Hackathon Assignment: Classroom Assignment Management System

Objective:

Build a web application for managing classroom assignments. The application should have separate interfaces for teachers and students with the following functionalities:

User Roles:

1. Teacher
2. Student

Requirements

1. Login Page

- User Interface:
  - Simple form with fields for Username, Password, and a role selection dropdown (Teacher/Student).
  - Submit button to log in.
- Functionality:
  - Validate user credentials against the database.
  - Redirect users to their respective dashboards based on their role.
  - Implement error handling for incorrect credentials.

2. Teacher Interface

A. Add a New Assignment

- User Interface:
  - Form with the following fields:
    - Title (Text)
    - Description (Text Area)
    - Due Date (Date Picker)
    - Attachments (File Upload, optional)
    - Class (Dropdown for 11th or 12th)
  - Submit button to create the assignment.
- Functionality:
  - Validate form inputs.

    - Save the assignment details to the database.
    - Provide feedback to the teacher upon successful creation.

B. View Submitted Assignments

- User Interface:
    - List view of submitted assignments with columns for:
        - Assignment Title
        - Student Name
        - Submission Date
        - Class (11th or 12th)
    - Filters for assignment title, student name, and class.
    - Option to view details of each submission.
- Functionality:
    - Fetch and display submissions from the database.
    - Implement filtering and searching capabilities.
    - Provide detailed view for each submission.

C. View All Assignments

- User Interface:
    - List view of all assignments created by the teacher with columns for:
        - Assignment Title
        - Class (11th or 12th)
        - Due Date
        - Actions (Edit, Delete)
    - Option to view details of each assignment.
- Functionality:
    - Fetch and display assignments from the database.
    - Implement edit and delete functionalities.
    - Provide detailed view for each assignment.

D. Dashboard

- User Interface:
    - Summary of total assignments created.
    - Number of submissions received.
    - Upcoming deadlines.
    - Charts/graphs for visual representation (optional).
- Functionality:
    - Fetch and display summary data from the database.
    - Optionally integrate charting libraries for visual representation.

3. Student Interface

A. Submit an Assignment

- User Interface:
    - List view of available assignments for the student's class (11th or 12th) with columns for:
        - Assignment Title
        - Due Date
    - Form to submit an assignment with the following fields:
        - Assignment Title (Dropdown)
        - Submission Text (Text Area)
        - Attachments (File Upload, optional)
    - Submit button to submit the assignment.
- Functionality:
    - Fetch and display available assignments from the database.
    - Validate form inputs.
    - Save the submission details to the database.
    - Provide feedback to the student upon successful submission.

B. View Submitted Assignments

- User Interface:
    - List view of assignments submitted by the student with columns for:
        - Assignment Title
        - Submission Date
        - Status (Submitted, Graded, etc.)
    - Option to view details of each submission.
- Functionality:
    - Fetch and display submissions from the database.
    - Provide detailed view for each submission.

Technical Requirements

Frontend:

- Use React or HTML/CSS/JavaScript.
- Ensure a responsive design for mobile and desktop views.
- Implement state management for handling form inputs and API responses.

Backend:

- Use Java with the SpringBoot framework.
- Create RESTful API endpoints for:
    - User authentication and authorization.
    - CRUD operations for assignments.
    - Fetching submissions and assignment details.
    - Dashboard data aggregation.
- Implement data validation and error handling.

Database:

- Use MySQL or PostgreSQL.
- Design schemas for:
    - Users (fields: id, username, password, role, etc.)
    - Assignments (fields: id, title, description, due_date, class, created_by, etc.)
    - Submissions (fields: id, assignment_id, student_id, submission_text, attachment_url, submission_date, etc.)

Authentication:

- Implement user authentication and authorization using JWT or session-based authentication.
- Protect API endpoints to ensure that only authorized users can access certain functionalities.

Bonus Features (Optional):

- Email notifications for assignment deadlines and submissions.
- Role-based access control for advanced security.
- Real-time updates using WebSockets or similar technology.
- Integration with external services (e.g., Google Drive for file storage).

Evaluation Criteria:

1. Functionality: Does the application meet all the specified requirements?
2. Code Quality: Is the code well-organized, readable, and maintainable?
3. UI/UX Design: Is the user interface intuitive and visually appealing?
4. Innovation: Are there any additional features that enhance the application?
5. Presentation: How well is the solution presented and demonstrated?