

# Full Stack Development – II

## Assignment

Name:- Nischaya Garg

UID:- 23BAI70359

Class:- 23AML-2 'A'

---

Q1) Summarize the benefits of using design patterns in frontend development?

Ans) Design patterns are consistent techniques and discipline followed while designing a full stack application that:-

- Improves Maintainability for future refactors
  - Improve consistency across teams
  - Makes the application scalable
  - Reduces chance of bugs
- 

Q2) Classify the difference between global state and local state in React?

Ans) Local state are resources that can be accessed by one component whereas Global state are resources that can be accessed by multiple components.

---

Q3) Compare different routing strategies in Single Page Applications (client-side, server-side, and hybrid) and analyze the trade-offs and suitable use cases for each?

Ans)

- Client Side Routing is where routing is handled in the browser using javascript. Major tradeoff is that initial load is heavier.

It's use cases are dashboards, admin panels etc.

- Server Side Routing is where each route triggers a server request. Major tradeoff is that the navigation is slower and it loads upon the server.

It's use cases are content heavy websites etc.

- Hybrid Routing is where above both are combined i.e. server side routing for first load and client side routing for subsequent loads.

It's use cases are web3 coop games, e-commerce websites etc.

---

Q4) Examine common component design patterns such as Container–Presentational, Higher-Order Components, and Render Props, and identify appropriate use cases for each pattern?

Ans) Design Patterns like:-

- Container-Presentational is where container components handle data fetching, state management etc. and presentational components focus on rendering UI based on data received by props.
- Higher-Order Components is a function that takes a component as input and returns a new component with enhanced or additional functionality, commonly used to share reusable logic such as authentication, authorization, logging etc.
- Render Props Pattern is a technique for sharing component logic by passing a function as a prop that returns a React element. It determines how the UI should be rendered and changed upon behaviour.

---

Q5) Demonstrate and develop a responsive navigation bar using Material UI components while applying appropriate styling and breakpoint configurations?

Ans)

- Steps to create a React application:-

Step 1: Prerequisites

Ensure the following are installed:

- Node.js (v18 or above)

- npm (comes with Node.js)

## Step 2: Create a New Vite + React Project

During the setup:

- Select React
- Select JavaScript (or TypeScript if required)

## Step 3: Install Default Dependencies

Install the base dependencies generated by Vite.

## Step 4: Install Runtime Dependencies

These packages are required during application execution.

## Step 5: Install Development Dependencies

These packages are used only during development and build time.

PTO→

• App.jsx code:-

```
return (
  <div className="App">
    <nav className="navbar fixed-top">
      <div className="container">
        <a className="navbar-brand" href="#"></a>
        <button className="navbar-toggler"></button>

        <div className="navbar-collapse">
          <ul className="navbar-nav">
            <li className="nav-item"><a className="nav-link" href="#"></a></li>
            <li className="nav-item dropdown">
              <a className="nav-link dropdown-toggle" href="#"></a>
              <ul className="dropdown-menu">
                <li><a className="dropdown-item" href="#"></a></li>
              </ul>
            </li>
          </ul>

          <form className="d-flex">
            <input className="form-control" />
            <button className="btn"></button>
          </form>

          <div className="d-flex">
            <button className="btn"></button>
            <button className="btn"></button>
          </div>
        </div>
      </nav>

      <main className="container">
        <section>
          <div className="row">
            <div className="col">
              <div className="card">
                <div className="card-body">
                  <h5></h5>
                  <p></p>
                  <button className="btn"></button>
                </div>
              </div>
            </div>
          </div>
        </section>
      </main>
    </div>
  );
```

---

Q6) Evaluate and design a complete frontend architecture for a collaborative project management tool with real-time updates. Include:

- a) SPA structure with nested routing and protected routes
- b) Global state management using Redux Toolkit with middleware
- c) Responsive UI design using Material UI with custom theming
- d) Performance optimization techniques for large datasets
- e) Analyze scalability and recommend improvements for multi-user concurrent access?

Ans)

a) SPA Structure:

- Feature-based folder structure ( auth, projects etc.)
- Protected routes using authentication guards.

b) Global State Management:

- Redux Toolkit for predictable state

Middleware:

- WebSocket middleware for real-time updates

c) Responsive UI with Theming:

- Material UI with custom theme
- Centralized color palette and typography

d) Performance Optimization:

- Memoization
- Lazy loading
- Debounced search and pagination

e) Scalability & Multi-User Access:

- WebSockets for real-time collaboration.
  - Conflict resolution and rate limiting.
-