Todo Web Application - Documentation

Project Overview:

The Todo Web Application is a simple and efficient task management platform built with Flask.

It allows users to create, update, delete, and mark tasks as completed. The application uses SQLite for persistent storage, ensuring that tasks are saved even after the application is closed. The goal of this project is to provide an intuitive interface for managing tasks and demonstrate my skills in full-stack web development.

Features:

- Task Management: Users can add, edit, delete, and mark tasks as completed.
- Persistent Data Storage: All tasks are stored in an SQLite database using Flask-SQLAlchemy.
- Responsive Design: The user interface adapts to various screen sizes, making it mobile-friendly.
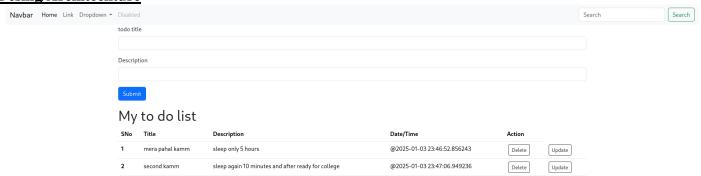- Production-Ready: The app is configured with Gunicorn for smooth deployment in production.

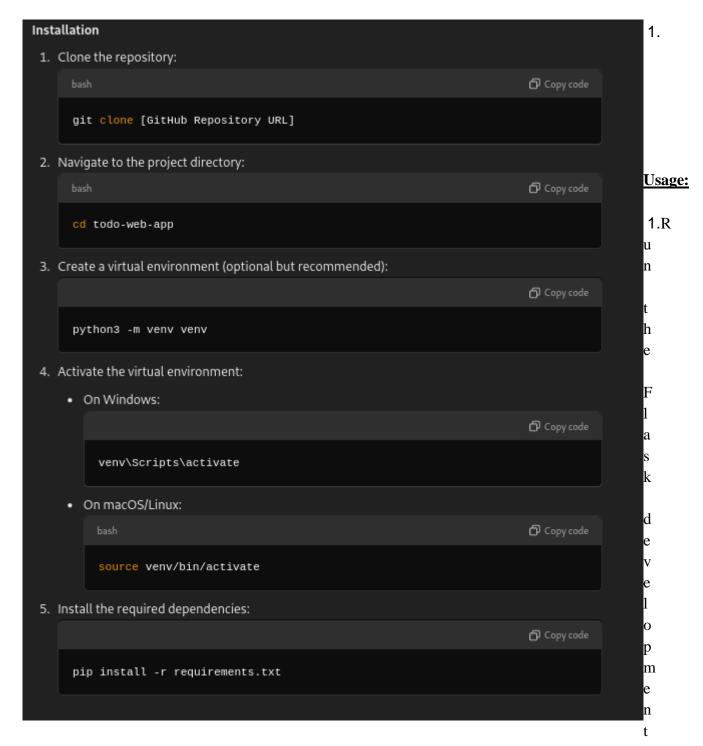## Tech Stack:
- Backend: Flask (Python)
- Frontend: HTML, CSS, JavaScript
- Database: SQLite with Flask-SQLAlchemy
- Deployment: Gunicorn for production

Desing/Architechture

SNo	Title	Description	Date/Time	Action	
1	mera pahal kamm	sleep only 5 hours	@2025-01-03 23:46:52.856243	Delete	Update
2	second kamm	sleep again 10 minutes and after ready for college	@2025-01-03 23:47:06.949236	Delete	Update

Installation:

server: flask run
The app will be accessible at http://127.0.0.1:5000.

2.For production deployment, use Gunicorn:
gunicorn -w 4 -b 127.0.0.1:8000 app:app
This will run the app with 4 worker processes on port 8000.

Project Structure:

Project Structure

```
todo-web-app/
|
├── app.py                  # Main application file
├── models.py               # Database models (Flask-SQLAlchemy)
├── templates/              # HTML templates for rendering views
|   ├── index.html          # Home page template
|   ├── layout.html         # Base template for consistent design
|
├── static/                 # Static files (CSS, JS)
|   ├── style.css           # Stylesheet for the app
|   ├── script.js           # JavaScript for interactivity
|
├── requirements.txt        # List of dependencies for the project
├── README.md               # Project documentation
```

How It Works:

1. Routes:
   - The main route (/) renders the homepage and displays a list of tasks.
   - Routes for adding, editing, deleting, and marking tasks as completed interact with the database.

2. Database:
   - SQLite stores tasks and their details (e.g., title, status).
   - Flask-SQLAlchemy is used to define the Task model and perform CRUD operations.

3. Frontend:
   - HTML templates define the structure of pages.
   - CSS ensures the app is visually appealing and responsive.

Future Enhancements:

- User Authentication: Add login/logout functionality for personal task management.
- Task Prioritization: Allow users to assign priority levels to tasks.
- Cloud Deployment: Deploy the app on platforms like Heroku or AWS for public access.

Conclusion:

The Todo Web Application is a foundational project for understanding full-stack web development.

It demonstrates the integration of Flask for backend, SQLite for database management, and a clean responsive frontend with HTML/CSS.

This project serves as a stepping stone for adding more advanced features like user authentication and cloud deployment.

Contact:

For any questions or contributions, feel free to reach out via the following:

- GitHub: **https://github.com/kamleshsande85/todo_Webapplication.git**
- Email: **Kamleshsande85@yahoo.com**