Todo App Documentation

Table of Contents

1. Introduction

This Web App is a modern, responsive web application designed to help users manage their tasks efficiently. It provides a clean, intuitive interface for adding, completing, and deleting tasks, as well as basic task filtering capabilities.

2. Features

- Add new tasks with title, description, and due date/time
- Mark tasks as complete
- Delete individual tasks
- Clear all tasks
- Filter tasks (All/Active/Completed)
- Responsive design for various screen sizes
- Local storage for data persistence
- Modern, visually appealing UI with animations

3. Technology Stack

- HTML5
- CSS3
- JavaScript (ES6+)
- Bootstrap 5.3.3
- Font Awesome 6.4.0

4. Setup and Installation

1. Ensure you have a modern web browser installed.
2. No additional installation is required as the app uses CDN links for Bootstrap and Font Awesome.

5. Usage Guide

Adding a Task

1. Fill in the task title in the "Task title" field.
2. (Optional) Add a description in the "Task description" field.
3. Set the due date and time using the datetime-local input.
4. Click the "Add Task" button.

Completing a Task

- Click the check (✓) button next to a task to mark it as complete.

Deleting a Task

- Click the trash (🗑️) button next to a task to delete it.

Filtering Tasks

1. Click the "Filter" button.
2. Enter one of the following options when prompted:
    - "all": Show all tasks
    - "active": Show only uncompleted tasks
    - "completed": Show only completed tasks

Clearing All Tasks

- Click the "Clear All" button to remove all tasks (after confirmation).

6. Code Structure

HTML (index.html)

- Defines the structure of the app
- Includes necessary CSS and JavaScript files
- Contains a template for task items

CSS (style.css)

- Defines the visual styling of the app
- Includes responsive design rules
- Defines animations and transitions

JavaScript (script.js)

- Manages the app's functionality
- Handles task addition, completion, deletion, and filtering
- Manages local storage for data persistence

7. Customization

Changing Colors

- Modify the CSS variables in the `:root` selector in `style.css`.

Adding New Features

- Extend the JavaScript functions in `script.js`.
- Update the HTML structure in `index.html` as needed.

8. Troubleshooting

Tasks Not Saving

- Ensure local storage is enabled in your browser.
- Clear browser cache and reload the page.

Styling Issues

- Make sure all CSS files are properly linked.
- Check for any console errors related to CSS loading.

9. Future Enhancements

- Implement drag-and-drop for task reordering
- Add categories or tags for tasks
- Implement due date notifications
- Add user accounts and cloud synchronization
- Improve accessibility features