</talentlabs>

Part 2: Deep-diving React with Contact Manager Application

Preface

This reading material is to explain to you how to best learn React in this part of the course. Please read it thoroughly before you move on to the next lecture.

Learning Approach

In this part, we are going to learn in a “learning by doing” approach. Instead of doing lengthy boring lectures with theories and book knowledge, we are going to show you how to build a hands-on practical project with React in a step-by-step way.

When you are learning, there are 3 things that you should keep in mind:

1. Expect that you will get stuck somewhere when you are following the course content. Just be patient and try to solve it slowly.
2. The focus should be put on “understanding” instead of “completion”. You should not just blindly copy the code from the video.
3. Try to write the code on your own instead of just following the instructions. Make changes and test it out.

Remember that the focus here is not speed of completion, but understanding.

</talentlabs>

Structure

In the next 4 lectures, we are going to build a “Contact Manager Application” with React. The process is going to be broken down into 4 parts.

Section 1: Building the Components

In this section, we are going to build the structure of the app and try to understand the actual usage of the React fundamentals concepts like project setup, structure, components, props etc.

Section 2: Routing

In this section, we are going to set up the routing of your React app, by using React Router. Apart from components, routing is another very important concept and skills in building web apps. It determines the user flow of your app.

Section 3: Hooking with APIs

In this section, we will start working on integrating the app with data. Most of the web apps will need to integrate with some sort of data and usually we would do it via APIs. In this section, we are going to work on the data and API integration with Axios (one of the most widely used API connection modules).

Section 4: Search Bar Implementation

We are going to build our first advanced interactive component for users to search for contacts. In this section, you will learn about how to build a Search bar component and implement the interaction on it. This is a very common component that you can see everywhere.

Section 5: Refactoring

Software applications require constant updates and upgrades. Sometimes, it is due to the underlying library upgrade, while it could be the changes in the best practices in the industry. In this case, we will need to make changes to our code and make it better. In the software engineering world, we call this “Refactoring”. In this final section, we will provide you a change to experience the refactoring process by upgrading the application with latest best practices and most updated version of libraries.

</talentlabs>

Learning Process

1. In each of the sections, we would have a pre-module document.To get started, you should read through the pre-module document first before watching the videos. Inside the pre-module document, there are two types of content:
    - **Pre-reads:** some documentations and knowledge that you need to learn before you can understand the video content.
    - **Content Updates:** the JavaScript libraries would keep changing and upgrading. Some of the video content might be outdated already. We would list out the parts that are not working anymore, and how to resolve that.

2. After reading the document, then you can start watching the video. We suggest you try implementing the code while watching the video. When in doubt or stuck, pause the video and make sure you understand what's happening and why.

3. The assignment for this part is really just a progress check to make sure that you have successfully built what we go through in the video. **Again, understanding is more important than completing the assignment.** If you don't understand, you won't be able to complete the next module smoothly.

4. At the end of each section, we would have a short summary article to help you wrap up your learnings in that section, so you would be able to assimilate the knowledge and apply them in the next module.

Solving Errors and Issues

As mentioned, it is expected that you will get stuck in the process. Here, we are going to provide you with a step-by-step guide of resolving the issues and get yourself unstuck.

1. Try to read the error messages and understand what it means
2. If you cannot understand that, try to Google search the error message
3. If still not able to resolve it, try to revert back your code, and make small changes step-by-step, and test your code step-by-step. Do not test a big chunk of changes in one go.
4. If you feel really frustrated, book a mentoring session and our mentor would be able to help you out on that.

— End of Preface —