Store.
- **Play Store:** A link to the project's mobile app on Google Play Store.

Projects Section:

- **Project Deployment:** Users can view their deployed projects. The app fetches project details using BigQuery and dynamically displays deployment links.
- **Data Refresh:** Users can refresh records by clicking a button, prompting the app to fetch updated deployment information.

Info-Hub Section:

- **Guidelines:** Provides tutorial videos and written guides on how to generate GitHub and Vercel tokens required for project deployment. It also has a demo video and pdf download for the user_tech_documentation.

5. What Problems It Solves in the Current Industry and Users

This application addresses several common pain points for developers and project creators:

- **Simplifies Project Deployment:** Many developers face challenges deploying their projects to platforms like GitHub and Vercel. This app streamlines the process by providing an easy-to-use interface for forking repositories, making modifications, and deploying portfolio into production.
- **AI-Powered Content Enhancement:** Creating engaging and high-quality project descriptions or personal profiles can be time-consuming. The AI-enhancer feature helps users refine their project descriptions, saving time and improving content quality.
- **Centralized Profile Management:** Managing multiple social media links (GitHub, LinkedIn, DevPost, etc.) in one place is difficult. This app consolidates social media profile management and validates the URLs, ensuring they are correct and up to date.
- **Customization:** Developers building apps or websites often need to customize the UI colors and layout. This app allows users to pick their theme colors and generate the necessary code to integrate it into a Flutter project, reducing manual coding.
- **User-Friendly Experience:** By leveraging Streamlit, the app provides an intuitive interface that even users with limited technical expertise can navigate. The app also provides helpful guides and tooltips, making it easier for users to follow through on complex tasks like creating GitHub and Vercel tokens.
- **Scalability for Multiple User:** The application can be used by multiple users, making it ideal for project teams. It simplifies collaborative processes such as project deployment, GitHub repo management, and profile building.