Day 2 CC1: Practical Coding Application: NodeJS & React Overview

Final Objective:

Develop a simple user management system where users can be added and displayed dynamically using React components and Node.js as the backend. The application will demonstrate key concepts like JSX, Virtual DOM, functional & class components, props, and event handling.

User Stories & Tasks

User Story 1: Setup & Initialization

As a developer, I want to set up a project with Node.js and React so that I can start building the application.

- Initialize a Node.js project with Express.
- Create a React project using Vite or Webpack manually.
- Set up Webpack and Babel for React compilation.

User Story 2: Backend API with Node.js

As a developer, I want to create an API in Node.js that provides user data so that my React frontend can fetch and display it.

- Set up an Express server.
- Create a simple REST API (/users) that returns a list of users.
- Enable CORS to allow React to fetch data from the backend.

User Story 3: Display Users in React

As a user, I want to see a list of users displayed dynamically so that I can view the available users.

- Create a functional component UserList to display user data.
- Use fetch or axios to get data from the backend.
- Render users using JSX and the Virtual DOM.

User Story 4: Add a New User

As a user, I want to add a new user through a form so that I can dynamically update the list.

- Create a UserForm component with an input field and a submit button.
- Use state to manage form input.
- On submit, send a POST request to the backend API to add a new user.
- Update the user list dynamically using React state.

User Story 5: Understanding Component Types & Props

As a developer, I want to implement both functional and class components so that I understand their differences.

- Convert UserList into a class component to demonstrate lifecycle methods.
- Pass user data as props from a parent component.
- Implement event handling for form submission.

Expected Outcome

By the end of this 120-minute activity, participants should have:

1. A working React-Node.js application that fetches and displays user data.
2. Experience in setting up Webpack and Babel manually for React.
3. Hands-on understanding of functional & class components, props, and event handling.
4. A clear understanding of how the Virtual DOM updates dynamically in React.
5. Successfully submitted and displayed user data through the application.
