

Requirements Document for To-Do Application

1. Introduction

This document outlines the requirements for a frontend web development project aimed at creating a simple to-do application. The application will be built using HTML, CSS, and JavaScript. The main functionalities include adding tasks, attaching details or notes to tasks, editing tasks, and deleting tasks.

2. Project Overview

2.1 Project Description

The objective of this project is to develop a user-friendly to-do application that allows users to manage their tasks efficiently. The application will enable users to add new tasks, attach additional information to tasks, edit existing tasks, and delete tasks they no longer need.

2.2 Project Scope

- Frontend development using HTML, CSS, and JavaScript.
- Creation of a simple to-do list application with CRUD (Create, Read, Update, Delete) functionalities.

3. Functional Requirements

3.1 User Interface

-

HTML Structure:

-

- Create a basic HTML structure to house the to-do application components.
- Include elements such as a header, main content area, and footer.

-

CSS Styling:

-

- Style the application using CSS to ensure a visually appealing and responsive design.
- Apply styles for different states of tasks (e.g., completed, active).

3.2 Task Management

-

Add Task:

- - Provide an input field for users to enter the task name.
 - Include a button to add the task to the list.
 - On clicking the add button, the task should appear in the task list.
-

Attach Details/Notes:

- - Allow users to attach additional details or notes to each task.
 - Provide a textarea or similar input field for entering notes.
-

Edit Task:

- - Allow users to edit the task name and attached details.
 - Include an edit button for each task.
 - Enable in-place editing or open a modal/dialog for editing.
-

Delete Task:

- - Provide a delete button for each task.
 - On clicking the delete button, the task should be removed from the list.

3.3 Task List

-

Display Tasks:

- - Display a list of tasks added by the user.
 - Show task name and attached details/notes.
-

Task States:

- - Differentiate between active and completed tasks.
 - Provide an option to mark tasks as completed.

3.4 Data Persistence

- **Local Storage:**
 - Use the browser's local storage to save tasks.
 - Ensure tasks persist even after the page is refreshed or the browser is closed.