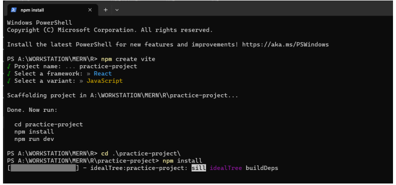
React js

02 April 2023 17:47

How Websites work and how React optimizes it

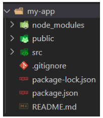
1. Multipage Application (commonly used, not optimized for scaling large usage)
   a. Server and client (Request Response)
      i. Client loads page ----> Request goes to server -----> Server Responds with HTML, CSS, JS Files for working of the requested page
         1. This process happens every time a new request is send for a new page(ex, About, Login, Signup, etc…)
         2. Every page has its own HTML, CSS and JS
2. Single Page applications
   a. Only once a request is send -----> Server Responds with HTML, CSS and JS ------> AFTER THAT JAVASCRIPT TAKES OVER and controls every request from within clients end and no further request is sent to server -------> Makes it faster and scalable for large usage

1. Using Vite to create a teplate app structure
   a.



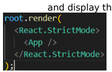
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS A:\WORKSTATION\MERN\R> npm create vite
√ Project name: ... practice-project
√ Select a framework: » React
√ Select a variant: » JavaScript

Scaffolding project in A:\WORKSTATION\MERN\R\practice-project...

Done. Now run:

  cd practice-project
  npm install
  npm run dev

PS A:\WORKSTATION\MERN\R> cd .\practice-project\
PS A:\WORKSTATION\MERN\R\practice-project> npm install
[                  ] - idealTree:practice-project: sill idealTree buildDeps
```

2. Use create-react-app https://create-react-app.dev/
   a. To create basic app folder structure consisting of all the necessary files for creating the app
   b. For permanently storing the library of react app structure
      i. >>nom install -g create-react-app <app_name>
   c. For using the libraries for work environment on temporary basis which will not use much of your space'
      i. >>npx create-react-app <app_name>
3. React apps are build using **Components(building blocks)**
   a. Each component has ability to render separately
   b. Use to create single page apps
   c. Seamless due to single component rendering rather than whole website render to change a small update
4.



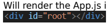
```
my-app
  node_modules
  public
  src
  .gitignore
  package-lock.json
  package.json
  README.md
```

   a. About these
   b. **Public**
      i. Includes files needed to build the site
      ii. Is publicly available, must NOT include sensitive information
   c. src/App.js
      i. This file is where you will create the app which will get rendered through Javascript and display the final output
5.



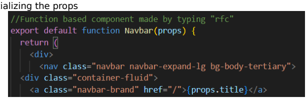
```
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

   a. Will render the App.js in root that's in the index.html
   b.



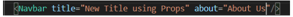
```
<div id="root"></div>
```

6. To start the app
   a. >> npm start
7. Prop and State
   a. PROPS (properties)
   b. Initializing the props
      i.



```
//Function based component made by typing "rfc"
export default function Navbar(props) {
  return (
    <div>
      <nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="/">{props.title}</a>
```

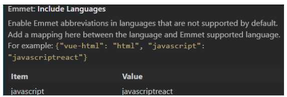
      ii. Setting the props value for different usage
         1.



```
<Navbar title="New Title using Props" about="About Us"/>
```

      iii. propTypes and defaultProps

8.



Emmet: **Include Languages**

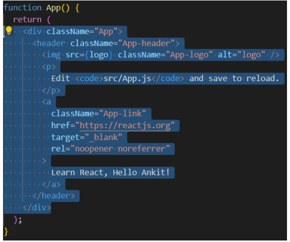
Enable Emmet abbreviations in languages that are not supported by default. Add a mapping here between the language and Emmet supported language. For example: {"vue-html": "html", "javascript": "javascriptreact"}

Item	Value
javascript	javascriptreact

   a. For better autocompletion of React wrt to javascript reserved names running with HTML

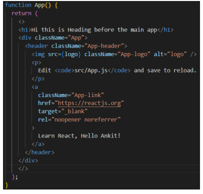
JSX

1. Lets you write HTML inside Javascript code giving you allthe features of Javascript to be used for HTML code(conditional, etc…)
2. Must always be wrapped inside a parent tag (<div></div> or <></>) as JSX ALWAYS RETURN SINGLE ELEMENT
3. {JS CODE} - curly brackets lets you write javascript in your HTML code
4. {/*Comment in JSX*/}
5. Two types of components in React
   a. Function Based & Class Based (old)
      i.



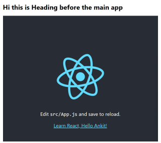
```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"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

      ii. Function based component function App() { JSX }
      iii. The highlighted part is JSX
         1. JSX Fragment??
            a. All app development need to be enclosed inside **return() under the function App().**
            b. We use JSX fragment when we need additional components that we need to run first before the main app
            c. We then enclose the whole thing inside blank tag <> new tag + function App()</> under return(<> The whole stuff </>)
               i.



```
function App() {
  return (
    <>
    <h1>Hi this is Heading before the main app</h1>
    <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, Hello Ankit!
        </a>
      </header>
    </div>
    </>
  );
}
```

               ii.

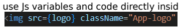


Hi this is Heading before the main app

Edit src/App.js and save to reload.

Learn React, Hello Ankit!

      iv. Why JSX is used??
         1. For developer ease of making the apps
            a. All the components in one place
            b. Can use Js variables and code directly inside HTML using cur brackets { }
               i.



```
<img src={logo} className="App-logo"
```

               ii. Here {logo} is Js defined variable
               iii. We use className instead of class for attribute "a" because **class** is reserved for Js.

Javascript essentials for React in brief: include js notes along with it

1. Datatypes
2. Functions
3. Objects with function value
4. Events (Event Listener)
5. String Methods (slice, substring, indexof,lastindexof)
6. Arrays(forEach, map, reverse)
7. Dates
8. Loops
9. Break Continue
10. Strict mode (To make sure that there's no bug due to illegal code, bad practices in variable declaration etc)
11. Callbacks
12. Asynchronous nature of jaavascript (Runs the code top to bottom and ensures smooth operation of code without blocking using callbacks)
13. Promises (either gets resolve or gets rejected)
14. **Default export & named export**

Using React RULES

1. Tags which does not have closing tag with it need to be closed
   a. <img src='#' />
   b. <Input type=text />
   c. Every hyperlink reference must be defined
      i. <a href="/"></a> NOT href="#"
2. Better to create a components folder INSIDE your "src" directory (../src/components)
   a. For better organised storing of all your components in your project
   b. Component names should always start with caps
      i. Navbar.js, Modal.js, Textarea.js…..