Cyber Security

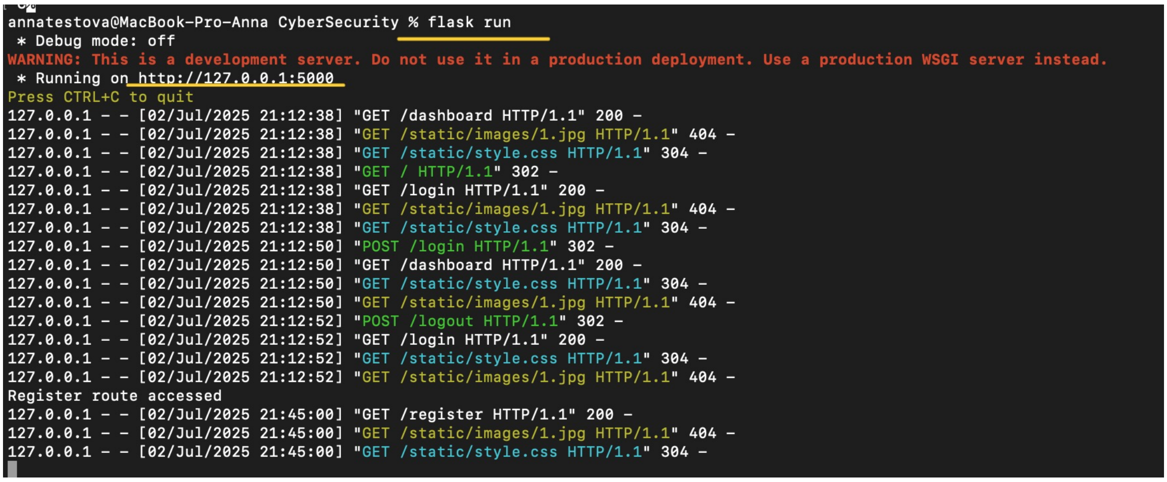
Idea 2: Secure Password Manager

With Encryption and Web Interface

Anna Testova
GH1025428
Link to the GitHub:https://github.com/AnnaTestova/annatestova.github.io.git

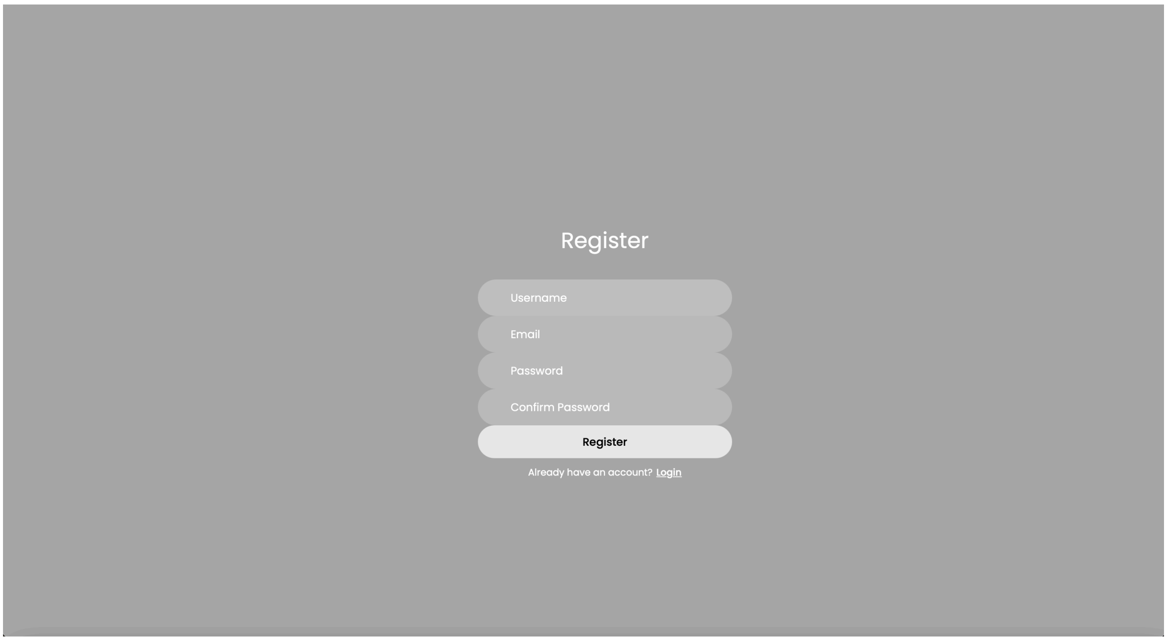
How to use my Programm:

I’m running my code through terminal on mac and I have flask-wtf and email-validator installed.



```
annatestova@MacBook-Pro-Anna CyberSecurity % flask run
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET /dashboard HTTP/1.1" 200 -
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET /static/images/1.jpg HTTP/1.1" 404 -
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET / HTTP/1.1" 302 -
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET /static/images/1.jpg HTTP/1.1" 404 -
127.0.0.1 - - [02/Jul/2025 21:12:38] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [02/Jul/2025 21:12:50] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [02/Jul/2025 21:12:50] "GET /dashboard HTTP/1.1" 200 -
127.0.0.1 - - [02/Jul/2025 21:12:50] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [02/Jul/2025 21:12:50] "GET /static/images/1.jpg HTTP/1.1" 404 -
127.0.0.1 - - [02/Jul/2025 21:12:52] "POST /logout HTTP/1.1" 302 -
127.0.0.1 - - [02/Jul/2025 21:12:52] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [02/Jul/2025 21:12:52] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [02/Jul/2025 21:12:52] "GET /static/images/1.jpg HTTP/1.1" 404 -
Register route accessed
127.0.0.1 - - [02/Jul/2025 21:45:00] "GET /register HTTP/1.1" 200 -
127.0.0.1 - - [02/Jul/2025 21:45:00] "GET /static/images/1.jpg HTTP/1.1" 404 -
127.0.0.1 - - [02/Jul/2025 21:45:00] "GET /static/style.css HTTP/1.1" 304 -
```

Just copy the http line and open it on the browser! Then it should look like this:



Report:

I’ve chose the second idea, because I’m interested in building web-applications and I wanted to learn more about them with the help of this project. I find this project quite complicated for my current skills, but I’ve tried to do my best.

My application allows users to register and log in, after which they can securely add and view their other passwords. All credentials are encrypted using cryptography library's Fernet symmetric encryption. So all users have their own unique generated encryption keys after registration.

This application uses Flask-WTF for the forms validation and it also includes CSRF(Cross-Site Request Forgery) protection to protect from web attacks. Flask ensures that only authenticated clients can access the dashboard and passwords. The simple user interface includes a simple dashboard shows stored credentials and also let users add new ones there. What is more, the logout securely ends the session and redirects users to the login page.

Due to the lack of my programming skills and the shortage of time I wasn't able to implement password retrieval feature into my current project.