React: A detailed survey

Varun Komperla, Pratiba Deenadhayalan, Poonam Ghuli, Ramakanthkumar Pattar
Department of Computer Science and Engineering, R V College of Engineering, Mysore Rd, RV Vidyaniketan Post, Bengaluru, India

ABSTRACT
With rapid developments in technology and the ever-growing number of web applications, it has become a necessity to create fast and scalable applications to cater to the current market. There are a plethora of frameworks available for web development, and a developer must choose the most appropriate framework for their use. In this paper, a detailed analysis of the history, prominent features and advantages of React, an open-source JavaScript library is presented. A discussion on React Native, a framework for building native applications is also given. This paper has provided an insight into the reason React is the leading web development framework in the world.

Keywords: JavaScript, JSX, React, React native, Single-page applications, Virtual document object model, Web frameworks

This is an open access article under the CC BY-SA license.

1. INTRODUCTION
In 2011, as Facebook was growing in size and popularity, its application was becoming more difficult to manage due to a large number of updates. Jordan Walke, a software engineer at Facebook, created a prototype of React called “FaxJS” [1] to make the process more efficient. He had taken inspiration from XHP, an HTML component library for PHP. React was first used in Facebook’s news feed in 2011, and later on Instagram in 2012. At JSConf US in 2013 [2] it was announced the React would be open-sourced, thus enabling everyone to develop smooth, interactive user interfaces for their applications. React Native, a framework that enables the development of native mobile applications for iOS and Android, was also created by Facebook and open-sourced in 2015. The most notable and differentiating feature of React was that of the virtual document object model (DOM) [3], which made it suitable for high performance applications. React also has a significant number of other advantages over other frameworks. It speeds up the development process due to its modular structure allowing developers to work in parallel on individual components, it’s flexible and easy to maintain thanks to self-contained components, it has been designed for high performance, and it allows for mobile development with React Native. Currently, React is one of the most popular JavaScript frontend frameworks with many popular organizations like Facebook, Instagram, Airbnb, Twitter, and Netflix using it extensively. In this paper, an in-depth analysis of React is presented.

2. RESEARCH
In this section, some related research and background has been presented. This will help to better understand the impact of the features that the React framework possesses. Some fundamental concepts of
React which make it so popular have also been described. An overview of React is presented, followed by the most popular features of React and finally, a short insight into React Native is given.

2.1. Related research
2.1.1. Single page applications

One needs to understand what a single-page application (SPA) is before one can truly appreciate the advantages of React. In the paper, Jhadav [4] an SPA is defined as a web application that is comprised of many individual components which can be replaced independently of each other without refreshing the entire page [5] thus eliminating the need to reload the page upon every user action. In the early days of the internet, the main purpose of traditional websites was to serve static pages to their clients [6]. For each user interaction, the entire webpage had to reload to reflect the result of the interaction. This resulted in slower response times, higher bandwidth usage and dissatisfaction amongst users. The solution to this is an SPA, where the initial page is loaded only once along with all of its resources like CSS, images, scripts, etc. and then only the appropriate components are updated dynamically based on user interactions. This causes every subsequent request, after the initial page is loaded, to take a significantly shorter amount of time since only a portion of the page is being refreshed rather than the entire page [7]. The generic Client-Server Request and Response interaction is shown in Figure 1.

2.1.2. Client-side routing

This raises the question, “How does one implement a single-page application?” client-side routing is the answer to this question. With client-side routing, routes are internally handled by a JavaScript file at the client-side and rendered to the front end. When a client clicks on an internal link, the URL is updated to show that there is an update happening to the page without reloading it. This provides users with a more intuitive sense of where they are within the application and also provides the ability to use deep links to load a specific view in the application. With client-side routing, only the newly required data is retrieved from the server and the appropriate portion of the page is dynamically updated with this data. There is no need to reload the page for every route in the application. It is important to understand that React can be used to build a variety of applications, with single-page applications being one of the more famous types of applications in the world today.

2.2. Overview of React

React is an open-sourced JavaScript library that is used to build interactive user interfaces (UIs), especially for single-page applications. The purpose of React is to build simple, fast, and scalable applications. React implements the view (V) part of the model-view-controller (MVC) architecture [9]–[13]. React gained popularity mainly due to its declarative and component-based nature [14]. One can create components that manage their own state and build more complex UIs by structuring these components together. React components are independent bits of code that enable re-usability. A component is conceptually similar to a JavaScript function. It accepts arbitrary inputs called “props” and returns some UI elements that will be rendered on the screen. With this design, a single component can be reused multiple times across various views within the application. React is a JavaScript library and thus supports
ECMAScript6 (ES6). React can also be rendered on the server-side using NodeJS. Furthermore, React can be used to power native mobile applications using React Native.

2.3. React features

React has many salient features that are the reason for its popularity. Some of these features are its simplicity, its component based nature and its support if JSX. There is also the virtual DOM which is the reason for improved performance of the user interface when using React, its capability of routing within an application and its nature of unidirectional data flow. All these features of React are described in the following sub-sections in more detail.

2.3.1. Simplicity

A very important reason why React is so popular among front end developers is due to its simplicity. One just needs to have an understanding of HTML and JavaScript. Developers migrating from another framework can easily adapt to React since it focuses only on building user interfaces.

2.3.2. Components

This is a key feature of React. Components are independent, reusable bits of code and are the building blocks of React code. They make it possible to split the UI into more manageable pieces and let one think about each piece in isolation. Data can be passed to components in the form of a “props” object which can be used to dynamically render elements in the DOM. Unlike the “props” object, which is passed from outside the component, each component has a “state” object that is managed within the component.

There are two types of components in React:
- Class component: a JavaScript class extending which has a render() method and extends the React.Component class.
- Functional component: a JavaScript function that returns JSX. They are also called stateless components since they do not possess their own state.

Each component also has its own lifecycle, consisting of three main stages—mount, update, and unmount. The component can be monitored and manipulated during various stages of its lifecycle using lifecycle methods. Some common React lifecycle methods are listed in Table 1. There are other lifecycle methods like shouldComponentUpdate(), getDerivedStateFromProps() and getSnapshotBeforeUpdate() but these are rarely used. The vast number of lifecycle methods provide great flexibility in manipulating components at various points in time.

<table>
<thead>
<tr>
<th>Lifecycle Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>render()</td>
<td>It is called during the mount and update phases of the component. For class components, it is a mandatory method. The state cannot be modified within render().</td>
</tr>
<tr>
<td>componentDidMount()</td>
<td>It is called after the component has mounted and is ready. The state can be changed within this function. Any API calls are usually initiated within this function.</td>
</tr>
<tr>
<td>componentDidUpdate()</td>
<td>It is called after the component has updated, usually in response to changes in the “props” or “state” objects. The state can be changed within this function.</td>
</tr>
<tr>
<td>componentWillUnmount()</td>
<td>It is invoked before the component is unmounted. Clean-up operations like aborting network calls, destroying objects that were created, etc. are usually carried out in this function. Since this is called just before a component is unmounted, the state cannot be modified.</td>
</tr>
</tbody>
</table>

2.3.3. JSX

JSX is what makes React a powerful library. JSX is an extension to JavaScript [16], analogous to how XHP is an extension of PHP. It allows one to write HTML within JavaScript. It enables the structuring of components in HTML, which most web developers are familiar with. JSX is used to create React elements. React components return elements to be rendered on the UI using JSX. For example:

```javascript
const Hello = <h1>Hello World!</h1>
```

as can be seen in the example, the HTML `<h1>` element is stored in a constant string which is using JSX syntax. JSX also prevents cross-site-scripting (XSS) attacks.

2.3.4. Unidirectional data flow

React uses one-way data bindings. This means that in React, data can only flow in a single direction. So data has only one way in which it can be passed to various parts of the application. As shown in Figure 2,
the state is passed to the view, that is, to a component and its children. When the state needs to be updated, an action is called by the view. The state is changed as a result of an action and is passed again to the view and its child components. As a result of one-way bindings, data cannot flow in the opposite direction (as opposed to two-way bindings in Angular). For example, the view cannot directly change its state. It must trigger an action to do so. A state is owned by a single component in React and any change in the state of a component can only affect its child components below it.

![Figure 2. Data flow in React [17]](image)

### 2.3.5. Virtual DOM

Traditionally, whenever an update needs to be made to an element in the DOM, the following steps occur:

- The browser parses the HTML to get the DOM.
- The required element is extracted from the DOM tree.
- The DOM is updated with the new value.
- The CSS is applied again to the parent and child, which involves calculations.
- The coordinates of each element in the layout are updated.
- Finally, traverse and render the DOM tree on the browser.

As seen in the steps above, updating the DOM has a lot of other operations attached to it. This is why it is a slow process. React maintains an in-memory data structure called a virtual DOM [3], [18] which is a lightweight representation of the actual DOM. Whenever a change is made to an element, the Virtual DOM gets created again, which is a fast process since nothing gets rendered on the browser. React maintains two Virtual DOMs at a time, one is the updated Virtual DOM and the other is the version of the Virtual DOM before the update. It uses a “diffing” algorithm [19] to compare the two Virtual DOMs and see what has changed after the update. React then applies only the required changes to update the real DOM in the minimum number of steps. This improves the performance [20], [21] of the user interface significantly and is the reason why React is desirable by developers around the world.

### 2.3.6. Routing

React router is the standard routing library used in React [22]. The standard way of introducing routing into a React application is to have the App component return a router component [23], which will wrap the Switch component containing all of the route components. An example of basic routing syntax is shown in Figure 3. The Switch wraps all the route components and renders the first route whose path matches the current URL. Each route component configures the route and wraps the components that will render for the configured route.

### 2.4. React Native

React Native is an open-source framework that is used to build native mobile applications. It can be used to develop applications for Android and iOS. React Native is popular because it is easy for developers to adapt to it, since it uses JavaScript—one of the most famous programming languages out there along with the regular web technologies (HTML). The difference is that applications developed in React Native compile to native code. Any developer with an understanding of React can pick up React Native as well, since they are very similar. The major appeal is that developers would be using familiar technologies to build mobile applications, which would otherwise require different domain knowledge. Another advantage of React Native is that most of the React Native APIs are cross-platform [24]–[27]. This means that an application will have consistent behaviour across Android and iOS platforms, and developers do not have to write code multiple times.
times for the same logic [28], [29]. If the need for platform-specific features arises, then React Native allows the use of platform-specific versions of each component as well. Thus, even with the use of platform-specific APIs, the portion of reusable code remains substantial [30]–[34].

```javascript
function App() {
    return (
        <Router>
            <div>
                <nav>
                    <ul>
                        <li><Link to="/">Home</Link></li>
                        <li><Link to="/about">About</Link></li>
                        <li><Link to="/docs">Docs</Link></li>
                    </ul>
                </nav>
                <Switch> 
                    <Route path="/about"> 
                        <About/>
                    </Route> 
                    <Route path="/docs"> 
                        <Docs/>
                    </Route>
                </Switch>
            </div>
        </Router>
    );
}
```

Figure 3. Basic routing application in React

3. METHOD OF TESTING
Unit Testing is an important stage of any software development project [35]–[41]. This applies to building applications in React as well. Some advantages of unit testing are:
- Improvement in design of solutions.
- Improvement in code quality.
- Aids the debugging process.
- Reduction in the cost of development.

Given the benefits of unit testing as mentioned above, there are two tools recommended by the developers of React, for testing. These are jest and React testing library [42]. Jest is a test runner in JavaScript that allows access to the DOM through jsdom. Jest can be used to mock React components without their children, and provides more control over the execution of the code. React Testing Library comprises of a set of helpers that enables the testing of react components while abstracting their implementation details. It allows for easy refactoring of code and also encourages developers to adopt the best accessibility practices. Since React is based on JavaScript, possessing extensive knowledge of testing tools [43], [44] can help programmers design simple, efficient and reliable interfaces using React as there is an abundance of testing facilities available for developers.

4. RESULTS AND DISCUSSION
In this section, the results of the survey have been presented concisely, along with a discussion. The advantages of React, which are the reasons for its success as the leading framework for front end development, have been listed in brief. This is followed by a discussion summarizing the motivation behind React, its features and the present state of the framework as well as its future.

4.1. Advantages of react
After looking at the various features of React, it is clear that React provides many advantages. These advantages are listed:
- With JSX, the process of writing components becomes much easier. It allows for simpler and cleaner source code while building custom components.
- With the Virtual DOM, React allows for faster rendering [45], quick application load times and a better user experience.
- Due to the unidirectional flow of data, React also ensures stable code.
- With React Native, the option to build native applications is available, as well as the capability to build cross-platform applications (iOS and Android).
- React also comes with a developer toolset that helps in the design and debugging processes.
- It boosts productivity. The ability to reuse system components is one of the best features of React, due to which updates for an application are relatively simple to develop.
- It is simple to set up server-side scripting in React, thus making it SEO [46]–[51] friendly.
- With React Routing, the code can be split up into logical segments which enables better management of the code.

4.2. Discussion
React was developed with the aim of building large scale applications whose data repeatedly changes with time, and it has addressed this well. React allows for easy creation of interactive UIs, supports JSX, has a component-based structure, and is much faster due to the virtual DOM. These features are attributed to the success of React as a front end framework. The latest version of React is React v17.0 and was released three years after the previous version. While it did not introduce any new developer features, the development team claimed that it is an incredibly important stepping stone for the future of React. It focuses on easing the process of upgrading the React version of an application. Different parts of the application can be upgraded to the latest React version gradually, instead of updating the whole application at once. It also introduces a new JSX Transform, which enables the use of JSX without React. The next version of React to be released is v18.

5. CONCLUSION
The React community has significantly grown over the last few years and has become the most widely used front end framework. React faces competition from other web development frameworks such as Angular and VueJS. With this in mind, and with the increasing popularity and user base of web applications, the React community needs to keep updating the React library according to the requirements of the modern web developer to remain the preferred front end development framework. In this paper, a thorough analysis of the history and overview of React is presented along with a detailed explanation of the salient features that make React unique and widely used in the industry. Finally, a brief discussion on testing and the tools available in React is presented and the paper is concluded with the advantages and a discussion of the present situation of React.

REFERENCES


**BIOGRAPHIES OF AUTHORS**

**Varun Komperla** has completed his Bachelor of Engineering from the department of Computer Science and Engineering, RV College of Engineering. His research interests are in the area of front end and back end software development. He can be contacted at email: varunkomperla.cs17@rvce.edu.in.

**Dr. Pratiba Deenadhayalan** is working as an Assistant professor in Computer Science and Engineering department at RVCE. She received her Ph.D degree from VTU. She has published over 40 research papers. She has worked on various research and consultancy projects sponsored by Cisco, Citrix and Samsung. She can be contacted at email: pratibad@rvce.edu.in.

**Dr. Poonam Ghuli** is an Associate Professor at the Department of Computer Science and Engineering, RV College of Engineering, Bengaluru. She has completed her PhD in the area of Machine Learning from Visvesvaraya Technological University, Karnataka, India in the year 2017. Her research interests are Data Analytics, Machine Learning and Computer Vision. She is actively involved in research and consultancy work supported by many leading industries. She can be contacted at email: poonamGhuli@rvce.edu.in.

**Dr. Ramakanthkumar Pattar** is a Professor and HOD in the Computer Science and Engineering department at RVCE. His research interests are Digital Image Processing, Pattern Recognition and Natural Language processing. He has published over 100 research papers. He has executed several funded research and consultancy projects sponsored by DRDO, ISRO, AICTE, GE India Pvt.Ltd, CABS, HP, Nihon Communication Solutions Pvt.Ltd etc. He can be contacted at email: ramakanthkp@rvce.edu.in.