Checkie by AngPhie

A Simple To-Do App with Sticky Notes

Anggela F Syaloomitha Taek (001202300153) & Shofiyyah Abidah Marpaung (001202300174)

Overview

Checkie by AngPhie is a simple and playful web-based to-do list application featuring a sticky notes wall. Designed for individuals seeking a fun, visual, and minimalistic task management tool, this application enables the creation, editing, archiving, and deletion of tasks, as well as the addition of colorful sticky notes to capture quick thoughts or reminders in an engaging way.

The development of this application was divided into two main responsibilities: frontend and backend. The frontend was developed by Shofiyyah using HTML, CSS perfect for lightweight applications. The backend was handled by Anggela, using Express and Node.js to efficiently manage API endpoints and handle CRUD operations. JSON was chosen as the data storage format because the format is simple, easy to read, and suitable for small-scale development applications such as Checkie by Angphie. Docker was implemented to simplify deployment, ensuring consistent environments across development and production by containerizing both the frontend and backend services.

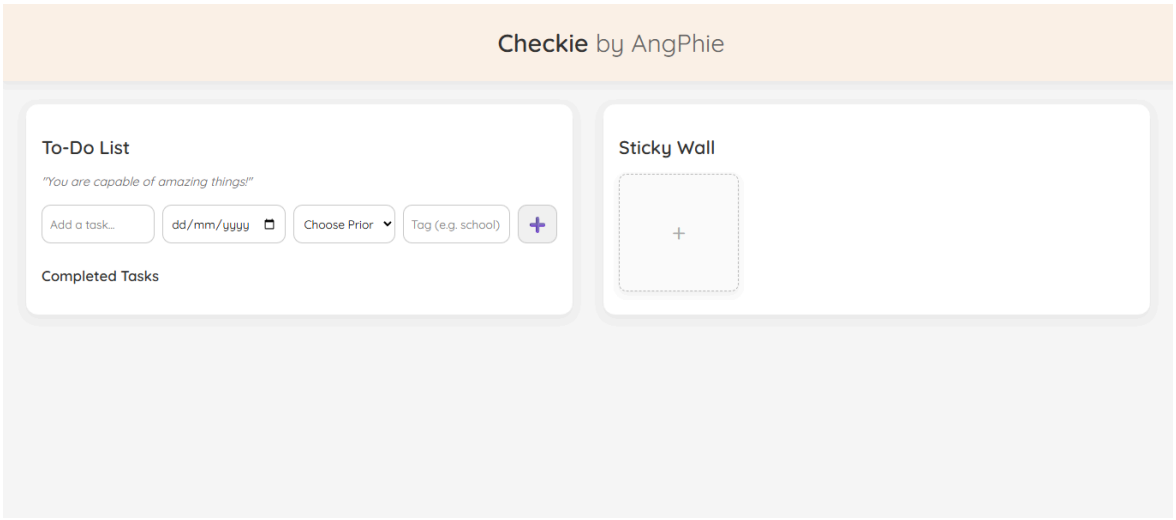
Tech Stack

- Frontend: HTML and CSS (by Shofiyyah)
- Backend: Node.js with Express.js (by Anggela)
- Data Storage: JSON
- Deployment: Docker (both)

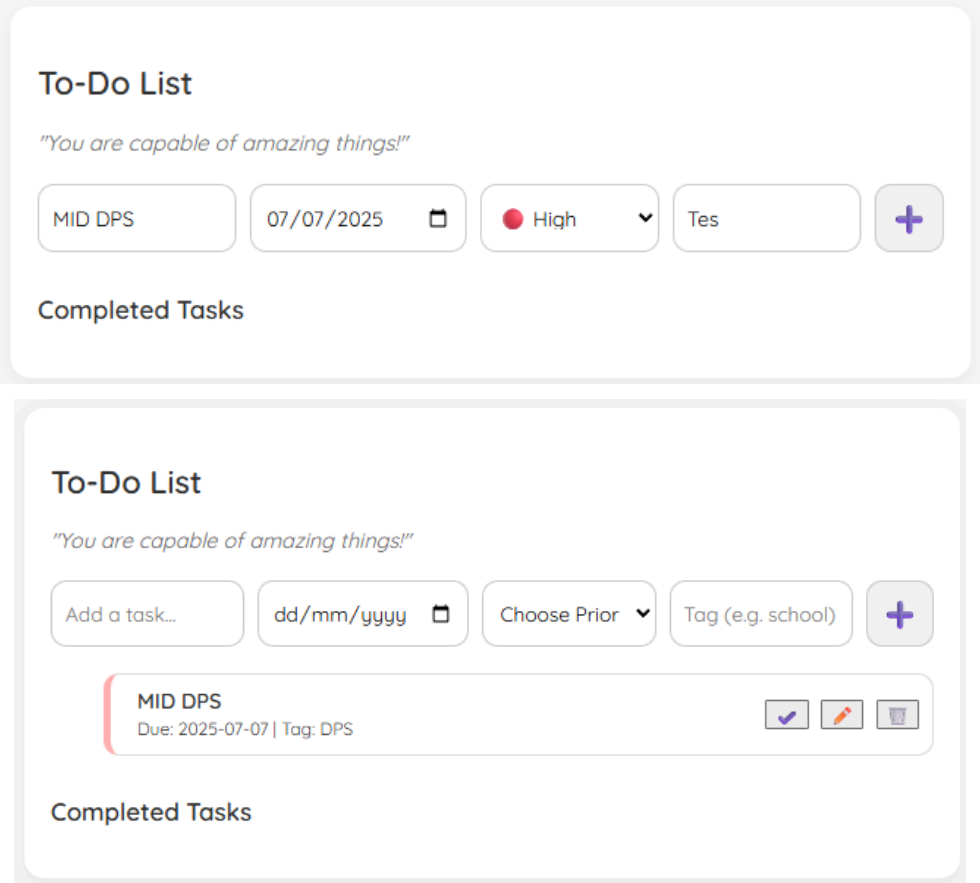
Features

- Add, edit, and delete tasks with due date, priority, and tags.
- Archive completed tasks.
- Add, edit, and delete sticky notes with random pastel colors.
- Clean and responsive UI design.

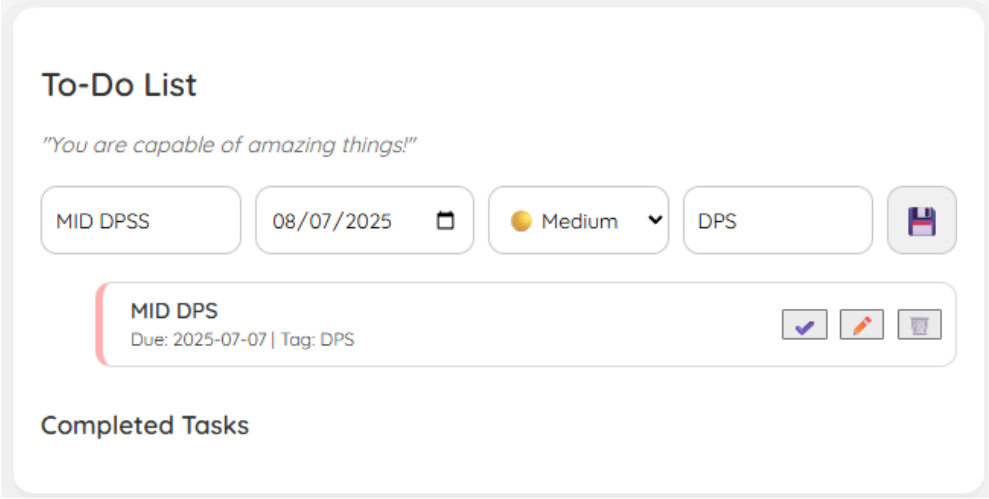
Documentation



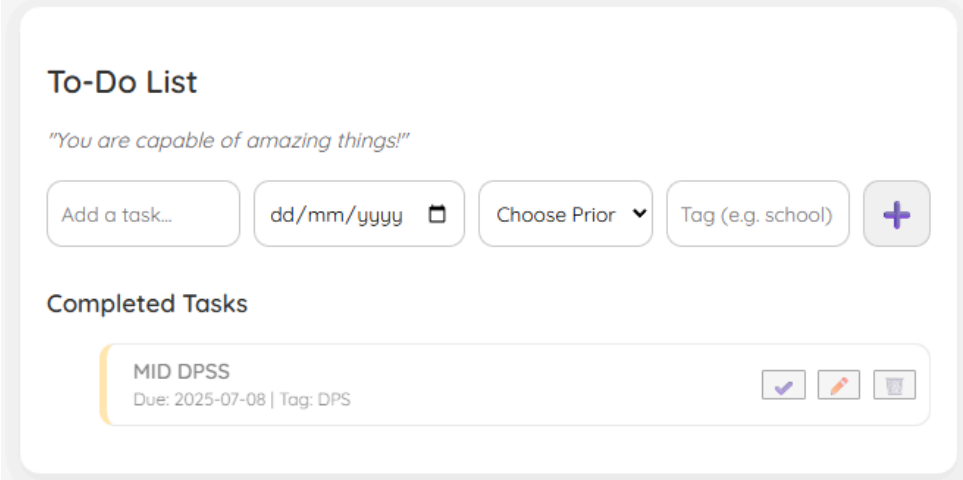
page overview



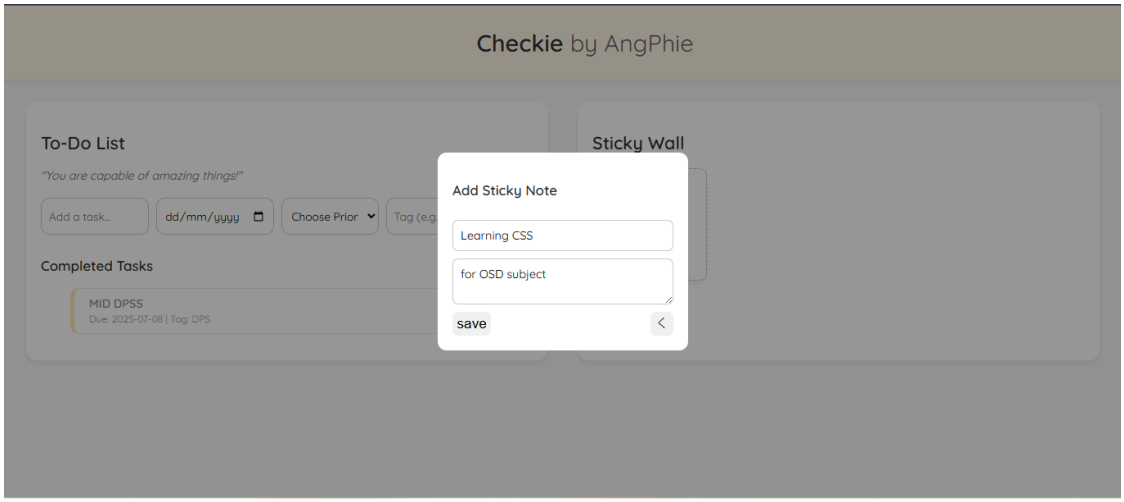
add a task



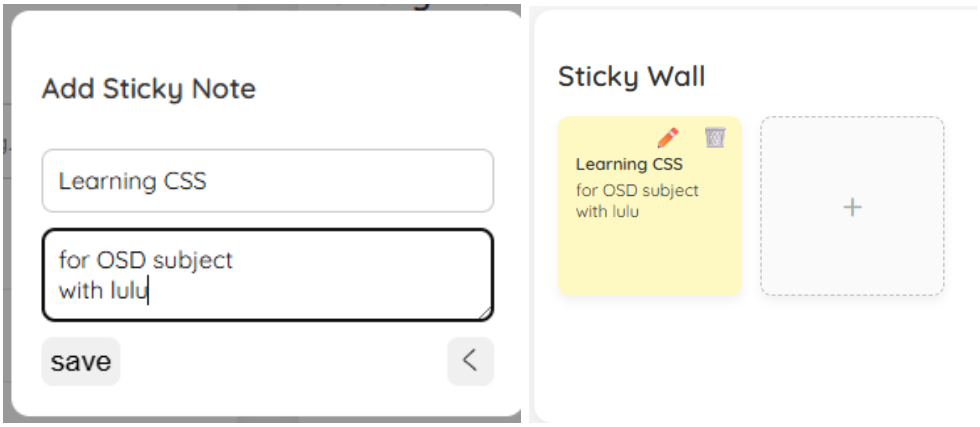
edit a task



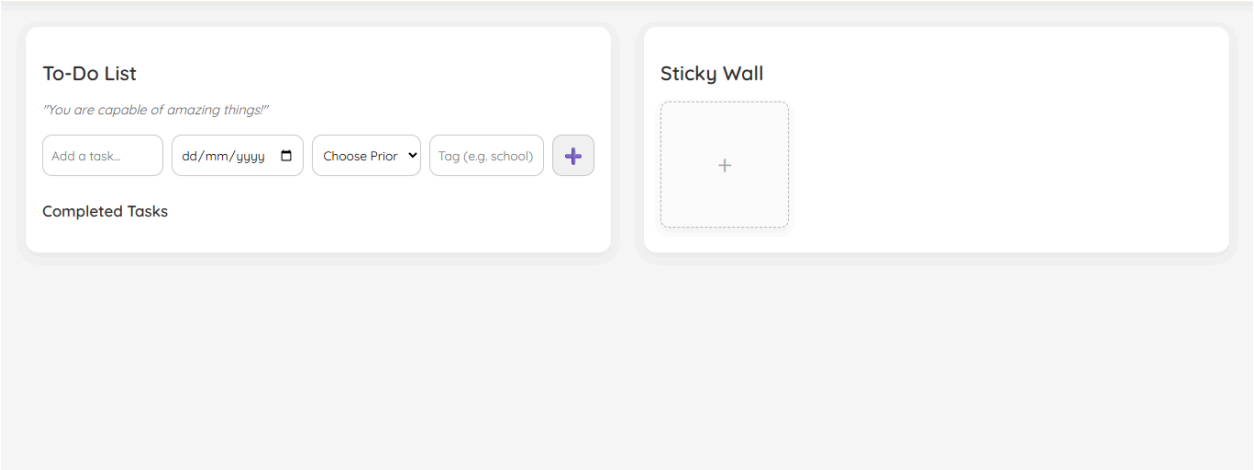
complete a task



add a sticky note



edit a sticky note



delete a task and a sticky note

Deployment with Docker

Step 1: Design the project structure for frontend and backend
Step 2: Create Dockerfile for frontend
Step 3: Create docker-compose.yml align with /frontend
Step 4: Define frontend service in docker-compose.yml
Step 5: Create Dockerfile for backend
Step 5: Define backend service in docker-compose.yml
Step 6: Build & run with docker-compose