# Section 1: Introduction

* Introduction to the course and the Task Manager application:

Welcome to our course on building a Task Manager application with React Monorepo architecture! In this course, we'll explore the fundamentals of React development within a Monorepo environment while creating a practical Task Manager application.

We'll begin by introducing the core concepts of React Monorepo architecture, discussing its benefits and how it enhances project organization and scalability. Then, we'll dive into the specifics of our Task Manager application, outlining its features and functionalities.

Throughout the course, you'll gain hands-on experience in structuring and managing a React project using the Monorepo pattern. You'll learn how to implement essential features such as task listing, creation, editing, and deletion, all while adhering to best practices in React development.

By the end of the course, you'll have not only built a fully functional Task Manager application but also acquired valuable skills in React Monorepo development that you can apply to future projects. Let's get started and bring our Task Manager application to life!

* Overview of what will be covered in the tutorial:

In this tutorial, we'll cover everything you need to know to build a Task Manager application using React Monorepo architecture. We'll start by discussing the basics of React Monorepo and its advantages, then delve into setting up our project environment.

Next, we'll explore the key features of our Task Manager application, including task listing, creation, editing, and deletion. We'll learn how to manage state using MobX, integrate Fluent UI for a polished user interface, and implement end-to-end testing with Cypress.

By the end of the tutorial, you'll have a comprehensive understanding of React Monorepo development and the skills to build your own Task Manager application from scratch. Let's dive in and get started!

# Section 2: Setting Up the Environment

* Installing Node.js and npm:

npm install -g npm@latest

* Installing Nx CLI globally:

npm install -g vite

* Installing Json Server:

npm install json-server

# Section 3: Creating React Monorepo Applications

* Creating a new Nx workspace for the Task Manager project:

npx create-nx-workspace@latest react-monorepo --preset=react-monorepo

npx nx g @nx/react:app inventory --directory=apps/inventory --dry-run

* Exploring the initial file structure of the React app
* Understanding the configuration files

# Section 4: Creating React Libraries

* Creating necessary react libraries for the task manager app:

nx generate @nx/react:library task-manager-components --directory=libs/task-manager/ components --unitTestRunner=vitest --bundler=vite

nx generate @nx/react:library task-manager-core --directory=libs/task-manager/core --unitTestRunner=vitest --bundler=vite

nx generate @nx/react:library task-manager-models --directory=libs/task-manager/models --unitTestRunner=vitest --bundler=vite

nx generate @nx/react:library shared-ui --directory=libs/shared/ui --unitTestRunner=vitest --bundler=vite

# Section 5: Creating React Components

* Creating necessary react components for the task manager app:

nx generate @nx/react:component task-list --directory=libs/task-manager/components/tasks/task-list --unitTestRunner=vitest --bundler=vite

nx generate @nx/react:component task-form --directory=libs/task-manager/components/tasks/task-form --unitTestRunner=vitest --bundler=vite

nx generate @nx/react:component task-details --directory=libs/task-manager/components/tasks/task-details --unitTestRunner=vitest --bundler=vite

# Section 6: Managing Task State with MobX

* Install MobX:

npm install mobx mobx-react –save

* Setting up MobX store for managing tasks.
* Creating MobX actions and observables for task manipulation.
* Connecting React components to the MobX store.

# Section 7: Integrating APIs

* Setting up a mock API server using JSON Server:

npm install json-server

* Install Axios Library:

npm install axios

* Fetching tasks data from the API using Axios.
* Implementing CRUD operations for tasks (Create, Read, Update, Delete) with MobX actions.

# Section 8: Testing with ViTest and CyPress

* Introduction to unit testing with ViTest.
* Writing unit tests for task management features.
* End to end testing with CyPress

# Section 9: Deployment of the application using Vite

* Building the Task Manager app for production.
* Configuring deployment options (e.g., Vercel, Netlify, etc.).
* Deploying the app to a hosting platform.

# Section 10: Conclusion

Recap of what was covered in the tutorial.

Encouragement for further exploration and learning.

Thanking the viewers for participating.