100xdevs > 0-100 > Week 5 - React >

5.4 | Notes for 5.2

Week 3

Week 4

Week 5 - React

5.1 | React Foundations ■

5.2 | React Project (Todo app) (Depreciated) ■

5.2 | React Project (Todo app) (Reupload) ■

5.3 | Notes for 5.1

5.4 | Notes for 5.2

Week 6

Week 7

Week 8 | Tailwind

Week 9 | TS and hooks

Week 10 - Postgres, Prisma

Week 11

Week 12 | Offline videos

Week 13

Week 14

Cohort 2.0 FullStack Open Source / ee

Download

Week 5.2

React Project —Todo Application

Up until now, our discussions have primarily revolved around theoretical concepts. In this lecture, Harkirat takes a `practical approach` by guiding us through the hands-on process of `building a to-do application.` We'll be applying the knowledge we've gained so far, specifically focusing on implementing the frontend using `React` and the backend using `Node.js`, `MongoDB`, and `Express` — creating a classic `MERN` stack application.

While there are `no specific notes` provided for this section, a mini guide is outlined below to assist you in navigating through the process of building the application. Therefore, it is strongly `advised to actively follow along` during the lecture for a hands-on learning experience.

Building a MERN Stack Todo Application with Zod

1] Environment Setup

- Install Node.js and npm on your machine.
- Set up a new React project using Vite: `npm create vite@latest` .
- Create a new Node.js project for the backend.

2] Frontend Development (React)

- Design the UI structure for your to-do app.
- Create React components for adding, displaying, and deleting todos.
- Utilize React Hooks (useState, useEffect) for managing the frontend state.
- Set up a clean and user-friendly interface for a seamless user experience.

3] Backend Development (Node.js, Express, MongoDB, Zod)

- Configure a Node.js and Express backend server.
- Integrate MongoDB for data storage.
- Implement Zod for backend data validation to ensure secure and valid inputs.
- Create API endpoints for handling todo operations like add, fetch, and delete.

4] Zod Integration

- Install Zod in your Node.js backend project using npm: `npm install zod` .
- Define schemas using Zod to validate incoming data.
- Integrate Zod validation within your API routes for robust data validation.

5] Connect Frontend to Backend

- Establish a connection between your React frontend and Node.js backend.

- Make API calls from the frontend to interact with the backend endpoints.

6] Run the Application

- Start both the React and Node.js servers.
- Open your to-do app in a browser and verify its functionality.

> By following these steps, you'll gain practical insights into building a MERN stack application with Zod integration, ensuring both frontend and backend components work seamlessly.

1 comments

Add a public comment...

Comment

Most upvotes All comments

Ak @Akshat Kharbanda 2 months ago

thank you for the notes!

👍 8 👎 1 ↩ 0 Reply

1