Student Grading System Assignment

Objective:

The purpose of this assignment is to develop a comprehensive Student Grading System that progresses through three stages of implementation. The goal is to assess students' understanding and proficiency in various technologies, frameworks, and concepts related to backend development, web application development, and Spring MVC/Spring REST.

Part 1: Command-line / Sockets and JDBC Backend Implementation

Task:

Build the initial version of the Student Grading System using the command-line interface, sockets, and JDBC in the backend.

Requirements:

- Design a MySQL database schema to store student information, courses, and grades.
- Create a command-line interface that allows students to log in and access their marks in different courses.
- Utilize JDBC to interact with the MySQL database, retrieve student data, and manage grades.
- Implement socket communication for any real-time requirements.

Assessment Criteria:

- Proper command-line implementation and handling of user input.
- JDBC integration to perform database operations.
- Demonstrated understanding of socket communication.
- Data security measures in place for user authentication and protection.

Part 2: Web App / Traditional MVC Servlets and JSPs Implementation

Task:

Enhance the Student Grading System by transforming it into a web application using traditional MVC Servlets and JSPs.

Requirements:

- Develop HTML and JSP files for user login, mark visualization, and display of statistical data (e.g., class average, median, highest, and lowest marks).
- Create servlets to handle user requests, manage sessions, and interact with the database using JDBC.
- Ensure secure user login functionality and access to their respective marks.

Assessment Criteria:

- Effective use of Servlets to manage user requests and responses.
- Accurate display of student marks and statistical data in JSPs.
- Proper implementation of user authentication and session management.
- Clear demonstration of traditional MVC design principles.

Part 3: Web App / Spring MVC and Spring REST Implementation

Task:

Advance the Student Grading System by migrating it to a web application utilizing Spring MVC and Spring REST.

Requirements:

- Configure Spring for the application and integrate Spring MVC and Spring REST modules.
- Replace traditional Servlets with Spring controllers to handle user requests and responses.
- Implement Spring Security for secure user authentication and authorization.
- Employ Spring JDBC template or JPA for database interaction.

Assessment Criteria:

- Accurate Spring configuration and module integration.
- Seamless migration from traditional MVC to Spring MVC.
- Effective implementation of Spring Security for user management.
- Proper use of Spring JDBC template or JPA for database operations.

Overall Assessment Criteria:

- Thorough understanding of backend development concepts, including database management, user authentication, and session handling.
- Effective implementation of different technologies and frameworks throughout the progression of the assignment.
- Proper error handling and feedback mechanisms for users.
- Well-structured video recording (8 mins max) showcasing each stage of the implementation, highlighting the development process, and explaining the rationale behind technological choices.
- Comprehensive written report demonstrating:

- Design: A clear and detailed explanation of the system's architecture, data model, and interaction flow between components.
- Implementation: Detailed documentation of the code, highlighting key functionalities, algorithms used, and important design decisions made during the development.
- Analytical Thinking: Critical evaluation of the strengths and weaknesses of each implementation stage, comparing different technologies, and justifying technology choices based on relevant criteria.
- Challenges and Solutions: Identification of challenges faced during the development process and well-described solutions to overcome them.
- Security Considerations: Analysis of the security measures implemented in the system and the rationale behind them.
- Future Enhancements: Insightful suggestions for potential improvements or features that could be added in future iterations of the system.
- The written report should be concise, well-organized, and written in a professional manner. It should demonstrate your ability to communicate complex technical concepts clearly and effectively.