

MIT WORLD PEACE UNIVERSITY

Full Stack Development
Third Year B.Tech, Semester 1

Design MERN Application for
CRUD Operation

ASSIGNMENT 7

Prepared By A1-02. Mayur Behere
Batch A

Aim:

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

Objective:

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 stack used in web development, comprised of four main technologies: MongoDB, Express.js, React, and Node.js. Each component of the stack plays a crucial role in developing dynamic web applications.

Here's a breakdown of each technology in the MERN stack:

- **MongoDB:** MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It's known for its scalability, flexibility, and ease of use, making it suitable for handling large volumes of structured or unstructured data.
- **Express.js:** Express.js is a minimalist and flexible Node.js web application framework. It provides a set of features for building web applications and APIs, simplifying the creation of robust and scalable server-side applications. Express.js handles server-side logic and routes requests made by the client.
- **React:** React is a JavaScript library developed by Facebook for building user interfaces. It allows developers to create reusable UI components that efficiently update and render when the underlying data changes. React's component-based architecture enables the creation of interactive and responsive web applications.
- **Node.js:** Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to execute JavaScript code on the server-side, enabling the development of scalable and high-

performance web applications. Node.js handles server-side logic and interacts with databases like MongoDB while supporting asynchronous, event-driven programming.

The MERN stack combines these technologies to create full-stack web applications where MongoDB serves as the database, Express.js manages the server-side application logic and routing, React builds the user interfaces, and Node.js powers the server environment. Developers often choose the MERN stack due to its flexibility, as JavaScript is used across the entire stack, allowing for code reuse, faster development cycles, and isomorphic JavaScript, which means code can run both on the server and client side. Additionally, the MERN stack is well-suited for building real-time applications, single-page applications (SPAs), and scalable web applications.

2] Use of Fetch API.

=>

The Fetch API is a modern JavaScript interface used for making network requests (such as fetching resources from a server). It provides a more powerful and flexible way to handle HTTP requests compared to older techniques like XMLHttpRequest (XHR).

Here's a basic example demonstrating how to use the Fetch API to make a GET request to retrieve data from a server:

A code editor window with a dark background and light-colored text. It contains a JavaScript code snippet that demonstrates how to use the Fetch API to make a GET request. The code includes comments for each step: fetching data from a URL, checking the response status, parsing the JSON body, logging the data, and handling errors. The code is as follows:

```
// Fetch data from a URL using a GET request
fetch('https://api.example.com/data')
  .then(response => {
    // Check if the request was successful (status
    // code 200-299)
    if (!response.ok) {
      throw new Error('Network response was not
ok');
    }
    // Parse the response body as JSON
    return response.json();
  })
  .then(data => {
    // Handle the JSON data
    console.log('Data received:', data);
    // Perform operations with the retrieved data
  })
  .catch(error => {
    // Handle errors that occurred during the fetch
    // request
    console.error('There was a problem with the
fetch operation:', error);
  });
});
```

In this example:

- The `fetch()` function initiates a GET request to the specified URL (`https://api.example.com/data`).
- The promise returned by `fetch()` is handled using `.then()`. The first `.then()` checks the response status and uses `response.json()` to parse the response body as JSON. This parsing is done asynchronously.
- If the response status is not in the range 200-299 (indicating a successful response), it throws an error that is caught by the `.catch()` block.
- The second `.then()` handles the JSON data returned from the successful request.

The Fetch API supports various options and configurations for making different types of requests (GET, POST, PUT, DELETE, etc.), setting headers, handling request and response bodies, and working with cookies. It uses promises for handling asynchronous operations, providing a cleaner and more concise way to work with network requests in JavaScript.

It's important to note that the Fetch API has widespread support in modern browsers, but for older browsers, you may need to use a polyfill or a library like Axios to ensure consistent behavior across different environments.

FAQ :

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

=>

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

1. **JavaScript Everywhere:** One of the primary reasons for its popularity is the use of JavaScript across the entire stack. Developers can write code in a single language (JavaScript), both on the client-side (React) and server-side (Node.js). This unification simplifies development, promotes code reusability, and reduces context-switching for developers.
2. **Flexibility and Scalability:** Each component of the MERN stack is designed to be flexible and scalable. MongoDB, a NoSQL database, allows for flexible schema designs and scalability. Express.js provides a lightweight and flexible web application framework. Node.js, being

event-driven and non-blocking, handles concurrent requests efficiently. React's component-based architecture allows for the creation of reusable UI components, promoting scalability in front-end development.

3. ****Rich Ecosystem and Community Support:**** The MERN stack benefits from a large and active developer community. This community contributes libraries, tools, extensions, and resources, providing extensive support and solutions to common challenges faced during development. This wealth of resources makes it easier for developers to find help and resources when building applications with the MERN stack.

4. ****Isomorphic/Universal JavaScript:**** MERN supports isomorphic/universal JavaScript, enabling code execution both on the server and client side. This approach enhances performance by allowing server-side rendering (SSR), resulting in faster initial page loads and better search engine optimization (SEO).

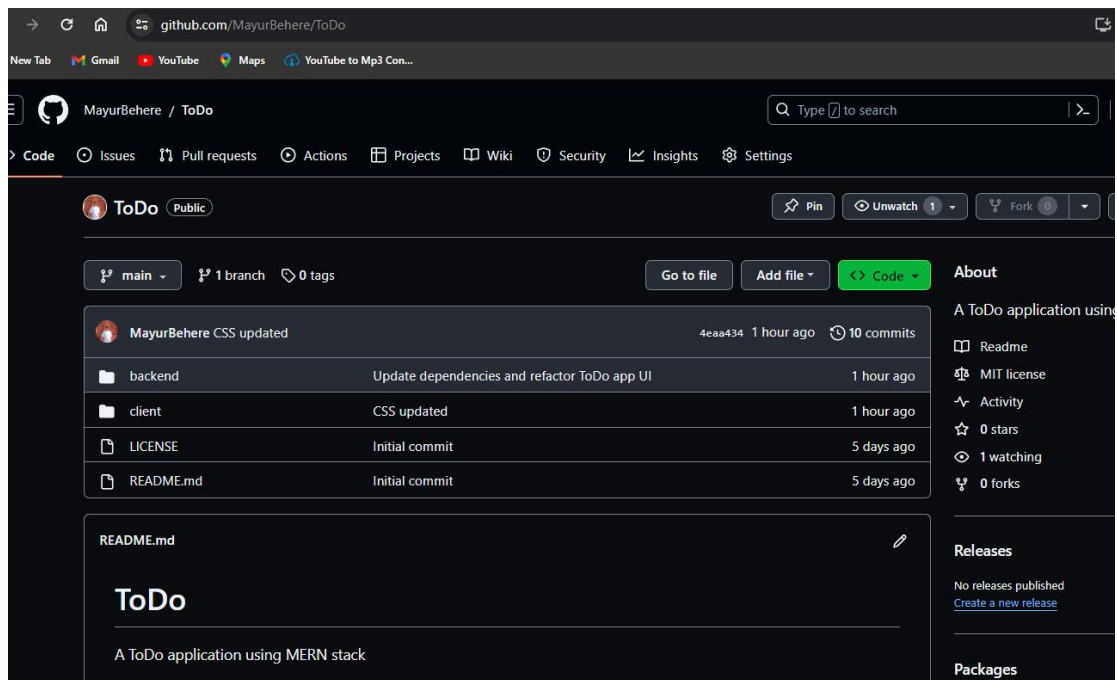
5. ****Rapid Development and Time-to-Market:**** The MERN stack's use of JavaScript, along with various libraries, tools, and a modular structure, accelerates the development process. With reusable components and an extensive set of tools, developers can build applications faster, reducing the time-to-market for new products or features.

6. ****Real-time Capabilities:**** MERN stack is well-suited for real-time applications due to technologies like WebSockets, enabling bidirectional communication between the client and server. This capability is particularly useful for building chat applications, collaborative tools, or any application requiring real-time updates.

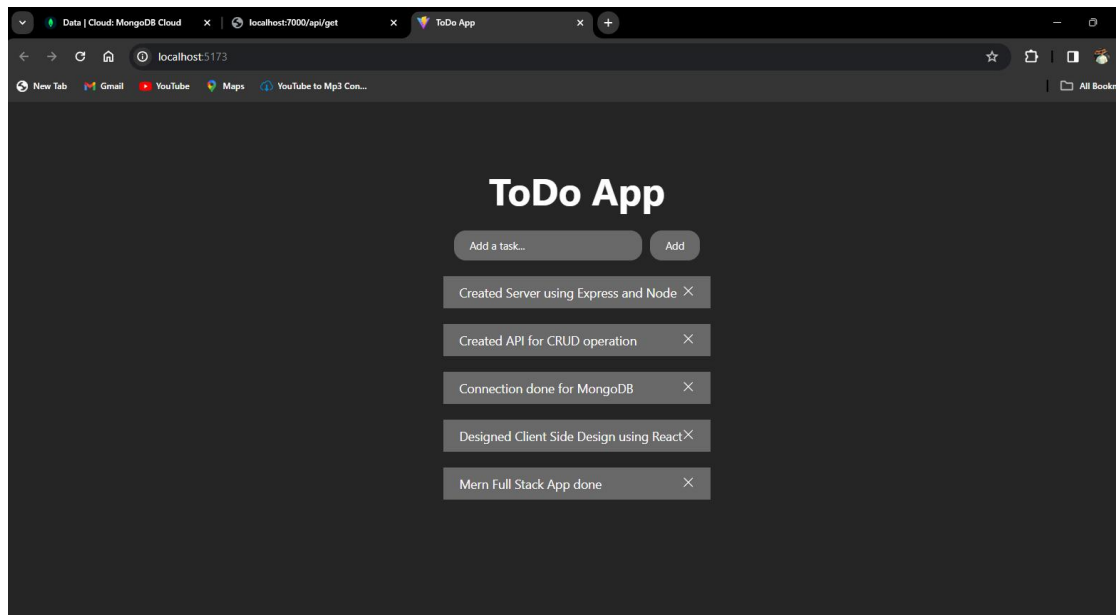
7. ****Industry Adoption and Demand:**** Many companies across different industries have adopted the MERN stack for building robust and scalable web applications. The demand for developers skilled in MERN technologies continues to grow, making it an attractive choice for developers seeking job opportunities in the field of web development. Overall, the MERN stack's combination of flexibility, scalability, rapid development capabilities, and an extensive ecosystem has contributed to its rapid growth and adoption within the web development community.

TASK : DESIGNED A TODO App USING MERN STACK.

GITHUB REPO : <https://github.com/MayurBehere/ToDo>

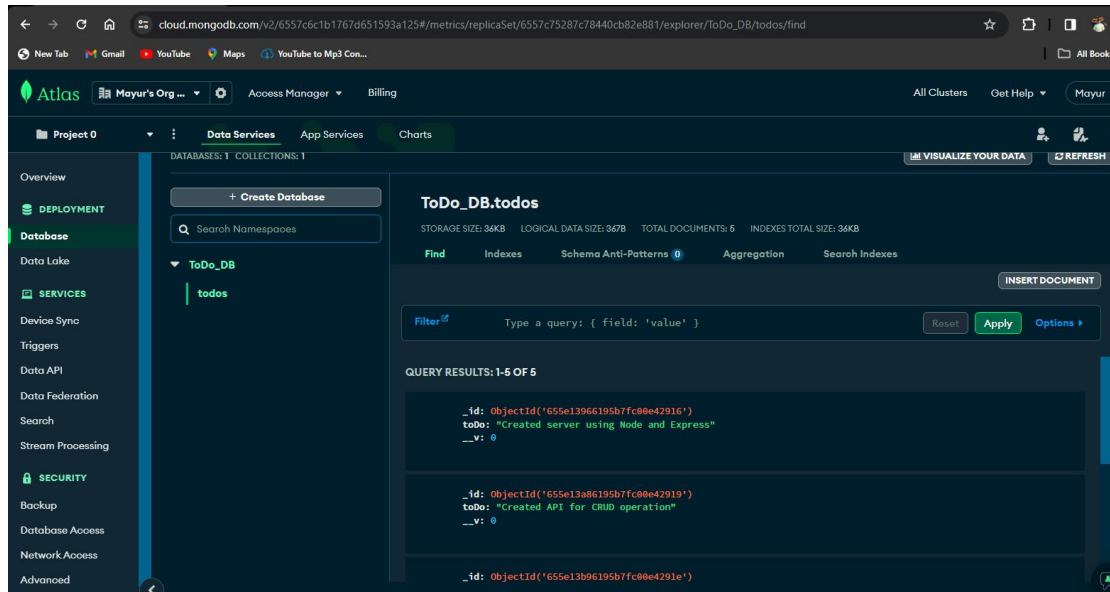


OUTPUT :

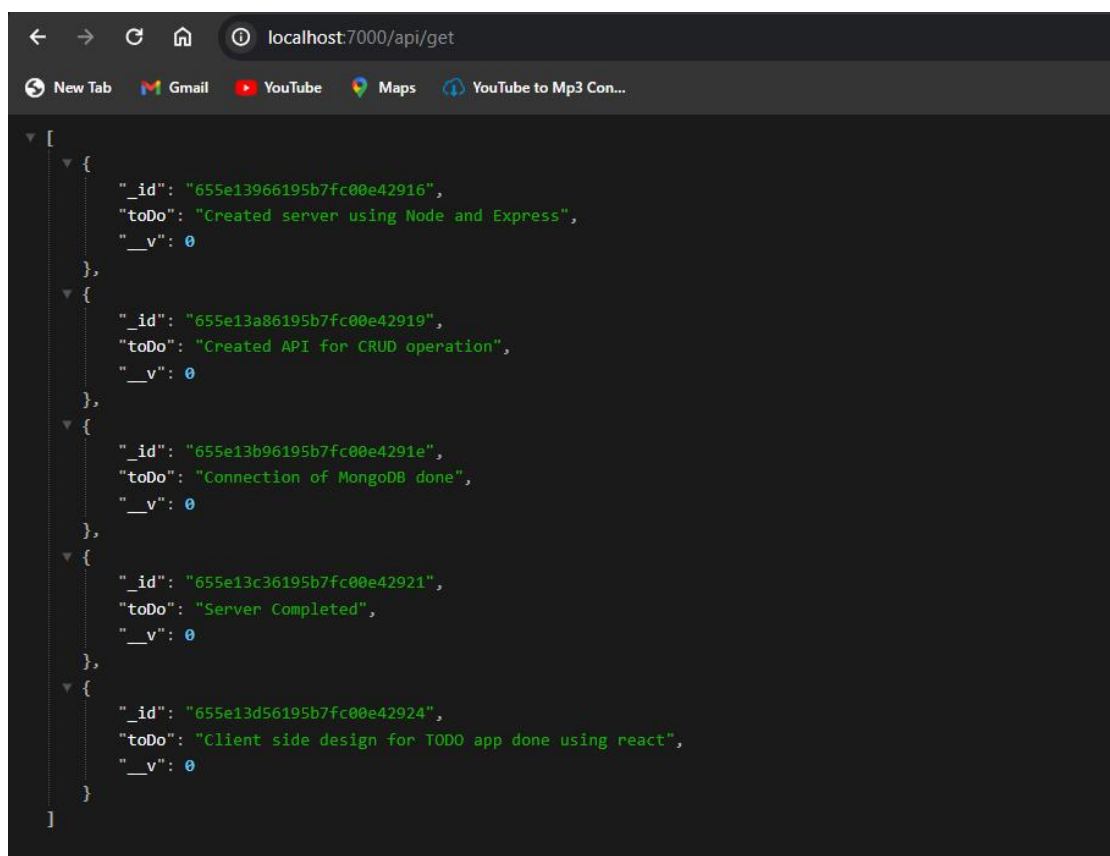


DEPENDENCIES:

USED MONGODB ATLAS AS DATABASE (CLOUD).



DESIGNED AN API FOR CRUD OPERATIONS.



Result:

Designed a full stack web application using MERN stack for performing CRUD operation