Project Documentation

Title:Course Enrollment & Assignment System (Django + MongoDB)

2. Introduction

The Course Enrollment & Assignment System is a backend module built using **Django, Django REST Framework, JWT Authentication, and MongoDB**. The system provides user authentication, course management, user self-enrollment, and admin-controlled assignment of courses to multiple users.

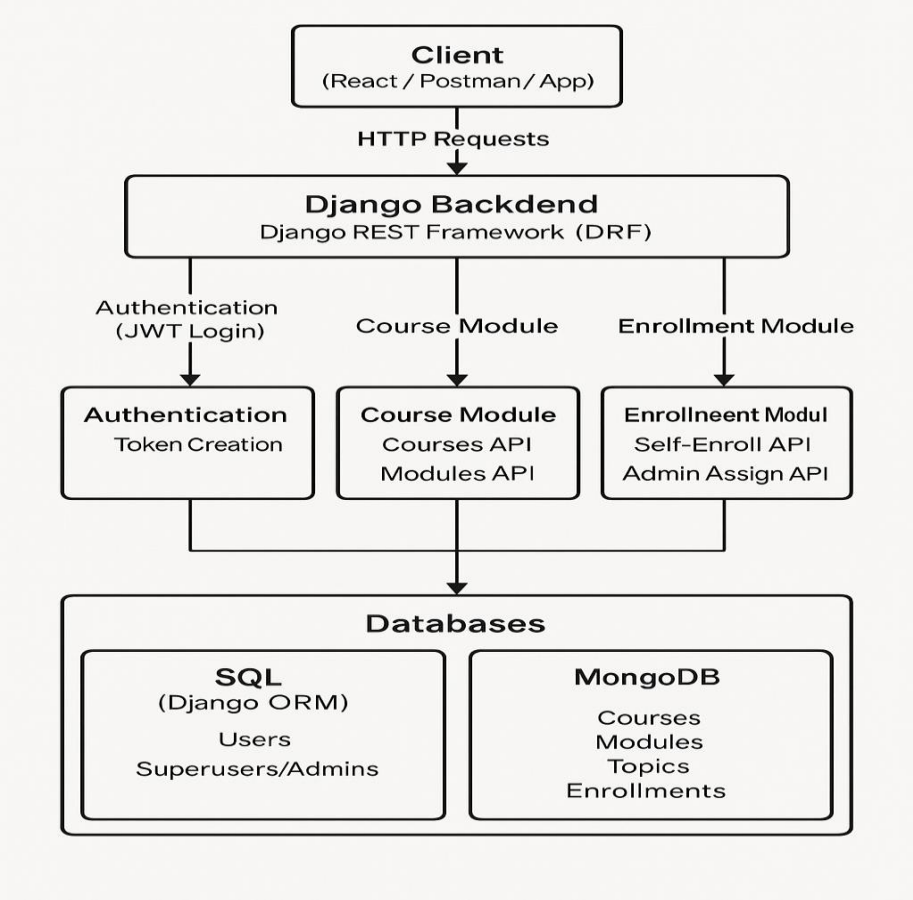
The goal of this project is to manage course data efficiently and provide a secure and scalable enrollment mechanism.

---

3. System Overview

This system integrates SQL + NoSQL databases:

- **Django User Model (SQL)** → Stores registered users



4. Technology Stack

Component	Technology
Backend	Django 5 + Django REST Framework
Authentication	JWT (Simple JWT)
Database (Users)	Django ORM (SQLite/PostgreSQL)
Database (Courses/Enrollments)	MongoDB
API Style	REST API
Tools	Postman, Python 3.11

---

5. Features

User Features

- Register and Login using JWT
- View available courses
- Self-enroll into a course
- View enrolled courses

Admin Features

- Assign a user to a course
- Assign **multiple users** to a course
- View all users and courses
- Control access based on permissions

---

6. Course & Enrollment Workflow

The system supports **two enrollment types**:

1. Self Enrollment (User)

A user logs in → sends POST /api/courses/<id>/enroll/ → added to course

2. Admin Assignment

Admin selects a course → selects one or many users → assigns them using:

- /assign/ (Single user)
- /assign-multiple/ (Multiple users)

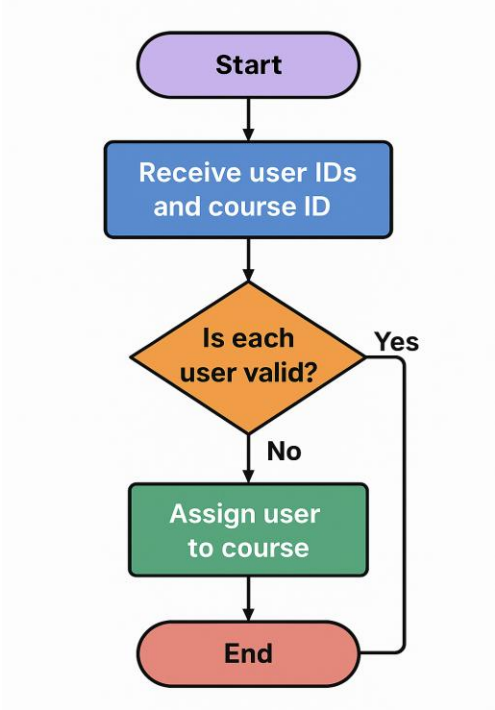
7. Detailed Enrollment Process

A. Self Enrollment Flow

1. User logs in via JWT
2. User hits enroll endpoint
3. Backend validates course
4. Enrollment document inserted in MongoDB
5. Course's enrollers count incremented
6. Username added to enrolled_users array

B. Admin Assignment Flow

1. Admin logs in (is_superuser = True)
2. Admin selects course
3. Admin assigns one/multiple users
4. Multiple enrollment documents created
5. Course updated accordingly



---

8. Database Design

The system uses **two database layers**:

8.1 Django User Database (SQL)

Stores registered users with fields:

- id
- username
- email
- phone
- password (hashed)
- is_superuser
- is_staff

8.2 MongoDB Collections (NoSQL)

courses

{

_id,

course_title,

course_description,

display_price: { amount, currency },

enrolled_users: [{ user_id, username }],

enrollers: <number>

}

enrollments{

_id,

user_id,

username,

course_id,

status,

enrolled_at

---

9. API Endpoints Documentation

Authentication

POST /api/auth/register/

POST /api/auth/login/

Course Management

GET /api/courses/

POST /api/courses/

GET /api/courses/<id>/

Enrollment

POST /api/courses/<id>/enroll/ (User self-enroll)

Admin Assign

POST /api/courses/<id>/assign/ (Assign single user)

POST /api/courses/<id>/assign-multiple/ (Assign multiple users)

---

10. Admin Assignment Module Explanation

The admin assignment feature allows admins to enroll users into courses without the users needing to enroll themselves.

Why this is needed?

- Useful for corporate onboarding
- Admin can bulk-assign new hires to required training
- Admin retains full control over course access
- Prevents duplicate enrollment logic
- Reduces user-side actions

How it works?

1. Admin sends list of user IDs
2. Backend validates each user
3. Creates enrollment documents in MongoDB
4. Updates course's enrollers & enrolled_users atomic operations
5. Returns assignment report

This module ensures speed, reliability, and avoids race conditions.

---

11. Conclusion

This Course Enrollment & Assignment backend provides a robust, scalable, and secure system for managing training workflows. The combination of Django + MongoDB enables flexible data storage, and JWT ensures secure access.
The admin assignment module makes the system enterprise-ready by supporting bulk operations and centralized control.