CSC Multicourse Biweekly Reports[1]

Your team's members:

Name	Email Address
NICHOLAS HAMILTON	hamiltonn@berea.edu

Select Report Date: Dec 31, 2024

Part 1: Weekly Progress Report

- **Accomplishments**: What did you accomplish since the last class meeting? *(up to 3 points)*

Set up PostgreSQL database - Installed and configured PostgreSQL server for secure data storage
Created initial database schema - Designed tables for users, projects, searches, and other core functionality
Enhanced database schema with security features - Added tables and columns for user authentication, session management, and rate limiting
Implemented DatabaseManager class - Created a Python class to handle all database operations securely
Set up Flask application structure - Created modular Flask application with blueprints for different functionality areas
Implemented secure password handling - Used bcrypt for password hashing and secure storage
Created user registration system - Built endpoint and database functions for new user creation with email verification
Implemented login system - Created secure login functionality with brute force protection and session management
Added authentication decorators - Created login_required decorator to protect routes from unauthorized access
Implemented session management - Added secure session handling with token generation and validation
Created rate limiting system - Implemented protection against brute force attacks and API abuse
Created frontend registration form - Built HTML form with client-side validation for user registration
Created frontend login form - Built HTML form with secure credential submission
Set up static file serving - Configured Flask to serve static files (CSS, JavaScript) from frontend directory
Set up template rendering - Configured Flask to serve HTML templates from frontend directory
Implemented CSRF protection - Added security against Cross-Site Request Forgery attacks
Set up environment variables - Created .env file for secure configuration management
Created directory structure documentation - Wrote detailed documentation explaining project organization
Set up logging system - Implemented comprehensive logging for security events and debugging
Implemented user activation system - Created email verification functionality for new accounts
Created password reset functionality - Implemented secure password reset with time-limited tokens
Set up throttling system - Added request rate limiting for sensitive endpoints
Created project management system - Built functionality for users to create and manage projects
Implemented search history - Created system to track and manage user searches
Added admin capabilities - Created admin-specific functionality and access controls
Set up error handling - Implemented comprehensive error handling and user feedback

[1] Detailed Weekly Report requirements can be found here: 24FA-CSC450 - Using Agile and Reporting Out

Created database backup system - Set up PostgreSQL backup and restore capabilities
Implemented database indexing - Added indexes for performance optimization and security
Set up version control - Initialized and organized Git repository for project management

- **Challenges**: What are your current roadblocks? *(up to 3 points)*

Need to test user activation functionality -Currently need to implement proper email verification testing methods and ensure tokens are working correctly
Session management security testing - Need to develop comprehensive tests for session token security and timeout mechanisms
Admin dashboard implementation - Need to create the admin interface for monitoring user activity, viewing security logs, and managing user accounts

- **Desired Discussion Points**: Do you have any desired discussion points that are not related to roadblocks? *(up to 2 points)*

Best practices for storing and managing API keys in the database - Looking for guidance on securely handling Census API keys for users
Recommendations for implementing real-time security alerts - Want to discuss options for notifying admins of suspicious activity (multiple failed logins, unusual access patterns)

- **Future Goal(s)**: What do you plan to accomplish before our next class meeting? These plans should be related to roadblocks or discussion points. If you plan to change direction, explain why. *(up to 2 points)*

Implement and test email verification system - Will set up email service integration and create verification workflow
Create basic admin dashboard - Will develop initial admin interface with user management and security log viewing capabilities

Part 2: Time Reporting

Make sure that as you fill out the first prompt, you include in enough detail in the summary. For example, "debugging" is vague, but "debugged function X to make sure that when user does action Y, it is called and returns the value Z" is better.

- **Time Spent**: Briefly explain how much time, outside of class, spent on your project. If you worked on multiple components, each should get a detailed summary. Make sure to add up all the hours and minutes correctly. *Add as many rows as you need to the table below. Please do not include time spent in the classroom. (up to 4 points)*

FIRST TWO WEEKS

START	FINISH	HOURS	DETAILED SUMMARY
10/22 13:00	10/22 15:30	2.5	Started PostgreSQL database setup, installed dependencies, configured initial connection
10/22 16:00	10/22 18:45	2.75	Created basic schema for users table, set up initial indexes and constraints
10/23 14:00	10/23 17:30	3.5	Implemented basic DatabaseManager class, added user creation functionality
10/23 19:00	10/23 21:15	2.25	Added user verification functions, tested database connections
10/25 13:30	10/25 16:45	3.25	Enhanced schema with security tables (sessions, login_history)
10/25 17:30	10/25 20:00	2.5	Added rate limiting and throttling tables, created security indexes
10/26 14:00	10/26 17:15	3.25	Implemented secure password handling with bcrypt, added password history
10/26 18:00	10/26 21:00	3	Created login system with brute force protection
10/28 13:00	10/28 16:30	3.5	Set up Flask application structure, created initial blueprints
10/28 17:00	10/28 19:45	2.75	Implemented registration routes and basic error handling
10/30 14:00	10/30 17:30	3.5	Created HTML templates for authentication forms
10/30 18:30	10/30 21:15	2.75	Added client-side form validation, improved error messages
11/01 13:30	11/01 16:45	3.25	Implemented user activation system, set up email verification
11/01 17:30	11/01 20:00	2.5	Created password reset functionality with secure tokens
11/02 14:00	11/02 17:15	3.25	Set up security logging system, implemented audit trails
11/02 18:00	11/02 20:30	2.5	Added request throttling implementation
11/03 13:00	11/03 16:30	3.5	Started admin functionality, created admin routes
11/03 17:30	11/03 20:45	3.25	Implemented user management capabilities for admins
11/04 14:00	11/04 17:15	3.25	Added security monitoring features and dashboard
11/04 18:00	11/04 20:30	2.5	Final testing and debugging of auth system
	WEEKLY TOTAL	54	

- **Total (Cumulative) Project Time Spent**: After the number of hours and minutes, make sure to briefly explain whether you are on track and if not, what you may need to do in order to achieve what you set out to accomplish. *(up to 2 points)*

I am mostly on track with the core security and authentication components of the project, having completed the essential database structure, user authentication system, and basic admin functionality. However, to fully achieve what I set out to accomplish, I still need to:

Complete the email verification system implementation with actual email service integration (estimated 5-6 hours needed)
Develop a more comprehensive admin dashboard with detailed security logging and user management interfaces (estimated 8-10 hours needed)
Implement thorough testing of all security features, particularly session management and rate limiting (estimated 6-8 hours needed)

Rubric:

The following rubric will be used, but they might change as needed.

Accomplishments (3 points)

1 point for a general description of progress, 2 points for specifics on progress, 3 points for specifics AND referring to previous targets and explaining how current accomplishments build on previous ones.

Challenges (3 points)

1 point for mentioning there are roadblocks, 2 points for specifics, 3 points for specifics AND what was done already to try to overcome them.

Desired discussion points (2 points)

1 point for at least one relevant discussion point as a general question, 2 points for relevant discussion points with specifics

Future Goals (2 points)

1 point for concrete future targets (i.e. "working more on the project" is a zero, but "working on getting component X to interface with component Y" suffices), 2 points for tying in the targets with what was hopefully discussed in the meeting.

Time Spent (4 points)

1 point for including general statements of how much time was spent ("4 hours on coding"), 2 points for splitting time into specific parts ("1.5 hours on research on component X, 1 hour coding, 2.5 hours debugging"), 3 points for specific parts and details on the pieces ("1.5 hours researching Turtle interface for drawing concentric circles given inputs from the user, 1 hour coding function X that used that interface, 2.5 hours testing function X by giving it multiple values and fixing errors for values A, B, C, and D"). 1 Point for totalling the hours correctly. Please do not include time spent in the classroom. Only report on time spent outside of classroom hours.

What happens if your time on a task is interrupted and you don't have a concrete (or discrete) end time? In this case put the start time in, and the word "interrupted" for the end time and include the task total time. Rounding to the nearest 15 minutes is acceptable. *(This makes adding up times easier, especially when you use decimal hours, i.e. 3.75 rather than 3 hours 45 minutes.)*

Total (Cumulative) Project Time (2 points)

1 point for summing the values correctly, 2 points for the total time AND reflection on progress (you are confident to fit the target and if not, what course corrections you anticipate needing to make)

Resources

Here's a link to this Weekly Report Template – Make a copy and use it:

CSC Multicourse Biweekly Report INDIVIDUAL TEMPLATE

SOURCES:

https://www.geeksforgeeks.org/what-is-__init__-py-file-in-python/