

Instructions to the candidates:

1. Question **No. 1** is compulsory
2. Attempt any **three** questions from remaining five questions
3. Assume suitable data if **necessary** and justify the assumptions
4. Figures to the **right** indicate full marks

Q1 Answer the following questions.

- A Explain how DNS works and the process it follows to resolve domain names to IP addresses. **05**
- B Differentiate between JSON and XML. Discuss their use cases and advantages in web development. **05**
- C Illustrate with an example how to handle form validation in JavaScript, including client-side validation for an email and password field. **05**
- D Create a simple React application that fetches data from an external API and displays it in a list format. Describe the steps involved. **05**

Q2 A Discuss the asynchronous nature of JavaScript and how Promises and async/await improve handling of asynchronous operations. Provide code examples **10**
B Explain the concept of state and props in React. How do they differ, and how are they used in components? **10**

Q3 A Compare and contrast the use of classes and inheritance in JavaScript with functional programming paradigms. Provide examples. **10**
B Create a simple Express application that integrates with React to display a list of items fetched from a server. Describe the steps involved and provide code snippets. **10**

Q4 A Explain how React's `useEffect` hook can be used to perform side effects in functional components. Provide an example where `useEffect` is used to fetch data from an API and display it in a component. **10**
B Create a web page using HTML and JavaScript where an image moves across the screen from left to right continuously. Provide the HTML and JavaScript code and explain the implementation. **10**

Q5 A Describe the Event Loop in Node.js. How does it handle asynchronous operations? Provide an example. **10**
B Explain how React Router can be used to create a single-page application (SPA). Provide an example of routing in React. **10**

Q6 A Describe how to manage state in a React application using Redux. Include an example to illustrate state management in a complex application. **10**
B Explain how Node.js handles asynchronous operations using callbacks, Promises, and `async/await`. Provide code examples for each method. **10**