WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Building dynamic web applications using ReactJS

Vivek Jha

"Department of Computer Science Engineering, Student of Computer Science & Engineering, Arya college of engineering and I.T., Kukas, Jaipur"

ABSTRACT:

In this rapid world of web development, building dynamic and interactive applications is important for improving user experience and engagement. In today's ever-changing technologies, React has been found as a go-to solution for developing robust and highly interactive user interfaces. React empowers developers to make some reusable UI components, supporting efficient rendering and state management, thanks to its component-based architecture. This paper explores the fundamental concepts behind React's functionality including JSX, state and props, hooks, virtual DOM, etc. It also covers integration of backend tech such as node. js and ExpressJS, to develop full stack applications that can communicate with SQL based databases. This paper focuses on best practices for performance optimization, state management, and routing with React Router, aiming to equip developers with actionable insights and strategies to build robust applications. This study evaluates the efficiency of React, a JavaScript library for building user interfaces, and its capacity to adhere to current trends.

1. Introduction

React, commonly referred to as a frontend JavaScript framework, is a JavaScript library created by Meta. With technological advancement, consumers increasingly require high interactivity and seamless experiences, leading to a rising demand for dynamic web applications. React has gained considerable popularity in frontend development owing to its component-based architecture and virtual DOM rendering. REACT — Scalable Dynamic Web Applications This project aims to explore the construction of dynamic web apps using REACT and using backend technologies for browser accessibility. JavaScript utilized alongside Express and communicating with databases such as SQL.

1.1 How does React Work?

React creates a VIRTUAL DOM in memory.
React, commonly referred to as a frontend JavaScript framework, is a JavaScript library created by Meta. With technological advancement, consumers increasingly require high interactivity and seamless experiences, leading to a rising demand for dynamic web applications. React has gained considerable popularity in frontend development owing to its component-based architecture and virtual DOM rendering. REACT — Scalable Dynamic Web Applications This project aims to explore the construction of dynamic web apps using REACT and using backend technologies for browser accessibility. JavaScript utilized alongside Express and communicating with databases such as SQL.

1.2 Brief History of React

The latest version of React.JS is V18.0.0, which was released in April 2022.
The inaugural public release (V0.3.0) took place in July 2013.
React.JS was first employed in 2011 for the News Feed feature of Facebook.
Jordan Walke, a Software Engineer at Facebook, was the developer of it.
The latest version of create-react-app is v5.0.1, which was released in April 2022.
Create React App includes integrated technologies such as Webpack, Babel, and ESLint.

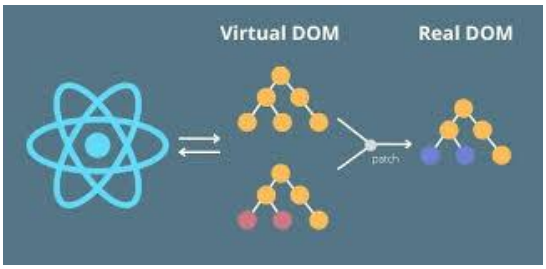
2. Key Features of React

ReactJS is increasingly becoming one of the foremost JavaScript frameworks among web developers. It serves a vital function in the front-end ecosystem. The following are the key features of ReactJS.

2.1 Virtual DOM

The Virtual DOM is a separate DOM created by React. The Virtual DOM represents the real DOM of the current HTML document. Upon alterations in the HTML text, React contrasts the updated virtual DOM with its previous state and exclusively modifies the disparities in the actual DOM. This improves

the rendering efficiency of the HTML document.



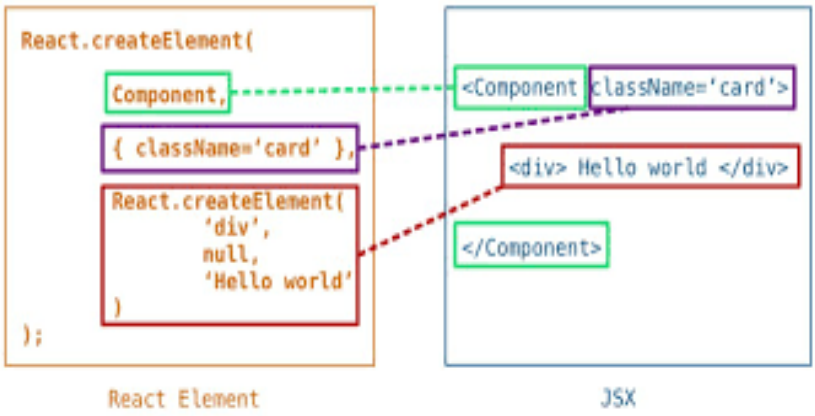
For example, when we create a React component to exhibit the current time by periodically refreshing it using the setInterval() method, React will update only the current time instead of the full component's content.

2.2 Components

React is based on the concept of components. Modern front-end frameworks are essentially grounded in component architecture. Component architecture enables developers to partition a large application into smaller components, which can subsequently be further broken into even smaller components. Disaggregating the application into smaller components improves its lucidity and controllability.

2.3 JSX

JSX is a JavaScript extension that enables the generation of arbitrary HTML elements using syntax similar to HTML. This will promote the creation of HTML documents and improve their clarity. React converts JSX into a JavaScript object that includes a call to the createElement() method of React before execution. It improves the application's efficiency. Moreover, React enables the creation of HTML pages through the use of the pure createElement() function, eliminating the need for JSX. This enables the developer to directly produce HTML documents in situations where JSX is inappropriate.



2.4 One way data binding

Unidirectional data binding restricts the backward transmission of data within a component. A component can transfer data solely to its child component. A component is incapable of transmitting data to its parent component under any circumstances. This will optimize data handling and reduce complexity. At first glance, two-way data binding may seem crucial; nevertheless, a further analysis reveals that one-way data binding is adequate, hence streamlining the application concept.

2.5 Scalable

React is proficient in creating apps of any magnitude. The architecture of React components, the Virtual DOM, and unidirectional data binding efficiently govern large applications within the requisite time limitations for front-end development. These characteristics make React a scalable solution.

2.6 Flexible

React provides a restricted set of essential principles for creating truly scalable applications. React does not place any restrictions on developers to follow a specific technique. This enables the developer to establish their own architecture based on the essential notion, hence augmenting its versatility.

2.7 Modular

Modular React components can be created in a separate JavaScript file and marked for export. This enables the developer to classify and organize particular components into a module for importation and exploitation as necessary.

3. ReactJS Lifecycle

Every React Component has a lifecycle of its own, lifecycle of a component can be defined as the series of methods that are invoked in different stages of the component's existence. React automatically calls these methods at different points in a component's life cycle. Understanding these phases helps manage state, perform side effects, and optimize components effectively.

3.1 Initialization

In this stage the component is created with the provided Props and default state. This is done in the constructor of a Component Class.

3.2 Mounting Phase

- Constructor: The constructor method assign the component. It's where you set up initial state and event handlers binding.
- render(): This method returns the JSX representation of the component. It's called during initial rendering and updates if made any.
- componentDidMount(): After the component is inserted into the DOM, this method is invoked. Use it for side effects like data fetching or setting timers.

3.3 Updating Phase

- componentDidUpdate(prevProps, prevState): Called after the component updates(modified) due to new props or state changes. Handle side effects here.
- shouldComponentUpdate(nextProps, nextState): Determines if the component should be re-rendered. Optimize performance by customizing these method.
- render(): Again, the render() method shows changes in state or props during updates and does exact same on UI.

3.4 Unmounting Phase

- componentWillUnmount(): Calling just before the component is deleted from the DOM. Clean up resources (e.g., event listeners, timers).

4. React Environment Setup

To run any React application, first we need to set up a ReactJS Development Environment. In this article, we will show you a step-by-step guide to installing and configuring a working React development environment. We will be discussing the following approaches to setup an environment in React.

Table of Content

- Method 1:By Using create-react-app (C.R.A. command)
- Method 2:By Using webpack and babel
- Method 3: By Using Vite build tool

Pre-requisite:

We must have NodeJS installed on our PC. So, the very first step is to install NodeJS. Once we have set up NodeJS on our computer, the next thing we do is to set up React Boilerplate.

- By installation of Node.js on Windows
- By installation of Node.js on Linux
- By installation of Node.js on mac

Method 1: Using create-react-app (C.R.A. command)

- **Step 1:** Navigate to the folder where you want to build the project and open it in terminal
- **Step 2:** In the terminal of the application directory write the following commands

npx create-react-app <<Application__Name>>

- **Step 3**: Go to the recently created folder using the command
- cd <<Application__Name>>
- **Step 4:** A default application will be generated with the following project structure and dependencies.

It will by default install some packages which can be seen in the dependencies in package.json file as written below:

```
"dependencies": {
  "@testing-library/jest-dom": "^5.17.1",
  "@testing-library/react": "^13.4.1",
  "@testing-library/user-event": "^13.5.1",
```

```
"react": "^18.2.1",
"react-dom": "^18.2.1",
"react-scripts": "5.0.1",
"web-vitals": "^2.1.4"}
```

Method 2: Using webpack and babel

For setting a react development environment using webpack and babel is a very very time consuming process and we have to include each package and create setup files ourselves.
We have to create the setup using *'npm init -y'* command and then import the required packages in the folder and then install react using the command.

```
->npm i react react-dom
```

To install the necessary packages in our project use these commands

```
->npm i webpack webpack-cli @babel/core@babel/preset-env@babel/preset-react
```

```
->babel-loader html-webpack-plugin webpack-dev-server --save-dev
```

Method 3: Using Vite build tool

- **Step 1:** Go to the folder where you want to build the project and open it in terminal or command prompt.
- **Step 2:**Type the following command In the terminal of the application directory.
- npm create vite@latest <<Application_name>>
- **Step 3:** Select the React Framework and then choose different framwork as JavaScript from options.
- **Step 4:** Navigate to the recent created folder using the command given below..

  cd <<Application_name>>

now, we can see the basic project
structure show in below image.

- **Step 5:** Use the below command in the cmd for windows and terminal for mac to install all demanded dependencies.

npm install

OR

npm i

After executing these steps or command we are able to see a new folder named "node_module" which contains all the dependencies in the project folder successfully.

5. Conclusion

Hence, React is cemented its own place in the list of library for building dynamic web applications. As developers build reusable UI components, it contributes to the maintainability and scalability. One of the key aspects that boosts performance is the use of virtual doms, that is, minimizing direct manipulation of the actual dom will help you render the content faster, providing smoother UI interactions to your user. Other libraries for state management for example redux, and for navigation React router makes it even more easier to develop complex applications that are also highly interactive and user-friendly. The flexibility to work with various backend technologies such as Node.js and Express makes it a vast unit for fullstack development. React is being used by various companies like Facebook, Instagram, and Netflix to implement their high-traffic applications which only proves its power to manage required workloads without compromising user experience. The huge community support and huge updates keep on making it as one of the frontiers of a wide range of web development that is well suited for a beginner stage for web development or an advanced developer stage however.

6. REFERENCES

1. Angular Minds. (n.d.). React: A crucial tool for web development Retrieved from Angular Minds.
2. 12Grids. (2024). Benefits of ReactJS for Dynamic Web Development Retrieved from 12Grids.
3. Adams, A. (2024). Mastering React. js Advanced Data Structures: Building Dynamic Web Applications with ease! Retrieved from Andrea Adams.
4. SPEC INDIA. (2024). Reasons to Choose React JS for Web Development Retrieved from SPEC INDIA.
5. Green, O. (n.d.). React Software Developmnet Meaning Modern Dynamic Web Applications Retrieved from LinkedIn.
6. Geekster. (n.d.). Advantages of React. Retrieved from Geekster.
7. Khan, Z. (n.d.). The Future of React. js: The Secret Behind Dynamic Web Development Retrieved from LinkedIn.