

React Basics | Lesson 3

Practice 2: React App

Instructions

You will create reusable components and practice passing data from a parent component to child components using props. This activity reinforces the concepts of props, dynamic rendering, and separation of concerns.

Part 1: Creating the Components

1. **Set Up a Components Folder**
 - Inside the `src` directory, create a folder named `components`.
2. **Create a `ProfileCard.jsx` File**
 - Define a `ProfileCard` functional component that accepts props for `name`, `age`, and `bio`.
 - Render the data dynamically using props.
3. **Create a `ProfileContainer.jsx` File**
 - Create a `ProfileContainer` functional component.
 - Render multiple instances of `ProfileCard` with different data for each profile.
4. **Export Components**
 - Ensure that both `ProfileCard` and `ProfileContainer` are exported.

Part 2: Combining Components in `App.jsx`

1. **Import Components**
 - Import `ProfileContainer` into `App.jsx`.
2. **Render the Container**
 - Add `ProfileContainer` inside the `App` component.

Part 3: Styling

1. Update App.css

- Add basic styles to center the content, add spacing, and style each card. For example:
 - Add a border and padding to each profile card.
 - Use flexbox or grid for alignment.

Deliverable A functional React app with:

- A `ProfileCard` component that dynamically displays:
 - A user's name.
 - Their age.
 - A short bio.
- A `ProfileContainer` component rendering multiple `ProfileCard` components with different data.

Example Output

When the app runs, you should see:

- A heading for the profile container (e.g., "User Profiles").
- Multiple styled profile cards displaying names, ages, and bios.

Example:

```
Name: Alex
Age: 25
Bio: A software engineer who loves hiking.
```

```
Name: Taylor
Age: 30
Bio: A designer passionate about photography.
```

Setup

1. Before starting this task, you will need to set up your React app:

- **Create a New React App:**

Use Vite to scaffold a new React project:

```
● npm create vite@latest
```

- Name the project static-todo-react-app.
- Choose the React framework and the JavaScript variant.

- **Install Dependencies:**

Navigate into the project directory:

```
● cd static-todo-react-app
```

- Install necessary dependencies:
 - npm install

- **Start the Development Server:**

Start your local server:

```
● npm run dev
```

- Open your browser and navigate to <http://localhost:5173>.
2. For this activity, you will create your own repository to store your code:
- Create a new repository.
 - Name it: practice-react-app[Your First Name]-[Your Last Initial]
 - Make sure the repository is public.
 - Clone the repository to your local machine.
 - Start working on the code.

Submission

Submit the url for your GitHub repo in the field below.