

## Local Food Community

### Objective:

Develop a "Local Food Community" platform using Django, Django REST Framework, and PostgreSQL. This application will serve as a digital platform for local chefs and food enthusiasts to share their recipes, culinary skills, and food-related events with the community. 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. Chef and Recipe Models:
  - Define models for chefs and recipes.
  - Each recipe should be linked to a chef and include fields like title, description, ingredients, instructions, creation date, and an image URL.
2. Recipe API:
  - Create RESTful API endpoints to list, create, update, and delete recipes.
  - Ensure these endpoints require authentication.
3. Chef Profile API:
  - Implement API endpoints to view and edit chef profiles.
  - Profiles should include the chef's name, bio, and a list of their recipes.
4. Event Model and API:
  - Define a model for food-related events, including fields like event name, description, date, location, and organizer (linked to a chef).
  - Create API endpoints to list, create, update, and delete events.
  - 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 (chefs) to register through an API endpoint, creating a new user and chef 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 recipes and events.
3. Allow chefs to edit their own profiles and manage their recipes and events only.

### **Database Design:**

1. Design the database schema with PostgreSQL, ensuring relationships between chefs, recipes, and events 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 "Local Food Community" 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)