React Lecture – 1

Introduction to React

What is React ?

React is a popular JavaScript library for building user interfaces. It's created and maintained by Meta (formerly Facebook) and a community of developers.

Features of React

ReactJS is a powerful JavaScript library known for its features that enhance UI development. Some key features of ReactJS include:

JSX (JavaScript Syntax Extension): JSX combines HTML and JavaScript, making code more readable and understandable. It simplifies the creation of React elements by allowing developers to embed JavaScript objects within HTML elements

Virtual DOM: React uses a virtual DOM to optimize performance by updating only the necessary parts of the actual DOM, leading to faster rendering and more efficient updates

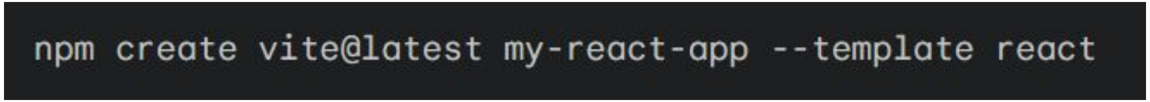
Components: React is component-based, allowing developers to create reusable and independent components. This architecture simplifies the management of complex UIs and enhances code maintainability

One-Way Data Binding: React enforces one-way data flow, ensuring that data is passed down from parent components to child components. This approach enhances predictability in data flow and simplifies debugging.

Declarative User Interface: React follows a declarative paradigm where developers describe the desired outcome, and React handles the process of updating and rendering the UI based on data changes. This feature makes React code more readable and user-friendly

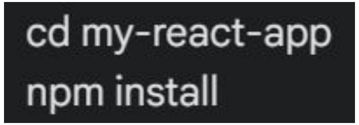
How to create a React Application

1. We can create React application using Vite



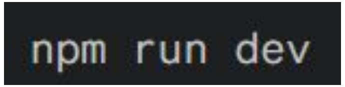
```
npm create vite@latest my-react-app --template react
```

2. Navigate to project and install dependencies



```
cd my-react-app
npm install
```

3. Run the application



```
npm run dev
```

React File Structure

1. **Root Directory**

   index.html: This is the entry point for your application. It loads the necessary scripts, including the main React component.

   main.jsx: This file acts as the initial entry point for your JavaScript code. It typically imports and renders the root React component (App.jsx).

   package.json: This file manages project dependencies, scripts, and other configuration details for your application.

   vite.config.js (optional): This file allows you to customize Vite's behavior for your project. It's optional and only present if you need specific configurations.

React File Structure

2. Source Directory (src): This directory houses the core source code of your React application:

App.jsx: This is the fundamental React component for your entire application. It serves as the starting point for your UI and can import other components to build the user interface.

Additional Components (src//*.jsx or src/**/*.tsx):** As your application grows, you'll likely create more React components to represent smaller, reusable UI elements. These components typically reside within the src directory, using subdirectories for further organization if needed (e.g., src/components, src/features).

Creating our First Component

- Define a function
  - React components are regular JavaScript functions, but their names must start with a capital letter or they won't work!
- Return a markup from the function
  - But if your markup isn't all on the same line as the return keyword, you must wrap it in a pair of parentheses:
- Export the function

How to use the created Component

1. Import the component where you want it to render
2. Use the component Ex: <**Component** />