GIT Repo URL: https://github.com/prateekp1304/FSD

Roll No: 08
Name: Prateek Patel
PRN: 1032210532

FSD Laboratory 07

Aim: Develop a full stack web application using MERN stack to perform CRUD operations.
Objectives:

1. To develop full-stack web projects using the MERN stack.
2. To learn database connectivity using fetch api.
3. To perform insert, update, delete and search operations on database.

Theory:

1. What is MERN stack?
   The MERN stack is a popular software development stack that consists of four key technologies, each representing a different aspect of building web applications. The components of the MERN stack are:
   - MongoDB: A NoSQL database that stores data in a flexible, JSON-like format.
   - Express.js: A web application framework for Node.js that simplifies the process of building web applications and APIs.
   - React: A JavaScript library for building user interfaces, particularly for single-page applications where data can change without the need for a full page reload.
   - Node.js: A JavaScript runtime that allows server-side execution of JavaScript, enabling the development of scalable and high-performance web applications.

   The MERN stack is known for its flexibility, as it allows developers to use JavaScript throughout the entire application stack.

2. Use of Fetch api.
   The Fetch API is a modern, built-in JavaScript interface for making network requests (e.g., fetching resources from a server). It provides a more powerful and flexible alternative to the older XML Http Request.
   Key features of the Fetch API include:
   - Promise-based: The Fetch API is based on Promises, making it easier to work with asynchronous code and handle responses.
   - Simple syntax: The API has a clean and simple syntax, making it easy to understand and use.
   - Support for various request and response types: It supports various HTTP methods (GET, POST, PUT, DELETE, etc.) and allows you to work with different types of data, including JSON.
   - Headers and authentication: Fetch allows you to set custom headers for requests and handle authentication.
   - Cross-origin resource sharing (CORS): It supports CORS, allowing you to make requests to different domains.

FAQ:

1. What makes MERN stack the fastest growing tech stack?

Ans: The MERN (MongoDB, Express.js, React, Node.js) stack has gained popularity and is considered a fast-growing tech stack for several reasons:

- JavaScript Throughout: One significant advantage of the MERN stack is that it uses JavaScript for both client-side (React) and server-side (Node.js) development. This allows for a unified language and data format (JSON) across the entire application stack, simplifying development and reducing the learning curve.

- Full Stack JavaScript Development: With MERN, developers can work on both the frontend and backend using JavaScript. This full-stack capability can streamline development workflows, improve collaboration between frontend and backend developers, and accelerate the overall development process.

- React's Component-Based Architecture: React, a key component of the MERN stack, utilizes a component-based architecture. This makes it easier to develop and maintain complex user interfaces, as developers can create modular, reusable components. This can lead to increased development speed and code maintainability.

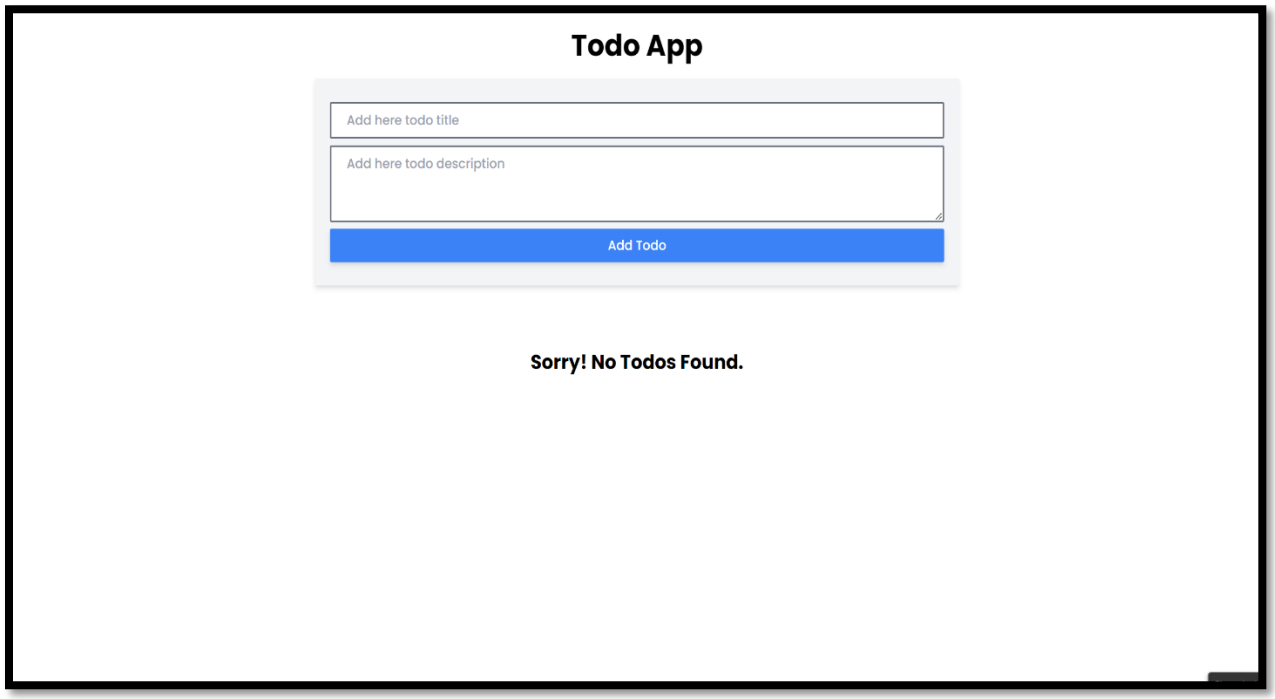
- Rich Ecosystem and Community Support: Each component of the MERN stack has a large and active community, which means extensive documentation, tutorials, and a wealth of third-party libraries. The availability of resources and community support can significantly accelerate development by providing solutions to common problems and challenges.

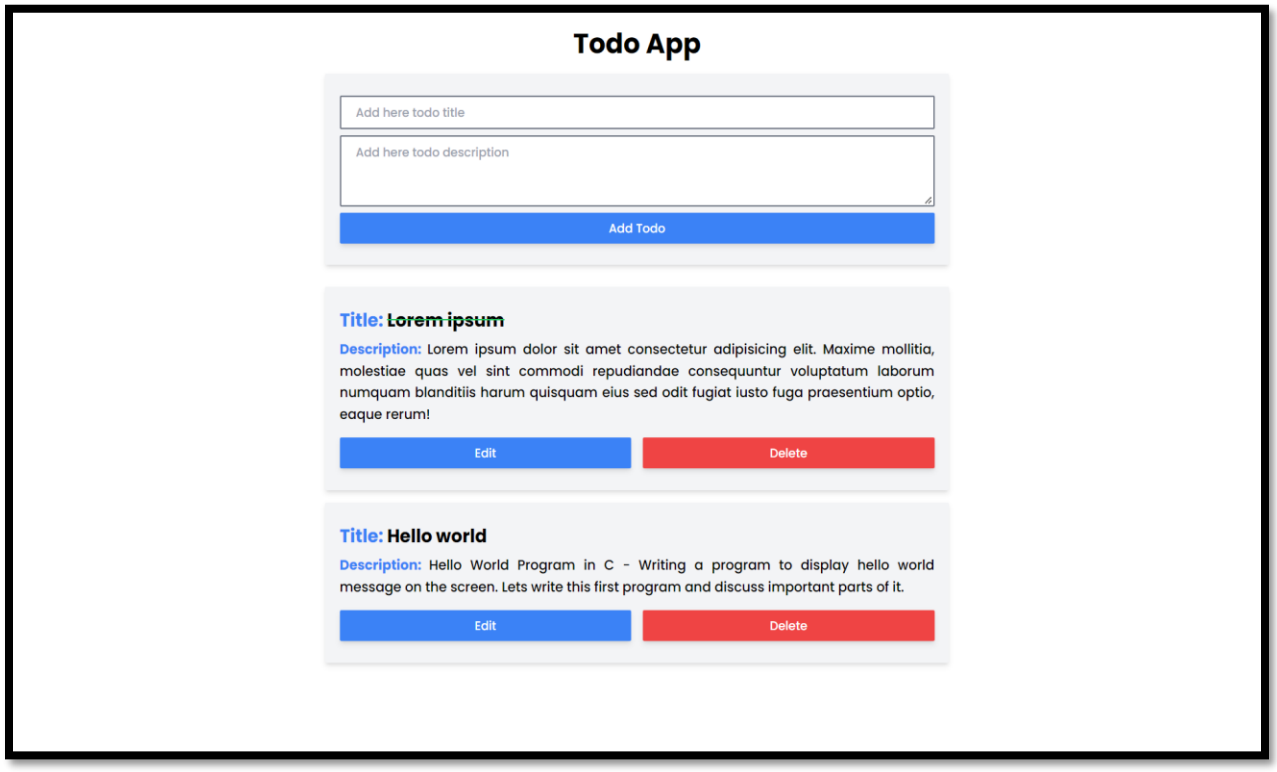
- Single Page Application (SPA) Development: MERN stack is well-suited for building Single Page Applications, where the entire application is loaded once, and subsequent interactions with the server happen asynchronously. This can result in a more seamless and responsive user experience, contributing to the stack's popularity.

- NoSQL Database (MongoDB): MongoDB, as the database component, is a NoSQL database that stores data in a flexible, JSON-like format (BSON). This schema-less approach allows for more agile development, as changes to the data model can be accommodated without significant modifications to the database.

It's important to note that the popularity of technology stacks can change over time, and new stacks may emerge based on evolving technologies and development trends. Always check for the latest trends and community feedback when making technology choices for a project.

Output Screenshots:





Sample Problem Statements:

CRUD Operations using MERN stack:

1.Student can create a React form or use existing/ implemented HTML form for Student's Registration System with the fields mentioned: First name, Last name, Roll No/ID, Password, Confirm Password,Contact number and perform following operations

1.Insert student details -First name,Last name, Roll No/ID, Password, Confirm Password,Contact number
2.Delete the Student records based on Roll no/ID
3.Update the Student details based on Roll no/ID- Example students can update their contact details based on searching the record with Roll no.
4.Display the Updated student details or View the Students record in tabular format.

2. Student can create a React form or use existing/ implemented HTML form for Library Management System with the fields mentioned: Book name, ISBN No, Book title, Author name, Publisher name and perform following operations

1.Insert Book details -Book name, ISBN No, Book title, Author name, Publisher name
2.Delete the Book records based on ISBN No
3.Update the Book details based on ISBN No- Example students can update wrong entered book details based on searching the record with ISBN No.
4.Display the Updated Book details or View the Book Details records in tabular format.

3. Student can create a React form or use existing/ implemented HTML form for Employee Management System with the fields mentioned: Employee name, Employee ID, Department_name, Phone number, Joining Date and perform following operations

1.Insert Employee details -Employee name, Employee ID, Department_name, Phone number, Joining Date
2.Delete the Employee records based on Employee ID
3.Update the Employee details based on Employee ID- Example students can update Employee details based on searching the record with Employee ID.
4.Display the Updated Employee details or View the Employee Details records in tabular format.

4. Student can create a React form or use existing/ implemented HTML form for Flight Booking Management System with the fields mentioned: Passenger name, From, to, date,Departure date,Arrival date, Phone number , Email ID and perform following operations

1.Insert Passenger details -Passenger name, From, to, date,Departure date,Arrival date, Phone number , Email ID
2.Delete the Passenger records based on Phone Number
3.Update the Passenger details based on Phone Number - Example students can update Flight Booking details based on searching the record with Phone Number.

4.Display the Updated Flight Booking details or View the Flight Booking Details records in tabular format.

Help Link:

https://www.mongodb.com/languages/mern-stack-tutorial