

## Gensec HW7 prompts

**Attempt 1**

Create a simple flask program that has one index page and authenticates with google oauth2. If the user is not yet authenticated, the page should tell them to click the login link. If they are authenticated, the page should display a welcome message.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/04cbc02595e6cf559c2e132d111da0406f1504b7>

*Changed some stuff with environment variables, but overall code would give me an error for "USER: ImportError: cannot import name 'url\_quote' from 'werkzeug.urls'". Decided to retry prompt.*

Create a simple flask program in a file called test.py that has one index page and authenticates a user with google oauth2. If the user is not yet authenticated, the page should tell them to click the login link. If they are authenticated, the page should display a welcome message. Use the python dotenv package to get the google environment variables. The redirect\_uri for the google auth should be <http://127.0.0.1:5000/login/authorized> and make sure to use oauth2.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/17a89961e096e7b360e661221ec7c5a40fe96c0e>

Redo test.py to use Oauth2 instead of Oauth and use the requests-oauthlib package.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/1c5b5999a7e42b7efa4e1227a2e3bec570c21a69>

*Replaced with oauth2 and solved issue with being able to run flask server. However, requests-oauthlib expected https.*

Add an environment variable change in the name guard of test.py to set the environment variable OAUTHLIB\_INSECURE\_TRANSPORT to 1.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/7130adc124487d8b807f0b58fb88b4c21e26bdd8>

Instead of sending pure html in the route returns of test.py, create a new file called index.html with a jinja2 template to display the email of the user with a welcome message. In a new file called index.py, create a flask view using MethodView to serve the index.html file with the user email. Reflect these changes in test.py.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/f07220d342d339e4d0e740e7936e5e2abc0d39b3>

Create a file called login.html and include a button for logging in. Create a file called login.py and create a MethodView for serving the login page if the user is not authenticated. Reflect these changes in test.py and index.py.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/c1042335fab7540f854e1863b3ad3ec23430d6ca>

Move redirect\_uri, authorization\_base\_url, and token\_url from test.py to config.py and change the variables in test.py to env calls.

<https://gitlab.com/skhodak/aider-oauth-app1/-/commit/ab1cf83595cf61c9bc3d2b419e92bf814d9f7ae5>

## Styling prompts

Add a stylesheet inside of a /static folder and apply styles to the index and login page to center the text and button in the middle of the screen, use helvetica font, make font size 20px, and make the background a dark grey with all text light gray.

Change styles to make button be underneath the text with padding in between, rather than on the same line.

Add flexbox around the anchor tag and h1 in login.html, and set the flex to be a column, with a 3rem gap in between the objects.

Add a /logout route in test.py to log the user out and redirect to index view. Add a button to index.html that says, "Log out", and apply the flex-container class around it and the h1, like earlier.

Remove the second .flex-container style in styles.css. The earlier one can be applied to multiple divs.

## Attempt 2

Create a flask webapp that authenticates the user with google oauth2, with requests-oauthlib. In a templates folder, create a main html page that welcomes the user with their email and includes a logout button, and create an html login page that has a button to log in with google if the user hasn't been logged in yet. The user should be redirected to the login html page automatically at root if they haven't logged in yet. Each of these pages should have an accompanying MethodView class that is routed to from a main file called app.py. Also, create a config.py file with dotenv that gets the necessary environment variables (client\_id, client\_secret, auth\_base\_url, token\_url, and a redirect\_url that is set to `http://127.0.0.1:5000/login/authorized` route after the user is authenticated). Make sure to have a name guard in app.py that sets the env `OAUTHLIB_INSECURE_TRANSPORT` to 1. Make the requirements.txt file with all the necessary imports.

<https://gitlab.com/skhodak/aider-oauth-app2/-/commit/f6f287b1993219eb24dcbf25f5d0c24636b306a4>

*Did everything really well, except doesn't even use the login page, redirects straight to login view with authorization url. Not what I wanted.*

This app currently redirects the user automatically to the login view and never uses the login.html page to direct to. Change the MainView to serve the login page if the user is not authenticated yet.

*Fixed it.*

Add a stylesheet inside of a /static folder and apply styles to the main and login page to center the text and change the anchor tag to be a button in the middle of the screen, use helvetica font, make font size 20px, and make the background a dark grey with all text light gray. Also, add a flex container around the text and button of each page, and set them in a column with a 3rem gap.