Chapter 04 - Talk is cheap, show me the code!

Assignment

- Is JSX mandatory for React?

  ➔ No, JSX is not mandatory. It just makes writing React code easier and cleaner.

- Is ES6 mandatory for React?

  ➔ No, but ES6 helps a lot. React code often uses ES6 features like arrow functions, destructuring, etc., for cleaner code.

- `{TitleComponent}` vs `{<TitleComponent/>}` vs `{<TitleComponent></TitleComponent>}` in JSX
  - {TitleComponent} → Treats it as a normal JavaScript function (value).
  - {<TitleComponent/>} or {<TitleComponent></TitleComponent>} → Actually **renders** the React component.

- How can I write comments in JSX?

  ➔ Write comments inside {} using /* */.
  Example: {/* This is a comment */}

- What is <React.Fragment></React.Fragment> and <></> ?

➔ These are used to group multiple JSX elements without adding extra nodes to the DOM.

  - `<React.Fragment>` is full form.
  - `<> </>` is short form.

- What is Virtual DOM?

  A Virtual DOM is a lightweight copy of the Real DOM. React updates it first and then efficiently updates only the changed parts in the real DOM.

- What is Reconciliation in React?

  It's the process where React compares the new Virtual DOM with the old one and updates only the changed parts.

- What is React Fiber?

  ➔ React Fiber is the new engine behind React that improves performance, allowing React to break work into chunks and prioritize updates.

- Why we need keys in React? When do we need keys in React?
  ➔ Keys help React **identify** which items changed, added, or removed in a list.
  ➔ We need keys when **looping** with .map() or **rendering lists** of components.

- Can we use index as keys in React?
  Yes, But it's not recommended because if the list changes, React may confuse items and cause bugs.

- What is props in React? Ways to
  Props are **inputs** to components — like arguments to a function.
  You can use props by **passing** values and **destructuring** them inside components.

- What is a Config Driven UI ?
  It's a UI built using **dynamic data** (config), not hardcoded. Example: UI changes based on data from an API.

Coding Assignment:

- Build a Food Ordering App
  - Think of a cool name for your app
  - Build a AppLayout
  - Build a Header Component with Logo & Nav Items & Cart
  - Build a Body Component
  - Build RestaurantList Component
  - Build RestaurantCard Component
  - Use static data initially
  - Make your card dynamic(pass in props)
  - Props - passing arguments to a function - Use Destructuring & Spread operator
  - Render your cards with dynamic data of restaurants
  - Use Array.map to render all the restaurants

PS. Basically do everything that I did in the class, in the same sequence. Don't skip small things.

References

- Code Link: https://bitbucket.org/namastedev/namaste-react-live/src/master/
- React without JSX: https://reactjs.org/docs/react-without-jsx.html
- Virtual DOM: https://reactjs.org/docs/faq-internals.html
- Reconciliation: https://reactjs.org/docs/reconciliation.html

- React Fiber Architecture: https://github.com/acdlite/react-fiber-architecture
- React Without ES6: https://reactjs.org/docs/react-without-es6.html
- Index Keys as Anti-Pattern: https://robinpokorny.com/blog/index-as-a-key-is-an-anti-pattern/