



**SCHOOL OF
ENGINEERING**

DAYANANDA SAGAR UNIVERSITY

**Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING
(DATA SCIENCE)
PROJECT REPORT**

BY

**B K KEERTHANA – ENG23DS0058
AMBATI SAMEEKSHA – ENG23DS0053
LAKSHMI . R - ENG23DS0017
HARSHITHA. Y – ENG23DS0047
DHANU SHREE – ENG23DS0009
SANITHA – ENG23DS0030**

“TO DO LIST”

Under the supervision of

**Dr. Shaila S G
Professor & Chairperson,
Department of CSE(Data Science)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING(DATA SCIENCE),
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
(2023-2024)**

DECLARATION

We , B K Keerthana(ENG23DS0058),Ambati Sameeksha (ENG23DS0053),Lakshmi . R(ENG23DS0017),Harshitha .Y(ENG23DS0047),Dhanu Shree(ENG23DS0009),Sanitha (ENG23DS0030) are the student of third semester B.tech in Computer Science and Engineering (Data science) at School of Engineering Dayananda Sagar University ,here by declare the Major Project titled “**TO DO LIST**” has been carried out by us during the academic year 2024-2025.

Student

B K KEERTHANA – ENG23DS0058
AMBATI SAMEEKSHA – ENG23DS0053
LAKSHMI . R - ENG23DS0017
HARSHITHA. Y – ENG23DS0047
DHANU SHREE – ENG23DS0009
SANITHA – ENG23DS0030

DATE : 25 November 2024

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a many individual who have been responsible for the successful completion of this project work. First , we take this opportunity to express our sincere graditude to the school of Engineering & Technology , Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution .

We would like to thank Dr Udaya Kumar Reddy K R, Dean, School of Engineering & Technology ,Dayananda Sagar University for his constant encouragement and expert advice .

It is a matter of immense pleasure to express our sincere thanks to Dr Shaila S G, Department Chairperson ,Computer Science and Engineering (Data Science), Dayananda Sagar University , for provding the right academic guidance that made our task possible .

We would like to thank our guide Dr Shaila S G, Professor & Chairperson, Dept of Computer Science and Engineering (Data Science), for sparing her valuable time to extend help in every step of our project work , which paved the way for smooth progress and fruitful culmination of the project.

We would like to thank our project Coordinator prof Mahindra Gowda and all the staff Member of Computer Science and Engineering (Data Science) for their support .

We are also grateful to our family and friends who provided us with every requirement Throughout the course.

We would like to thank one and all who directly or indirectly helped us in the project work .

CERTIFICATE

This is to certify that the Major project word titled “**TO DO LIST**”
B K Keerthana(ENG23DS0058),Ambati Sameeksha
(ENG23DS0053),Lakshmi . R(ENG23DS0017),Harshitha
.Y(ENG23DS0047),Dhanu Shree(ENG23DS009),Sanitha
(ENG23DS0030)are 3rd semester b.tech in computer science and
engineerging (data science) , at school of Engineering Dayanada
University ,Banglore.

Mr.Mahindra Gowda	Dr. Shaila S G	Dr. Udaya Kumar Reddy K R
Assitant Professor Dept.of CSE (Data Science), School of Engineering Dayananda Sagar University	Professor & Chairperson Dept. of CS&E(Data Science), School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date: 22 NOVEMBER 2024	Date:22 NOVEMBER 2024	Date:22 NOVEMBER 2024

Name of the Examiner

- 1.
- 2.

Signature of Examiner

TABLE OF CONTENT

Introduction

Introduction

- 1.1 Background of the Project
- 1.2 Purpose and Objectives
- 1.3 Scope of the Project

Project Overview

- 2.1 Project Name
- 2.2 Project Description
- 2.3 Features
- 2.4 Target Audience
- 2.5 Technologies Used
- 2.6 Goals and Objectives

Technology Stack for Responsive To-Do List Website

- 3.1 Frontend Technologies

System Design and Architecture for Responsive To-Do List Website

- 4.1 Architecture Overview
- 4.2 System Flow Diagram

LIST OF ABBRIVATION

Abbreviation	Full Form
UI	User Interface
UX	User Experience
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
API	Application Programming Interface
DB	Database
SQL	Structured Query Language
CRUD	Create, Read, Update, Delete
HTTPS	Hypertext Transfer Protocol Secure
MVC	Model-View-Controller
JSON	JavaScript Object Notation
REST	Representational State Transfer
SSL	Secure Sockets Layer
SEO	Search Engine Optimization
CMS	Content Management System
QA	Quality Assurance
SSL/TLS	Secure Sockets Layer / Transport Layer Security
POS	Point of Sale
CDN	Content Delivery Network
HTTP	HyperText Transfer Protocol
SMTP	Simple Mail Transfer Protocol
AJAX	Asynchronous JavaScript and XML

Abbreviation	Full Form
PWA	Progressive Web App
UI/UX	User Interface/User Experience
GDPR	General Data Protection Regulation
OTP	One-Time Password
IDE	Integrated Development Environment
CI/CD	Continuous Integration/Continuous Deployment

ABSTRACT

This project demonstrates the creation of a simple, interactive To-Do List application using HTML, CSS, and JavaScript. The primary goal of the application is to allow users to add, remove, and manage tasks effectively in a user-friendly interface. The structure of the application is built with HTML, providing the basic layout for the input fields, buttons, and list display. CSS is used to style the interface, making it visually appealing and enhancing user experience with smooth transitions and dynamic feedback. JavaScript is employed to handle the core functionality of the application, such as adding new tasks, marking tasks as completed, deleting tasks, and saving the list in the browser's local storage to persist the data across page refreshes.

This To-Do List application is designed to be simple, yet functional, offering a basic but effective solution for task management. It serves as an ideal project for those learning the fundamentals of web development, including DOM manipulation, event handling, and basic JavaScript logic. By integrating these technologies, the project aims to provide a seamless and responsive experience, demonstrating the practical use of front-end development skills.

INTRODUCTION

In today's fast-paced world, effective task management is essential for personal productivity and efficiency. People often juggle multiple tasks, and keeping track of them can become overwhelming. Traditional methods, such as pen-and-paper lists or physical planners, can be limiting and less efficient, especially in the digital age.

1.1 Background of the Project

With the widespread use of smartphones and computers, digital task management tools have become more popular, offering users the ability to organize and manage tasks seamlessly. This **To-Do List** project is a simple, yet effective, solution for managing daily tasks, using basic web technologies (**HTML, CSS, and JavaScript**) to create an interactive, user-friendly interface.

1.2 Purpose and Objectives

The primary purpose of this To-Do List application is to create an intuitive, interactive platform for users to manage and track their tasks. By using this application, users will be able to:

- Add new tasks to a list with ease.
- Mark tasks as completed to indicate progress.
- Delete tasks that are no longer needed.
- Store tasks persistently using local storage, allowing tasks to remain available even after the page is refreshed.

The project aims to provide a simple, responsive interface that can be used by anyone, regardless of technical skill. The core objective is to help users manage their day-to-day responsibilities with an easy-to-use digital tool.

1.3 Scope of the Project

The scope of this project is focused on creating a basic yet functional To-Do List application with the following features:

- **Task Management:** Users can add, edit, mark as complete, and delete tasks.
- **Persistence:** Using browser local storage to store tasks, ensuring that they persist across page reloads.
- **User Interface:** A simple and clean layout that focuses on usability and user experience. The interface will be built using HTML for structure, styled with CSS, and powered by JavaScript for functionality.

Project Overview: Responsive To-Do List Website

2.1 Project Name:

Responsive To-Do List Website

2.2 Project Description:

The **Responsive To-Do List Website** is a dynamic web application designed to help users organize, track, and manage their daily tasks efficiently. The website utilizes **HTML, CSS, and JavaScript** to create an intuitive interface that is easy to use and can adapt to various screen sizes, ensuring seamless usability across devices like desktops, tablets, and smartphones.

This to-do list website allows users to:

- **Add new tasks** with a simple input interface.
- **Mark tasks as complete** or incomplete.
- **Delete tasks** once they are finished or no longer needed.
- **Save tasks persistently** using browser local storage, so the list remains intact even after page reloads.
- **View tasks** in an organized list, with clearly defined action buttons for task management.

The website is designed with **responsiveness** in mind, ensuring that users can access and use the application smoothly, regardless of the device they are on.

2.3 Features:

1. Task Management:

- Add, edit, and delete tasks.
- Mark tasks as completed with a visual indicator.

2. Responsive Design:

- The layout automatically adjusts to fit various screen sizes, from desktops to mobile devices.
- Simple, clean, and modern design for easy navigation.

3. Persistent Data:

- Tasks are stored in the browser's **local storage**, ensuring that the list remains intact between sessions.

4. **Interactive User Interface:**

- Real-time updates: Tasks are instantly added, completed, or deleted.
- User-friendly controls with clear buttons and intuitive actions.

5. **CSS Styling:**

- Stylish interface with a focus on minimalism and readability.
- Smooth animations and transitions for a pleasant user experience.

6. **JavaScript Functionality:**

- DOM manipulation to handle task interactions.
- Event listeners to add, complete, and delete tasks.
- Local storage implementation for persistence.

2.4 **Target Audience:**

This project is designed for individuals who need a simple and efficient way to manage tasks on a daily basis. Whether it's for work, study, or personal life, this website aims to improve productivity and task organization. The responsive design ensures that users can access and update their tasks on any device, making it convenient for people on the go.

2.5 **Technologies Used:**

- **HTML:** For the structure of the to-do list and task interface.
- **CSS:** For styling and ensuring the website is responsive and visually appealing.
- **JavaScript:** For handling the functionality, including adding tasks, marking them as completed, deleting tasks, and utilizing local storage.

2.6 **Goals and Objectives:**

1. **User Experience:** Create an easy-to-use platform that allows users to manage their tasks effectively.

2. **Responsiveness:** Ensure the website adapts to different screen sizes and works seamlessly on both desktop and mobile devices.
3. **Persistence:** Store tasks in local storage to maintain data between sessions and page reloads.
4. **Simplicity and Efficiency:** Focus on essential features to provide users with a straightforward and functional tool for task management.

This responsive to-do list website serves as both a practical tool for task management and a learning project for understanding web development concepts like DOM manipulation, event handling, and responsive design.

Technology Stack for Responsive To-Do List Website

The Responsive To-Do List Website is built using a combination of essential web technologies that ensure it is functional, user-friendly, and adaptable across different devices. Below is an overview of the primary technologies used in the project:

3.1 Frontend Technologies:

HTML (HyperText Markup Language)

- **Role:** HTML provides the foundational structure of the website. It is used to create the essential elements of the To-Do List, such as the input fields, buttons, and task display list.
- **Features:**
 - Creates the layout of the website with semantic tags.
 - Provides a clear, organized structure for tasks and interactive elements.

CSS (Cascading Style Sheets)

- **Role:** CSS is used for styling the website, making it visually appealing and responsive. It defines the look and feel of the website, including colors, fonts, layout, and spacing.

- **Features:**
 - **Responsive Design:** Using CSS media queries, the website adapts to various screen sizes, ensuring a smooth experience on desktops, tablets, and mobile devices.
 - **Animations & Transitions:** Adds smooth transitions for task actions (adding, completing, deleting) to enhance user experience.
 - **Flexbox/Grid Layout:** For flexible and responsive layouts to ensure proper alignment of elements across different screen sizes.

JavaScript

- **Role:** JavaScript is used to provide the dynamic functionality for the To-Do List. It enables interaction with the task items, such as adding, deleting, and marking tasks as completed.
- **Features:**
 - **DOM Manipulation:** Allows for real-time updates to the task list (e.g., adding new tasks, marking tasks as completed).
 - **Event Handling:** Handles user actions like button clicks (to add tasks or delete them) and user interactions.
 - **Local Storage:** Used for storing the task data in the browser, ensuring tasks persist even after the page is refreshed.

System Design and Architecture for Responsive To-Do List Website:

The design and architecture of the **Responsive To-Do List Website** focus on delivering an efficient, user-friendly, and scalable solution for task management. It is a **client-side application** that is built using fundamental web technologies like HTML, CSS, and JavaScript. Below is a breakdown of the system design and architecture:

4.1 Architecture Overview:

The architecture follows a **Model-View-Controller (MVC)** pattern, which is common for building web applications. However, it is implemented in a simplified form because the app is client-side only and does not involve a server-side component.

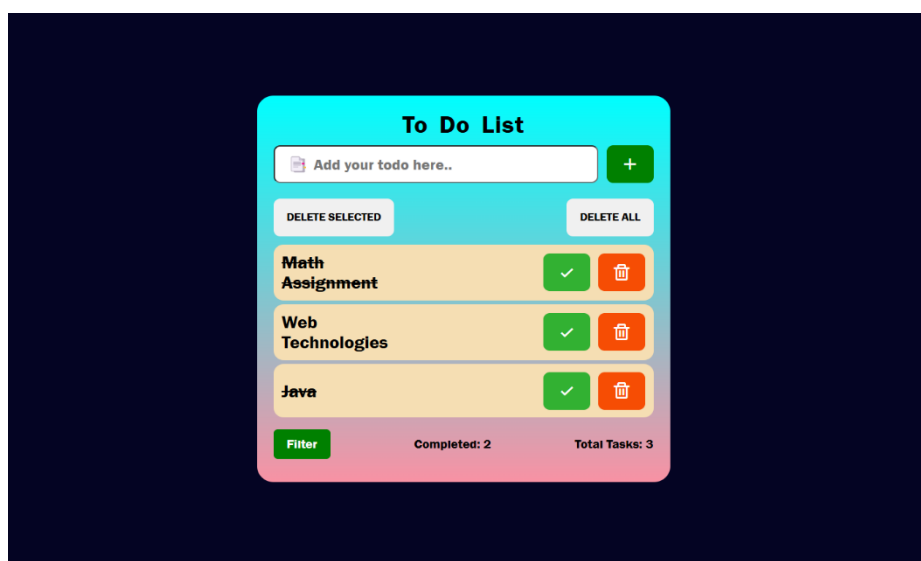
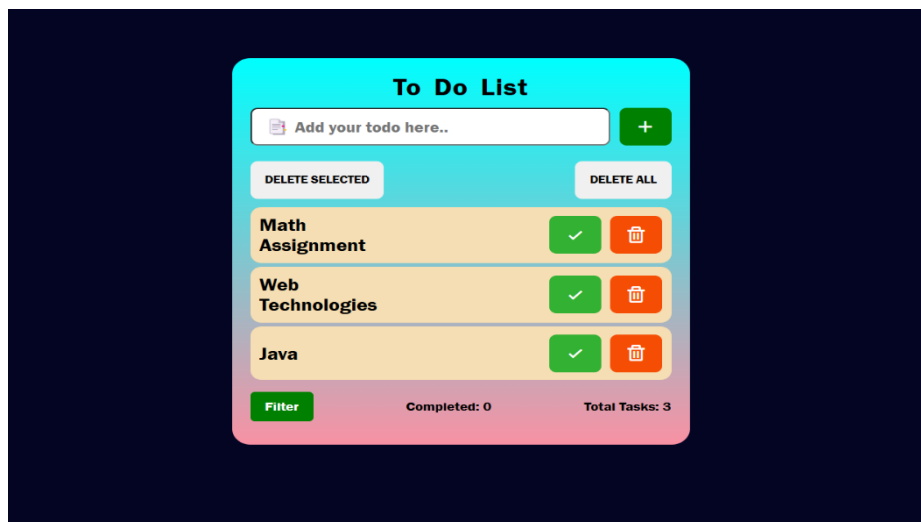
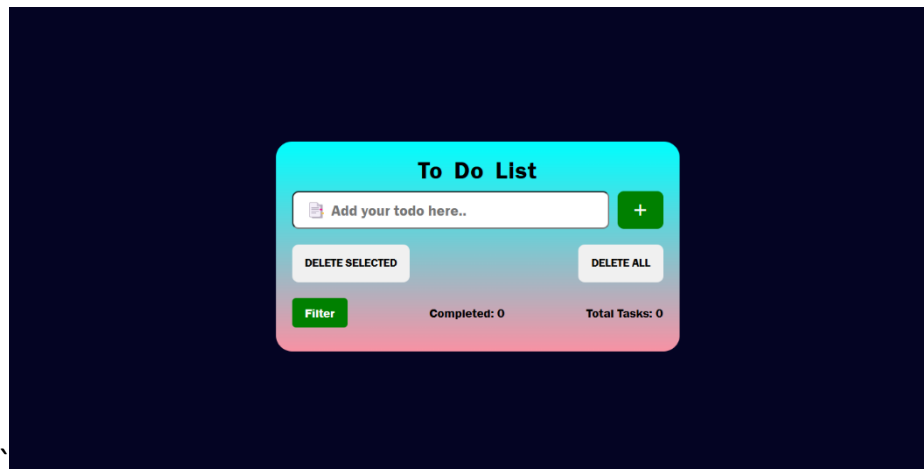
- **Model:** This is where the data is stored and managed. In this case, the tasks are stored in the **browser's local storage**, allowing them to persist even after page refreshes.
- **View:** The view represents the user interface, which is composed of HTML for structure and CSS for styling. It displays the task list and interacts with the user.
- **Controller:** JavaScript acts as the controller, responding to user interactions (e.g., adding, marking, or deleting tasks) and updating the model (local storage) and view (HTML elements).

4.2 System Flow Diagram:

The system flow can be broken down as follows:

1. **Initial Load:** The application loads from the browser.
 - JavaScript checks if there is any existing data in local storage.
 - If data exists, it is loaded into the task list.
 - If no data exists, the list is empty.
2. **User Interaction:**
 - **Add Task:** The user enters a task into the input field and clicks "Add".
 - JavaScript updates the task list on the page and in local storage.
 - **Complete Task:** The user clicks the checkbox of a task to mark it as completed.
 - The task is visually updated and the change is saved in local storage.
 - **Delete Task:** The user clicks the delete button next to a task.
 - The task is removed from the page and local storage.
3. **Persistence:** On every page load, tasks are retrieved from local storage and displayed.

SAMPLE OUTPUT:



SUMMARY:

The **Responsive To-Do List Website** follows a **client-side architecture** with a simple **Model-View-Controller (MVC)** structure, using **HTML** for structure, **CSS** for responsive design, and **JavaScript** for dynamic task management. The tasks are stored persistently in **local storage**, and the interface is designed to be responsive across all devices. This ensures that the user has a seamless, intuitive, and efficient experience for managing daily tasks.

The application features a responsive design, making it accessible and functional on a variety of devices, including desktops, tablets, and smartphones. The task list updates dynamically based on user interactions, with clear visual cues for completed tasks and easy-to-use controls for managing the list.

The project serves as an introductory exercise in web development, teaching key concepts such as **DOM manipulation**, **event handling**, and **local storage** management. Though simple in scope, the To-Do List offers a practical solution for personal task management and can be easily extended with additional features like task categorization or user authentication.

Overall, the To-Do List Application provides an effective, efficient, and responsive platform for organizing daily tasks, with potential for further enhancements in future iterations.

CONCLUSION:

The **To-Do List Application** successfully demonstrates the use of core web technologies—**HTML, CSS, and JavaScript**—to create a functional and interactive tool for task management. By implementing a simple yet effective design, the application provides users with an intuitive interface for adding, marking, and deleting tasks, ensuring that they can stay organized and productive.

The project highlights the importance of **responsiveness** and **user experience** in web development, ensuring that the application works seamlessly across various devices and screen sizes. The use of **local storage** to persist tasks across sessions further enhances the usability of the tool, offering convenience for users who need to track tasks over time.

This To-Do List project serves as a foundational learning experience for aspiring web developers, offering practical insights into **DOM manipulation, event handling, and local storage management**. Although simple, the application is an excellent starting point for more complex task management systems that could integrate features like categorization, user authentication, and cloud storage.

In conclusion, the To-Do List application is a powerful example of how basic front-end technologies can be leveraged to create

useful, everyday tools. It provides an accessible solution for personal task management while offering ample opportunities for future enhancements and scalability.

WEBLIOGRAPHY:

Below is a collection of useful resources that can aid in the development of the To-Do List application using HTML, CSS, and JavaScript.

1. MDN Web Docs (Mozilla Developer Network)

- **HTML Documentation:**
<https://developer.mozilla.org/en-US/docs/Web/HTML>
Comprehensive guide to HTML elements and structures for building the foundation of your To-Do List.
- **CSS Documentation:**
<https://developer.mozilla.org/en-US/docs/Web/CSS>
Detailed guide to styling and formatting the To-Do List, including responsive design techniques using CSS.
- **JavaScript Documentation:**
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
Learn JavaScript fundamentals, including DOM manipulation and event handling, crucial for the functionality of the To-Do List.
- **Local Storage Documentation:**
<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>
Understand how to store data locally in the browser to persist tasks across sessions.

2. W3Schools

- **HTML Tutorial:**
<https://www.w3schools.com/html/>
A beginner-friendly resource for learning HTML and building the basic structure of the To-Do List.
- **CSS Tutorial:**
<https://www.w3schools.com/css/>
Learn the essentials of CSS for styling the To-Do List and creating a responsive layout.

- **JavaScript Tutorial:**

<https://www.w3schools.com/js/>

Explore JavaScript concepts such as variables, loops, functions, and DOM manipulation to build the app's interactive features.

3. CSS-Tricks

- **Responsive Web Design:**

<https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

Learn how to implement responsive web design and media queries to make the To-Do List mobile-friendly.

- **Flexbox Layout:**

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

A guide to Flexbox, a powerful layout system used to create flexible, responsive layouts for the To-Do List.

4. Stack Overflow

- **JavaScript Local Storage Questions:**

<https://stackoverflow.com/questions/tagged/localstorage>

A helpful community for troubleshooting issues related to local storage in JavaScript.

5. GitHub

- **Sample To-Do List Projects:**

<https://github.com/search?q=to-do+list>

Explore various open-source To-Do List projects for inspiration, code examples, and best practices for your own project.

These resources will support your learning and provide you with the necessary tools to develop a robust, user-friendly To-Do List application, from the basics of HTML and CSS to advanced JavaScript and local storage techniques.

