Django Project Report: Little Lemon

Index

---

1. Introduction

The Little Lemon project is a Django-based web application that follows the Model-View-Template (MVT) architecture. This report documents the project structure, key components, and commands used in development.

---

2. Project Directory Structure

```
/Django-Project
|— db.sqlite3      # SQLite database (stores all project data)
|— manage.py       # Django command-line tool (runserver, migrations, etc.)
|— /littlelemon/   # Main Django project folder (Global settings & configurat
|— /restaurant/    # Django App (Handles restaurant-specific functionality)
```

Inside `/littlelemon/` (Project Configurations)

```
/littlelemon/
|— __init__.py    # Marks this as a Python package
|— asgi.py        # ASGI configuration (for async requests)
|— settings.py    # Core Django settings (database, apps, middleware)
|— urls.py        # Global URL routing (directs requests to apps)
|— wsgi.py        # WSGI configuration (for deployment)
```

Inside `/restaurant/` (Django App)

```
/restaurant/
|— __init__.py        # Marks this as a Python package
|— admin.py           # Django Admin panel configurations
|— apps.py            # App configuration
|— models.py          # Defines database structure (tables)
|— tests.py           # Unit tests
|— urls.py            # App-specific URL routing
|— views.py           # Handles request logic (what happens when a user visits a page)
|— /migrations/       # Tracks database changes
|— /templates/restaurant/
|       |— home.html  # Template for the homepage
```

---

3. Explanation of Each Component

1. `db.sqlite3`

- Stores the project's database.
- SQLite is used by default in Django.

2. `manage.py`

- The command-line tool to interact with the Django project.
- Used for running the server, migrations, and more.

3. `/littlelemon/` (Main Project Directory)

File	Purpose
`settings.py`	Configures database, installed apps, middleware, templates, and static files.
`urls.py`	Routes URLs to views.
`asgi.py`	Handles ASGI deployments.
`wsgi.py`	Handles WSGI deployments.

4. `/restaurant/` (App Directory)

File	Purpose
`models.py`	Defines database tables using Django ORM.
`views.py`	Contains Python functions to handle user requests.
`urls.py`	Defines app-specific URL patterns.
`templates/`	Stores HTML templates for rendering views.

---

4. Django Commands Used

Below are the commands executed during development:

Project Setup Commands

```
# Create Django Project
django-admin startproject littlelemon

# Create Django App
python manage.py startapp restaurant
```

Database Migration Commands

```
# Create migration files
python manage.py makemigrations

# Apply migrations
python manage.py migrate
```

Run Development Server

```
python manage.py runserver
```

Creating a Superuser for Admin Panel

```
python manage.py createsuperuser
```

Check for Issues

```
python manage.py check
```

---

5. Key Concepts & Best Practices

1. Model-View-Template (MVT) Architecture

- **Model**: Defines database structure (`models.py`).
- **View**: Handles business logic (`views.py`).
- **Template**: Renders the frontend (`home.html`).

2. Virtual Environment (`venv`)

- A separate Python environment for dependencies.
- Created using:

```
python -m venv venv
```

3. URL Routing

- **Project URLs (`littlelemon/urls.py`)** include app URLs.
- **App URLs (`restaurant/urls.py`)** handle app-specific routes.

---

6. Conclusion

The Little Lemon Django project is structured according to Django best practices, following the MVT pattern. The use of virtual environments, migrations, and Django commands ensures an organized and scalable web application.