Data Integrity and Authentication Final Project

A Secure Document Vault with Authentication, Integrity, and Encryption

SecureDocs is a secure web platform where users can upload, sign, store, and manage documents, with full protection through:

- Modern authentication (OAuth 2.0 & Okta)
- Multi-Factor Authentication (2FA)
- Document encryption and digital signatures
- Secure transmission via HTTPS
- Role-based access control (RBAC)
- Data integrity verification

This system simulates a real-world document management application often used in legal, HR, or enterprise settings.

Roles & Permissions

Admin:

- Add/edit/delete users
- Manage roles
- View system logs
- Upload/edit/delete any document

User:

- Register/login
- Upload and download own documents
- Sign documents
- View and update own profile

Core Functionalities

1. **Authentication & Access Control**
    a. OAuth 2.0 Login via Google/GitHub
    b. SSO Login via Okta
    c. Enforced 2FA after login (Google Authenticator)
    d. Session-based login with token expiration
    e. Role-based UI (admin panel appears only to Admins)
2. **Document Vault**
    a. Upload documents (PDF, DOCX, TXT)
    b. AES-encrypted storage
    c. Automatically hash documents with SHA-256

d. Verify integrity on download using HMAC or CRC
e. Digitally sign documents using OpenSSL
f. Signature verification on download

3. **Profile Management**
    a. View and edit user profile
    b. Admins can view all user profiles and modify roles
    c. Enforce secure password policies
4. **HTTPS & Certificate Management**
    a. Configure local SSL/TLS using OpenSSL
    b. Run app over HTTPS
5. **Security Audit Simulation**
    a. Simulate MITM attacks using Wireshark
    b. Demonstrate protection via HTTPS
    c. Screenshots showing intercepted vs. protected traffic
6. **UI Requirements**
    a. General Guidelines:
        i. Clean, modern UI using Bootstrap, Tailwind, or similar
        ii. Responsive (works on desktop and mobile)
        iii. Route protection based on roles (e.g., Admin Panel visible only to admins)
    b. Pages to Design:
        i. Login / Register: OAuth buttons (Google, GitHub), optional fallback login
        ii. 2FA Setup: QR code generation & verification (TOTP)
        iii. Dashboard: Welcome, document stats, role-based UI widgets
        iv. Upload Document: Upload + encryption + signing form
        v. Documents List: Table with download, delete, integrity check buttons
        vi. Profile: View and edit user info
        vii. Admin Panel: View all users, roles, and logs
        viii. Audit Logs (Admin): View login history, file uploads, and suspicious actions

Proposed Folder Structure

```
securedocs/
├── app.py
├── templates/
│   ├── login.html
│   ├── dashboard.html
│   ├── upload.html
│   ├── profile.html
│   └── admin.html
├── static/
│   ├── css/
│   └── js/
├── uploads/
├── certs/
│   ├── server.crt
│   └── server.key
├── .env
├── requirements.txt
└── README.md
```

Project Submission Guidelines & Deadline

Submission Deadline:

The final deadline for submission is Wednesday, May 21, 2025, at 11:59 PM via the designated Google Form.

Team Requirements

Each team must consist of exactly 5 students.

Teams with fewer members will not be accepted.

Project Discussion

All teams will present and discuss their projects on Thursday, May 22, 2025 as follows:

- Group 1: at 9:00 AM
- Group 2: at 11:00 AM

Submission Requirements

You must submit the following:

1. Source Code (GitHub Repository)
    a. The code must be well-commented
    b. The implementation must follow secure coding practices
2. PDF Report
    a. A clear explanation of the encryption and authentication flow
    b. Screenshots showing each implemented feature
    c. A Wireshark capture summary demonstrating secure communication

Important Note and Disclaimer

It is highly recommended that all team members are from the same tutorial group to avoid scheduling conflicts with other sections or lectures.

Any student who attends the project discussion outside their assigned time does so at their own responsibility.