Full Stack Development with AI

[Module 7: Week 8: Frontend Frameworks]

Submitted By: **Jijin Pavithran**

1. Identify Key Concepts (5 Marks)

a) Three key concepts of modern front-end frameworks:

- **Component-Based Design:** UI is divided into reusable, self-contained components, this will simplifying code management and maintenance.
- **Client-Side Rendering (CSR):** Rendering content in the browser reduces the load on server end and latency.
- **Declarative Programming:** simplifies front-end development by allowing developers to define the desired state of the UI, and the framework will take care of the rendering. we describe what the UI should look like given certain data, and the framework will handle the actual update. This abstraction significantly reduces the complexity of managing UI state.

b) Benefits in real-world applications:

- **Component-Based Design:** In an e-commerce site, product cards can be reused across the homepage, search results, and recommendations.
- **Client-Side Rendering (CSR):** Social media feeds like Facebook load quickly without full-page reloads, providing a smooth browsing experience.
- **Declarative Programming:** Interactive dashboards update in real-time, with the framework efficiently managing UI changes.

2. Component Creation and Reusability (5 Marks)

a) Using CodeSandbox, create two reusable components, such as a Button and a Card.

App.js Button.jsx Card.jsx

src > Card.jsx > ...

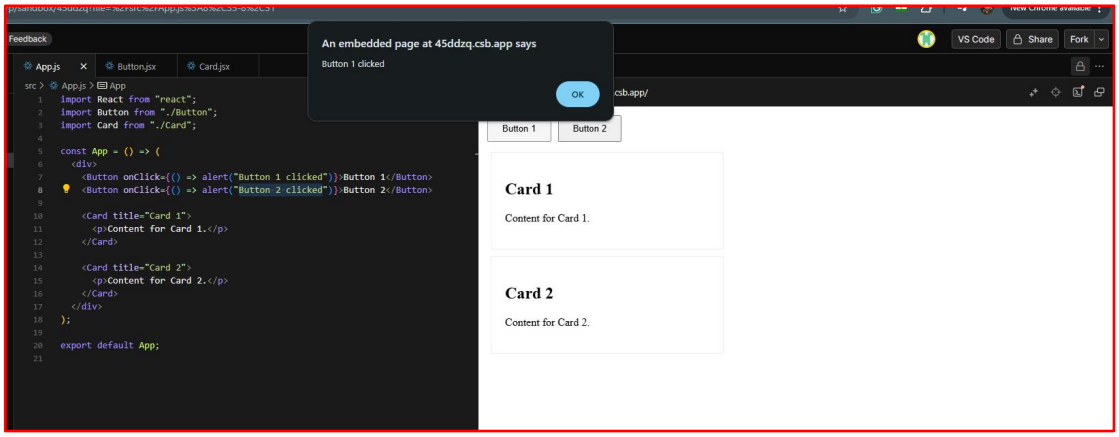
```jsx
// Card.jsx
import React from "react";

const Card = ({ title, children }) => (
  <div
    style={{
      border: "1px solid #eee",
      padding: "20px",
      margin: "10px",
      width: "300px",
    }}
  >
    <h2>{title}</h2>
    <div>{children}</div>
  </div>
);

export default Card;
```

b) Demonstrate how each component can be reused within different parts of a simple web page. For instance, use the Button in multiple sections to

trigger different actions, or display the Card component with different content.



3. CSR & SSR (5 Marks)

a) Explain CSR & SSR

- **Client-Side Rendering (CSR):** CSR fetches the application and via JavaScript, generating the UI dynamically.
- **Server-Side Rendering (SSR):** SSR generates HTML on the server and sends complete pages to the client. Each change comes from server side need refresh on client end.

b) Difference between CSR and SSR with advantages

- **CSR:**
  - Lower data usage and enhanced user experience.
  - Longer initial loading times and requires capable client devices.
- **SSR:**
  - Generates HTML on the server, minimizing client-side computation.
  - Faster initial page load and better for SEO.

4. Single Page Application (SPA) Structure (5 Marks)

a) Definition:

- **A Single Page Application (SPA) is a web application that loads a single HTML page and dynamically updates the content as the user interacts within the application.** The key characteristics of a single page application is that it does not require a full-page reload to change views or perform actions. In traditional web applications, each view typically resides in its own separate physical page. **In SPA, however, there is only one physical HTML file**. T**he illusion of multiple pages is achieved by dynamically swapping the view of the single page during runtime**. JavaScript plays a crucial part by handling routing and view updates.

b) Two advantages of SPA structure:

- **Faster interactions:** Reduce the number of server requests needed for navigation.
- **Smoother transitions:** Provide a more responsive and engaging user interface.