

Homework Platform

Developing platform for managing homework assignments. The platform will serve students, teachers, and administrators, providing functionalities such as homework submission, grading, and administration. The project must include a backend with a database, a RESTful API, and optional frontend components. Integration of AI services for features like automatic grading or plagiarism detection is encouraged. Additionally, consider implementing microservices architecture for scalability and modularity.

User Roles and Permissions

I. Students

1. Description: Students are the primary users of the platform who submit homework assignments and view their grades.

2. Functions:

- Submit homework assignments through the platform.
- View and download homework assignments.
- Check their grades and feedback on submitted assignments.
- Each student could see only own grades from all assigned classes.

3. What Students Cannot Do:

- Create or modify homework assignments.
- View grades or submissions of other students.
- Perform administrative tasks such as user management or grade assignments.

II. Teachers:

1. Description: Teachers are responsible for creating, assigning, and grading homework assignments.

2. Functions:

- Create new homework assignments with instructions and deadlines.
- Assign homework to specific classes or groups of students.
- Grade submitted assignments and provide feedback.
- Each teacher can see all students and classes assigned to him/her.
- Group assignments by category (i.e. assignments for different classes)
- Create/print grading reports (i.e. printing the grades of all/selected students of his/hers assigned students)

3. What Teachers Cannot Do:

- View grades or submissions of students not assigned to their classes.
- Modify grades of assignments once they are submitted.
- Access personal information of students beyond what is necessary for grading purposes.

- Each teacher cannot see grades from not assigned classes or students.
- Create grading reports for students which are not assigned to him/her.

III. Administrators:

1. Description: Administrators have access to administrative functions and oversee the overall operation of the platform.

2. Functions:

- Manage user accounts, including registration, authentication, and role assignments.
- Monitor platform usage and activity.
- Modify grades.
- Assign students and teachers to classes.
- Delete users.
- Print grading reports for all/selected students

3. What Administrators Cannot Do:

- Interfere with the grading process or modify grades provided by teachers without proper authorization.
- Access personal information of students beyond what is necessary for administrative purposes.

Features

I. Features (Mandatory):

1. User Registration and Authentication:

- Some users should be able to register for an account and log in securely.
- Implement authentication mechanisms to ensure user data security.

2. Homework Submission:

- Students should be able to submit their homework assignments through the platform.
- Implement functionality to upload files or enter text for homework submission.

3. Homework Management:

- Teachers should be able to create, assign, and manage homework assignments.
- Include features for setting deadlines, grading criteria, and attachments.

4. User Roles and Permissions:

- Define different user roles (students, teachers, administrators) with appropriate permissions.
- Ensure that each role can access relevant functionalities while maintaining data security.

5. Database Integration:

- Utilize a database to store user data, homework assignments, user roles, etc.

- Implement database queries and optimizations for efficient data retrieval and storage.

6. RESTful API:

- Create RESTful API endpoints for user authentication, homework submission, and management.
- Ensure proper handling of HTTP methods and status codes for each endpoint.

7. Grading Report:

- Teachers: Generate grading reports for their assigned classes and students.
- Administrators: Generate comprehensive grading reports for all students or selected groups.

II. Features (Optional):

1. Frontend Interface:

- Develop a user-friendly frontend interface for interacting with the platform.
- Include features for browsing homework assignments, viewing grades, etc.

2. AI Integration for Grading:

- Integrate AI models for automatic grading of homework assignments.
- Utilize machine learning algorithms to analyze submitted work and provide feedback.

3. Plagiarism Detection:

- Implement a plagiarism detection system to identify copied content in submitted homework.
- Integrate third-party APIs or develop custom algorithms for plagiarism detection.

4. Batch Processing for Uploads:

- Allow users to upload multiple homework assignments in bulk.
- Implement a batch processing system to extract relevant information from uploaded files.

Architecture

I. Backend (Mandatory):

1. Use of Microservices:

- Implement a microservices architecture for the backend to enhance scalability and maintainability.
- Break down the application logic into smaller, independent services.

2. Database Integration:

- Utilize a database (e.g., MySQL, PostgreSQL) to store user data, homework assignments, etc.
- Implement database migrations and ensure data integrity and consistency.

3. RESTful API Development:

- Create RESTful API endpoints for user registration, authentication, homework submission, etc.
- Ensure proper request handling, error responses, and authentication mechanisms.

II. Frontend (Optional):

1. User Interface Design:

- Develop a responsive and intuitive frontend interface for interacting with the platform.
- Implement features for browsing homework assignments, viewing grades, etc.

2. Customizable Templates:

- Provide users with customizable templates for homework submission and viewing.
- Allow users to choose different layouts and designs for their assignments.

3. Preview Mode:

- Include a preview mode to allow users to preview their homework assignments before submission.
- Ensure that the preview accurately reflects the final appearance of the assignment.

REST Endpoints

These REST endpoints are guidance(examples) to what is needed in order to cover the functionalities required for user authentication, homework management, submission, grade management, user management, class management, and optional plagiarism detection. Each endpoint is designed to handle specific actions within the platform, ensuring proper separation of concerns and adherence to RESTful principles.

1. Authentication:

- /register (POST): Allows users to create an account.
- /login (POST): Allows users to log in to their account.
- /logout (GET): Allows users to log out of their account.

2. Homework Management:

- /homeworks (POST): Allows teachers to create a new homework assignment.
- /homeworks/{id} (GET): Allows users to view a specific homework assignment.
- /homeworks/{id} (PUT): Allows teachers to update a specific homework assignment.
- /homeworks/{id} (DELETE): Allows teachers to delete a specific homework assignment.

3. Homework Submission:

- /submit-homework (POST): Allows students to submit homework assignments.
- /submitted-homeworks/{id} (GET): Allows students to view their submitted homework assignments.

4. Grade Management:

- /grades/{id} (GET): Allows users to view the grade of a specific homework assignment.
- /grades/{id} (PUT): Allows teachers or administrators to update the grade of a specific homework assignment.
- /reports/generate (POST): Allows teachers or administrators to generate a new grading report based on specified criteria.

- /reports/{reportId} (GET): Allows users to view the status of a specific grading report.
- /reports (GET): Allows users to list all generated reports they have access to.

5. User Management:

- /users/{id} (GET): Allows users to view their own profile information.
- /users/{id} (PUT): Allows users to update their own profile information.
- /users/{id} (DELETE): Allows administrators to delete a specific user account.

6. Class Management:

- /classes (GET): Allows users to view all classes.
- /classes/{id} (GET): Allows users to view a specific class.
- /classes/{id}/students (GET): Allows teachers to view all students enrolled in a specific class.
- /classes/{id}/teachers (GET): Allows administrators to view all teachers assigned to a specific class.
- /classes/{id}/assign-student (POST): Allows teachers or administrators to assign a student to a specific class.
- /classes/{id}/assign-teacher (POST): Allows administrators to assign a teacher to a specific class.
- /classes/{id}/remove-student (POST): Allows teachers or administrators to remove a student from a specific class.
- /classes/{id}/remove-teacher (POST): Allows administrators to remove a teacher from a specific class.

7. Plagiarism Detection (Optional):

- /detect-plagiarism (POST): Allows users to check for plagiarism in submitted homework assignments.