Secure Portfolio Web Application

Author: Jamison Stalter
Project Type: Secure Full-Stack Portfolio Website
Technologies Used: Flask, Python, SQLite, HTML/CSS, Jinja2, Gunicorn, Render
Live URL: https://secure-portfolio-site.onrender.com
Repository: https://github.com/JayStalt/secure-portfolio-site

---

1. Introduction

The Secure Portfolio project is a full-stack web application built with Flask. It serves as both a professional portfolio and a demonstration of secure development practices. This platform includes robust user authentication, role-based admin access, project management capabilities, security microtools, and centralized logging. The site is deployed with Render using a production-ready Gunicorn WSGI server.

---

2. Objectives

- Build a secure, professional portfolio with user login and registration
- Implement role-based access control for admin-specific features
- Enable project management (create/edit/delete) through an admin interface
- Log all security-related actions (logins, project changes, etc.)
- Showcase cybersecurity tools through interactive microtools
- Deploy to a cloud platform with persistent logging and environment variables

---

3. Features

- **User Authentication:** Login, logout, and registration with bcrypt-hashed passwords
- **Admin Dashboard:** Role-based interface for managing all projects and viewing logs

- **Project Categories:** Public project display with tabs for:
    - Cybersecurity
    - Full Stack Development
    - Game Projects
    - Creative Writing
- **Security Logging:** Real-time logging of login attempts, project changes, and system actions to `security.log`
- **Cybersecurity Microtools:**
    - **Header Analyzer** – Analyzes response headers from a given URL
    - **JWT Decoder** – Decodes and displays contents of a JWT
    - **Threat Simulation** – Simulates user-inputted threat data to test log integrity
- **Live Deployment:** Deployed via Render with full environment separation

---

4. Architecture

- **Backend:** Flask, Flask-Login, Flask-WTF, SQLAlchemy
- **Frontend:** HTML, CSS, Jinja2 templating
- **Database:** SQLite (configured via SQLAlchemy)
- **Security:** bcrypt for password hashing, CSRF protection, manual access control
- **Logging:** Custom logging module writes events to a persistent `security.log` file
- **Hosting:** Render platform using Gunicorn WSGI server, Profile

---

5. Security Highlights

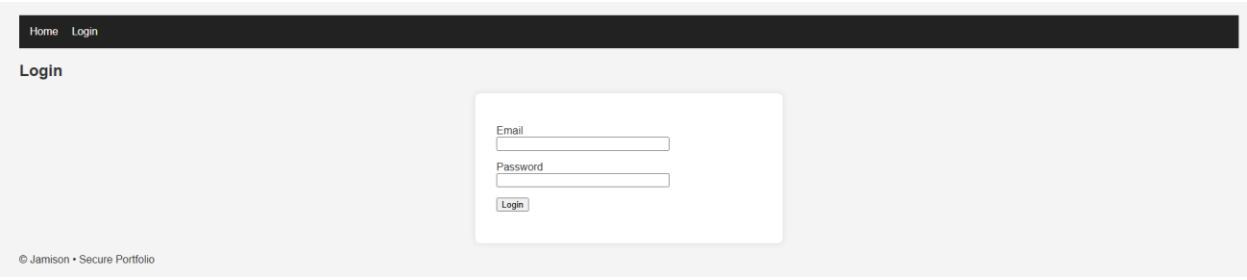
Control	Implementation
Password Hashing	Bcrypt (via Flask-Bcrypt)
SQL Injection Prevention	SQLAlchemy ORM – no raw SQL used
CSRF Protection	Enabled by default through Flask-WTF forms
Session Management	Flask-Login handles session authentication
Role-Based Access	Admin-only routes gated via email-based checks
Event Logging	Custom logger tracks key actions and outputs to file

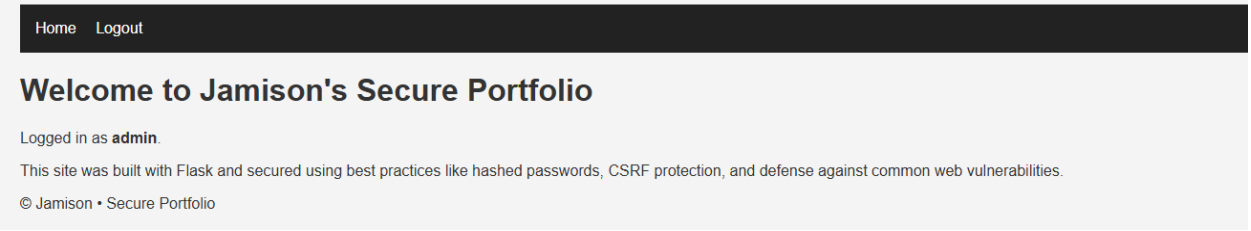
---

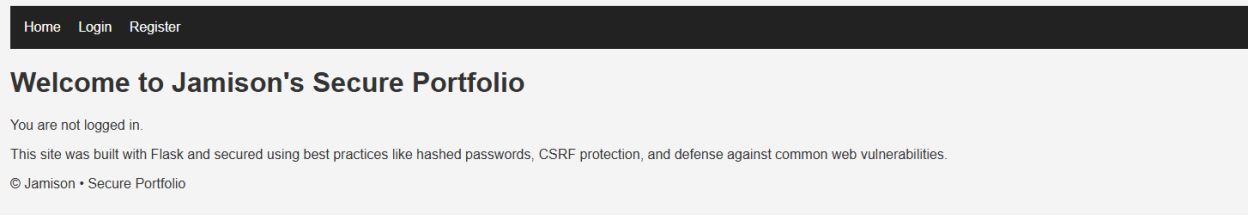
6. Deployment Instructions

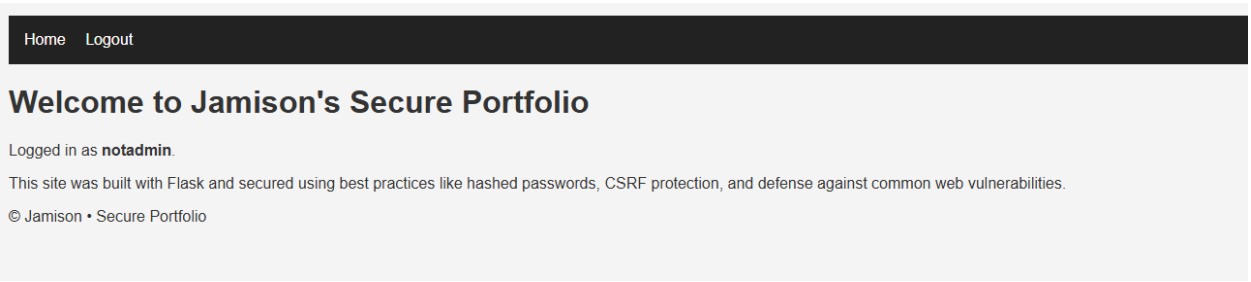
- **Local Setup:**
  - git clone https://github.com/JayStalt/secure-portfolio-site.git
  - cd secure-portfolio-site
  - python -m venv venv
  - source venv/bin/activate # On Windows: venv\Scripts\activate
  - pip install -r requirements.txt
- **Create `.env` File:**
  - SECRET_KEY=your-secret-key
  - DATABASE_URL=sqlite:///site.db
- **Initialize the Database:**
  - flask shell
  - >>> from app import db
  - >>> db.create_all()
  - >>> exit()
- **Run Locally:**
  - python run.py

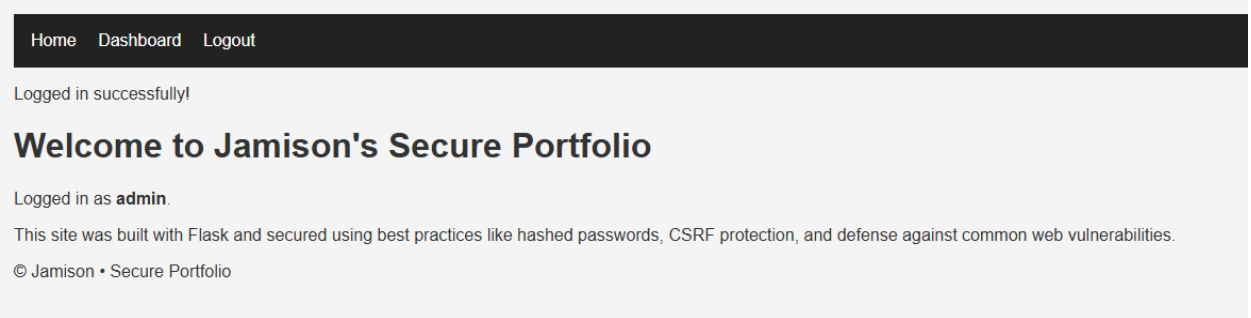
7. Screenshots











Home Dashboard Logout

Welcome, Admin

You have access to the secure admin dashboard.

- Manage Projects (Coming Soon)
- View Activity Logs (Coming Soon)
- Site Configuration (Coming Soon)

Home Projects Dashboard Logout

My Projects

Test Project

to test "Add New Project"

Home Projects Dashboard Logout

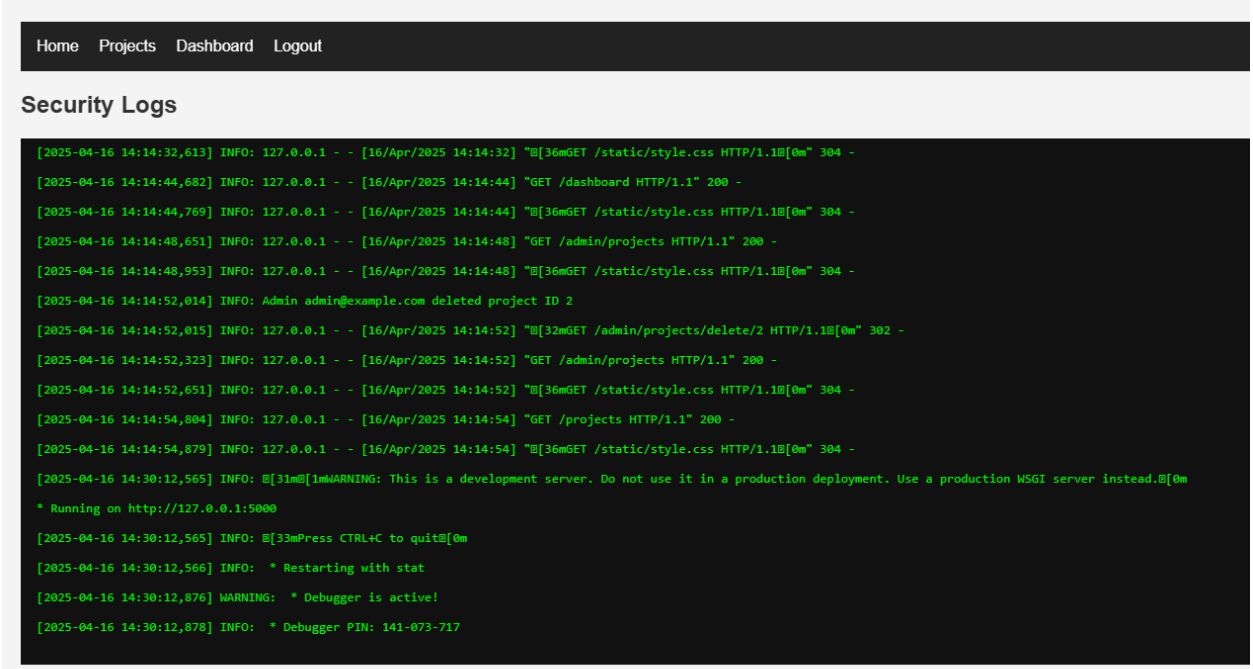
Project updated successfully!

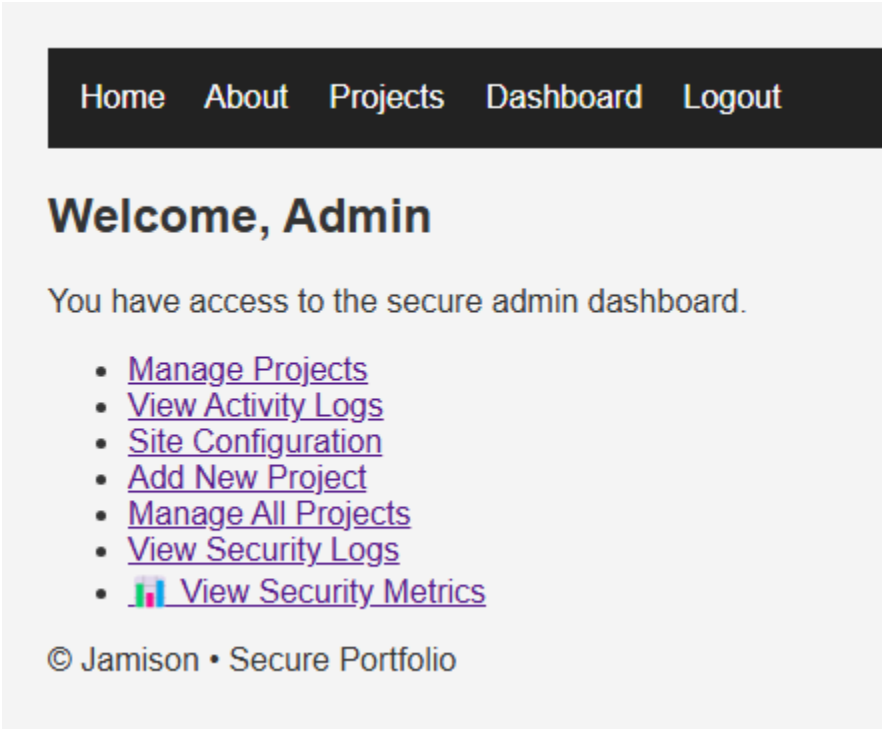
Manage Projects

Test Project *Updated*

to edit "Add New Project"

Edit | Delete


Home Projects Dashboard Logout
Security Logs
[2025-04-16 14:14:32,613] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:32] "[36mGET /static/style.css HTTP/1.1[0m" 304 -
[2025-04-16 14:14:44,682] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:44] "GET /dashboard HTTP/1.1" 200 -
[2025-04-16 14:14:44,769] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:44] "[36mGET /static/style.css HTTP/1.1[0m" 304 -
[2025-04-16 14:14:48,651] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:48] "GET /admin/projects HTTP/1.1" 200 -
[2025-04-16 14:14:48,953] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:48] "[36mGET /static/style.css HTTP/1.1[0m" 304 -
[2025-04-16 14:14:52,014] INFO: Admin admin@example.com deleted project ID 2
[2025-04-16 14:14:52,015] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:52] "[32mGET /admin/projects/delete/2 HTTP/1.1[0m" 302 -
[2025-04-16 14:14:52,323] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:52] "GET /admin/projects HTTP/1.1" 200 -
[2025-04-16 14:14:52,651] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:52] "[36mGET /static/style.css HTTP/1.1[0m" 304 -
[2025-04-16 14:14:54,804] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:54] "GET /projects HTTP/1.1" 200 -
[2025-04-16 14:14:54,879] INFO: 127.0.0.1 - - [16/Apr/2025 14:14:54] "[36mGET /static/style.css HTTP/1.1[0m" 304 -
[2025-04-16 14:30:12,565] INFO: [31m[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
* Running on http://127.0.0.1:5000
[2025-04-16 14:30:12,565] INFO: [33mPress CTRL+C to quit[0m
[2025-04-16 14:30:12,566] INFO: * Restarting with stat
[2025-04-16 14:30:12,876] WARNING: * Debugger is active!
[2025-04-16 14:30:12,878] INFO: * Debugger PIN: 141-073-717
© Jamison • Secure Portfolio


Home About Projects Dashboard Logout
Welcome, Admin
You have access to the secure admin dashboard.
Manage Projects
View Activity Logs
Site Configuration
Add New Project
Manage All Projects
View Security Logs
View Security Metrics
© Jamison • Secure Portfolio


Security Metrics Dashboard
Total Login Attempts: 5
Successful Logins: 4
Failed Logins: 1
Registered Users: 1
Unique Emails Logged In: 2
Last Log Entry: [2025-04-20 09:42:43,793] INFO: 127.0.0.1 - - [20/Apr/2025 09:42:43] "[36mGET /admin/metrics?__debugger__=yes&cmd=resource&f=console.png&s=qMKnNZi2EzHsskmoF6vY HTTP/1.1[0m" 304 -
© Jamison • Secure Portfolio

Home About Projects Dashboard Logout

HTTP Header Analyzer

https://example.com

Analyze

Headers:

- **Accept-Ranges:** bytes
- **Content-Type:** text/html
- **ETag:** "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
- **Last-Modified:** Mon, 13 Jan 2025 20:11:20 GMT
- **Vary:** Accept-Encoding
- **Content-Encoding:** gzip
- **Content-Length:** 648
- **Cache-Control:** max-age=1395
- **Date:** Sun, 20 Apr 2025 16:48:38 GMT
- **Alt-Svc:** h3=":443"; ma=93600,h3-29=":443"; ma=93600,quic=":443"; ma=93600; v="43"
- **Connection:** keep-alive

Home About Projects Dashboard Logout

JWT Decoder

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ1c2VyX2lkIjoxMjMsInJvbGUiOiJhZG1pbiJ9.
c2lnbmF0dXJl
```

Decode

Header:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload:

```
{
  "role": "admin",
  "user_id": 123
}
```

Signature:

```
c2lnbmF0dXJl
```


Home About Projects Dashboard Logout

Threat Simulation

This tool simulates how the app reacts to suspicious inputs like XSS or SQL Injection.

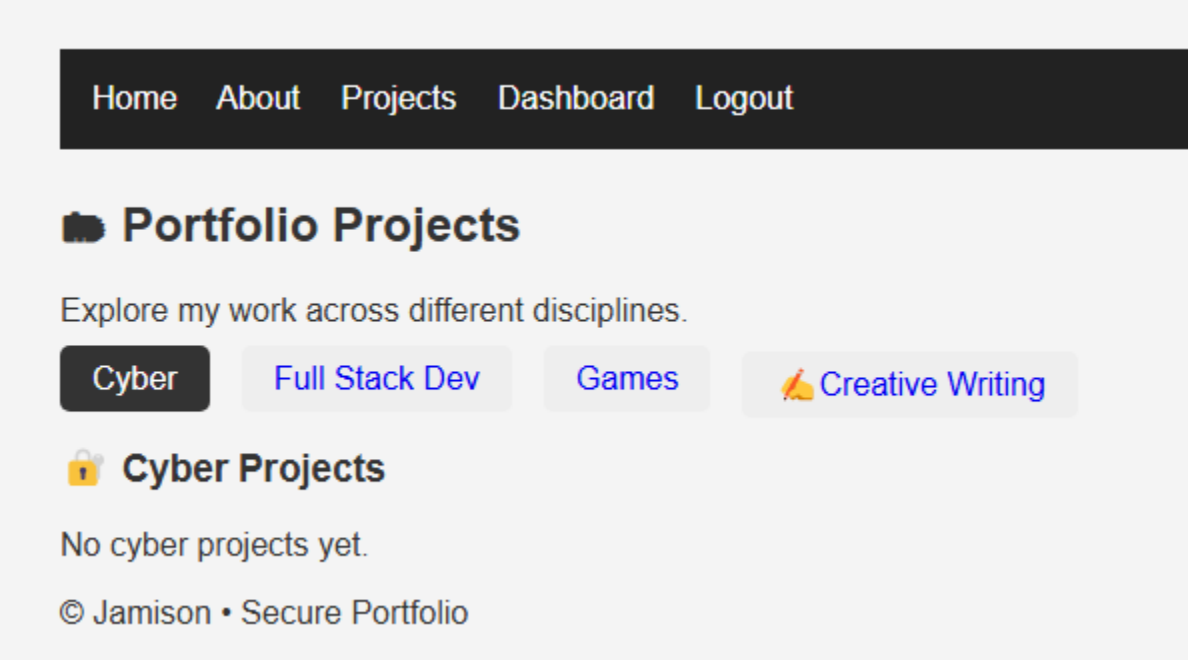
```
<script>alert('xss')</script>
```

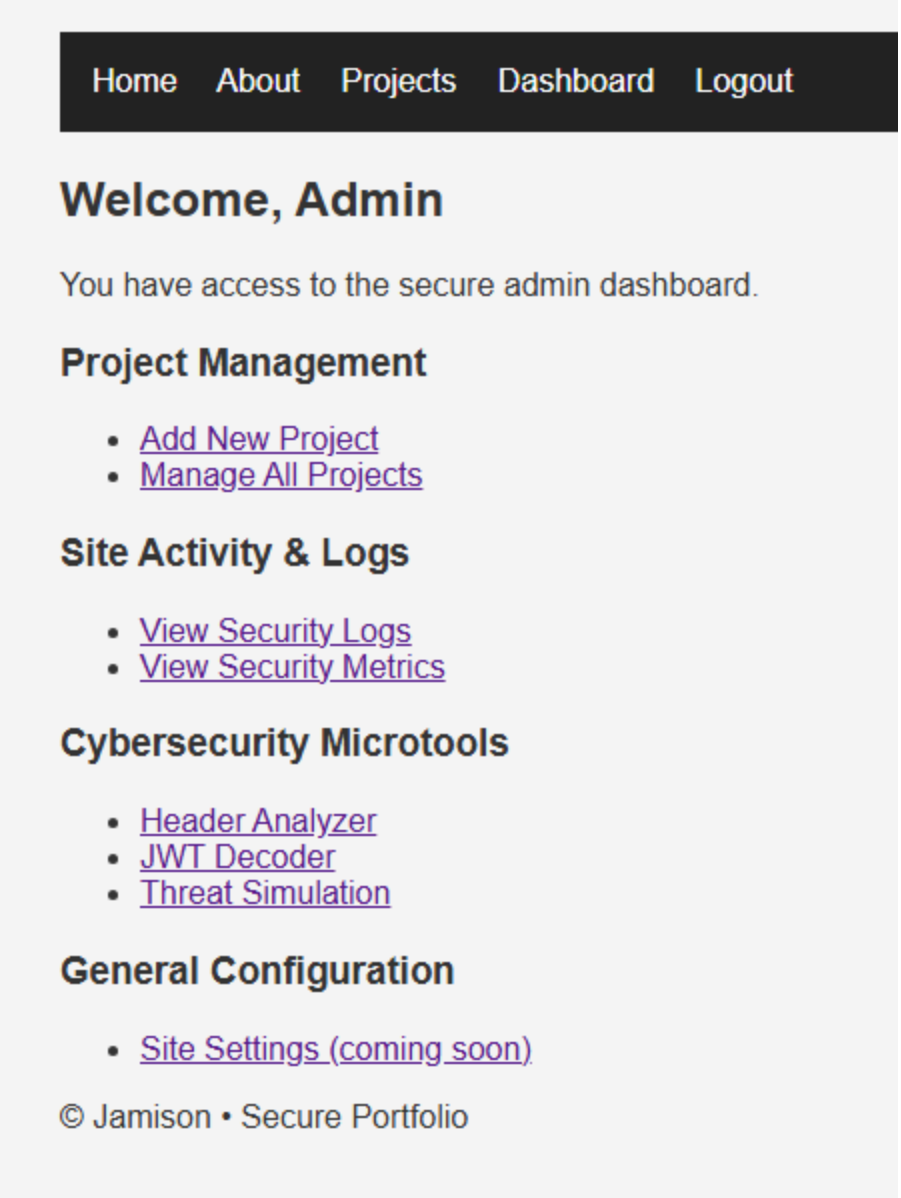
Submit

Threat simulation detected: <script>

Security Logs

```
[2025-04-20 09:58:51,362] INFO: Simulated threat input from admin@example.com: <script>alert('xss')</script>
[2025-04-20 09:58:51,367] INFO: 127.0.0.1 - - [20/Apr/2025 09:58:51] "POST /tools/threat-sim HTTP/1.1" 200 -
[2025-04-20 09:58:51,457] INFO: 127.0.0.1 - - [20/Apr/2025 09:58:51] "␛[36mGET /static/style.css HTTP/1.1␛[0m" 304 -
[2025-04-20 09:58:58,470] INFO: Simulated threat input from admin@example.com: <script>alert('xss')</script>
[2025-04-20 09:58:58,470] INFO: 127.0.0.1 - - [20/Apr/2025 09:58:58] "POST /tools/threat-sim HTTP/1.1" 200 -
[2025-04-20 09:58:58,780] INFO: 127.0.0.1 - - [20/Apr/2025 09:58:58] "␛[36mGET /static/style.css HTTP/1.1␛[0m" 304 -
[2025-04-20 09:59:09,281] INFO: Simulated threat input from admin@example.com: ' OR '1'='1
[2025-04-20 09:59:09,281] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:09] "POST /tools/threat-sim HTTP/1.1" 200 -
[2025-04-20 09:59:09,292] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:09] "␛[36mGET /static/style.css HTTP/1.1␛[0m" 304 -
[2025-04-20 09:59:21,206] INFO: Simulated threat input from admin@example.com: DROP TABLE users;
[2025-04-20 09:59:21,206] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:21] "POST /tools/threat-sim HTTP/1.1" 200 -
[2025-04-20 09:59:21,510] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:21] "␛[36mGET /static/style.css HTTP/1.1␛[0m" 304 -
[2025-04-20 09:59:28,803] INFO: Simulated threat input from admin@example.com: <?php echo "hacked"; ?>
[2025-04-20 09:59:28,804] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:28] "POST /tools/threat-sim HTTP/1.1" 200 -
[2025-04-20 09:59:28,816] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:28] "␛[36mGET /static/style.css HTTP/1.1␛[0m" 304 -
[2025-04-20 09:59:36,366] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:36] "GET /dashboard HTTP/1.1" 200 -
[2025-04-20 09:59:36,671] INFO: 127.0.0.1 - - [20/Apr/2025 09:59:36] "␛[36mGET /static/style.css HTTP/1.1␛[0m" 304 -
```





---

8. Conclusion

The Secure Portfolio Web App showcases the fusion of software engineering and cybersecurity. From access control to custom logging and interactive tools, this project demonstrates secure-by-design practices across development, architecture, and deployment. It's an active platform to highlight projects, tools, and personal achievements in both tech and storytelling.

Prepared by:
Jamison Stalter
Cybersecurity MS Candidate
USMC Veteran
Live URL: https://secure-portfolio-site.onrender.com
Repository: https://github.com/JayStalt/secure-portfolio-site