Ticket Tracking System (TTS)

Migrating from SQLite3 to PostgreSQL

To migrate your Django project from SQLite3 to PostgreSQL for production, follow these steps:

1. Install PostgreSQL

Ensure PostgreSQL is installed on your system. You can install it using the package manager for your operating system:

- **Ubuntu/Debian**:

  ```
  sudo apt update
  sudo apt install postgresql postgresql-contrib
  ```

- **MacOS (Homebrew)**:

  ```
  brew install postgresql
  ```

- **Windows**: Download and install PostgreSQL from the official website: https://www.postgresql.org/download/

2. Create a PostgreSQL User for Django

After installing PostgreSQL, create a new user for your Django project:

1. Switch to the PostgreSQL user:

   ```
   sudo -i -u postgres
   ```

2. Open the PostgreSQL shell:

   ```
   psql
   ```

3. Create a user (e.g., `django_user`) with a password:

   ```sql
   CREATE USER django_user WITH PASSWORD 'your_password';
   ```

4. Grant the user permission to create databases:

   ```
   ALTER USER django_user CREATEDB;
   ```

5. Exit the PostgreSQL shell:

   ```
   \q
   ```

3. Create a PostgreSQL Database for Django

Now, create a database owned by the new PostgreSQL user:

1. Log back into the PostgreSQL shell if needed:

   ```
   psql -U postgres
   ```

2. Create the database and assign ownership to `django_user`:

   ```
   CREATE DATABASE django_db OWNER django_user;
   ```

3. Exit the PostgreSQL shell:

   ```
   \q
   ```

4. Update `settings.py` for PostgreSQL

Update your Django project's `settings.py` file to configure PostgreSQL:

```
# settings.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'django_db',
        'USER': 'django_user',
        'PASSWORD': 'your_password',
        'HOST': 'localhost',
        'PORT': '5432',  # Default PostgreSQL port
    }
}
```

5. Install PostgreSQL Dependencies

Ensure the necessary PostgreSQL driver is installed for Django:

```
pip install psycopg2
```

6. Apply Migrations

Now that your PostgreSQL database is configured, apply the migrations to create the necessary tables in the new database:

```
python manage.py migrate
```

7. Migrate Data from SQLite3 to PostgreSQL

If you have existing data in your SQLite3 database, you can migrate it as follows:

1. Dump the data from the SQLite3 database into a JSON file:

   ```
   python manage.py dumpdata --natural-primary --natural-foreign > data.json
   ```

2. Apply the migrations for the new PostgreSQL database (if not done already):

   ```
   python manage.py migrate
   ```

3. Load the data into PostgreSQL:

   ```
   python manage.py loaddata data.json
   ```

8. Test the New Setup

Run the development server to ensure everything is working with PostgreSQL:

```
python manage.py runserver
```

Open your web browser and navigate to http://localhost:8000 to verify that the app is working properly with the PostgreSQL database.