■ REACT • VITE • NODE INSTALL

■ HOW REACT PROGRAM STARTS

Whenever we write React code, the entry point is always **index.js** (CRA) or **main.jsx** (Vite). These files create the root of the React project. From **index.html**, React takes the **<div id="root">** element (in the public folder) and makes it the root of the React app. Whatever we return inside the **App** component is finally rendered into this root. React then converts the JSX code into browser-understandable JavaScript.

■ 1. Create React App using CRA

Step 1:
Install CRA (if not installed)

```
npm install -g create-react-app
```

Step 2:
Create the project

```
npx create-react-app my-app
```

Step 3:
Go inside folder

```
cd my-app
```

Step 4:
Start the app

```
npm start
```

✔ **Runs on: http://localhost:3000**

✔ **Entry file: src/index.js**

src/index.js

```
import React from 'react'; import ReactDOM from 'react-dom/client';
import App from './App'; const root =
ReactDOM.createRoot(document.getElementById('root')); root.render(<App
/>);
```

■ What happens internally?

CRA uses Webpack as a bundler. Webpack loads index.js as the entry file. React renders App.js. Browser loads public/index.html with a **<div id="root">**.

■ 2. Create React App using Vite

Step 1:
Create project using Vite command

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

Step 2:
Choose:
✔ React
✔ JavaScript

Step 3:
Go inside folder

```
cd my-app
```

Step 4:
Install dependencies

```
npm install
```

Step 5:
Start the app

```
npm run dev
```

✔ Runs on: http://localhost:5173

✔ Entry file: src/main.jsx

main.jsx in Vite

```
import React from "react"; import ReactDOM from "react-dom/client";
import App from "./App";
ReactDOM.createRoot(document.getElementById("root")).render( <App /> );
```

■ What happens internally?

Vite does not bundle everything at the start. It loads files using native ES modules. It is very fast because it compiles only what is needed. Browser loads index.html directly from the root folder.

■ When do we use npm init -y?

We use **npm init -y** when we want to create a Node.js project (backend). Examples:
- Express server
- Node scripts
- Any backend app Command:
`npm init -y` This creates the **package.json** file. After that, you install the dependencies you

need (for example: Express). Example backend start:

```
npm install express
node server.js
```

Note: This PDF was generated to match the user's requested formatting: bold headings, colors, code blocks, and clear step-by-step instructions.