Online School

> Django is my playground. Let’s create something remarkable...

Online School is an e-learning platform that offers courses to learners. It provides features such as course modules, quizzes, resources, and instructor interaction. Users can enroll in courses, learn, and track their progress.

System Overview

The *Online School* is a web-based e-learning platform where :

- **Teachers** can create and manage courses.
- **Students** can browse, purchase, and learn from these courses.
- **Admins** can oversee all activities, manage courses, users, and view platform statistics.

User Roles

Role	Description	Permissions
Admin	Platform manager	Full control over users, courses, and statistics
Teacher	Instructor	Create, update, and delete their own courses
Student	Learner	Browse, enroll, and access courses

1. Authentication Endpoints

This will handle user Registration, Login, and Token management

HTTP Method	Endpoints	Description
POST	**/auth/users/**	**Register a new user**
POST	**/auth/jwt/create**	**Login and obtain token**
POST	**/auth/logout/**	**Logout the current user**
POST	**/auth/jwt/refresh/**	**Refresh an expired JWT token**

2. Course Endpoints

Manage and Retrieve Courses

HTTP Method	Endpoints	Description
GET	**/api/v1/courses/**	**List of all course**
GET	**/api/v1/course/<id>/**	**Retrive a specific course**
GET	**/api/v1/course/?search=**	**Search course by Name**
GET	**/api/v1/course/?department=**	**Filter by department**

3. Teacher Endpoints

Manage and Retrieve Teacher

HTTP Method	Endpoint	Description
GET	/api/v1/teacher/courses/	List all courses created by the teacher
POST	/api/v1/teacher/courses/	Create a new course
PUT	/api/v1/teacher/courses/<id>/	Update a course created by the teacher
DELETE	/api/v1/teacher/courses/<id>/	Delete a course created by the teacher
GET	**/api/v1/teacher/courses/<id>/students**	View all students enrolled in a specific course

4. Student Endpoints

Manage and Retrieve Student

HTTP Method	Endpoint	Description
GET	/api/v1/student/enrollments/	View all courses enrolled by the student which payment done
POST	/api/v1/student/enroll/	Enroll in a course but not paid
POST	/api/v1/student/enroll/<enroll_id>/make_payment	Make Payment for a enroll course
GET	/api/v1/student/enrollments/<course_id>/progress	View progress for a specific course

5. Department Endpoints

Manage and Retrieve Department

HTTP Method	Endpoint	Description
GET	/api/v1/departments/	List all departments
POST	/api/v1/departments/	**(Admin)** Create a new department

HTTP Method	Endpoint	Description
PUT	/api/v1/departments/<id>/	**(Admin)** Update department
DELETE	/api/v1/departments/<id>/	**(Admin)** Delete department

6. User Profile Endpoints

Manage and Retrieve User Profile

HTTP Method	Endpoint	Description
GET	**/auth/users/me**	Retrieve the Login User and Update current user
POST	**/auth/users/**	Create a User (Teacher,Student)

7. Admin Dashboard Endpoints

Manage and Retrieve Admin

HTTP Method	Endpoint	Description
GET	/api/v1/admin/dashboard/	Admin dashboard overview (stats summary)
GET	/api/v1/admin/courses/	View all courses
POST	/api/v1/admin/courses/	Add a new course
PUT	/api/v1/admin/courses/<id>/	Update a course
DELETE	/api/v1/admin/courses/<id>/	Delete a course

Models

Here is a details plan of models, Including the field name, data types and relationships. This cover all entities and there interactions for online e-learning projects.

1. User Model

Represents all platform users (Admin, Teacher, Student)

Field Name	Data Type	Description
id	Primary Key	Unique identifier for each user
first_name	CharField	User’s first name
last_name	CharField	User’s last name
email	EmailField (Unique)	Used for login and email verification
address	TextField	User address
phone_number	CharField	User phone number
password	CharField	Hashed password for authentication
role	CharField (choices= [‘teacher’, ‘student’])	Defines user type
is_active	BooleanField	Activated after email verification
date_joined	DateTimeField	Timestamp when the user registered

Relationship:

- **One-to-Many:** A **Teacher** can create many **Courses**
- **One-to-Many:** A **Student** can have many **Enrollments**

2. Department Model

Represents academic departments (e.g., Computer Science, Business)

Field Name	Data Type	Description
id	Primary Key	Unique identifier for each department
name	CharField (Unique)	Department name
description	TextField	Description of the department

Relationship:

- **One-to-Many:** A **Department** can have many **Courses**

3. Course Model

Represents a course created by a teacher

Field Name	Data Type	Description
id	Primary Key	Unique identifier for each course
title	CharField	Course title
description	TextField	Detailed course description
department	ForeignKey(Department, on_delete=models.CASCADE)	Department the course belongs to
teacher	ForeignKey(User, on_delete=models.CASCADE, limit_choices_to={'role': 'teacher'})	The teacher who created the course
price	DecimalField	Course price (for purchase)
created_at	DateTimeField	When the course was created
updated_at	DateTimeField	Last updated timestamp
is_active	BooleanField	To deactivate a course if needed

Relationship:

- **Many-to-One:** Belongs to one **Department**
- **Many-to-One:** Created by one **Teacher**
- **One-to-Many:** Can have many **Enrollments**

4. Enrollment Model

Represents a student’s enrollment in a course

Field Name	Data Type	Description
id	Primary Key	Unique identifier for each enrollment
student	ForeignKey(User, on_delete=models.CASCADE, limit_choices_to={'role': 'student'})	The student who enrolled
course	ForeignKey(Course, on_delete=models.CASCADE)	The course the student enrolled in
enrolled_on	DateTimeField	Date and time of enrollment
progress	FloatField (default=0.0)	Progress percentage (0–100%)
is_completed	BooleanField (default=False)	Whether the student has completed the course

Relationship:

- **Many-to-One:** A student can enroll in many courses
- **Many-to-One:** A course can have many enrolled students

Role Summary

Admin

- Full control over **users**, **departments**, **courses**, **enrollments**
- Can view **system-wide statistics** (sales, top students, etc.)
- Responsible for **platform management**

Teacher

- Can **create, edit, delete** own courses
- Can **view enrolled students** and their progress
- Cannot modify others' courses
- Has limited access to analytics (only their courses)

Student

- Can **browse, search, and filter** courses
- Can **enroll, pay, and learn**
- Can **track their own progress**
- Cannot create or modify any course