

# **Full Stack – II | Question Bank for MST 1**

## **UNIT – I: Advanced Front-End Development**

### **SECTION A: Very Short Answers**

1. What is meant by global state in a React application?
2. Why is state management important in large-scale React projects?
3. Define Context API and mention one use case.
4. What is Redux Toolkit?
5. What is the role of a Redux store?
6. What is the purpose of the useContext() hook?
7. Define performance optimization in React.
8. What is memorization?
9. Name any one hook used for memorization.
10. What is React.memo()?
11. What causes unnecessary re-rendering in React?
12. What is Material UI?
13. Mention any two advantages of using Material UI.
14. What is Ant Design primarily used for?
15. What are Styled Components?
16. How are Styled Components different from traditional CSS?
17. What does PWA stand for?
18. Mention one key feature of Progressive Web Apps.
19. What is a service worker in a PWA?
20. What is React Testing Library?

### **Section B: Short Answers**

1. Explain the concept of state management in React applications.
2. Describe how Context API helps in avoiding prop drilling.
3. Explain the working of Redux Toolkit with an example scenario.
4. Differentiate between Context API and Redux Toolkit with suitable points.
5. What are common performance issues faced in React applications?
6. Explain how memorization improves React performance.
7. Distinguish between useMemo() and React.memo().
8. Explain the importance of component reusability in UI design.
9. Describe the role of Material UI in modern front-end development.
10. How does Ant Design support enterprise-level UI development?
11. Explain the concept of CSS-in-JS with reference to Styled Components.
12. Why is mobile-first design important in React applications?
13. What is a Progressive Web App? Explain its advantages.
14. How do service workers enable offline functionality?
15. Explain unit testing using Jest.

16. How does React Testing Library focus on user-centric testing?
17. What is snapshot testing and where is it useful?
18. Explain the process of debugging React applications using Chrome DevTools.
19. What information is provided by Lighthouse performance reports?
20. List and explain best practices for writing maintainable front-end code.

## **Section C: Long Answers**

1. Explain advanced state management techniques in React using Context API and Redux Toolkit, highlighting their use cases.
2. Discuss various performance optimization strategies used in React applications.
3. Explain the architecture of Redux Toolkit and its advantages over traditional Redux.
4. Compare Material UI and Ant Design in terms of design philosophy, component availability, and real-world usage.
5. Describe how Styled Components help in building scalable and maintainable React applications.
6. Explain the complete workflow of developing a Progressive Web App (PWA) using React.
7. Discuss different front-end testing approaches using Jest and React Testing Library.
8. Explain how Chrome DevTools can be used to debug and analyze React application performance.
9. Describe the role of Lighthouse audits in improving front-end performance and user experience.
10. Write a detailed note on best practices for building reliable, testable, and high-performance front-end applications.