Volunteer Coordination Platform

Objective:

Develop a "Volunteer Coordination Platform" using Django, Django REST Framework, and PostgreSQL. This application will serve as a digital platform for non-profit organizations to post volunteer opportunities and for volunteers to sign up and manage their participation. The core of this task is to evaluate your proficiency in backend development with Django, understanding of RESTful API design, and implementation of Token-based authentication.

Requirements:

Project Setup and Structure:

1. Initialize a new Django project and configure it to use PostgreSQL as the database.
2. Use Django REST Framework for creating API endpoints.
3. Organize the project in a clean and modular architecture, separating concerns appropriately (e.g., models, serializers, views).

Core Features:

1. Organization and Opportunity Models:
    - Define models for organizations and volunteer opportunities.
    - Each opportunity should be linked to an organization and include fields like title, description, date, location, and required skills.
2. Opportunity API:
    - Create RESTful API endpoints to list, create, update, and delete volunteer opportunities.
    - Ensure these endpoints require authentication.
3. Organization Profile API:
    - Implement API endpoints to view and edit organization profiles.
    - Profiles should include the organization's name, mission, contact information, and a list of their volunteer opportunities.
4. Volunteer Registration and Participation Model and API:
    - Define a model for volunteer registrations, including fields like registration date, status, opportunity (linked to a volunteer opportunity), and volunteer details.
    - Create API endpoints to list, create, update, and delete volunteer registrations.
    - Ensure these endpoints require authentication.
5. User Authentication:
    - Implement a custom user model and use Django REST Framework to create a login API.
    - Authentication should be based on tokens, returning an access token and a refresh token upon successful login.
6. Registration API:
    - Allow new users (volunteers) to register through an API endpoint, creating a new user profile.

Authentication and Permissions:

1. Utilize Token for managing user sessions and securing API endpoints.
2. Ensure that only authenticated users can create, update, or delete volunteer opportunities and registrations.
3. Allow organizations to edit their own profiles and manage their opportunities only.
4. Allow volunteers to manage their own registrations.

Database Design:

1. Design the database schema with PostgreSQL, ensuring relationships between organizations, opportunities, and volunteer registrations are efficiently modeled.
2. Implement migrations for your database models.

Coding Practices and Documentation:

1. Write clean, modular, and reusable code.
2. Use Django best practices for models, views, serializers, and URL routing.
3. Document your API endpoints and provide a README file with instructions on how to set up and run your project, including database setup and any initial configuration steps.

Prohibited Use of AI Services:

The use of external AI services or assistance for code generation or problem-solving in this task is strictly prohibited. Your submission will be reviewed for originality and compliance with this rule.

Deliverables:

1. A GitHub repository containing the source code for the "Volunteer Coordination Platform".
2. Documentation covering setup instructions, API endpoint details, and a brief explanation of your project structure and design choices.
3. Unit test cases for your models and API endpoints. (Optional)