DevifyX

Django Job Assignment

Online Debate Platform for Students

Assignment Deadline: 7 Days

Assignment Submission Form: `https://forms.gle/17V2DA2P1VUR4nou5`

Objective

Design and implement the backend (API-focused) of an **Online Debate Platform for Students** using Django. The platform should enable students to participate in moderated debates, manage debate topics, and interact through a secure, scalable API.

Core Features

1. **User Registration and Authentication:** Implement secure student registration, login, and logout endpoints.
2. **User Roles:** Support at least two roles: *Student* and *Moderator*, each with appropriate permissions.
3. **Debate Topic Management:** Allow moderators to create, update, and delete debate topics.
4. **Debate Session Management:** Enable creation and scheduling of debate sessions linked to specific topics.
5. **Participant Management:** Allow students to join or leave scheduled debate sessions.
6. **Real-time Messaging:** Provide endpoints for posting and retrieving debate arguments/messages within a session.
7. **Moderation Tools:** Allow moderators to mute, remove, or warn participants during debates.
8. **Debate History:** Store and retrieve transcripts of past debate sessions for review.

Bonus Features

- **Voting System:** Allow students to vote for the best arguments or winning side after each debate.

- **User Profiles:** Implement extended profiles with statistics (e.g., debates participated, wins, etc.).
- **Notification System:** Notify users of upcoming debates, session changes, or moderation actions.
- **API Documentation:** Provide interactive API documentation (e.g., Swagger/OpenAPI).

Technical Requirements

- Use **Django** (latest stable version) and **Django REST Framework**.
- Database: **PostgreSQL** (preferred) or SQLite for local development.
- Implement authentication using **JWT** or Django's token authentication.
- Use **class-based views** and **serializers** for API endpoints.
- Follow RESTful API best practices.
- Write clear and maintainable code, following PEP8 standards.
- Use **Git** for version control; include a clear commit history.
- Include a `README.md` with setup instructions and API usage examples.

Deliverables

- Complete Django project source code (backend only).
- Database migration files and sample data for testing.
- `README.md` with setup, usage, and API documentation.
- (If implemented) API documentation (e.g., Swagger/OpenAPI schema).

Use of AI Tools

> You are **permitted and encouraged** to use AI-based coding tools such as **GitHub Copilot**, **ChatGPT**, or similar platforms to assist with code generation, debugging, and documentation. However, the final submission should reflect your own understanding and structure.

Submission

- Upload your code to a public GitHub repository.
- Submit the repository link via the assignment submission form: `https://forms.gle/17V2DA2P1VUR4nou5`
- Ensure the repository is accessible and includes all required files.

Evaluation Criteria

- **Feature Completion:** All core features are implemented and functional.
- **Code Quality:** Clean, modular, and well-documented code.
- **API Design:** RESTful, intuitive, and secure endpoints.
- **Security:** Proper authentication, authorization, and data validation.
- **Documentation:** Clear setup instructions and API usage examples.
- **Bonus Features:** Implementation of any bonus features will be considered.
- **Git Usage:** Meaningful commit messages and logical history.
- **Testing:** Presence of basic tests for critical endpoints (optional but recommended).

We look forward to your submission!
— **DevifyX Team**

Click here to read our Terms and Conditions