**Lecture 1: Creating a React Project with Create React App or Vite**

**1. React Project Creation Options**

* **Create React App (CRA)**
  + Command: npx create-react-app my-app
  + **Explanation**: create-react-app is a popular tool used to create React applications quickly. However, it can be **bulky** and **slow** due to the many unnecessary files and dependencies included in the setup.
  + **Downsides**: Takes time to install, has many default configurations and files you may not need for simple projects.
* **Vite (Bundler)**
  + Vite is a fast, lightweight alternative to Create React App. It skips some of the heavy initial setups, allowing for faster app creation.
  + Command: npm create vite@latest my-app
  + After creating a Vite app, you install dependencies with:
  + **Benefits**: Faster startup, minimal boilerplate, modern build tool. No unnecessary files are included at the start, and it's optimized for performance.

**2. React to React-DOM**

* **In a Web App**: React works with **React-DOM**, a package that manages how React renders components on a web page.
* **In a Mobile App**: For mobile development, you use **React Native** instead, which is tailored for mobile platforms (iOS and Android).

**3. npm vs npx**

* **npm**: Stands for **Node Package Manager**. Used to manage packages (install, update, uninstall) in your project.
* **npx**: Stands for **Node Package Executor**. Used to execute packages or commands without globally installing them. Example: npx create-react-app allows you to create a new React project without having to install the create-react-app globally.

**4. Package Management**

* **npm install** or npm i: Used to install project dependencies listed in package.json.

**5. package.json**

* **Purpose**: This file acts as the **entry point** for your project. It contains:
  + Project **name**, **version**, and **dependencies** (all the libraries and frameworks your project relies on).
  + **Scripts**: Commands to run the project (e.g., npm run start to start the project).
  + **Browserlist**: Specifies which browsers the project supports.

**6. Starting the Project**

* Command: npm run start / npm start
  + This starts the development server.

**Lecture 2: Understanding Project Structure**

**1. How React Integrates into HTML**

* **index.html**: This is the main HTML file where the React app gets injected. React essentially **ejects** into this HTML file and uses it as the foundation of the app.
* <noscript>: This tag inside the HTML file displays a message if JavaScript is disabled, advising users to enable it for the React app to work.

**2. React Virtual DOM**

* **Virtual DOM**: React creates a virtual copy of the DOM (Document Object Model). When changes happen in the app, React updates the virtual DOM first, compares it with the real DOM (using a process called **reconciliation**), and efficiently updates only the necessary parts of the real DOM.
* **Why Important?**: This ensures fast and optimized updates, enhancing app performance.

**3. JSX (JavaScript XML)**

* **JSX** allows you to write HTML-like syntax directly inside JavaScript.
* Example:

const element = <h1>Hello, world!</h1>;

* **Custom Tags**: You can create your own custom components and use them like HTML tags. React components are written in JSX and must start with a **capital letter** (e.g., <MyComponent />).
  + Example:

**function MyComponent() {**

**return <div>This is a custom component!</div>;**

**}**

**4. Other Important Files**

* **package-lock.json**: Locks the specific versions of all installed packages, ensuring consistent behavior across different environments.
* **manifest.json**: Used in Progressive Web Apps (PWA) or mobile apps to manage how the app is installed and looks when saved on a home screen.
* **web-vitals.js**: Tracks the app's performance, helping developers optimize loading times and responsiveness.