To-Do List Application - Documentation

Developer: R. A. Y. Indeewari

Submission Date: February 27, 2025

Company: FITI

1. Project Overview

This is a simple To-Do List application built using Django, providing users with task management features such as adding, updating, and deleting tasks. The application supports user authentication, ensuring that each user's tasks are private.

2. Technologies Used

- **Frontend:** HTML, CSS, Bootstrap
- **Backend:** Django (Python)
- **Database:** SQLite
- **Authentication:** Django's built-in authentication system
- **Framework Used:** Django

3. Features

- User authentication (Signup & Login)
- Add, update, and delete tasks
- Each user has a private task list
- UI designed using Bootstrap for responsiveness
- A database to store users and their tasks

4. Setup Instructions

Step 1: Install Python & Virtual Environment

Ensure Python is installed. If not, download it from [Python's official website](#).

Then, create a virtual environment:

```
python -m venv venv
```

Activate it:

- **Windows:**
  - venv\Scripts\activate
- **Linux/Mac:**
  - source venv/bin/activate

Step 2: Install Dependencies

Navigate to the project folder and install required packages:

pip install -r requirements.txt

Step 3: Apply Migrations

Run the following commands to set up the database:

python manage.py makemigrations
python manage.py migrate

Step 4: Create a Superuser (Optional - Admin Access)

python manage.py createsuperuser

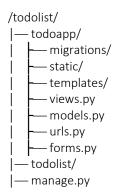Follow the prompts to set up an admin account.

Step 5: Run the Server

Start the development server:

python manage.py runserver

The application will be available at:
http://127.0.0.1:8000/

5. Folder Structure

```
/todolist/
|— todoapp/
|   |— migrations/
|   |— static/
|   |— templates/
|   |— views.py
|   |— models.py
|   |— urls.py
|   |— forms.py
|— todolist/
|— manage.py
```

|— db.sqlite3
|— requirements.txt
|— README.md

6. Known Issues & Fixes

- **Ensure all dependencies are installed** before running the project (pip install -r requirements.txt).
- **Django database migrations must be applied** using makemigrations and migrate commands.
- If Reverse for 'delete' not found occurs, check the urls.py to ensure the correct path name is used.