Introduction to React

- What is React?
- Working of DOM
- Problems with JS
- Working of React
- JS Vs React
- Intro to Components

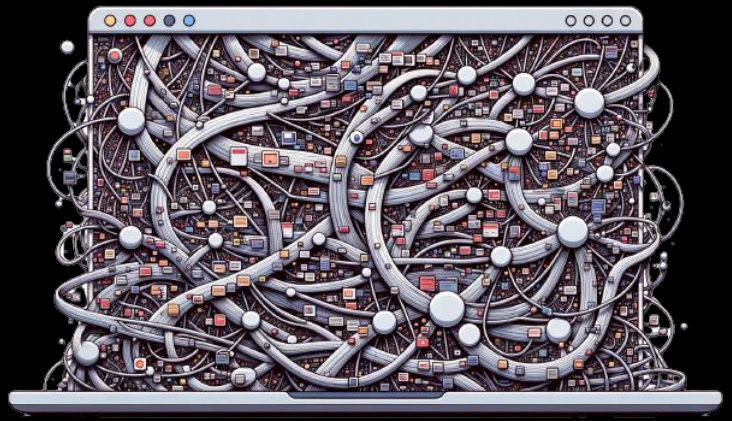
React

What is React

1. JavaScript library to build Dynamic and interactive user interfaces
2. Developed at Facebook in 2011.
3. Currently most widely used JS library for front-end development.
4. Used to create single page application (page does not re-load).

React

Working of DOM



1. Browser takes HTML and create DOM.
2. JS helps us modify DOM based on user actions or events.
3. In big applications, Working with DOM becomes complicated.

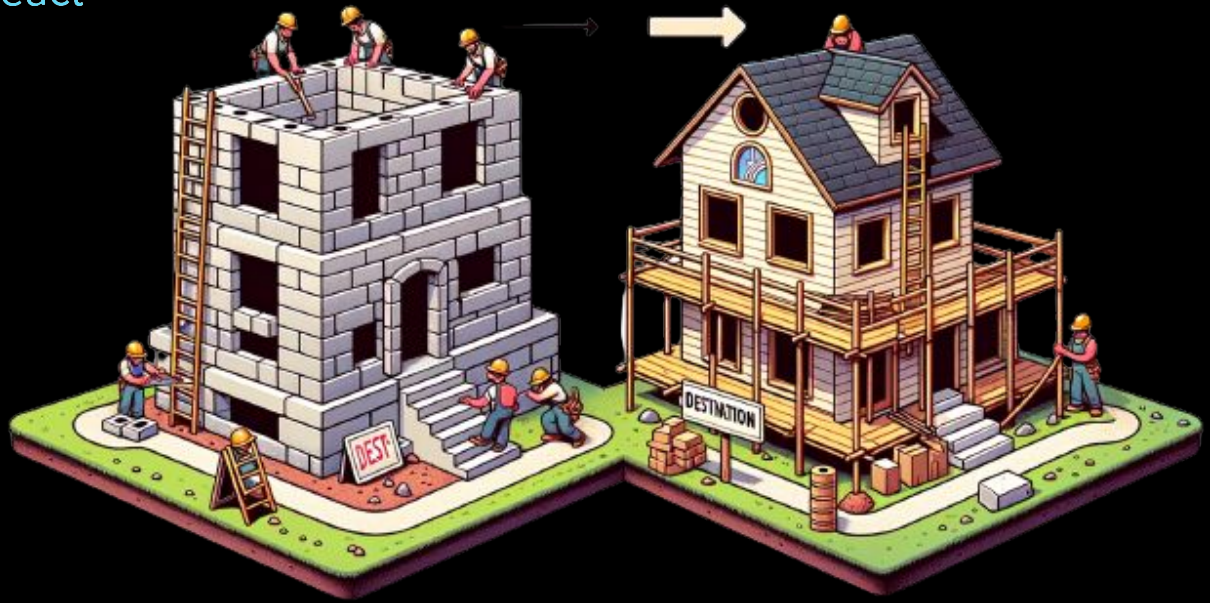
Problems with JavaScript

1. React has a simpler mental model
2. JS is cumbersome
3. JS is Error-prone
4. JS is Hard to maintain

Working of React

1. No need to worry about querying and updating DOM elements.
2. React creates a web page with small and reusable components
3. React will take care of creating and updating DOM elements.
4. IT saves a lot of time, cheezein aasan hai, pahele se likhi hui hain

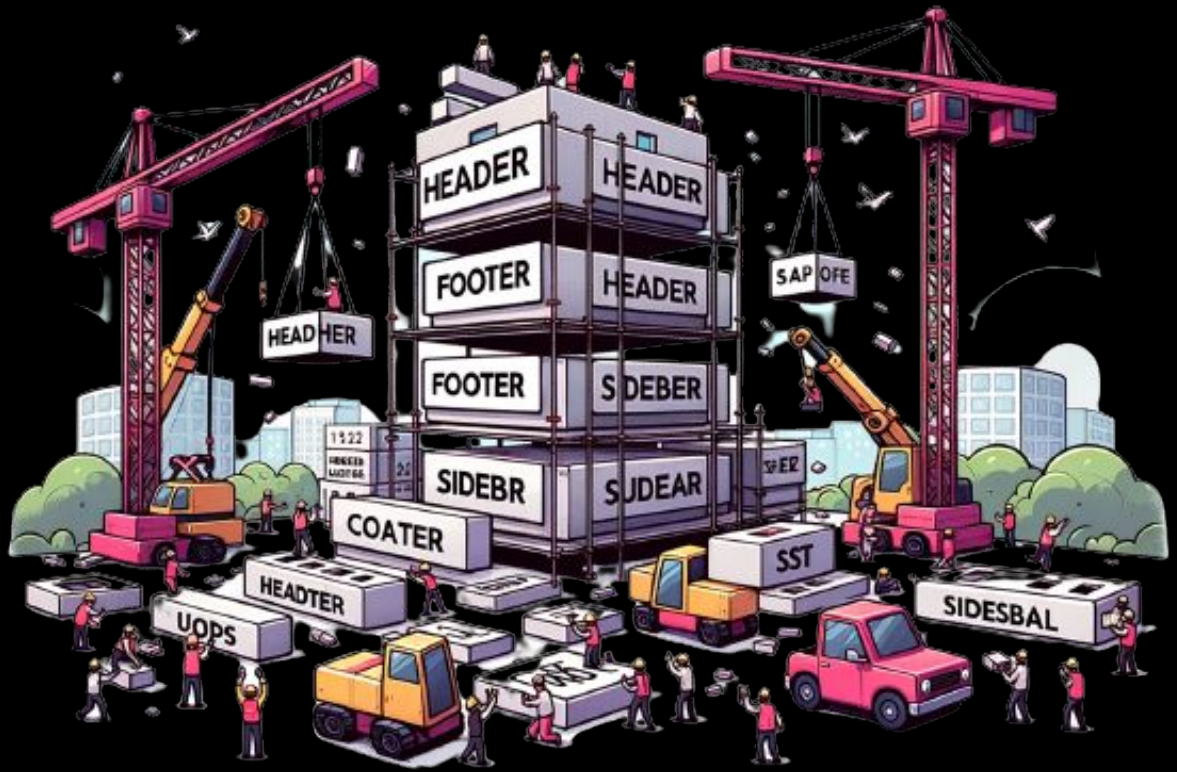
JS Vs React



1. JS is imperative: You define steps to reach your desired state.
2. React is Declarative: You define the target UI state and then react figures out how to reach that state.

React

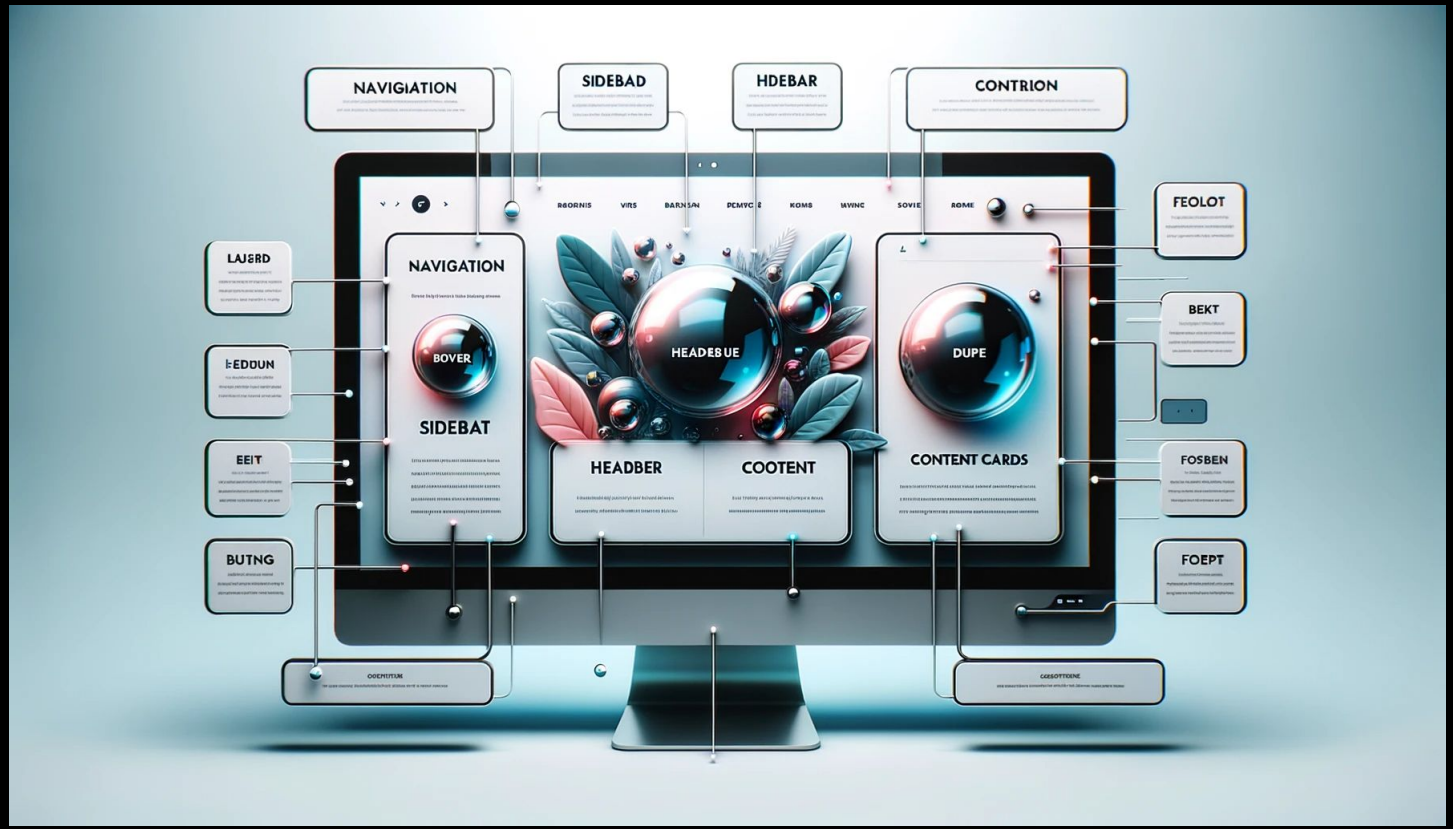
Introduction to Components



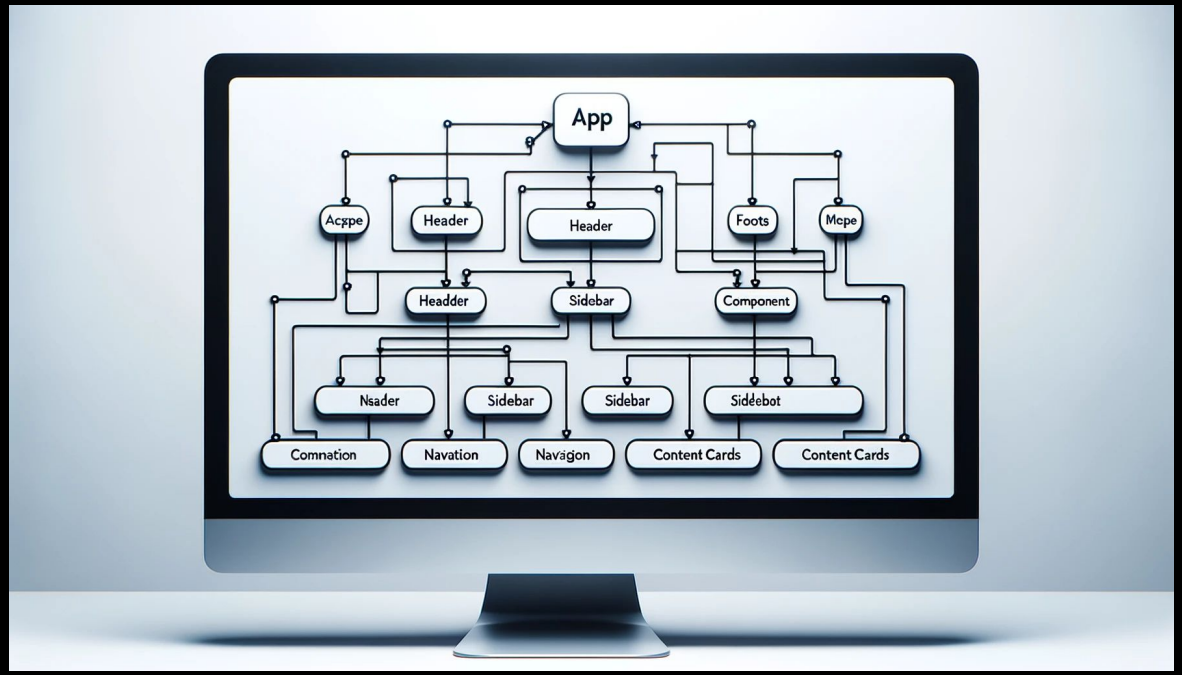
Components help us write reusable, modular and better organized code.

React

Introduction to Components



Introduction to Components



React application is a tree of components with App Component as the root bringing everything together.

Create a React App

Official tool is CRA(Create React APP)

Quick Start

```
npx create-react-app my-app
cd my-app
npm start
```

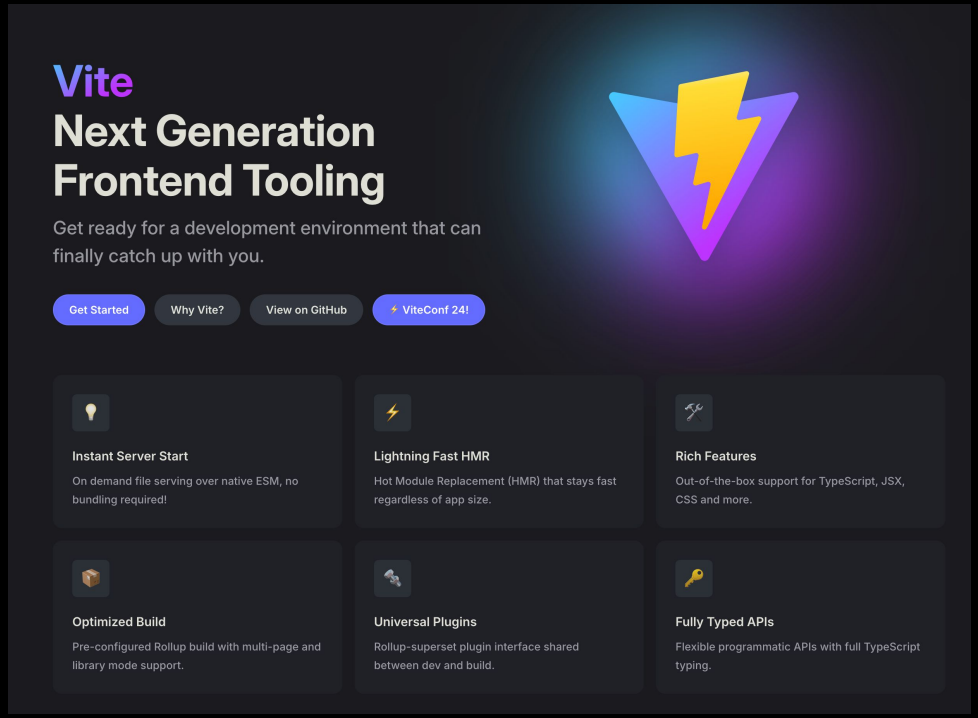
- npm: You need to install a package first (using npm install) before you can use it.
- npx: You can directly run a package without installing it, using npx <package-name>.

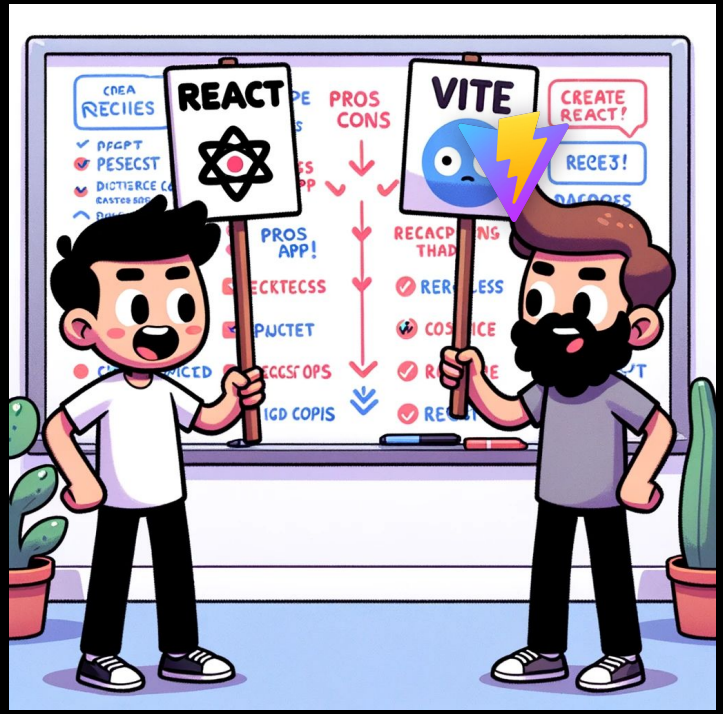
Project Structure

React

- node_modules
- public
  - favicon.ico
  - index.html
  - logo192.png
  - logo512.png
  - manifest.json
  - robots.txt
- src
  - App.css
  - App.js
  - App.test.js
  - index.css
  - index.js
  - logo.svg
  - reportWebVitals.js
  - setupTests.js
- .gitignore
- package-lock.json
- package.json
- README.md

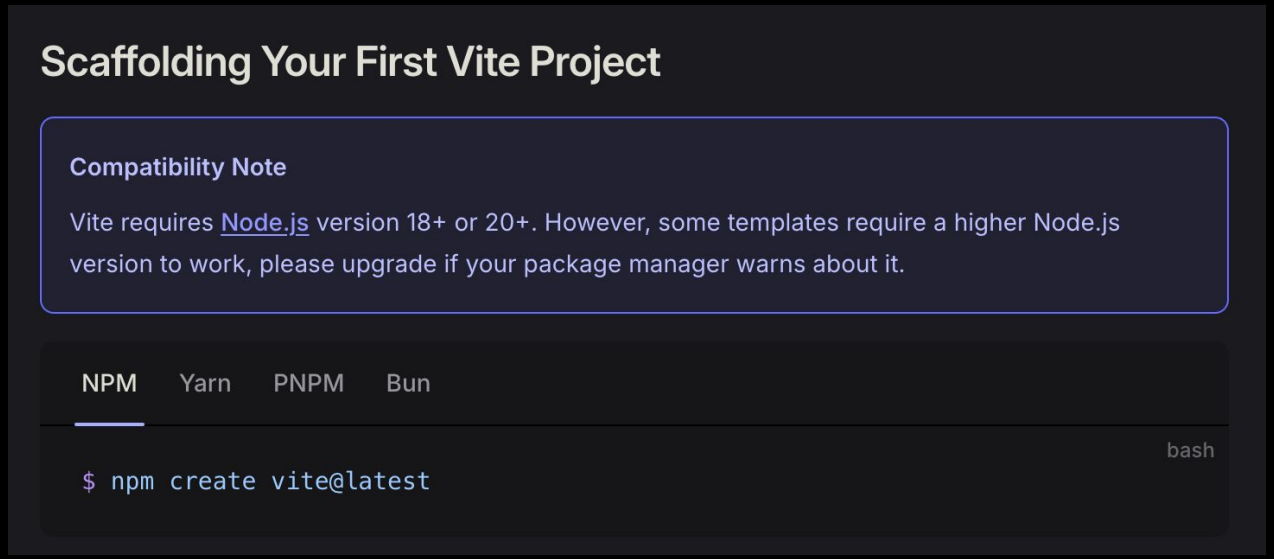
Create a React App





1. Vite is a modern tool to create React Project.
2. Vite produces Quick and Small bundle size.

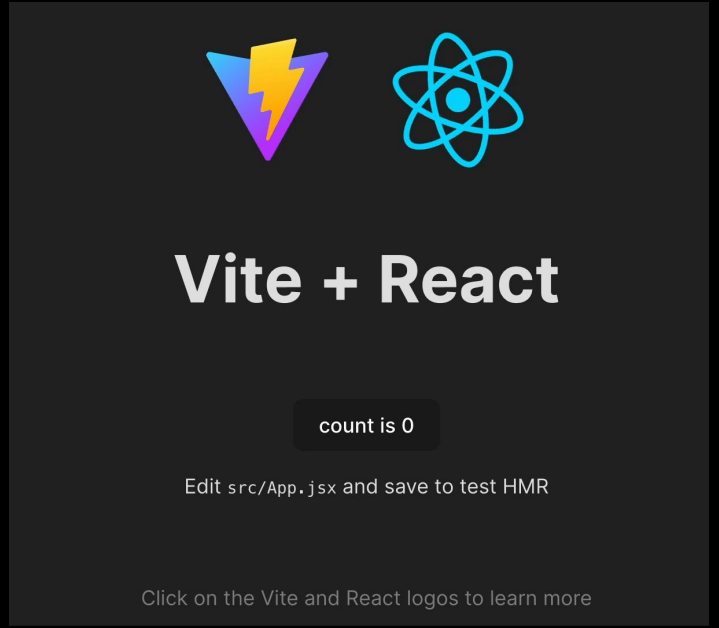
Create a React App



1. Vite is a modern tool to create React Project.
2. Vite produces Quick and Small bundle size.

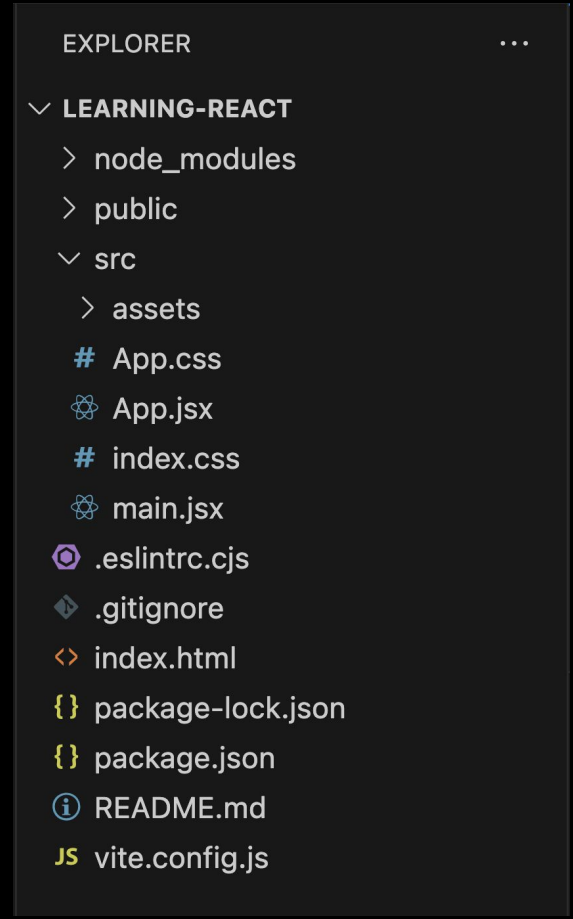
Create a React App

```
Scaffolding project in /Users/prashantjain/workspace/Test Project/react/test2...

Done. Now run:

  cd test2
  npm install
  npm run dev

prashantjain@Prashants-Mac-mini react % cd test2
prashantjain@Prashants-Mac-mini test2 % npm install

added 215 packages, and audited 216 packages in 26s

99 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
prashantjain@Prashants-Mac-mini test2 % npm run dev

> test2@0.0.0 dev
> vite
```



Project Structure

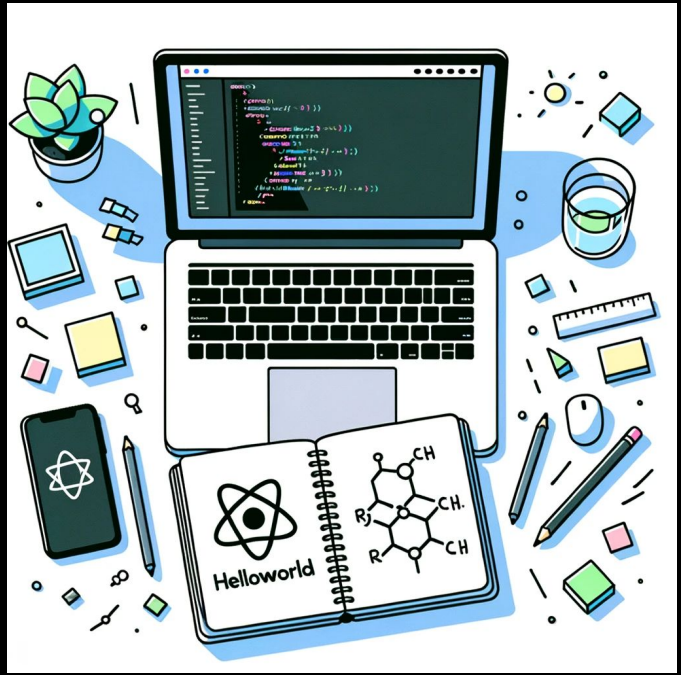
1. node_modules/ has all the installed node packages
2. public/ Directory: Contains static files that don't change.
3. src/ Directory: Main folder for the React code.
    1. components/: Reusable parts of the UI, like buttons or headers.
    2. assets/: Images, fonts, and other static files.
    3. styles/: CSS or stylesheets.
4. package.json contains information about this project like name, version, dependencies on other react packages.
5. vite.config.js contains vite config.




React

First React Component

- File Extensions
- What is JSX?
- Class vs Function Components
- Exporting component
- Other important Points
- Dynamic Components
- Reusable Components



React

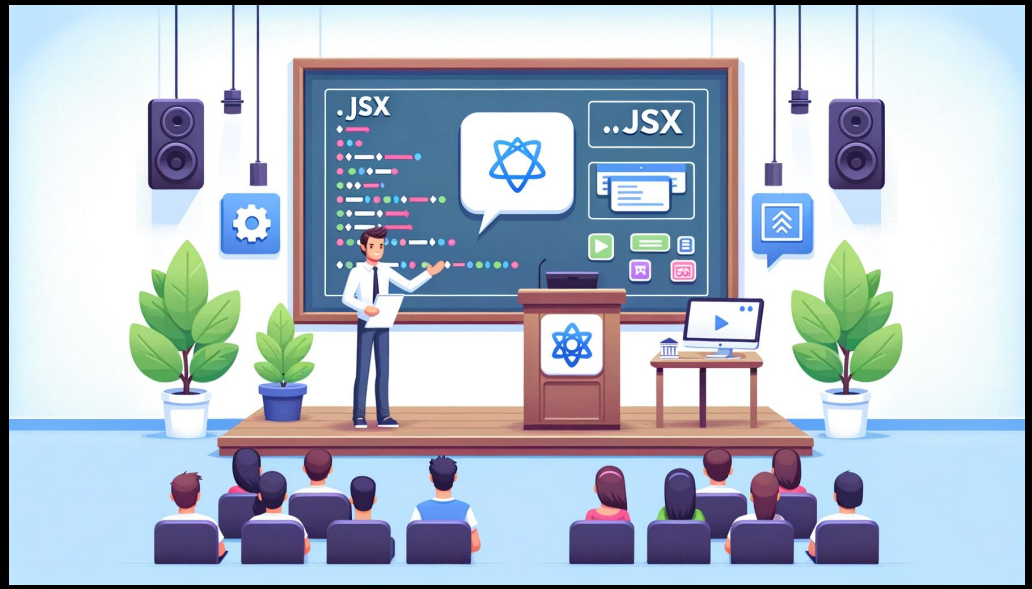
File Extensions

.JS

- Stands for JavaScript
- Contains regular JavaScript code
- Used for general logic and components

.JSX

- Stands for JavaScript XML
- Combines JavaScript with HTML-like tags
- Makes it easier to design UI components

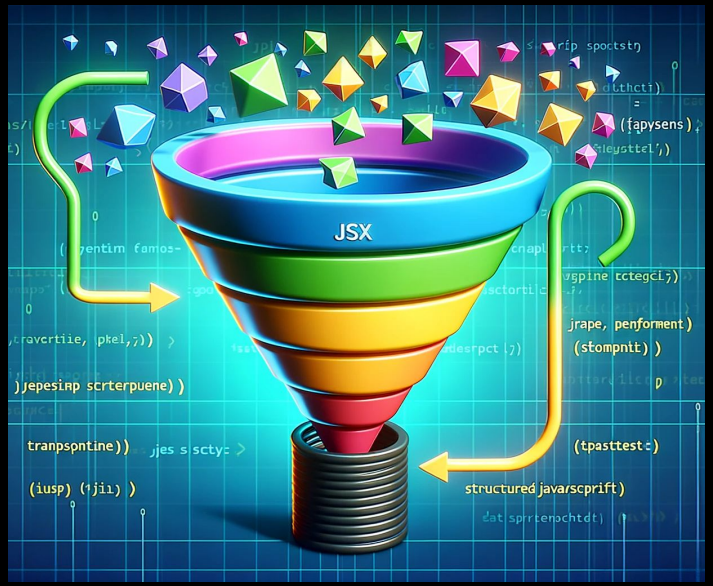


Technically, there is no difference between .js and .jsx files in terms of functionality. The separation is purely a convention to differentiate between files that predominantly contain JSX from those that are purely JavaScript.

What is JSX?

React

1. Definition: JSX determines how the UI will look wherever the component is used.
2. Not HTML: Though it resembles HTML, you're actually writing JSX, which stands for JavaScript XML.
3. Conversion: JSX gets converted to regular JavaScript.
4. Babeljs.io/repl is a tool that allows you to see how JSX is transformed into JavaScript.



JSX Example

```jsx
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
```

React

Class vs Function Components

Class Components

- Stateful: Can manage state.
- Lifecycle: Access to lifecycle methods.
- Verbose: More boilerplate code.
- Not Preferred anymore.

Functional Components

- Initially stateless.
- Can use Hooks for state and effects.
- Simpler and more concise.
- More Popular.

Class Components

```
import React, { Component } from 'react';

class Heading extends Component {
  render() {
    return <h1>{this.props.title}</h1>;
  }
}

export default Heading;
```

Function Components

```
import React from 'react';

const Heading = (props) => {
  return <h1>{props.title}</h1>;
};

export default Heading;
```

Using CSS

est1 > src > # KGButton.css > ...

```css
.custom-button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  font-size: 16px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.custom-button:hover {
  background-color: #45a049;
}
```

```jsx
import React from 'react';
import './KGButton.css';

const KGButton = () => {
  return (
    <button className="custom-button">
      Click Me!
    </button>
  );
};

export default KGButton;
```