**IP Workshop Manual**

***Speaker:*** *Ninad Walke, B.E. Computer Engineering*

***Team:***

*Shweta Shimpi, B.E. Computer Engineering*

*Shreya Patil, B.E. Computer Engineering*

*Mandaar Pawar, B.E. Computer Engineering*

* GitHub link for the IP-Workshop Repository:

[*https://github.com/NinadWalke/ip-workshop*](https://github.com/NinadWalke/ip-workshop)

**Commands**

1. Create React App: npx create-react-app frontend
2. Create Vite React: npm create vite@latest
3. Install Axios for Frontend: npm install axios
4. Install Backend Dependencies: npm install nodemon express cors body-parser

**Links**

1. NodeJS: [*https://nodejs.org/en/download*](https://nodejs.org/en/download)
2. VSCode: [*https://code.visualstudio.com/docs/?dv=win64user*](https://code.visualstudio.com/docs/?dv=win64user)

**Theory**

1. Features of React.js:

* **Virtual DOM** – React uses a virtual representation of the DOM to optimize updates. Instead of re-rendering the entire UI, it only updates the parts that changed, improving performance.
* **Component-Based Architecture** – UI is built using reusable, self-contained components. This makes code modular, easier to maintain, and encourages reusability.
* **JSX (JavaScript XML)** – React allows writing HTML-like syntax directly in JavaScript, which makes the code more readable and easier to understand.
* **One-Way Data Binding** – Data flows in a single direction from parent to child components, making the app more predictable and easier to debug.
* **Declarative UI** – You describe what the UI should look like for a given state and React handles the rendering. This reduces bugs and makes UI logic easier to follow.
* **React Hooks** – Functional components can use state and other React features without writing class components, making code simpler and more modern.