

To Do List App

Name: Devesh Mahajan

ID: UMIP25776

Domain: Full Stack Web Development

Type: Basic Project

Sr. No.	Section	Page No.
1	Introduction	2
2	Objective	2
3	Technologies Used	3
4	Features	6
5	Working	7
6	Summary	9

Introduction:

This project is a simple yet powerful To-Do list web application designed to help users manage their tasks efficiently. Built using the MERN stack, which stands for MongoDB, Express.js, React, and Node.js, this app demonstrates the full-stack capabilities of modern web development. This application serves as a perfect example of how to combine the power of JavaScript with the MERN stack to build full-stack web applications.

Objective:

- The objective of this project is to build a simple yet functional To-Do list web application using the MERN stack (MongoDB, Express.js, React, Node.js).
- The primary goal is to create a user-friendly interface that allows users to efficiently manage their tasks. This includes the ability to add, update, delete, and mark tasks as completed.
- By the end of this project, the To-Do list app will be a fully functional, easy-to-use tool for users to track and manage their daily tasks with minimal complexity.
- We will be developing a dynamic and responsive user interface using React to interact with the backend and present tasks in a clear, organized manner.

Technologies Used:

- **JavaScript:** JavaScript is a high-level, versatile, and interpreted programming language primarily used to create interactive and dynamic content on websites. It is a core technology of the web, alongside HTML and CSS, enabling developers to build responsive and feature-rich user interfaces.
- **HTML:** HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It uses a system of tags to define elements such as headings, paragraphs, links, images, and more.
- **CSS:** CSS (Cascading Style Sheets) is a language used to style and layout HTML content on web pages. It controls the appearance of elements, such as colors, fonts, spacing, and overall design.
- **Vite:** Vite is a modern front-end build tool and development server designed for fast and efficient development. It is particularly popular for projects using frameworks like React, Vue, and Svelte.
- **ReactJS:** ReactJS is a popular JavaScript library for building dynamic, interactive, and reusable user interfaces, especially for single-page applications (SPAs). It was developed by Facebook and is maintained by Meta and an open-source community.
- **React-icons:** The React Icons library is a popular collection of customizable and scalable icons that are easy to use in React applications. It provides a set of high-quality, vector-based icons that can be easily incorporated into React components, making it simpler to add icons without having to deal with complex SVG or font icon libraries.

- **Axios:** Axios is a popular, promise-based JavaScript library used to make HTTP requests, such as fetching or sending data to a server. It is often used in web development, particularly in React applications, to interact with APIs. Axios provides a simple and clean API for making requests and handling responses, and it works both in the browser and in Node.js environments.
- **Cors:** CORS (Cross-Origin Resource Sharing) is a security feature implemented by web browsers that allows or restricts web applications running at one origin (domain) to make requests for resources from a different origin. It is designed to prevent malicious websites from accessing resources and data from other domains without permission.
- **Dotenv:** Dotenv is a zero-dependency module for managing environment variables in Node.js applications. It loads environment variables from a .env file into process.env, which is the global object in Node.js that stores environment variables. By using dotenv, you can keep sensitive information like API keys, database credentials, or configuration settings out of your codebase, improving security and flexibility.
- **Express:** Express is a minimal and flexible web application framework for Node.js, designed to simplify the process of building web applications and APIs. It provides a robust set of features that make it easy to handle routing, middleware, and HTTP requests and responses, allowing developers to focus on building their application logic rather than managing low-level details.
- **Mongoose:** Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a straightforward way to interact with MongoDB from a Node.js application by mapping MongoDB documents to

JavaScript objects. Mongoose helps to manage the relationship between data, provides schema validation, and simplifies CRUD operations.

- **MongoDB:** MongoDB is a popular NoSQL database that stores data in a flexible, document-oriented format using collections and documents instead of traditional tables and rows, as in relational databases. MongoDB is designed to handle large volumes of data with high scalability and flexibility, making it suitable for modern web applications, real-time analytics, and big data projects.

Features:

- **Responsiveness:** Responsiveness in websites refers to the design and development approach that ensures a website looks and functions well on a wide variety of devices and screen sizes, including desktops, tablets, and mobile phones.
- **State Management:** State management in this web application refers to the management of the data and states (e.g., UI, application logic, or session data) across various parts of the application. Since MERN stack apps typically involve multiple components, handling state effectively is crucial to ensure smooth user experience, consistency, and ease of maintenance.
- **Easy To Use UI:** An easy-to-use User Interface (UI) is a design that is intuitive, efficient, and enjoyable for users to interact with. It ensures that people can achieve their goals with minimal effort, regardless of their technical skill level.
- **Data Persistency:** Persistence is one of the most significant advantages of using a database. Unlike in-memory data (which is lost when the server restarts), a database ensures that data is saved permanently or until it is explicitly deleted or modified. This is crucial for storing user data, application configurations, and transactional records.



Working:

a) Basic Design



TO DO APP

Add



Workout



Study



Cleaning





b) Adding New task



TO DO APP

Add



Workout



Study



Cleaning



c) Updating Existing Task

TO DO APP

Workout

Update

Workout



Study



Cleaning



Summary:

This report outlines the development of a To-Do list web application built using the MERN stack (MongoDB, Express.js, React, Node.js). The application is designed to help users efficiently manage their tasks, with features like adding, updating, deleting, and marking tasks as completed. It leverages key technologies such as React for building an interactive user interface, Axios for making API requests, and MongoDB for persistent data storage. The project also incorporates essential tools like dotenv for managing environment variables and CORS for enabling secure cross-origin requests. The app emphasizes responsiveness, state management, and ease of use, ensuring a smooth user experience across different devices.

Source code: <https://github.com/deveshm019/todo-list.git>