**Module 7 Assignment**

Apply the foundational knowledge of frontend frameworks covered in the module. You will compare traditional and modern frontend development approaches, identify the core features of component-based and declarative design, and create a simple Single Page Application (SPA) using a cloud-based development environment.

**1. Identify Key Concepts (5 Marks)**

a) List and briefly define three key concepts of modern frontend frameworks, such as component-based design, client-side rendering, and declarative programming.

b) For each concept, provide a short example or scenario where it would be beneficial in a real-world application.

**2. Component Creation and Reusability (5 Marks)**

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

b) Difference between CSR and SSR with advantages

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

a) Define SPA

b) List two advantages of SPA structure for modern web applications

**Frontend Frameworks Assignment**

**1. Identify Key Concepts**

a) Three Key Concepts of Modern Frontend Frameworks:

1.Component-Based Design:

This refers to breaking down of the UI into independent, reusable components that manage their own state and rendering.

2.Client-Side Rendering (CSR):

This refers to web pages rendered directly in the browser rather than receiving fully rendered HTML from the server.

3.Declarative Programming:

This refers to describing what the UI should look like based on the current state rather than imperatively defining each step to update the UI.

b) Real-World Examples:

1.Component-Based Design:

An e-commerce site like Amazon.com can have reusable components that display different products with the same styling and layout but different content (image, title, price).

2.Client-Side Rendering:

A social media platform like Twitter uses CSR to dynamically load new tweets as the user scrolls without refreshing the page, creating a smoother user experience.

3.Declarative Programming:

A weather app that declaratively renders different UI components based on the current weather state (showing sun icon when sunny, rain icon when rainy) without manually manipulating the DOM.

**2. Component Creation and Reusability (5 Marks)**

**Button:**

function Button({ onClick, children, variant = "primary" }) {

return (

<button

className={`btn ${variant}`}

onClick={onClick}

>

{Clickme}

</button>

);

}

export default Button;

Notes:

onClick: a function to run when clicked.

Clickme: content inside the button.

* variant: allows changing the style (primary, secondary, etc).

**Card:**

function Card({ title, content, image }) {

return (

<div className="card">

{image && <img src={image} alt={title} />}

<h3>{title}</h3>

<p>{content}</p>

</div>

);

}

export default Card;

Notes:

* title, content: to display the card’s information.
* image: optional image to include.

 Can be reused to show different product or content blocks.

**Reuse:**

import Button from './Button';

import Card from './Card';

import './App.css';

function App() {

return (

<div className="container">

<h1>My Online Shop</h1>

{/\* Section 1: Using Button with different functions \*/}

<section>

<Button onClick={() => alert('Added to cart!')}>

Add to Cart

</Button>

<Button variant="secondary" onClick={() => console.log('Item Saved')}>

Save Item

</Button>

</section>

{/\* Section 2: Using Card with different product details \*/}

<section>

<h2>Featured Products</h2>

<Card

title="Mobile Phone"

content="High Specs Mobile Phone."

/>

<Card

title="Wireless Earbuds"

content="Crystal clear sound with long battery life."

/>

</section>

{/\* **Reusing both Button and Card again** \*/}

<section>

<h2>Daily Recommendations</h2>

<Card

title="SmartTV"

content="Netflix, Internet, Shopping."

/>

<Button onClick={() => alert('Viewing details...')}>

View Details

</Button>

</section>

</div>

);

}

export default App;

**3 different sections** reuse Button and Card in various contexts:

* Buttons with different actions

Cards with unique titles/content

**3. CSR & SSR (5 Marks)**

**CSR (Client-Side Rendering)**: The browser downloads a minimal HTML page and JavaScript bundle, then the JavaScript renders the content dynamically in the browser.

**SSR (Server-Side Rendering)**: The server generates the complete HTML for each page request and sends it to the client.

**CSR Advantages**:

1. More interactive once the page loads
2. Smoother switching between pages or views
3. Offline capabilities with service workers

**SSR Advantages**:

1. Better for search engines (SEO)
2. Faster first-time page load
3. Can still show content even if JavaScript is turned off

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

An SPA is a website that loads just one main page and updates parts of the page as you use it, without reloading the whole page.

b) Two Advantages:

Smoother Experience: SPAs change the content quickly without refreshing the entire page, so it feels faster and more like a mobile app.

Less Server Work: After the first load, the app only asks the server for small pieces of data instead of loading whole pages, which saves bandwidth and reduces server work.