Mini Project

The mini-project task is designed to reinforce the concepts you have learnt in React over the last few sessions, as well as give you an opportunity to practise advanced topics that will be covered in forthcoming sessions. The objective of the mini project is to give you the confidence of building a full-stack web application from scratch (React frontend, Node.js backend and SQL database).

Note: The tasks in this document are a guideline only, and you are encouraged to be creative. These guidelines may be updated to reflect the progress the class is making on a weekly basis to ensure everyone is keeping up as well as being challenged.

Homepage

Summary: Practise building simple React components that take static props.

Guidelines:

- Build a homepage that displays your Name, Bio and Photo
- Use a component for each item listed above, and pass in the data to be displayed as a prop
- Display a bullet-point list of topics and skills you have covered in the bootcamp
- Add a button that will navigate to another page

Books page

Summary: Display a Grid of your favourite books that will be retrieved using a HTTP request

Guidelines:

- A BookGrid component that encapsulates the logic of retrieving and displaying books
    - A BookItem component to handle display logic of each book on the grid
    - Include the name, author and thumbnail of the book
- First retrieve the list of books from a local file, then from a HTTP request, then from a SQL database
- Advanced: Show a like button on each Book Item
- Advanced: Store the number of likes in a database

Single book page

Summary: Define a page that can dynamically retrieve data based on url parameters

Guidelines:

- Use the Next.js router to retrieve the Book ID

- Use the Book ID to dynamically get book details from an API
- Display a loading state whilst data is loading
- Add a link to purchase the book on Amazon
- Add a button to copy the Amazon link using *react-copy-to-clipboard*
- Advanced: Use the Book ID to dynamically get book details from a Database

Search page

Summary: Dynamically query an API endpoint with user input

Guidelines:

- Create a search page with a search box, search button and list of results
- Retrieve search results from an API when the search button is clicked
- Advanced: retrieve search results from a SQL database when the search button is clicked
- Advanced: retrieve search results in real-time as the user's input changes
- Advanced: handle no results

Contact page

Summary: Use a HTML form to submit data, and refetch updated results once submitted

Guidelines:

- Create a contact page with a simple form
- Submit contact messages to the Database
- Show all contact messages below the form, update the list everytime a new form is submitted
- Advanced: show a loading state
- Advanced: Display an animation when a user submits a form
- Super Advanced: Display a popup modal to confirm the form has been successfully submitted