# EdTech Assignment Tracker – System Design

Prepared by: Ankit

## 1️⃣ System Architecture

Frontend:

- Simple HTML/CSS/JavaScript pages  
- Forms for assignment creation, submission, and viewing submissions

Backend (Django REST API):

- Django with REST Framework (DRF)  
- JWT Authentication using djangorestframework-simplejwt  
- SQLite as Database

Client ↔ API Server ↔ Database

## 2️⃣ Core Entities and Relationships

|  |  |  |
| --- | --- | --- |
| Entity | Attributes | Relationships |
| User | id, username, email, password, role (student/teacher) | A user can be a student or teacher |
| Assignment | id, title, description, due\_date, created\_by (teacher) | One assignment created by one teacher |
| Submission | id, assignment\_id, submitted\_by (student), content, submitted\_on | One student can submit one assignment |

## 3️⃣ API Endpoints

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint | Method | Role | Description |
| /api/signup/ | POST | All | Register as student or teacher |
| /api/login/ | POST | All | Login & get JWT token |
| /api/assignments/ | POST | Teacher | Create an assignment |
| /api/assignments/ | GET | Teacher | View all assignments (created by teacher) |
| /api/assignments/<id>/submissions/ | GET | Teacher | View all submissions for an assignment |
| /api/assignments/<id>/submit/ | POST | Student | Submit assignment by student |

## 4️⃣ Authentication Strategy

- JWT-based Authentication

- Role-based access control  
 - Student: Can signup/login, view assignments, submit assignments  
 - Teacher: Can signup/login, create assignments, view submissions

## 5️⃣ Scalability Suggestions

- Migrate from SQLite to PostgreSQL for production  
- Use Redis + Celery for background tasks (like file processing)  
- Deploy using Docker + Gunicorn + Nginx  
- Use AWS S3 or similar for file uploads  
- Implement pagination for API responses  
- Role-based permission classes at the API level

✅ Ready for Submission — Let me know if you want any more edits!