**To-Do List Application Documentation**

**1. Project Overview**

**1.1 Project Purpose**

The To-Do List Application is a simple yet effective tool designed to help users manage their daily tasks. It allows users to add, edit, and delete tasks while keeping track of tasks to be completed and tasks that have been completed. The application uses local storage to persist tasks across sessions, ensuring that user data is not lost upon refreshing the page.

**1.2 Key Features**

- Add Tasks: Users can add new tasks to their list.

- Complete Tasks: Users can mark tasks as completed using a radio button.

- Delete Tasks: Users can delete tasks from their list.

- Task Counters: The application shows the number of tasks to complete and completed tasks.

- Persistent Storage: Tasks are saved in the browser's local storage, making them available even after refreshing the page.

- Motivational Note: A motivational message is displayed when all tasks are completed.

**1.3 Technology Stack**

- HTML: For creating the structure of the application.

- CSS: For styling the application, with a focus on a purple-themed, aesthetic look.

- JavaScript: For functionality, including CRUD operations and local storage management.

**2. Setup Instructions**

**2.1 Prerequisites**

- A modern web browser (e.g., Chrome, Firefox, Edge).

- Basic understanding of HTML, CSS, and JavaScript (optional, for customization).

**2.2 Installation**

1. Download the Project:

- Clone the repository or download the project ZIP file from [your repository link] and extract it to your desired location.

2. Open the Project:

- Open the `index.html` file in your web browser to view and interact with the application.

**2.3 File Structure**

|-- index.html # Main HTML file

|-- style.css # CSS for styling the application

|-- script.js # JavaScript for application functionality

|-- README.md # Project documentation

**3. Detailed Code Explanation**

**3.1 HTML Structure (`index.html`)**

The HTML file forms the backbone of the application, containing all the elements needed for the to-do list.

- Container Div: Wraps the entire application to center it on the page.

- Input Container: Contains the task input field and the "Add Task" button.

- Task List: An unordered list (`<ul>`) where tasks will be dynamically inserted by JavaScript.

- Task Summary: Displays the "Tasks to Complete" and "Completed Tasks" counters.

- Motivation Note: A message area that displays motivational text when all tasks are completed.

**3.2 CSS Styling (`style.css`)**

The CSS file is responsible for the visual appearance of the application, focusing on a purple-themed aesthetic.

- Background & Font Colors: Various shades of purple are used for the background, buttons, and text to maintain a consistent and visually pleasing design.

- Task Item Styles:Styling for individual task items, including hover effects and a strikethrough for completed tasks.

- Responsiveness: The layout is simple and responsive, ensuring usability across different screen sizes.

**3.3 JavaScript Functionality (`script.js`)**

The JavaScript file handles the core functionality of the application, including task management and local storage operations.

- Task Storage: Uses the browser's local storage to save tasks, ensuring persistence across sessions.

- Task Management Functions:

- Add Task: Adds a new task to the list and saves it in local storage.

- Toggle Task Completion: Marks a task as complete or incomplete, with a visual strikethrough for completed tasks.

- Delete Task: Removes a task from the list and updates local storage.

- Update Task Summary: Recalculates the number of tasks to complete and completed tasks, updating the display accordingly.

- Event Listeners: Respond to user interactions such as clicking the "Add Task" button or marking a task as complete.

**3.4 CRUD Operations**

The application implements all four CRUD operations:

- Create: Adding a new task.

- Read: Displaying tasks on the screen.

- Update: Marking tasks as complete or incomplete.

- Delete: Removing tasks from the list.

**4. Usage Guidelines**

**4.1 Adding a Task**

1. Type your task into the input field.

2. Click the "Add Task" button, or press "Enter."

3. Your task will appear in the task list.

**4.2 Marking a Task as Completed**

1. Click the radio button next to the task you want to mark as completed.

2. The task will be strikethrough to indicate completion, and the task counter will update.

**4.3 Deleting a Task**

1. Click the trash can emoji button next to the task you want to delete.

2. The task will be removed from the list, and the task counter will update.

**4.4 Viewing Task Statistics**

- The number of tasks to complete and completed tasks are displayed below the task list.

- Once all tasks are completed, a motivational message will appear.

**5. Customization**

**5.1 Changing Colors and Styles**

- Open the `style.css` file to change colors, fonts, and other styles. Adjust the CSS variables to suit your preferred color scheme.

**5.2 Adding New Features**

- You can expand the JavaScript file with additional features like task editing, deadlines, or priority levels.

**5.3 Extending Functionality**

- Consider adding a task search feature, filters (e.g., view only completed tasks), or integration with external APIs for task management.

**6. Troubleshooting & FAQs**

**6.1 Task List Does Not Persist After Refresh**

- Ensure that local storage is enabled in your browser. Check the console for any JavaScript errors.

**6.2 Task Completion Message Not Appearing**

- Make sure all tasks are marked as complete for the message to appear. If the problem persists, verify the logic in the `updateTaskSummary` function in `script.js`.

**6.3 Application Layout Issues**

- Check your browser's zoom settings and reset them to default. Ensure that you haven't modified the CSS in a way that disrupts the layout.

**7. Conclusion**

This To-Do List Application serves as a simple yet powerful tool for task management. With its easy-to-use interface, customizable design, and persistent storage, it is a perfect project for both personal use and as a learning exercise in web development.