Online School

You are required to implement the following features for this project:

Functional Requirements

1. User Authentication and Authorization
    - User registration and login (students, instructors, administrators).
    - Password reset functionality.
    - Role-based access control to restrict access to specific features.
2. User Roles and Permissions
    - Administrator: Full access to manage users, courses,can add instructors..
    - Instructor: Ability to create, update, and delete their own courses, add assignments into courses.
    - Student: Ability to view and enroll in courses,submit assignments.
3. Course Management
    - CRUD operations (Create, Read, Update, Delete) for courses.
    - Course details including title, description, syllabus, and instructor.
    - One instructor in one course
4. Enrollment Management
    - Students can enroll in available courses.
    - Instructors can view the list of enrolled students.
5. Content Delivery
    - Instructors can upload and manage course materials (Images,descriptions,documents, etc.).
    - Students can access and download course materials.
6. Assignments and Assessments
    - Instructors can create and manage assignments (Simple forms).
    - Students can submit assignments.
7. Progress Tracking
    - Students can track their course progress.
    - Instructors can monitor student performance.

Technical Stack

1. Backend
    - Django: Web framework for the application.
    - Django Rest Framework (DRF): To build the RESTful API.
2. Database
    - PostgreSQL (or any other relational database).
3. Authentication
    - Django's built-in authentication system.

    - or, JWT or OAuth2 for token-based authentication.
4. Deployment
    - Render or any other cloud provider for deployment.

API Endpoints (Examples, Can be changed,added)

1. User Authentication
    - `POST /api/auth/register/`: Register a new user.
    - `POST /api/auth/login/`: Authenticate a user and return a token.
    - `POST /api/auth/logout/`: Logout a user.
2. Course Management
    - `GET /api/courses/`: List all courses.
    - `POST /api/courses/`: Create a new course.
    - `GET /api/courses/{id}/`: Retrieve a course by ID.
    - `PUT /api/courses/{id}/`: Update a course.
    - `DELETE /api/courses/{id}/`: Delete a course.
3. Enrollment Management
    - `POST /api/enrollments/`: Enroll in a course.
    - `GET /api/enrollments/`: List all enrollments for the authenticated user.
    - `DELETE /api/enrollments/{id}/`: Unenroll from a course.
4. Assignments
    - `POST /api/courses/{id}/assignments/`: Create an assignment for a course.
    - `GET /api/courses/{id}/assignments/`: List all assignments for a course.
    - `GET /api/assignments/{id}/`: Retrieve an assignment by ID.
    - `POST /api/assignments/{id}/submissions/`: Submit an assignment.

Implementation Steps

1. Set up a Django project and apps.
2. Configure Django Rest Framework.
3. Design database models for users, courses, enrollments, and assignments.
4. Implement authentication and authorization using DRF.
5. Develop API endpoints for all functionalities.
6. Create serializers and views for handling API requests and responses.
7. Deploy the application