GOOGLE LOGIN DOCUMENTATION

Problem statement:

Create a web application that implements Google Login functionality. Once a user logs in using their Google account, the dashboard should display a personalized message saying "Welcome [user name]" and include a logout button.

Requirements:

Use the Google Sign-In API to enable users to authenticate with their Google accounts.

After successful login, retrieve the user's name from the Google account information.

Display a dashboard page that includes a welcome message with the user's name.

Include a logout button that allows the user to sign out of their Google account.

Note: Use Django to implement the above and use proper git commits as in when required.

Project Overview:

The project is a web application that implements Google Login functionality. Its purpose is to provide users with a seamless and secure login experience using their Google accounts. Once logged in, users are greeted with a personalized dashboard that includes a welcome message displaying their name and a convenient logout button.

Key Features:

1. Google Login: Users can authenticate using their Google accounts, leveraging the Google Sign-In API.

2. Personalized Dashboard: After successful login, the application retrieves the user's name from their Google account information and displays a welcome message on the dashboard.

3. Logout Functionality: A logout button is available for users to sign out of their Google accounts and securely end their session.

With this application, users can enjoy the benefits of single sign-on with their Google accounts while accessing a personalized dashboard tailored to their name.

Technologies Used:

- Django Framework (Version: 4.2.1)
- Python (Version: 3.11.0)
- Google Sign-In API
- Google Cloud Platform
- HTML5/CSS

Installation and Setup:

Prerequisites:

- Python (version 3.6 or higher)
- Django (version 3.0 or higher)
- Git

Steps:

1. Open a terminal or command prompt and navigate to the directory where you want to store the project.
2. Create a Virtual Environment Navigate to the project directory and run the following command to create a virtual environment:

   >>> virtualenv mac

   >>> mac\Scripts\activate

   >>> pip install Django

   >>> django-admin startproject googlelogin

   >>> cd googlelogin

   >>> python manage.py startapp main

3. Configure Google Sign-In API Credentials:
   - Visit the Google Developers Console (https://console.developers.google.com/).
   - Create a new project or select an existing one.
   - Enable the "Google Sign-In API" for your project.
   - Obtain the client ID and client secret for your application.
   - Update the Django settings file with the obtained credentials.
4. Run Migrations: Apply the database migrations to set up the required database tables.
   >>> python manage.py makemigrations
   >>> python manage.py migrate

5. Start the Development Server:
   >>> python manage.py runserver

6. Access the Application: Open a web browser and visit http://localhost:8000 to access the application.

Git commands used:

>>>git bash here

For Configuring

>>> git config --global user.name "AmannSingh02"

>>> git config --global user.email "amansingh941941@gmail.com"

>>> git add comm.txt(will add this file)

>>> git commit -m "First Commit"(will commit)

>>> git push -u origin master (will push the file to remote repo)

django version 4.2.1

Enhancements and scope for improvements in future:

- User Profile Customization: Allow users to customize their dashboard by adding personalized widgets, changing themes, or rearranging the layout.
- Social Sharing Integration: Enable users to share content or updates from the dashboard on social media platforms, such as Twitter or Facebook.
- Multi-Provider Authentication: Extend the authentication options to include other popular providers, such as Facebook or Twitter, to give users more flexibility in logging in.
- User Data Analytics: Implement analytics features to track and analyse user behavior within the application, providing insights for improving user experience and making data-driven decisions.