EduTech-Assignment Tracker

Introduction

In the modern digital learning ecosystem, managing academic assignments efficiently is crucial for both educators and students. The EdTech Assignment Tracker is a simplified system designed to streamline the assignment lifecycle from creation and distribution by teachers to submission and tracking by students. This system addresses common challenges in education platforms such as missed deadlines, lack of centralized tracking, and limited communication around assignments.

The goal of this assignment tracker is to provide a minimal yet functional solution that supports key operations like posting assignments, submitting responses, and monitoring status updates. It leverages RESTful API design principles to enable smooth interaction between the front-end (user interface) and back-end (data and logic).

Objective

Design and implement a simplified assignment tracking system for an EdTech platform that allows teachers to post assignments and students to submit them.

System Architecture

This system is designed to facilitate seamless assignment management between teachers and students. It begins with user authentication, where both teachers and students can register and log in to their respective dashboards. Each user has a profile section that they can view and update as needed. Teachers are assigned specific classes, and students are linked to a particular class based on the information provided during registration. This class association forms the basis for assignment visibility and submission tracking.

Once logged in, teachers can create assignments targeted at specific classes. These assignments include details like the title, description, creation date, and due date. When an assignment is created for a class, it becomes automatically visible to all students in that class on their dashboards. Students can then view the assignment, complete it, and submit their work before the deadline. Each submission is linked to both the assignment and the submitting student, allowing for easy organization and retrieval.

Teachers can view all submissions for each assignment through their dashboards. They can download submitted files, review them, and provide feedback in the form of remarks or grades. This creates a full cycle of assignment distribution, submission, and evaluation within a single platform. The system ensures that only relevant assignments are shown to each user, and the data is securely managed through four core tables: Teacher, Student, Assignment, and Submission, enabling efficient data handling and user interaction.

Front End Technologies: HTML, CSS, Javascript.

Back End Technologies: FastAPI(Python).

Database: SQLite.

Core Entities and Relationship

1. Student Table

- Stores information about students like name, email, roll number, year, and class.
- Each student belongs to a class_name and can submit multiple assignments.
- Key fields: id, email, class_name.

2. Teacher Table

- Stores teacher details such as name, email, and employee ID.
- Each teacher can create multiple assignments.
- Key fields: id, email, emp_id.

3. Assignment Table

- Represents assignments created by teachers.
- Linked to a teacher_id (foreign key from teachers) and assigned to a particular class_name.
- Students from that class can see the assignment and submit it.
- Key fields: id, teacher_id, class_name, title.

4. Submission Table

- Stores student submissions for assignments.
- Linked to both assignment_id and student_id.
- Contains the file path of the submission and the submission time.
- Key fields: id, assignment_id, student_id.

Tabular Form Entities

Entity	Attributes
Student	id (PK), name, email, password, roll_number, year, class_name
Teacher	id (PK), name, email, password, emp_id
Assignment	id (PK), subject, class_name, title, file_path, assignment_type, created_at, teacher_id (FK)
Submission	id (PK), assignment_id (FK), student_id (FK), file_path, submitted_at

Tabular Form Entities Relationships

Relationship	Type	Connected Entities	Description
CREATES	1:N	Teacher → Assignment	A teacher can create multiple assignments.
SUBMITS	1:N	Student → Submission	A student can submit multiple assignments.
BELONGS_TO	M:1	Assignment → Teacher	Each assignment belongs to one teacher.

Relationship	Type	Connected Entities	Description
IS_FOR	M:N (via class)	Assignment ↔ Student	Students see assignments if class_name matches.
HAS_SUBMISSION	1:N	Assignment → Submission	One assignment can have many submissions.

API endpoints

1. Teacher Creates Assignment

This API endpoint allows a teacher to create a new assignment. It uses the POST method and is accessible via /api/assignments/. The teacher must provide the subject, class name for which the assignment is intended, title of the assignment, file path (which can be a file upload or a link), type of assignment (e.g., homework, project), and their own teacher ID to associate the assignment. Once the request is processed successfully, the server responds with a confirmation message and the newly created assignment ID.

2. Student Submits Assignment

This endpoint enables a student to submit an assignment and is accessed through the POST method at /api/submissions/. The student needs to provide the ID of the assignment they are submitting, their student ID, and the file path or file upload of the submission document. The system records the submission time automatically. If the submission is successful, the system returns a message confirming the submission along with a unique submission ID for reference.

3. Teacher Views Submissions

To allow a teacher to review all student submissions for a specific assignment, this GET endpoint is used at the path /api/assignments/{assignment_id}/submissions. Here, the teacher provides the assignment ID as a path parameter. The API responds with a list of all submissions related to that assignment, including details such as submission ID, student ID and name, file path, and submission timestamp. This allows teachers to access and review each student's work efficiently.

Authentication Strategy

Teacher Authentication Strategy: The authentication system for teachers ensures that each teacher signs up with essential credentials including their name, email ID, password, and a unique employee ID. The email serves as the unique identifier to prevent duplicate accounts. During the sign-up process, the system validates that the provided email doesn't already exist in the database. Once registered, a teacher can log in using their email and password. Upon successful login, they are redirected to a dedicated teacher dashboard where they can create assignments, view student submissions, and manage their profile. Authentication is typically handled using secure password hashing (e.g., bcrypt), and session or token-based methods (e.g., JWT) can be used to maintain login sessions securely.

Student Authentication Strategy: For students, the authentication process begins during sign-up, where they provide their name, email ID, password, class name, and year of joining. The system checks for the uniqueness of the email ID to avoid account duplication. Once registered, a student can log in using their email and password credentials. Upon successful authentication, the student is taken to a separate student dashboard where they can view assignments relevant to their class, submit assignments, and update their profile. Like with teachers, secure password handling is essential, and sessions or tokens are used to protect student sessions and authorize access to protected routes within the student interface.

Future Scope & System Scaling Suggestions

1. **Role-Based Access Control (RBAC):** Enhance the authentication system by implementing robust role-based access control. This would allow adding more roles in the future, such as administrators, parents, or academic coordinators, with customized permissions for each role.
2. **Multi-Class and Subject Support for Teachers:** Allow teachers to be associated with multiple classes or subjects. Currently, the system assumes one class per teacher, but

scaling this will support a real-world teaching environment where a teacher handles many subjects across multiple classes.

3. **Assignment Grading and Feedback System:** Add a structured grading and feedback system where teachers can give numerical grades, comments, or rubric-based assessments. Students could then view their performance history over time.

4. **Notification System:** Integrate email and in-app notifications for new assignments, upcoming deadlines, and submission confirmations. This keeps students and teachers informed in real time.

5. **File Storage with Cloud Integration:** Move uploaded assignments and submissions to cloud storage services like AWS S3, Google Cloud Storage, or Firebase. This improves scalability, performance, and file security.

6. **Mobile Application Support:** Develop Android and iOS apps using React Native or Flutter so students and teachers can manage assignments on the go.

7. **Scalable Architecture with Microservices:** Refactor the backend into microservices architecture and use containerization tools like Docker with orchestration via Kubernetes for high availability and horizontal scaling.

8. **Multi-Tenant Support:** Add support for multiple schools or educational institutions by designing the system as a multi-tenant platform, each with its own data and branding.