README for Setting Up Django, PostgreSQL, and REST API (with Detailed PostgreSQL Setup and Folder Structure)

Folder Structure

After setting up your Django project, the folder structure will look like this:

```
project_name/         # Root project directory
│
├── manage.py         # Django's command-line utility for administrative tasks
├── db.sqlite3        # Default SQLite database (can be ignored if using
PostgreSQL)
├── requirements.txt  # List of Python dependencies (if generated)
├── app_name/         # Your Django app
│   ├── migrations/   # Database migrations folder
│   ├── __init__.py   # Marks the directory as a Python package
│   ├── admin.py      # Django admin configuration
│   ├── apps.py       # App configuration
│   ├── models.py     # Database models
│   ├── tests.py      # Test cases for the app
│   ├── views.py      # Views for handling requests
│   └── urls.py       # URL routing for the app
│
├── project_name/     # The project configuration directory
│   ├── __init__.py   # Marks the directory as a Python package
│   ├── asgi.py       # ASGI configuration for asynchronous deployment
│   ├── settings.py   # Project settings (database, installed apps, etc.)
│   ├── urls.py       # URL routing for the project
│   └── wsgi.py       # WSGI configuration for production deployment
│
└── env/              # Virtual environment directory (if created locally)
```

Step 1: Install Prerequisites

Ensure you have the following installed:

- Python (version 3.10 or higher)
- PostgreSQL (version 12 or higher)
- Pip (Python package manager)
- Virtualenv (optional but recommended)

Step 2: Install and Configure PostgreSQL

1. **Download and Install PostgreSQL:**
    - Download PostgreSQL from the official website.

- Follow the installation steps provided for your operating system.

2. **Set Password for `postgres` User During Installation:**
   - During installation, you'll be prompted to set a password for the default superuser `postgres`. Choose a strong password and **remember it**, as you'll need it to configure your Django project.
   - Example: Set the password to `mypassword` (replace this with your actual password).
3. **Verify Installation:**
   - After installation, open the PostgreSQL terminal (psql) or pgAdmin to ensure it's installed correctly.
   - Access the database using the command line:

     ```
     psql -U postgres
     ```

   - Enter the password you set during installation when prompted.
4. **Create a New Database:**
   - Log in to the PostgreSQL terminal (psql) and run:

     ```
     CREATE DATABASE your_database_name;
     ```

5. **Create a New User:**
   - Create a new user with a password:

     ```
     CREATE USER your_username WITH PASSWORD 'your_password';
     ```

6. **Grant Permissions:**
   - Grant the new user all privileges on the newly created database:

     ```
     GRANT ALL PRIVILEGES ON DATABASE your_database_name TO your_username;
     ```

7. **Exit PostgreSQL:**

   ```
   \q
   ```

Step 3: Set Up Django

1. **Create a Django Project:**
   - Install Django:

```
pip install django
```

   - Create a new project:

```
django-admin startproject project_name
```

2. **Add a New App:**
   - Inside your project folder, create a new app:

```
python manage.py startapp app_name
```

Step 4: Configure Django for PostgreSQL

1. **Update `settings.py`:**
   - Open the `settings.py` file in your Django project and update the `DATABASES` setting:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'your_database_name',  # Replace with the name of your
database
        'USER': 'your_username',       # Replace with your PostgreSQL
username
        'PASSWORD': 'your_password',   # Replace with your PostgreSQL
password
        'HOST': 'localhost',           # Default host for PostgreSQL
        'PORT': '5432',                # Default PostgreSQL port
    }
}
```

2. **Apply Migrations:**
   - Run migrations to initialize the database:

```
python manage.py makemigrations
python manage.py migrate
```

Step 5: Add REST API

1. **Install Django REST framework:**

```
pip install djangorestframework
```

2. **Add `'rest_framework'` to `INSTALLED_APPS`:**

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',  # Add this line
]
```

3. **Create a Simple API Endpoint:**

   - `views.py`:

```
from django.http import JsonResponse

def api_test(request):
    return JsonResponse({"message": "Hello, API!"})
```

   - `urls.py`:

```
from django.urls import path
from .views import api_test

urlpatterns = [
    path('api/test/', api_test, name='api_test'),
]
```

4. **Run the Server:**

   - Start the development server:

```
python manage.py runserver
```

5. **Test the API:**
    - Visit the API endpoint in your browser:

```
http://127.0.0.1:8000/api/test/
```

Recap of PostgreSQL Setup

1. **Install PostgreSQL** and set the password for the `postgres` user during installation.
2. **Create a database and user** using the PostgreSQL terminal:

```
CREATE DATABASE your_database_name;
CREATE USER your_username WITH PASSWORD 'your_password';
GRANT ALL PRIVILEGES ON DATABASE your_database_name TO your_username;
```

3. **Update Django settings** with the database credentials:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'your_database_name',
        'USER': 'your_username',
        'PASSWORD': 'your_password',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

FounderMatching

Access GitOps Guidelines here (VinUni credentials required).