



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 5, May 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.165



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

React JS: Open Source Java Script Library for Frontend Development

Vivek Toppo, Arun Sahu, Kuldeep Uikey, Daneshwar Chandravanshi, Savita Sahu

BE Scholar, Department of Computer Science and Engineering, Government Engineering College, Bilaspur(C.G), India

BE Scholar, Department of Computer Science and Engineering, Government Engineering College, Bilaspur(C.G), India

BE Scholar, Department of Computer Science and Engineering, Government Engineering College, Bilaspur(C.G), India

BE Scholar, Department of Computer Science and Engineering, Government Engineering College, Bilaspur(C.G), India

Assistant Professor, Department of Computer Science and Engineering, Government Engineering College, Bilaspur(C.G), India

ABSTRACT: This article explores the React JS library, which is based on the JavaScript Frontend library for web and mobile app development. React is a powerful JavaScript library and highly customizable framework for user interaction. Because there are many frameworks for developing the best applications and libraries of the time. The most popular web technology React JS has proven to be a very fast forwarding library. It focuses on the view part of the MVC pattern and is widely used to enhance extensible applications.

After it was made open source, many contributors are adding new features in it day by day. It has been quite successful in providing the best user experience when dealing with large amounts of data and users. Along with Facebook, some other large organizations and applications also use React JS and React Native for development. Instagram, Netflix and Airbnb are just a few of the well-known companies that seamlessly serve millions of users around the world. These big names prove that React JS works very well. Throughout the study, the main goal was to evaluate the library and prove that React JS is an interoperable platform that can be used when you have multiple options to choose from. In this study, the basics, features, data processing techniques, popularity, and underlying architecture were discussed. React JS has no restrictions on its use compared to other frameworks, but it is recommended to adopt and use it as a new web technology depending on the nature of the application you want to create.

KEYWORDS: React JS, Components, JSX, Virtual DOM, JavaScript, HTML, Library, Front-end, Framework.

I. INTRODUCTION

A tendency is developing to move software or applications that we used to use in office machines on the web. Several JavaScript-based frameworks and libraries are used to develop various apps. There are React JS, Angular JS, Ember JS, Meteor JS, Vue JS, Knockout JS and many more at the moment in production. React is a popular open source JavaScript library built by Facebook. React is very popular in the developer community due to its simplicity and simple yet effective development process. React makes it easier to create interactive user interfaces. It efficiently updates through rendering the exact components to the view of each state and makes the data changes in the application. In React JS, every component manages their own state and composes them to the user interfaces. This concept of components instead of templates in JavaScript, plenty of data can easily be passed to the app and thus keep the state out of the DOM.

Using Node React can also be rendered on the server side. Alongside web apps, to build mobile applications we can use React Native as well. The purpose of the thesis is to carry out an in-depth research of the React JS library based on JavaScript. The fundamental concepts, characteristics, characteristics, development processes, basic architecture and market research and compatibility. The aim is to provide a solid understanding of the ReactJS library.

II. LEARN REACT

Today, developers and engineers choose React because it allows them to spend more time focusing on product development than spending less time learning and learning the framework. A React application is a collection of discrete

components, each representing a single view. The idea of individual view components makes it easy to iterate in product development, because making changes to a single view or single component doesn't need to be viewed. consider the whole system. When an app is built using React, the code is often predictable, this is because React wraps the mutable and imperative API of the DOM with a declarative API, which enhances the level of abstraction and simplicity. programming modeling. It's a simpler way to develop apps, React is basically faster because you don't have to tell the app how to represent state - you just have to stay what you want and it will happen. out. It's quick and easy and leaves less room for human error. The web's rapid iterative cycle has spawned some great products that include many of Facebook's components. A great JavaScript framework called Relay has also been built on top of React, which simplifies data fetching at scale learning.

2.1 A Short and Simple Learning Curve

React has an easier learning curve than other programming languages since developers don't have to spend as much time relearning the language. In other words, React is based on JavaScript, which developers are already familiar with. React is the greatest toolkit for making web development easier to implement and iterate, thanks to its straightforward design, usage of JSX (an HTML-like syntax), and extensive documentation.

2.2 React is quick and adaptable.

React JS supports one-way unidirectional data flow between application states and layers. This means that data only flows in one direction between application states and layers. When using two-way data binding, such as Angular, when a model is changed, the view changes as well, and vice versa. React renders DOM updates much faster than other frameworks and has a much smaller library. The term DOM refers to the document object model. As a result, selecting the tools for the job is simple.

2.3 React Introduced JSX.

JSX stands for JavaScript XML, which allows us to write HTML in React. JSX makes it easier for us to write and add HTML to React. During the compilation process it performs optimization because it is faster. Most errors are caught during compilation and it is also type safe. If you are good at HTML then it makes writing templates easier and faster. With the help of JSX, you can specify the DOM elements before the components write inside JavaScript files.

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;
```

Figure 1.0 Screenshot of JSX

III. REACT CORE ARCHITECTURE

3.1 React Virtual DOM

DOM stands for Document Object Model. DOM manipulation could be very vital for contemporary-day interactive net technology. It is regularly known as the coronary heart of the contemporary-day net. It is an abstraction of the established text. But it really works slower than different JavaScript operations due to the fact that maximum JavaScript frameworks typically replace the DOM even though they no longer want to do it. In that approach, the one's updates aren't always required to carry out the moves however they nevertheless do through default. For example, allow us to count on 9 objects that were installed in a purchasing basket in an internet net store. Now allow us to say simplest: the primary object is wanted to shop for and continue to checkout. Here, maximum technology could rebuild the complete listing that has been positioned inside the basket. In this approach, the framework has to unnecessarily paint ten instances more. Because of the simplest alternative, the device has to rebuild the listing precisely the way it was before. React no longer invented Virtual DOM, however, makes use of and presents it to the developer network for free. Virtual Dom is genuinely an abstraction of HTML DOM. React has a corresponding digital DOM item for each DOM item like a correspondent or a lightweight copy. Virtual DOM is likewise characterized by comparable residences to an actual DOM. However, it can't make any modifications immediately to the view. DOM manipulation is a pretty sluggish process. But manipulating Virtual DOM is

quicker as it has not anything to do with the view component and does now no longer make any modifications to the screen. Figure 13, reprinted from stackoverflow.com, is an example of Virtual DOM inside the memory.

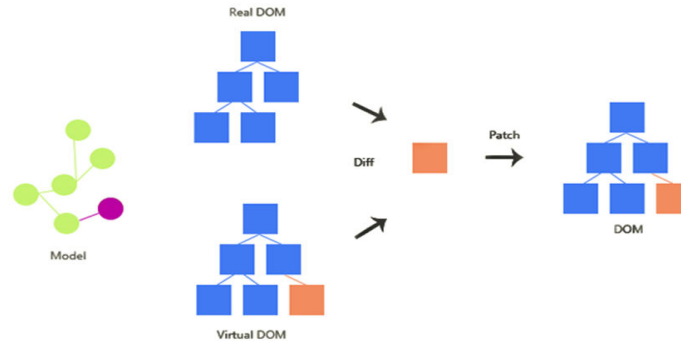


Figure 3.0 React Virtual DOM in memory

As proven in Figure 13, a React digital DOM inside the reminiscence is a lightweight reproduction of the actual DOM. React makes use of a way called “diffing” because of this renders a JSX detail that receives each unmarried Virtual DOM up to date. This would possibly sound inefficient but, in fact, it does not do anything as Virtual DOM is pretty speedy to get up to date and does now no longer make any effect inside the procedure. After the DOM receives up-to-date React compares the up-to-date DOM with a pre-up to date nation of the DOM and determines which digital DOM has been modified. Once React detects the modified DOMs, React updates the handiest of the items to the actual DOM.

Thus, React makes the updates quicker thru Virtual DOM. In the above-cited example, React could have up-to-date handiest the checked object from the listing and depart the relaxation of the objects alone. This makes the difference whilst updating an internet web page in a software program while React can simplest make adjustments to the important additives of the DOM. This digital DOM manipulation procedure is one of the essential motives why React is gaining an awful lot of recognition in a number of developer communities.

3.2 React Components

A component is a fundamental building block in React. In other words, every application you generate in React will be made up of pieces known as components. Components make it much easier to create user interfaces. You can see a UI broken down into multiple individual components and work on them independently before combining them all into a parent component, which will be your final UI. Components are like JavaScript functions. They literally perform the same task but in different environments and different approaches. Like functions, they take inputs called props and return React elements.

Creating a React Components

A React component can be written as a simple JavaScript function. This function takes arguments and returns a React element. They are referred to as functional components. A component can also be defined using an ES6 class.

```
import React from 'react';
import ReactDOM from 'react-dom';

// This is a functional component
const Welcome={()=>
{
    return <h1>Hello World!</h1>
}

ReactDOM.render(
    <Welcome />,
    document.getElementById("root")
);
```

Figure. 3.1 Creating a React Components



3.3 Introducing JSX components

JSX is not a unit of characters or HTML. Mathematically a syntax extension in JavaScript. It is similar to the language directed at something designed to work on modern web browsers. JSX is recommended for use with React to design and build user visuals. Although it comes with full JavaScript capabilities it may seem like a template language at first though. React element is produced by JSX. It can be provided in React Virtual DOM.

3.4.1 JSX characteristics

JSX has some unique features that made the JSX very popular among React and React Native engineers. At first, it may seem daunting, but in the long run, getting JSX can be easier. First, fast: While the JSX source code is embedded in JavaScript, it shows up a well-executed result. Compared to the same JavaScript code, the JSX-generated code works faster. JSX proved to be 12% faster on iOS and 29% faster on Android. Second, more secure: Unlike JavaScript, JSX is mathematically written and usually typesafe. The quality of the apps gets higher when they are upgraded using JSX, as many errors will be caught during the merger. It also provides debugging features at the compiler level as well. Third, it's easy: JSX offers a robust Java-like class system, which frees developers from working with a given prototype legacy system, with JavaScript. Speeches and statements, however, are similar to JavaScript, so it is easy for JavaScript programmers to start using JSX. There are also programs available Language editors' resources / IDEs, for example coding to make coding easier.

3.4.2 Practical JSX

Expressions can be embedded in JSX though it is an expression too. The function written below is an expression.

```
function formatName(user) {
  return user.firstName + ' ' + user.lastName;
}
const user = {
  firstName: 'Ajey ',
  lastName: 'Nagar'
};
const element = (
  <h1>
    Hello,{formatName(user)}!
  </h1>
);
ReactDOM.render(
  element,document.getElementById('root')
);
```

JSX tags can also contain children.

```
const element = (
  <div>
    <h1>Hello!</h1>
    <h2>Is React JSX worthy or not?</h2></div> );
```

Objects in React are also represented by JSX. Here are two different code examples written below but they are identical.

```
const element = (
  <h1 className="greeting">
    Hello, Ajey!
  </h1>);
const element = React.createElement(
  'h1',
  {className: 'greeting'},
  'Hello, Ajey!'
);
```

Requirements can be specified in JSX in a number of ways. JavaScript expressions, string literals can also be passed as props. If, Else, Do, While and For loops are not considered expressions in plain JavaScript, so they cannot be used directly in JSX, but not in the environment.

```
function NumberDescriber(props) {
  let description;
  if (props.number % 2 == 0) {
    description = <strong>even</strong>;
  }
  else {
    description = <i>odd</i>;
  }
  return <div> {props.number} is an {description} number </div>;
}
```

IV DISCUSSION

This chapter of this study provides a brief assessment of the issue. The purpose of this research work on the ReactJS library was to examine its features and have a clear understanding of what it offers, the key concepts behind its architecture, how it differs from other libraries and frameworks, data management and other features. The reason I chose ReactJS as a topic was to get to know you better by doing research and enriching your personal learning. Research shows that ReactJS is a less complex library than other frameworks. It is a very fast and flexible rendering library under development. Learning and mastering React is just a library, not a framework, so it takes less time than others. Frameworks typically take longer to become familiar with the framework and learn the terminology. Components are also a core concept in React. Everything you see on the frontend is just one component. A user interface is a set of components. Changes to one component do not affect the other components. This makes it simpler and easier to update applications when data changes. To perform this function, React introduces a modified DOM concept into the virtual DOM. The unidirectional flow of data for data management is another strength of React. The data can be manipulated anywhere in the application layer. Because data flows in one direction, this provides strong control between model and application state. Therefore, when applications developed for a large number of users work with large amounts of data, ReactJS is recommended for faster and smoother user experience.

V RESULTS

- With ReactJS, you can build web applications with less time and effort. Components, states, properties, etc. make web applications easier to use.
- Open Notepad on your system, type HTML code and save it in .html format. Yes, it's basically a website. To add some style and functionality, create .css and .js files, add code and include the generated HTML file.
- So why do you need a library like React.js? In fact, you can code any type of website using HTML, CSS, and JS. Setting up a large project in this way, however, took a lot of time and effort. Several libraries and frameworks are available.
- ReactJS is one of them. You can use React to break your application into components. I recommend ReactJS for building web apps instead of websites. This is because when your React web application loads, the package that contains all the UI structures that the browser should have is loaded first. It took some time. However, after downloading the package, it is very fast

VI.CONCLUSION

The aim of this thesis was to research and review an open-source external JavaScript-based library called ReactJS. Facebook developed ReactJS for its own purposes and later made it open source. In a very short time after its launch, ReactJS has gained tremendous popularity among both developers and the technology industry. In summary, this article provides detailed instructions on how to get started with React, clear guidelines for React's features and capabilities, and examples of alternatives to React and data architecture management systems to consider. Reacting is a complex and important skill to learn. Learning this skill through further research will help enrich a person's proficiency in this skill. Based on this, the topic was chosen. Developing applications with React would be great as it enriches you with practical skills, but I've been learning React extensively over the last few months due to some limitations. In conclusion, ReactJS is a



learnable skill and definitely worth considering for production applications. It brings a new dimension to web application development. The fast render library improves the efficiency of your application, the future of React is bright and learning React is worth the effort. There are 20 reviews and ratings. There is already a solid concept for this.

REFERENCES

1. A JavaScript library for building user interfaces <https://reactjs.org/>
2. React Virtual DOM online URL: <http://stackoverflow.com/questions/21109361/why-is-reacts-concept-of-virtual-dom-said-to-be-more-performant-than-dirty-mode>
3. ReactJS Tutorial by Geeks For Geeks <https://www.geeksforgeeks.org/reactjs-tutorials/>
4. ReactJS by W3Schools <https://www.w3schools.com/react/default.asp>
5. A JavaScript Library for Building User Interfaces Online <https://facebook.github.io/react/>
6. “Learning React Functional Web Development with React and Redux” by Alex Banks and Eve Porcello.
7. “End-to-end E-commerce web application, a modern approach using MERN stack” by Hung Viet Nguyen.
8. “React JS: An Open-Source JavaScript Library for Front-end Development” by Naimul Islam Naim.
9. ReactJS library by wikipedia. [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
10. ReactJS by Javatpoint. <https://www.javatpoint.com/reactjs-tutorial>
11. ReactJs in Visual Studio Code with NodeJs .<https://code.visualstudio.com/docs/nodejs/reactjs-tutorial>
12. “The Complete Beginner’s Guide to React” By Kristen Dyrr.



INNO  SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

 **doi**[®]
cross **ref**

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details