Interactive Web development

Practical Assignment on React JS

Year3/CS/SoICT/CST/UR

3 students per group

Practical work

Physical presentation on computers

Duration of the assignment 8 days

Deadline: 25 November 2024

Note:

4 questions randomly be chosen for evaluation. Each question will be marked out of 5 marks totaling 20marks for this assignment.

Each student in a group will be required to explain into details the response for the selected question. Failure to this will impact the overall marks of the group unless the later has been reported before as non-participating student.

This assignment is intended to let students understand the basics of React, covering components, events, forms, routing, memoization, and more. At the end of these exercises, students will be to work with react for any kind of project requiring experts as front-end developers.

1. Components

1. Create a functional component that displays a welcome message and another component that displays the current date.
2. Build a parent component with a list of hobbies and a child component that displays each hobby.
3. Create a reusable button component with customizable text and color props.

4. Design a profile card component that accepts `name`, `age`, and `email` as props.
5. Build a component to display a list of items using the `.map()` function.

2. Events

6. Create a button that toggles between "ON" and "OFF" states when clicked.
7. Build a counter component with increment and decrement buttons.
8. Create a component that changes the background color of a `div` when hovered over.
9. Build a form that logs the user's input to the console when the form is submitted.
10. Create a dropdown menu component that shows and hides options when clicked.

3. Forms Handling

11. Create a simple login form with fields for username and password.
12. Build a controlled form with a single input field and display the entered value below the form.
13. Design a form with validation for email and password fields.
14. Create a multi-step form where users can fill in details step-by-step (e.g., name, address, and payment information).
15. Build a form with multiple checkbox inputs and display the selected options.

4. Routing

16. Set up a basic React Router with three pages: Home, About, and Contact.
17. Create a route for a product details page that accepts a product ID as a URL parameter.
18. Build a navigation bar with links to different routes using React Router.
19. Create a "Not Found" page that displays when a user navigates to an undefined route.
20. Implement nested routes for a blog with a main blog page and individual post pages.

5. React.memo

21. Create a parent component that passes a prop to a child component. Use React.memo to prevent the child component from re-rendering unnecessarily.
22. Build a component that displays a counter. Use React.memo to optimize a list of unrelated items from re-rendering.
23. Create a "heavy calculation" component that uses React.memo to optimize performance.
24. Build a todo list app where the list component is memoized to prevent re-renders when the input changes.
25. Implement a component that shows live time updates but prevents unnecessary re-renders of static UI parts.

Registration Forms

Here are **5 registration form scenarios** for specific use cases:

26. Registration Form for Lecturers

- Fields: Name, Email, Subject, Phone Number
- Add validation to ensure the email is in the correct format and phone number is numeric.

27. Registration Form for Students

- Fields: First Name, Last Name, Email, Student ID, Date of Birth
- Validate the Student ID to ensure it contains only alphanumeric characters.

28. Registration Form for Drivers

- Fields: Name, License Number, Phone Number, Vehicle Type
- Add a dropdown to select the vehicle type (e.g., car, truck, motorcycle).

29. Registration Form for Books

- Fields: Book Title, Author, ISBN, Published Year
- Validate that the Published Year is a four-digit number.

30. Registration Form for Modules

- Fields: Module Name, Module Code, Description, Credits
- Ensure the Credits field accepts only numeric values and is required.

How to Approach Each Exercise

1. **Setup**:
    - Create a React app using npx create-react-app.
    - Install necessary dependencies (react-router-dom, etc.) where applicable.
2. **Implement**:
    - Use functional components and hooks like useState and useEffect as needed.
    - Use props and state to manage and display data.
3. **Test**:
    - Test interactions like button clicks, form submissions, and routing.
    - Verify form validations and conditional rendering logic.

React JS References:

https://reactforbeginners.com/

https://react.dev/

https://www.frontendmentor.io/

https://www.w3schools.com/react/default.asp

https://www.youtube.com/watch?v=j942wKiXFu8&list=PL4cUxeGkcC9gZD-Tvwfod2gaISzfRiP9d