Project Title: Securing an Existing Banking Application - Security Assessment and Improvement of a Web-Based Banking Application

I. Project Goal:

To collaboratively assess, improve, and document the security of an existing web-based banking application, demonstrating an understanding of secure development principles and vulnerability assessment techniques.

II. Existing Application Features:

The existing Simple Banking App already implements:

1. **User Authentication:**
    a. Login with username/password
    b. Registration of new users
    c. Password recovery mechanism
2. **Account Management:**
    a. Display of account balance
    b. View of transaction history
3. **Fund Transfer:**
    a. Transfer money to other registered users
    b. Confirmation screen before completing transfers
    c. Transaction history updated after transfers
4. **User Role Management:**
    a. Regular user accounts
    b. Admin users with account approval capabilities
    c. Manager users who can manage admin accounts
5. **Location Data Integration:**
    a. Philippine Standard Geographic Code (PSGC) API integration
    b. Hierarchical location data selection

III. Security Improvement Focus Areas:

1. Secure Data Storage: Review and enhance how sensitive data is stored on the server.
2. Input Validation: Strengthen validation on both client-side and server-side to prevent injection attacks.
3. Authentication and Authorization: Improve existing authentication mechanisms and role-based access controls.
4. Session Management: Enhance session management to prevent session hijacking and fixation.
5. Cross-Site Request Forgery (CSRF) Protection: Review and strengthen existing CSRF protections.
6. Error Handling: Improve error handling to prevent information leakage.
7. Output Encoding: Enhance output encoding to prevent XSS vulnerabilities.
8. Dependency Management: Update all software components to the latest security patches.
9. Rate Limiting: Review and improve existing rate limiting mechanisms.
10. Secure Communication: Ensure all communication is properly encrypted.
11. GitHub Workflow: Use GitHub for version control, issue tracking, and pull requests for code review.

IV. Technology Stack (Already Implemented):

1. **Web Application:**

a. Frontend: HTML, CSS, Bootstrap 5
b. Backend: Python with Flask
c. Database: MySQL with SQLAlchemy ORM

2. **Security Libraries:**
   a. Flask-Bcrypt for password hashing
   b. Flask-Login for authentication
   c. Flask-WTF for CSRF protection
   d. Flask-Limiter for API rate limiting

V. Security Assessment (choose as many as you can):

1. **Tools:**
   a. OWASP ZAP (Zed Attack Proxy): For web application security scanning
   b. Burp Suite: For more advanced web application security testing
   c. Nmap: For network scanning
   d. Nikto: For web server vulnerability scanning
   e. Browser Developer Tools: For inspecting HTTP traffic and client-side code
2. **Testing Areas:**
   a. Authentication: Test for weak passwords, password cracking, and bypassing authentication
   b. Session Management: Test for session hijacking and session fixation vulnerabilities
   c. Data Storage: Verify that sensitive data is stored securely
   d. Network Communication: Verify that all communication is encrypted using HTTPS
   e. Input Validation: Test for injection vulnerabilities (SQL injection, XSS, command injection)
   f. Authorization: Test for improper access control
   g. CSRF: Test for CSRF vulnerabilities
   h. Clickjacking: Test for clickjacking vulnerabilities
   i. Known Vulnerabilities: Check for known vulnerabilities in the frameworks and libraries

VI. Deliverables:

1. **GitHub Repository:** A project repository
2. README.md File (Complete Documentation):
   a. Project Title
   b. Group Members
   c. Introduction
   d. Objectives
   e. Original Application Features
   f. Security Assessment Findings: Vulnerabilities identified in the original application
   g. Security Improvements Implemented: Detailed description of improvements made
   h. Penetration Testing Report: Summary of vulnerabilities identified, exploitation steps, and recommendations
   i. Remediation Plan: Steps taken to address identified vulnerabilities
   j. Technology Stack: Updated list of technologies used
   k. Setup Instructions: Instructions on how to set up and run the improved application
   l. Live web application link in www.pythonanywhere.com: see this tutorial

3. Presentation: A presentation summarizing the security assessment, vulnerabilities identified, and security improvements implemented (.pptx)

VII. Evaluation Criteria:
   1. Security Assessment Quality: How thorough and effective was the security analysis?
   2. Security Improvements: How effectively were security vulnerabilities addressed?
   3. Vulnerability Assessment: How thorough and effective was the penetration testing process?
   4. Documentation: How clear, comprehensive, and well-organized is the documentation?
   5. Collaboration: How effectively did the group members collaborate using GitHub?
   6. Presentation: How well does the student communicate their understanding of the security improvements?