

# Art Gallery

## Objective:

Develop a "**Art Gallery**" platform using Django, Django REST Framework, and PostgreSQL. This application will serve as a digital platform for local artists to share their artworks, including paintings, digital art, and sculptures, 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:

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

### Core Features:

- Artist and Artwork Models: Define models for artists and artworks. Each artwork should be linked to an artist and include fields like title, description, creation date, and an image URL.
- Artwork API: Create RESTful API endpoints to list, create, update, and delete artworks. Ensure these endpoints require authentication.
- Artist Profile API: Implement API endpoints to view and edit artist profiles. Profiles should include the artist's name, bio, and a list of their artworks.
- 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.
- Registration API: Allow new users (artists) to register through an API endpoint, creating a new user and artist profile.

### Authentication and Permissions:

- Utilize Token for managing user sessions and securing API endpoints.
- Ensure that only authenticated users can create, update, or delete artworks.
- Allow artists to edit their own profiles and manage their artworks only.

### Database Design:

- Design the database schema with PostgreSQL, ensuring relationships between artists and artworks are efficiently modeled.
- Implement migrations for your database models.

**Coding Practices and Documentation:**

- Write clean, modular, and reusable code.
- Use Django best practices for models, views, serializers, and URL routing.
- 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:**

- A GitHub repository containing the source code for the "Local Artwork Showcase" platform.
- Documentation covering setup instructions, API endpoint details, and a brief explanation of your project structure and design choices.
- Unit test cases for your models and API endpoints. (Optional)