# React Js Framework

**Mohammed Habib Rassiwala[1] Dr.S.S.Agrawal[2]**
[2]Guide
[1,2]Department of Computer Science Engineering
[1,2]Shri Shivaji College of Engineering and Tech, Akola, MS, India

*Abstract*— Internet has become a busy hub for searching information and doing different tasks virtually that used to be done manually before the internet age. There are enormous numbers of mobile and web applications that have made it easier to do different tasks. A big part of our everyday task can be done on the internet at the current age. Faster internet along with fast performing devices demands faster applications. A trend is growing to shift software or applications we used to use in desktop machines to the web. There are plenty of applications that are usable from web and mobile devices. Several JavaScript based frameworks and libraries are used to develop different applications. There are ReactJS, AngularJS, EmberJS, MeteorJS, VueJS, KnockoutJS and many more at this moment in production. ReactJS is one of them for the front-end development of applications. React is a popular open-source front-end JavaScript library developed by Facebook. React is widely popular among developer communities because of its simplicity and easy but effective developing 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 ReactJS, 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 in-depth research of the ReactJS library based on JavaScript. The fundamental concepts, characteristics, features, development processes, core architecture and market research as well as compatibility will be covered in the thesis. The aim is to provide a solid understanding of the ReactJS library.

***Keywords:*** React, Open Source, DOM, Node

## I. INTRODUCTION

React seems to be everywhere. Companies and projects large and small are using it to build their applications. The popularity comes from the fact that React builds on core web development skills. That's not to say you will learn it in a day or that every feature is easy to understand on the first try. Instead, React excels precisely because it minimizes the amount of React specific knowledge you need. You don't need to learn about templates or controllers or complex patterns. Instead, most of the code you write will be JavaScript combined with standard HTML. It can get complicated from there. The HTML, for example, is really a markup language called JSX that is parsed by React before going into the DOM. But as you take each step in your learning you will be building on solid foundations of web development. That means you gain a double benefit as your learn React. Not only will you be building world class applications, you will be strengthening your own knowledge of JavaScript and web standards. You will develop transferable skills that you can use in any future web-based application whether it's built with React or not.

## II. WHY LEARN REACT?

React was introduced to the world two years ago, and since then it has seen impressive growth, both inside and outside Facebook. New web projects at Facebook are commonly built using React in one form or another, and it is being broadly adopted across the industry. Developers and engineers are choosing React because it al-lows spending more time to focus more on the product development and less time spent on fighting and learning to the *framework.*

## III. SHORT AND EASY LEARNING CURVE

Unlike some other JavaScript libraries where it takes a lot of time to learn about the frameworks, in React it does not take much of an effort to start building an application. React is comprised of many strong features. Readability is one of the greatest strengths of React. It is easily readable even to those who are unfamiliar to it. While other frameworks require learning many concepts about the framework itself, ignoring the language fundamentals, React does the absolute opposite. [3] For example, let's consider how different it is in React and Ionic (AngularJS) rendering a portion of an employer's list. In Ionic, it requires to use the directive called ngRepeat. Let's assume an array of employers. Each of them contains the following fields: first_name, last_name, is_married. The target is to show only employers who are married. The following Figure 1 shows a screenshot of a function written in Ionic framework

```
function EmployerCtrl($scope) {
    $scope.employers = [
        {
            first_name: 'Naimul',
            last_name: 'Naim',
            is_married: true,
        },
        {
            first_name: 'Karar',
            last_name: 'Habib',
            is_married: true,
        },
        {
            first_name: 'Tanmoy',
            last_name: 'Zadid',
            is_married: false,
        }
    ];
}
```

Fig. 1: a function is written called EmployerCtrl where it shows some specific information of the employers

 

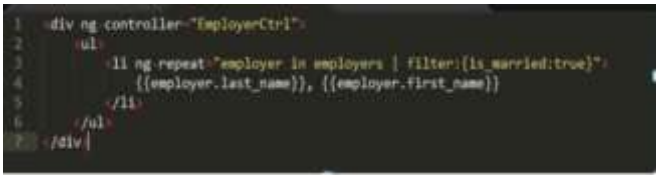


Fig. 2: Written in AngularJS framework.

If one is not familiar with Ionic/Angular, this code snippet may raise some immediate questions of what is the $scope and what is the specific syntax here for the filter. But in React, one can use the existing knowledge of language fundamentals. The above functionality can be done using filter and map functions in React. The following

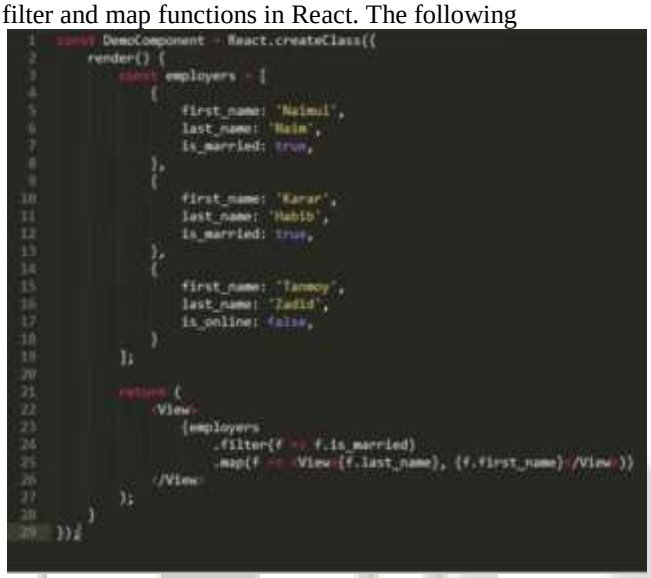


Fig. 3: Screenshot of how to use filter and map function in ReactJS

## IV. Environment Setup

There are plenty of text-editors to start working with. Most of them are open source and free of cost. Atom is one of them. It is a very useful text-editor. It has a great community of developers around it and they have enough useful plugin updates constantly. It is usable in every platform including Windows, Mac and Linux. For Windows users it is needed to install Git-bash while mac and Linux users can do the job from the terminal. The reason behind using Git bash in Windows is so that you can have access to the same commands that are available in Linux environments like Ubuntu distribution or on a mac laptop.

## V. Installing Nodejs Bundles

To install Nodejs, it is needed to go to the website called nodejs.org and there are couple of download facilities available for different operating systems. Node can be downloaded from there. This download will install a couple of things. First it will install nodejs. Nodejs allows creating a web server so that React components can be used locally and can be deployed to the web directly. It also installs node package manager called npm which will let us install various third party modules like React into our applications. Figure 4 shows a screenshot of some initial setup.


Fig. 4: Screenshot of installation of Nodejs and npm package manager.

## VI. Creating Web Server

To start working with React it is necessary to create a simple webserver beforehand. Without a webserver, there is no way to see the files in the browser. After opening the terminal we can use npm-init to create a new node project. Here first we create a folder in the desktop to store the project named HelloReact. We run npm-init comand from the terminal and it creates one file in the project. It gives a little introduction of what exactly it is doing and then asks a few basic questions. Figure 5 will show how to create a package.json file for the project.


Fig. 5: Screenshot of Creating package.json file in project folder.

As shown in Figure 5, running npm-init command in HelloReact project lets us walk through the setup for the package.json file. After answering all those questions including the file name, version, author and license, a little file prints out telling that it is about to write a file into our HelloReact project called package.json. This file is used for not only the node server but also to manage React dependencies. Now, if the new folder called HelloReact is opened inside the Atom editor it shows the package.json file there looks exactly what has been printed out in the terminal.

 

Fig. 6: shows the package.json file in the editor inside HelloReact Folder.

Now the first module is to be installed. A module is a third-party code or library we will use in our application. The following command has to be typed in the terminal to install the module.

```
npm install express@4 -save
```

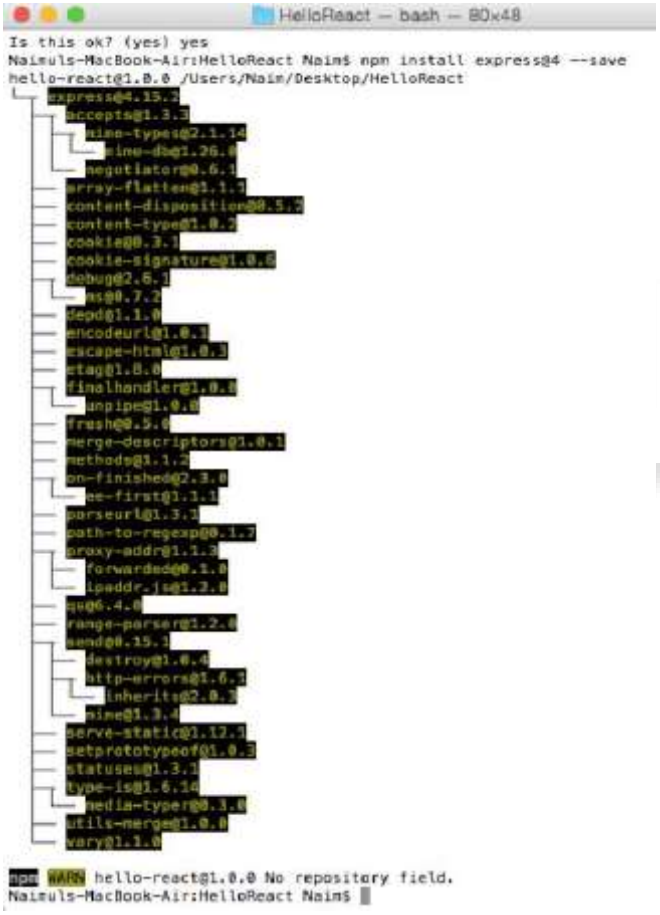


Fig. 7: is a screenshot of installing an express module in the project.

As shown in Figure 7, when the express module is saved in the project folder all the dependencies are also automatically downloaded. To write the -save flag is important because without saving the flag it is not going to update the package.json file which stores all the dependencies. Inside Atom there is a node-module folder which has plenty of files in it. And also in the package.json file an express module is already installed.

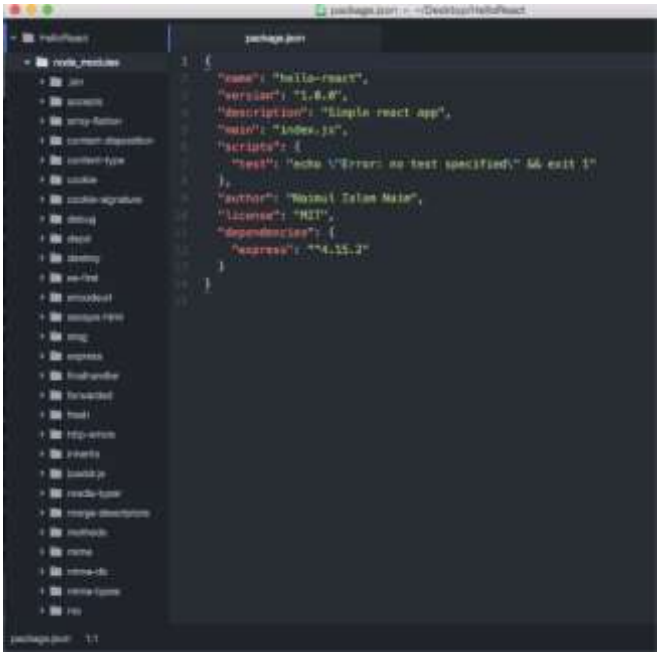

Fig. 8: is a screenshot of a node module folder with a plenty of files inside it.

As shown in Figure 8, node_modules in the project folder contain all the important files with dependencies. This same action can be done only with npm install command as long as the "express" module is present in the package.json file. Only command npm install can do the job as well. Now in the root of the project a new file called server.js has been created and let us run it in localhost port 3000. Figure 9 is a screenshot of the file server.js inside the project folder as follows.

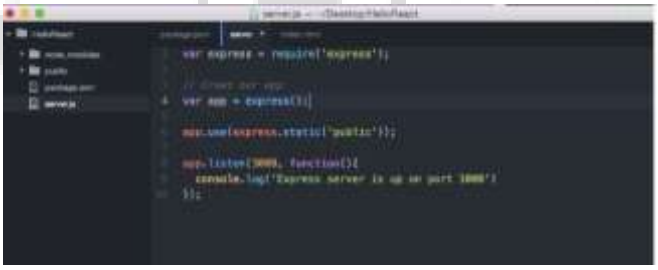

Fig. 9: Screenshot of Server.js file in the public folder.

As shown in Figure 9, the public folder has a server.js file in it. A public folder and index.html has been created in the root of the application. Index.html is the default file of the application. Figure 10 shows a public older in the editor is a screenshot.

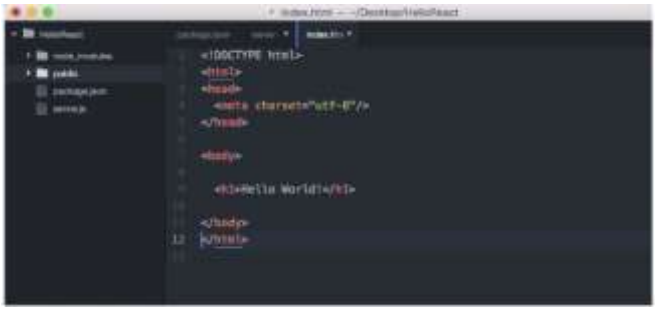

Fig. 10: Screenshot of Index.html as the default file of the project.

As shown in Figure 10, the public folder inside the main project folder contains the package,json and server.js

file. Now if the command ls is run in the terminal it can be seen that all the files and folders are showing up in terminal. Figure 11 shows if every file and folder is placed inside the main project folder properly.

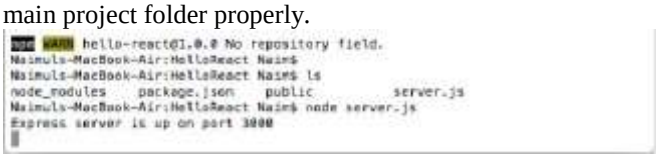


Fig. 11: Screenshot of all files showing up in the same project folder.

As can be seen in Figure 11, ls command shows all the files and folders inside the project folder and the express server is ready to run on port 3000. Now let us run the server in web typing node server.js command in the terminal.


**Hello World!**

Fig. 12: Screenshot of project deployment in localhost with node server.

As Figure 12 shows, the project is deployed in the localhost with node server at port 3000.

In this section, the procedure of using ReactJS in the local machine was described thoroughly. There are several other text editors available to start writing code. Atom is one of them and it is a good one. It has many plugins available and is easy to follow.

## VII. React virtual DOM

DOM stands for Document Object Model. DOM manipulation is very important for modern interactive web technologies. It is often called the heart of the modern web. It is an abstraction of the structured text. But it works slower than other JavaScript operations because most JavaScript frameworks usually update the DOM even if they do not need to do it. That means those updates are not necessarily required to perform the actions but they still do by default. For example, let us assume nine items have been put in a shopping basket in an online web store. Now let us say only the first item is needed to buy and proceed to checkout. Here, most technologies would rebuild the entire list that has been put in the basket. This means the framework has to unnecessarily work ten times more. Because of only one change the system has to rebuild the list exactly how it was before. React did not invent Virtual DOM but uses and provides it to the developer community for free. Virtual Dom is simply an abstraction of HTML DOM. React has a corresponding virtual DOM object for every DOM object like a correspondent or a lightweight copy. Virtual DOM is also characterized with similar properties to a real DOM. However, it cannot make any changes directly to the view. DOM manipulation is quite a slow process. But manipulating Virtual DOM is faster because it has nothing to do with the view part and does not make any changes to the screen. Figure 13, reprinted from stackoverflow.com, is an illustration of Virtual DOM in the memory.

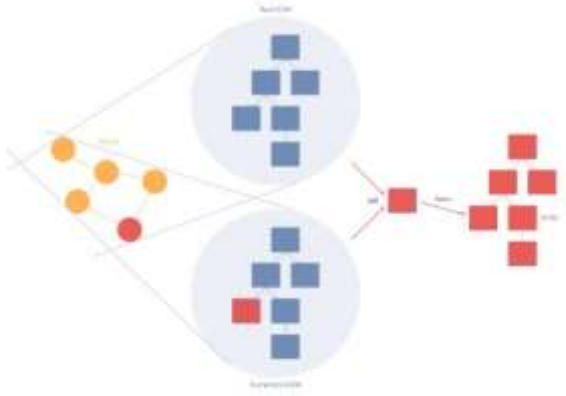

Fig. 13: Screenshot of React virtual Dom in memory. Reprinted from stackoverflow.com (2017).

As shown in Figure 13, a React virtual DOM in the memory is a lightweight copy of the real DOM. React uses a method called "diffing" which means rendering a JSX element gets every single Virtual DOM updated. This might sound inefficient but, in fact, it costs nothing as Virtual DOM is quite fast to get updated and does not make any impact in the process. After the DOM gets updated React compares the updated DOM with a pre-updated state of the DOM and determines which virtual DOM has been changed. Once React detects the changed DOMs, React updates only those objects to the real DOM. Thus, React makes the updates faster through Virtual DOM. In the above-mentioned example, React would have updated only the checked item from the list and leave the rest of the items alone. This makes the difference when updating a page in an application while React can only make changes to the necessary parts of the DOM. This virtual DOM manipulation process is one of the main reasons why React is gaining much popularity among the developer communities.

## VIII. React Virtual DOM Pros and Cons

Among the many advantages of the ReactJS library, a few of the key advantages are described here.

- The diffing algorithms written in React is quite fast and efficient
- Inclusion of JSX and hyperscript let us build multiple frontends for the same application.
- It is very lightweight and capable to be run in every mobile device
- Lots of tractions and mindshare
- It can also be used without React as an independent engine

A few disadvantages of React are as follows.

- It occupies quite much of the memory. Full in memory copy of the DOM.
- Static and dynamic elements don't bring much of a difference.

## IX. One-Way Data Flow

Frameworks like Angular and Ember use two-way data binding. In a two-way data binding for example in Angular, if a model is changed, the view also automatically gets changed and vice versa. An input field in the model can also

 

mutate the model. It performs well in most of the applications but sometimes it may lead to cascading updates and changing to one model may cause updates in other models. Again, since the state is mutable by both view and controller, the data flow can be unpredictable in some cases. Flux or Redux with React can be a better solution to avoid those uncertainties since both architectures follow one-way data flow. One-way data flow does not make cascading updates and changes in view. One way data flow ensures that data flows throughout the application in a single direction to offer more control between the states and models in an application. One way data flow also makes the architecture less complicated and understandable. Flux architecture is a functional approach. Here the view is considered as a function of the application state. Eventually, if the state gets some changes the view also gets re-rendered automatically. Moreover, a similar view is generated from the states and gives a better understanding and predictability to the application. To make it more predictable, in an application, data from parent to child flows in a single direction. Any data can be updated from any view, anytime in this approach. In case something goes wrong, debugging is also made less complicated in this way.

## X. REACT COMPONENTS

Components are very important for React. It is often considered as the heart of React, which is a collection of components. It is small reusable UI element that provides data to the view and changes over time. [6] To create the entire UI, those small components are then composed together, nested inside one another. Components let the UI (userinterface) to be split into small pieces and to design and build in a comprehensive way. UI stands for user interface, i.e. what is shown on the screen. Components are like JavaScript functions. They literally perform the same task but in different environment and different approaches. Like functions, they take inputs called props and return React elements. Those elements describe what the user sees in the interface on the screen. React components can be used to build the entire interface or even a part of it. Creating a React Component A React component can be simply written as a JavaScript function. This function accepts props and returns a React element. They are called as functional components. ES6 class can also be used to define a component. function Welcome(props){return <h1>Hello, {props.name}</h1>;} A React component can also be created in several other ways. To extend or to inherit or to derive a class from the main component which it attached to object is another wayto create a component. Functional components can also be stateless. Rendering every component builds the user interface experiencing faster and efficient.

## XI. INTRODUCING JSX SYNTAX

JSX is neither a string nor HTML. It is statically typed syntax extension to JavaScript. It is similar to an object-oriented language which is designed to run on modern web browsers. JSX is recommended to be used with React to design and build the user interface. While it comes with the full power of JavaScript it might even seem as a template language too at the first glance though it is not. The React element is produced by JSX. It can be rendered to the React Virtual DOM.

## REFERENCES

[1] Stefanov Stoyan, editor. React: Up and Running: Building web Applications. First Edition; 2016.
[2] Horton Adam. Vice Ryan, author. Mastering React; February 23; 2016.
[3] Stein Johannes, author. ReactJS Cookbook. December 6; 2017.
[4] Masiello Eric, author. Mastering React Native. January 11; 2017.
[5] A javaScript Library for Building User Interfaces [Online] URL: https://facebook.github.io/React/.
[6] React Native Bringing Modern Web Techniques to mobile. URL:https://code.facebook.com/posts/1014532261909640/React-native-bringing-modern-web-techniques-to-mobile/.
[7] Github React Native [A framework for building native apps using React native] [Online] URL: https://facebook.github.io/Reactnative/.

 