**React JS**

**Module-1**

**What is React?**

The React.js framework is an open-source JavaScript framework and library developed by Facebook. It’s used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

Back in 2011, Facebook had a massive user base and faced a challenging task. It wanted to offer users a richer user experience by building a more dynamic and more responsive user interface that was fast and highly performant.

Jordan Walke, one of Facebook’s software engineers, created React.

**React with CDN**

Link: <https://legacy.reactjs.org/docs/cdn-links.html>

Both React and ReactDOM are available over a CDN.

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

**First React Code with CDN**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>React CDN</title>

  </head>

  <body>

    <div id="root"></div>

    <script

      crossorigin

      src="https://unpkg.com/react@18/umd/react.development.js"

    ></script>

    <script

      crossorigin

      src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"

    ></script>

    <script>

      const message = React.createElement("h1", {}, "Hello form React");

      const root = ReactDOM.createRoot(document.getElementById("root"));

      root.render(message);

    </script>

  </body>

</html>

**Module-2**

**Setting up NPM**

**NPM:** NPM, which stands for Node Package Manager, is essentially a giant toolbox for JavaScript developers. It's actually two things working together:

1. **A giant online repository:** This repository contains a massive collection of pre-written code snippets, called packages, that developers can easily borrow and use in their own projects. There are millions of these packages available, so you can find code for just about anything you can imagine.
2. **A handy downloader and manager:** NPM includes a command-line tool that you can use to search for packages in the repository, download them to your project, and keep track of which ones you're using. It also helps manage any other packages that your downloaded packages rely on (dependencies). This saves you a ton of time and effort compared to writing all that code yourself.

NPM is free to use and comes bundled with Node.js, a popular runtime environment for running JavaScript code outside of a web browser. So if you're working on a JavaScript project, there's a good chance you'll be using NPM!

**Initialise NPM:**

F:\WebDevelopment\React.JS\React form Scaler\Lession\_1> npm init -y

Wrote to F:\WebDevelopment\React.JS\React form Scaler\Lession\_1\package.json:

{

"name": "lession\_1",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC"

}

**Setup React Project using Vite**

Link: <https://vitejs.dev/guide>

Here's a detailed guide on setting up a React project using Vite, a blazing-fast development server:

**Prerequisites:**

* **Node.js and npm (or yarn):** Ensure you have Node.js (version 14.18+ or 16+) and npm (Node Package Manager) or yarn installed on your system. You can verify their installation by running node -v and npm -v (or yarn -v) in your terminal. If not installed, download them from the official Node.js website (<https://nodejs.org/en/download/package-manager>).

**Steps:**

1. **Create a New Project Directory:**

Open your terminal or command prompt and navigate to your desired workspace directory. Create a new project directory using:

Bash

mkdir my-react-vite-project

cd my-react-vite-project

1. **Initialize the Project using Vite:**

Use either npm or yarn to initialize the project with Vite's template:

**Using npm:**

Bash

npm create vite@latest my-react-vite-project --template react

**Using yarn:**

Bash

yarn create vite@latest my-react-vite-project --template react

This command will create a new project directory (my-react-vite-project) and initialize it with the React template. Vite will prompt you with some basic configuration options, such as package manager and variant selection (choose the defaults for most cases).

1. **Install Dependencies:**

After the initialization is complete, Vite will install the necessary dependencies for your React project. This may take a few minutes depending on your internet speed.

1. **Start the Development Server:**

Once the dependencies are installed, you can start the development server using:

Bash

npm run dev

**Using yarn:**

Bash

yarn dev

This will start Vite's development server, typically running at http://localhost:5173/ by default (the exact port number might vary). You can open this URL in your browser to see the default React app created by Vite.

**Project Structure:**

Vite creates a well-structured project directory with the following key components:

* index.html: The main HTML file that serves as the entry point for your React app.
* src folder: Contains your React components, styles, and other application code.
* package.json: The project's manifest file that lists dependencies and scripts.
* public folder (optional): Intended for static assets like images or fonts that are directly accessible by the browser without bundling.

**Hot Module Replacement (HMR):**

Vite offers a great feature called Hot Module Replacement (HMR). This allows you to see changes you make to your code reflected in the browser almost instantly, without the need for a full page reload. This significantly improves the development experience for React apps.

**Additional Considerations:**

* **Customizing Configuration (Optional):** You can optionally customize Vite's configuration in the vite.config.js file for advanced usage like setting custom aliases, configuring a proxy server, or adding plugins.
* **Deployment:** Refer to the Vite documentation for specific instructions on building production-ready bundles of your React app.

By following these steps, you'll have a basic React project set up and running using Vite. Enjoy the fast development experience with HMR as you build your React applications!