**Standard Operating Procedure (SOP) for Running Streamlit Career Exploration App and Generating PDF Reports**

**Objective:**

This SOP provides detailed instructions for two processes:

1. Running a Streamlit web application for career exploration and college details.
2. Generating PDF reports containing college and job details using the Python script.

**Purpose :**

This SOP aims to guide users, developers, and administrators in effectively executing and maintaining the Streamlit career exploration application and PDF report generation process. It ensures consistency, accuracy, and ease of use.

**Scope:**

This SOP applies to users, developers, and administrators involved in utilizing and managing the Streamlit career exploration application and PDF report generation. It covers the end-to-end process from running the app to creating PDF reports.

**Responsibilities**

**Users:**

Run the Streamlit career exploration application using AWS link.

Select college and job details within the application.

Download PDF reports containing selected information.

**Developers:**

Maintain and update the Streamlit application code.

Ensure the application functions smoothly and accurately.

Provide support for AWS S3 integration and PDF generation.

**Administrators:**

Oversee the AWS S3 configuration and access.

Monitor the application's performance and availability.

**Procedure for Developer**

Step 1: Running the Streamlit Career Exploration App

**Prerequisites:**

Python installed (version 3.6 or higher).

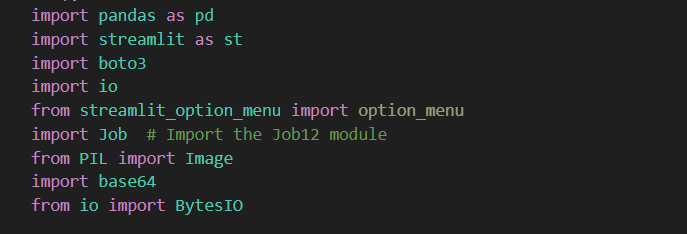
Required Python libraries installed (pandas, streamlit, boto3, PIL, etc.).

AWS S3 account with credentials.

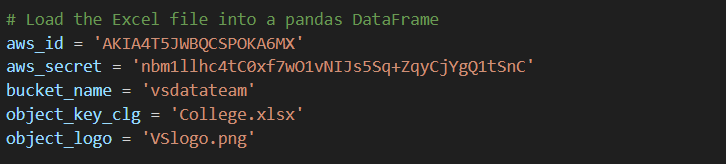
Internet access for external resources.

**Steps:**

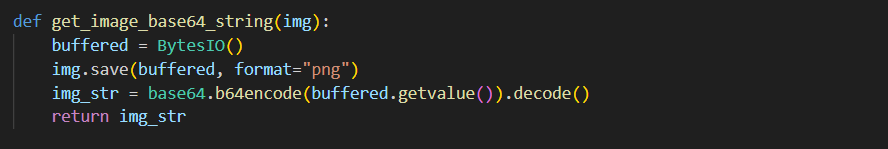
**Step 1: Importing Libraries**

1. Open a Python environment or code editor.
2. Import the necessary libraries using the provided script. 

**Step 2: Loading Data from AWS S3**

1. Replace **aws\_id**, **aws\_secret**, **bucket\_name**, **object\_key\_clg**, and **object\_logo** with your AWS credentials and S3 bucket information.
2. Load data from AWS S3 using the provided code. 

**Step 3: Converting Image to Base64 String**

* 1. Use the provided function to convert the logo image to a base64-encoded string. 

**Step 4: Creating the Streamlit Interface**

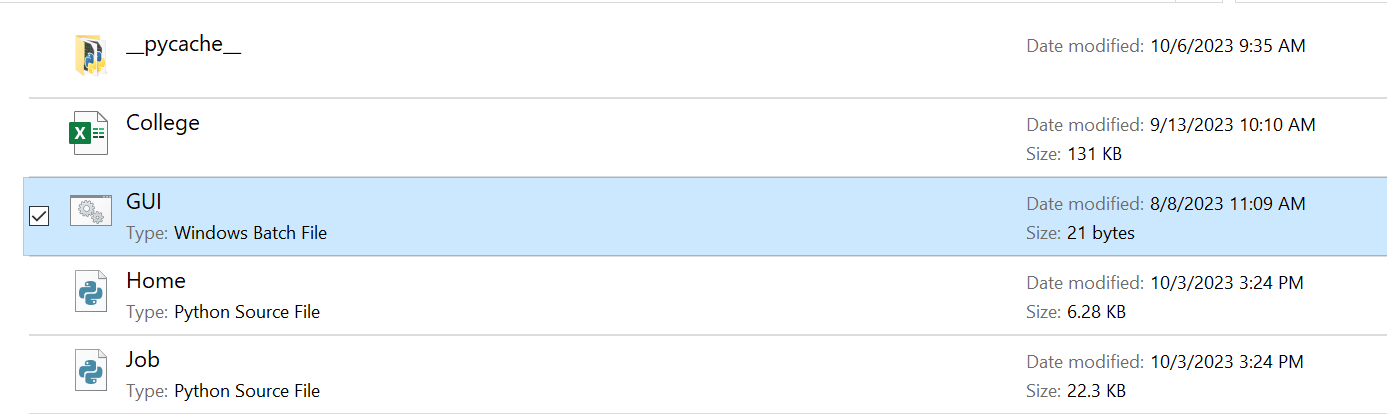
1. Define the Streamlit interface within the **main()** function. This includes the title, user inputs, and college details display.
2. Implement user inputs and error handling as specified in the script.

**Step 5: Handling the Next Page**

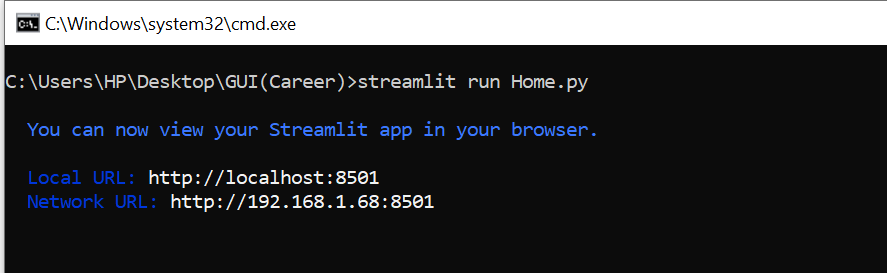
1. Check if the next page should be displayed and, if so, display the "Job12.py" page using the **Job.main()** function.

**Step 6: Running the Application**

**Method 1: Launching the Application using the GUI.bat File**

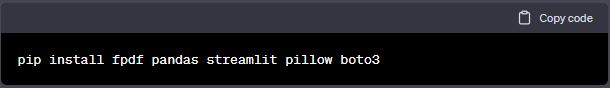
1. Locate the GUI.bat file for your application. 
2. Directly double-click on the GUI.bat file to launch the application.
3. Alternatively, you can create a shortcut to the GUI.bat file and place it on your desktop for convenient access. Clicking on this shortcut will open the GUI.

**Method 2: Launching the Application using the Streamlit Script**

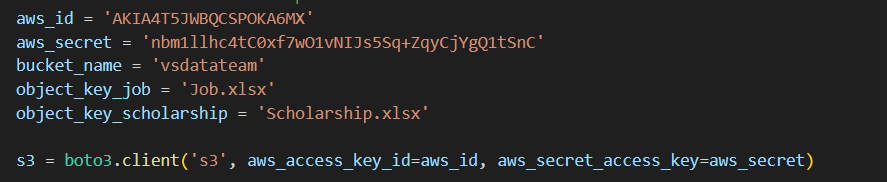
1. Go to the location where you have the Streamlit application script installed.
2. Open your command prompt or terminal.
3. Navigate to the directory where the Streamlit application script is located by typing **cd path/to/your/application** and press Enter.
4. In the command prompt, type the following command and press Enter:
5. Here, Home.py is assumed to be your main Python script for the application. 
6. After executing the command, you will see a black command prompt window.
7. Your GUI home page will become visible in your default web browser.
8. By following either Method 1 or Method 2, you can easily launch the application and access its graphical user interface (GUI).
9. Open a web browser and go to the URL provided by Streamlit (usually [http://localhost:8501](http://localhost:8501/)) to access the application.
10. Enter your name and select your desired degree, field, subfield, and college. 
11. Click the "Explore Career" button to proceed to the next page. 

**Part 2: Generating PDF Reports from College and Job Details**

**Step 7: Setup Environment**

1. Open your command-line interface or terminal.
2. Navigate to the directory containing the Python script.
3. Ensure you have Python installed and set up a virtual environment if needed.
4. Install the required Python libraries using **pip**: 

**Step 8: AWS S3 Configuration**

1. Obtain your AWS S3 access credentials (AWS ID and AWS Secret Access Key). 
2. Ensure you have access to the S3 bucket containing the Excel files with college and job details.

**Step 11: Select College and Job Details**

1. After selecting college and job details in the Streamlit application, scroll down to view the generated content.

**Step 12: Generate the PDF Report**

1. Click the "Download PDF" button to download the PDF report containing the selected college and job details.

**Step 13: Repeat as Needed**

1. You can repeat the process as many times as needed to generate PDF reports with different college and job details.

**Step 14: Close the Streamlit Application**

1. Once you have generated the desired PDF reports, close the Streamlit application by stopping it in the terminal (usually by pressing **Ctrl+C**).

**6. Conclusion**

By following this comprehensive SOP, users can effectively run the Streamlit career exploration application and generate PDF reports. Developers can maintain the application, and administrators can oversee AWS S3 configurations, ensuring a seamless and efficient process.

**7. References**

Streamlit Documentation: [streamlit.io/docs](https://streamlit.io/docs)

Python Documentation: [python.org](https://www.python.org/)

AWS S3 Documentation: [aws.amazon.com/s3](https://aws.amazon.com/s3/)

FPDF Documentation: [pyfpdf.readthedocs.io](http://pyfpdf.readthedocs.io/)

**Recommendation and feedbacks from Gayathri :**

To enhance the user interface (UI) of your Streamlit Career Exploration App and implement the requested changes, you can follow these guidelines:

1. **Eliminate Degree Dropdown**:
   * Remove the degree dropdown menu.
   * Replace it with two buttons: "Indian Careers" and "Foreign Careers" to allow users to choose their desired career path.
2. **Industry Selection**:
   * Create an interactive interface with icons representing various industries.
   * Users can click on an icon to select an industry they are interested in.
   * Use tooltips to provide brief descriptions of each industry when users hover over the icons.
3. **Top 10 Companies**:
   * Display logos of the top 10 companies for the selected industry.
   * Users can click on a company logo to explore job opportunities within that company.
   * Include emojis or visual icons next to each company logo to represent the company's culture or values.
4. **Visuals Over Text**:
   * Emphasize the use of visuals and icons throughout the app.
   * Replace lengthy text explanations with concise visual representations.
   * Use images, icons, and graphics to convey information whenever possible.
5. **Bullet Points**:
   * Summarize large paragraphs into three bullet points.
   * Use emojis or icons to represent key points within each bullet point.
6. **Role Models**:
   * Include pictures of role models or successful individuals in the selected career field.
   * Provide brief descriptions of their achievements and contributions.
   * Use emojis to highlight key skills or characteristics that made them successful.
7. **Career Comparison**:
   * Create a comparison feature that allows users to compare different careers side by side.
   * Use visual charts or graphs to display salary, job satisfaction, or growth prospects.
   * Add emojis or icons to represent the advantages and disadvantages of each career option.