Feathr Jr. Engineer Work Assignment

Welcome to your work assignment for the Junior Engineer role at Feathr! Your mission is to help Feathr keep user data safe by adding a secure authentication flow to this web application called "whoami".

What does whoami do? It's quite simple - when you log in, it tells you who you are! That part is basically done. What's not done is the authentication.

The basic skeleton of the application is in place, review the files in the `whoami` directory to get an understanding of the application architecture. It's a python web server built with the Flask microframework, which happens to be how we build most of our API servers here at Feathr.

You can implement your user authentication flow however you want, use whatever backing databases you want, just make sure to follow security best-practices to the best of your ability.

Modify and add files to this directory as needed, but maintain the existing routes and their accepted HTTP methods as defined.

As you work on this project, keep in mind the Feathr's core values, PACTS:

- Practicality - Prefer incremental improvement over delayed perfection
- Ambition - Learn and do what is necessary to solve real challenges
- Clarity - Make sure your code is understandable and maintainable
- Trust - Be sincere and trust others to do the same
- Service - Ask more from yourself than from others

Try to exemplify our PACTS in the design and implementation decisions you make as you work on this project.

Use whatever resources, tutorials, documentation, whatever you need, to complete the project. We google/stackoverflow stuff when we get stuck too. :)

If you can't finish the whole assignment in time, that's fine! We want to see how far you get and how you did it. It's about the process, not the destination.

Getting Started

We used `poetry` to setup this project. `poetry` provides packaging and dependency management for python projects. You can find documentation for how to use `poetry` at the project website (see the Resources section below). You will need to install poetry with pip first so you can use it to manage dependencies for this project.

```
pip install poetry
```

Then you'll need to install the project dependencies. In this directory, run

```
poetry install
```

When adding dependencies, running the development server or tests, make sure you have the virtual environment active. To activate the virtual environment for the project, run

```
poetry shell
```

Once the virtual environment is active, you will have access to the locally installed dependencies like the `flask` CLI, in your shell.

With the dependencies installed and your virtual environment active, you can run the server by navigating to the `whoami` directory and running

```
flask run
```

To run tests, just run

```
pytest
```

By default, this will find and run each function that begins with `test_` in each file that begins with `test_` in the `tests` directory. Consult the pytest documentation (see the link in the Resources section below) for more details and options.

Requirements

- A new user should be able to sign up with username and password at the `/signup` page.
- After signing up, a user should be able to login with username and password at the `/login` page.
- Once logged in, the user should be taken to `/me` where they can see their username on the page.
- Navigating to `/logout` should log the user out and take them back to `/login`.
- Users should not be able to access `/me` without logging in.
- A test for each aforementioned route should be written in `tests/test_whoami.py`
- Fill in the "Implementation" and "Limitations/Future Work" sections of the `NOTES.rst` file

Submit the finished project as a zip archive.

Additional Notes

In order to store user profiles and credentials, you'll need to add some sort of persistence layer to whoami. You can do this however you choose: with a backing database like MongoDB or MySQL, with python's built in sqlite3 support, or even by just writing to a file.

Resources

- Poetry documentation
- Flask documentation
- pytest documentation