Assignment Title: Building a Secure User Authentication System with HTML and CSS

Assignment Description:

Objective: The purpose of this assignment is to assess the candidate's ability to create a complete and user-friendly user authentication system using PHP, HTML, and CSS.

Instructions:

You are tasked with building a user authentication system for a web application. The system should be secure, visually appealing, well-structured, and follow best practices in PHP, HTML, and CSS development. You should also incorporate features to prevent common security vulnerabilities such as SQL injection and cross-site scripting (XSS).

Requirements:

- User Interface Design:
  - Create visually appealing HTML and CSS for the user registration, login, and dashboard pages.
  - Ensure the user interface is responsive and works well on both desktop and mobile devices.
- User Registration:
  - Integrate the registration form into the HTML layout.
  - Create a visually appealing registration form with the following fields:
    - Username (unique)
    - Email (unique)
    - Password
    - Confirm Password
  - Implement server-side validation for the registration form to ensure data integrity.
  - Hash and securely store passwords in the database.
- User Login:
  - Integrate the login form into the HTML layout.
  - Create a visually appealing login form with fields for username/email and password.
  - Implement user authentication using PHP sessions.
  - Protect against brute force login attacks by implementing rate limiting or a captcha.
  - Implement a "Remember Me" feature.
- Password Reset:
  - Integrate the password reset functionality into the HTML layout.

- Allow users to request a password reset link by email.
- Implement a visually appealing and user-friendly password reset process.

User Dashboard:

- After logging in, users should have access to a dashboard with a visually appealing layout.
- Implement user profile picture upload functionality with proper security measures.

Security:

- Implement protection against common security threats, including but not limited to SQL injection and XSS.
- Use prepared statements for database interactions.
- Implement CSRF protection.
- Store sensitive data securely, such as API keys and database credentials.

Logging:

- Implement logging for important security events and error handling.
- Log login attempts, password reset requests, and other relevant activities.

Documentation:

- Provide clear and concise documentation for the code, including comments and explanations of key decisions made during development.
- Include instructions on how to set up the project locally.

Testing:

- Write unit tests for critical parts of the authentication system.

Scalability and Performance:

- Optimize the code for scalability and performance.

Deliverables:

A well-structured PHP project with all the necessary files and folders.
A secure, functional, and visually appealing user authentication system that meets the requirements outlined above.
Documentation explaining the code, including HTML and CSS, and setup instructions.
Any necessary database schema or setup scripts.
Unit tests for critical components.

Evaluation Criteria:

HTML/CSS Design: How well the candidate designs and styles the user interface.
Security: How well the authentication system protects against common security threats.
Functionality: Does the system meet all the specified requirements?
Code Quality: Is the code well-structured, organized, and easy to understand?

Documentation: Is the code well-documented with clear explanations?
Testing: Are there adequate unit tests, and do they pass successfully?
Performance: Does the system perform well and efficiently?

This extended assignment will evaluate the candidate's ability to create an aesthetically pleasing and secure user authentication system while also assessing their HTML and CSS skills.