Team Members:

Aeslyn Brougthon, Shalim Castro, John Munoz

Libaries:

Library: `werkzeug.security`
Purpose: Secure password hashing and verification
Installation Command: `pip install werkzeug`

Library: `secrets`
Purpose: Generate cryptographically secure salts
Installation Command: **Built-in** (No installation needed)

Library: `hashlib`
Purpose: PBKDF2-based key derivation for encryption
Installation Command: **Built-in** (No installation needed)

Library: `base64`
Purpose: Encoding and decoding encrypted content
Installation Command: **Built-in** (No installation needed)

Library: `os`
Purpose: Retrieve environment variables for encryption keys
Installation Command: **Built-in** (No installation needed)

Library: `re`
Purpose: Regular expressions for password validation
Installation Command: **Built-in** (No installation needed)

Changes:

401K/ Retirement

- Configure Logging
- Define Blueprint
    - User Accounts with Predefined Funds
    - Rate Limit Tracker { “user”: {“endpoint”: [timestamps]} }
    - Logging Function
    - Rate Limiting Function
    - Balance Route (Limit: 10 per min)
    - Contribution Route (Limit: 3 per min)
- Prevent simultaneous transactions (Lock funds during processing)
- Final balance check before deducting funds

- Reset Route (Limit: 1 per min)

Admin Portal

- Forced the admin to change their default password on first login by redirecting to a password update page.
- Implemented password strength policies requiring:
    - At least 8 characters
    - At least 1 uppercase letter
    - At least 1 number
    - At least 1 special character from !@#$%^&*()_+=
- Created a password validation function to enforce security policies before allowing a password change.
- Prevented unauthorized users from accessing the password change page by restricting it to authenticated admins only.
- Replaced plaintext password storage with hashed passwords using werkzeug.security.generate_password_hash.
- Improved session security by ensuring that an admin who has not changed their password cannot proceed to other admin functions.
- Updated error handling to provide meaningful responses when:
    - Passwords do not match
    - Passwords do not meet security criteria
    - The user is unauthorized to change the password

Company News

- Fixed XSS vulnerability in /fetch by sanitizing user input.
- Added input validation for the filter parameter to prevent malformed input.
- Escaped user-generated content before rendering to prevent injection attacks.
- Implemented a response for detected XSS attempts, displaying an error message instead of processing malicious input.
- Improved error handling for:
    - Invalid JSON input
    - Request failures
    - Unexpected errors
- Replaced direct string interpolation in query parameters with a safe request structure.

File Upload

- Configure logging
- Allowed extensions and MIME types
    - Monitoring Metrics

    - MIME type validation (Stronger security)
    - Generate a unique filename and save the file
- Only supported file types supported

Login/Register New Acount

Under register:

- Implemented password strength policies requiring:
    - At least 8 characters
    - At least 1 lowercase letter
    - At least 1 uppercase letter
    - At least 1 number
    - At least 1 special character from !@#$%^&*()_+=
- New password confirmation.
- Added salting to the existing Hash values for added security.

Notes

SQL Injection prevention.
Restricted users access to only view personal notes

Added Log Monitoring tool to detect User activity for malicious activity

Areas for Future Improvements

Overall, a better logging system for the site would be beneficial.

401K/Retirement

Prevent user from viewing other user info
Prevent user from constant resetting and issue an error message when attempted

Admin Portal

A prompt for admin to change password from the hard coded default.
Extra multifactor authentication would be beneficial

Company News

No obvious areas for improvement.

File Uploads

No obvious areas for improvement.

Login Registration

No obvious areas for improvement.

Notes

In the notes section, there are still some concerns about XSS. However, future patches can use the code under the “Company News” section to reduce or eliminate it.

There is also room for improvement for encrypting the Notes data while at rest, so if outside users do access the database, the contents will be secure.