

friendly. This is especially important for single-page applications (SPAs), where traditional client-side rendering may not be optimal for SEO.

8. Easy Integration with Other Technologies

React is often used in conjunction with other libraries and frameworks to handle various tasks such as state management (Redux), routing (React Router), and form validation. Its flexibility and ease of integration make it a powerful tool for building complex applications.

Workflow of ReactJS

Step 1: Install NodeJS

- **NodeJS** is a JavaScript runtime that allows running JavaScript code outside the browser. It is essential for developing React applications as it manages dependencies through **npm (Node Package Manager)**.
- Visit <https://nodejs.org/en> and download the **LTS (Long-Term Support)** version for stability.
- Follow the installation wizard and ensure **npm** is installed alongside NodeJS (default setting).

Step 2: Install Visual Studio Code (VS Code)

- **VS Code** is a lightweight yet powerful code editor widely used for React development due to its features and extensions.
- Download it from <https://code.visualstudio.com/download>.
- Install it by following the on-screen instructions.

Step 3: Configure VS Code with Extensions

Extensions enhance the coding experience in VS Code. Install these:

1. **npm**: Helps with Node.js package management directly in the editor.
2. **Prettier**: Automatically formats your code to ensure consistency.
3. **ESLint**: Highlights errors and enforces coding standards.
4. **Bracket Pair Colorizer**: Makes nested brackets easily distinguishable.
5. **ReactJS Snippets**: Provides ready-to-use code snippets for React components.
6. **Live Server**: Allows real-time preview of static web pages.

How to Install Extensions:

- Open **Extensions Marketplace** (Ctrl+Shift+X in VS Code).
- Search for the extension and click **Install**.

Step 4: Check NodeJS and npm Versions

After installing NodeJS, confirm its installation by checking versions:

- Open the terminal in VS Code (Ctrl+` shortcut).
- Run the following commands:
 1. `node -v` - Displays the installed NodeJS version.
 2. `npm -v` - Displays the npm version.
 3. `npx -v` - Verifies if npx (package executor) is installed.

Step 5: Create a ReactJS Application

- Use the command `npx create-react-app lprac` in the terminal.
 - **npx** ensures you're using the latest React template without globally installing it.
 - **Practical** is the name of your project folder.
- This command creates a ReactJS project with a predefined structure and installs all required dependencies.

Step 6: Open and Navigate the App Directory

- Navigate to the project directory using:
`cd Practical`
- Open the directory in VS Code:
`code .`
 - **code .** launches the VS Code editor for the current folder.

Step 7: Understand the Folder Structure

- In the opened project, explore the folders and files:
 - **src Folder:** Contains the main logic and components.
 - **App.js:** Entry point for adding React logic.
 - **node_modules:** Auto-generated folder containing all dependencies.
 - **package.json:** Manages project metadata and dependencies.

Step 8: Install Additional Dependencies

- Install required dependencies by running:
npm install
 - This ensures all default packages are updated.
- For specific functionality, install additional plugins like:
npm install @babel/plugin-private-property-in-object
 - **Babel** is used to transpile modern JavaScript into a format compatible with older browsers.

Step 9: Write Code in App.js

- The **App.js** file in the **src** folder is the main component.
- Modify or add logic here to implement features like managing data, displaying lists, or handling user interactions.

Step 10: Run the React Application

- Use the command:
npm start
 - This starts the development server and opens the app in your browser at <http://localhost:3000>.

Scope of React JS

1. Web Development

ReactJS is widely used for creating dynamic and responsive single-page applications (SPAs). Its component-based architecture allows developers to build reusable UI components, making development faster and more efficient. Websites like Facebook and Instagram heavily rely on React for their front-end.

2. Mobile App Development

Using React Native, a framework built on ReactJS, developers can create cross-platform mobile applications for iOS and Android with a single codebase. It provides near-native performance and access to native device features.

3. Enterprise Applications

Many businesses choose ReactJS to build scalable and maintainable enterprise-level applications. Its modular architecture allows easy scaling, while tools like Redux help manage complex state management needs.

4. Progressive Web Applications (PWAs)

ReactJS facilitates the development of high-performance Progressive Web Applications that provide a seamless user experience. PWAs built with ReactJS work offline and offer faster load times, enhancing user engagement.

5. E-Commerce Platforms

ReactJS is often used for building e-commerce platforms due to its ability to create highly interactive and dynamic interfaces. Features like real-time product updates and smooth navigation enhance the shopping experience.

6. Dashboard Development

ReactJS is ideal for building data-intensive dashboards and analytics tools. Libraries like Recharts and D3.js can be integrated with React to create visually appealing and interactive data visualizations.

7. Gaming Applications

If npx not working, execute the following command in PowerShell in Administrator node:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```