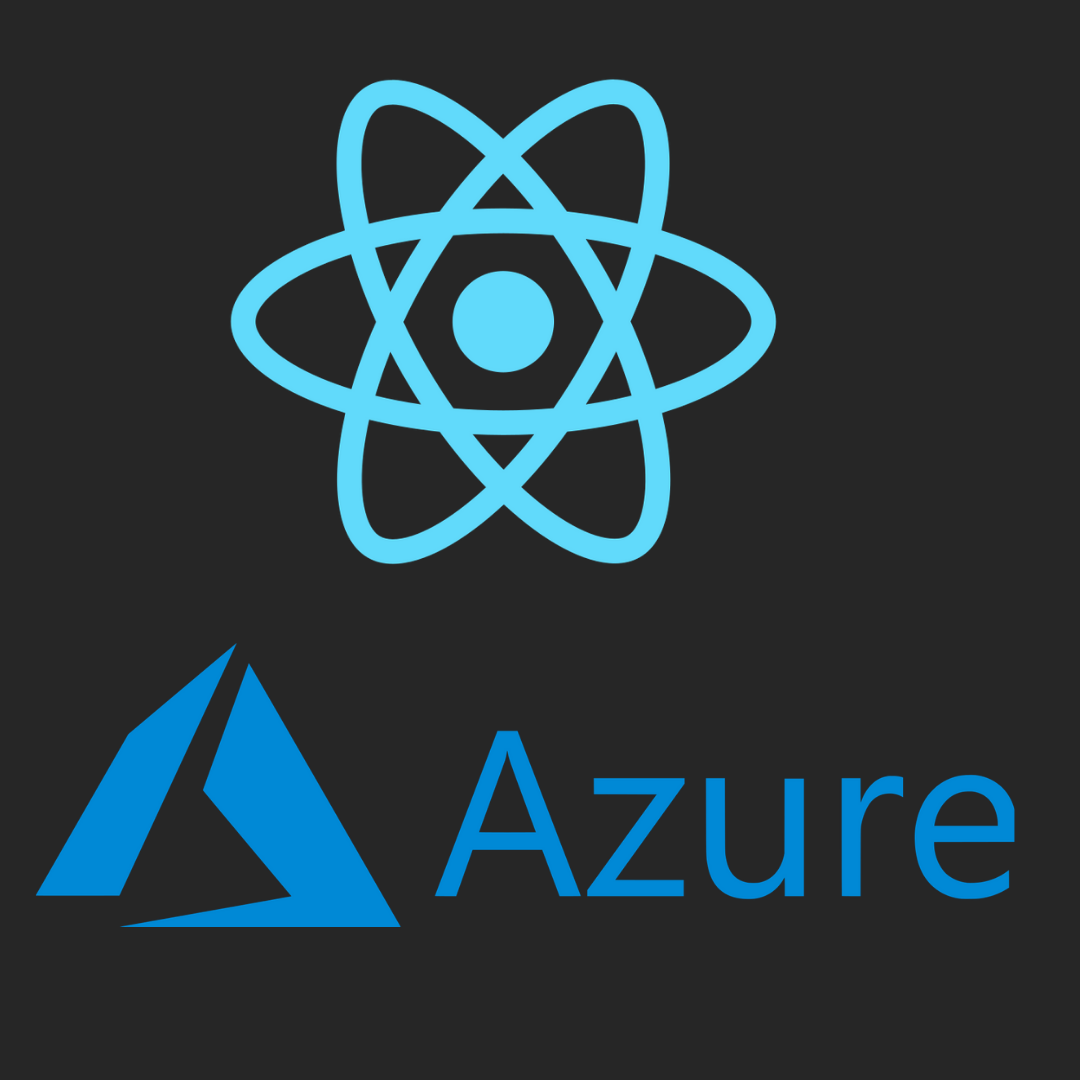
Plan & Document Analysis

Utilizing React

Marjorie Silva



# Introduction

In the following essay, I have developed a plan and document style architectural analysis on utilizing React. React is the leading JavaScript frontend framework today and its popularity just keeps increasing. According to Google Trends, React has surpassed other frameworks like VueJS and AngularJS in a matter of a year.

Graphical user interface, text, application

Description automatically generated

For this reason, I chose to analyze React by researching the following topics:

* What is React used for?
* What are the advantages and disadvantages?
* Examples of where React is utilized.
* Deploying React to Azure.
* Follow up on deployment to Azure.

Writing this analysis will give me the background I need to be a successful team player for the final project we are about to being. My team and I have decided to utilize React to complete our final project and we will gain productive knowledge from this plan and document style architectural analysis to lead us to success. At this moment, we are struggling with coming up with project ideas so I decided to research popular applications that utilize React to give us an idea of good ways to implement React.

## React

React is an open-source JavaScript library that is used for building user interfaces. It handles the view layer for web and mobile applications. One of the most important principles is that it allows us to create reusable user interface components. The main objective for React was to develop user interfaces that implement fast, simple and scalable applications. Utilizing React allows developers to create large web applications that changes data without having to reload the page. React has a great reputation on its own, however, other JavaScript libraries and frameworks can be integrated with it. According the reactjs.org website,

“React can be used in any web application. It can be embedded in other applications and, with little care, other applications can be embedded in React.”

Having flexible design principles is what every developer hopes for when building applications. React is the “view” part of the MVC pattern when used. Next we will discuss some advantages and disadvantages of React.

## Advantages & Disadvantages

Just like everything else in the world, React has some advantages and disadvantages that developers should be aware of when getting started. Throughout my research, I found that the following advantages and disadvantages were the most important and common ones discussed by many developers. Below are some advantages of React:

1. Easy to learn
   1. Basic JavaScript, HTML and CSS knowledge will allow you to easily get started.
2. Simplicity for dynamic web applications
   1. Used JSX(JavaScript Extension) that allows the mix of HTML and JavaScript.
3. Reusable components
   1. Helps applications to be easier to develop and maintain.
4. React Native
   1. Creates mobile applications for IOS, Android and web applications.
5. Rich library
   1. Allows for more flexibility
6. Virtual DOM
   1. Updates changes applied without affecting other parts of the interface which increases performance.
7. One direction data flow
   1. Makes code stable because it uses downward data binding where changes made to child structures do not affect parent data.

These are just a few advantages I encountered but there are many more that can be discussed in the future. I found that there were more advantages than disadvantages, we will move on to discuss some disadvantages below:

1. High pace of development
   1. Changes are frequently made making developers learn new ways regularly. Developers Michael Jackson and Ryan Florence described this by saying,

“In case you didn’t notice we’re driving a car here with two flat tires, the hood just flew up in front of the windshield, and we have no clue what its going on anymore!”

1. Poor documentation
   1. This is due to the fact of the constant and accelerated updates. It does not allow proper time for documentation.
2. JSX
   1. Can be a problem for junior developers who will need to implement HTML and JavaScript which can lead to confusion.

React has proven itself efficient enough to gain the popularity it has and be supported by large companies. Now, we will look at some of those companies who use React today to enhance their high performance and good quality for their business.

## React Applications

React was first introduced by Jordan Walke, a software engineer at Facebook in 2011. The objective was to create a library with a new way of rendering webpages, making them highly dynamic and have faster response time for user input while bettering the user interface. It did just that, so it is with no surprise that other major applications quickly adopted the React framework. Below are other applications using React today:

1. Instagram
   1. Discussing the global popularity of Instagram would consist of many hours. Instagram can thank React library that for the wonderful features it has allowed the app to implement. For example, an article from brainhub.edu said,

“Instagram is completely bases on the ReactJS library and has let fans fully adopt to its amazing features… A proof for that is the numerous features including geolocations, Google Maps API’s, search engine accuracy as well as tags that pop out without hashtags.”

1. Netflix
   1. Netflix utilize React because they found that it helps with startup speed, runtime performance and modularity to name a few.
2. WhatsApp
   1. The new WhatsApp web application utilizes React in a similar manner to Facebook.
3. Twitter
   1. In recent year, Twitter switched their mobile front-end and built their user interface with React.

All of these applications and more receive great benefits by upgrading to React and according to reactjs.org on October 20th, 2020, React 17 is being released with some upgrades BUT without new features, which is a disadvantage we discussed earlier on the fast evolving, high pace development.

## Deploying React to Azure

There are many articles and tutorials on deploying React to Azure that give great information to move forward with this step. The steps to follow for deploying to Azure using GitHub are:

1. Create a new App Service on Azure.
2. Create new GitHub repository.
3. Clone GitHub repository.
4. Create a new React App
   1. npx create-react-app azure-react-demo
5. Install react-router as a dependency
   1. npm i react-router-dom
6. Verify you are in the right directory
   1. cd azure-react-demo
7. Start the app
   1. npm start (should be directed to localhost with React logo)
8. Modify create-react-app app.
9. Upload to GitHub.
10. Deployment Center in Azure to connect Local Git/GitHub
11. Add node version to package.json
12. npm build in terminal

The above procedure is similar to what how we have deployed Node.js to Azure. I decided to also include another deployment in this analysis. Below are steps to deploy React through Azure DevOPs:

1. Create a new react app
2. Create build pipeline in Azure DevOps
   1. Create new project
   2. Push local code to Azure repo
   3. Copy code from “Push existing repo from command line” and paste to terminal
   4. Pipeline 🡪 Create pipeline 🡪Classic Editor 🡪Azure Repo Git🡪 Empty Job
   5. Add agent jobs: npm install, npm build, publish artifact drop
   6. Build and run
   7. Triggers
3. Create Azure Web page
4. Create new pipeline 🡪 empty job 🡪stage name Dev🡪enable trigger
5. New tasks: Azure web app
6. Make a change in React app and push
   1. Check if build is triggered, and build process and deployment should work
7. Go to Azure webpage to check site.

## Follow Up on Deployment

In most of the articles, a common discussion was how it is not a simple task to deploy React to Azure. We may come across many problems mainly client and server side routing, dependencies in package.json, folder issues, and even just not knowing exactly what everything does like I came across when using Azure DevOps. There are many pros to deploying azure, one being that it will get our app running live and once that’s done, it is simple to make changes. However there are also many cons especially for junior developers who just getting started with deploying live sites.

## Inclusion in Final Project

My team and I will be utilizing React, and as I mentioned earlier, this analysis was a great start to get us on the right track to make a good unique app. We now have basic knowledge of the background of React, its uses, pros and cons, and some ways to deploy. Even though I had trouble and spent a lot of time trying to figure out the deployment part of this analysis, I also learned a lot from it. I will continue to figure out the answers and have a deployed app on Azure very soon.

I believe that implementing React into courses at Lewis University is a great idea. For example, maybe in courses that deal with more web development. I have never taken a web development course at Lewis so I am not positive if it is already being implemented.

## Agile Spike

If I was going to conduct an agile functional Spike on the React app I recently built, I would try to estimate the timing of it to be completed in one sprint. During that time, researching should be conducted on the uncertainties of the app and ideas of how to advance or develop a better product.

“Deploy our React app to Azure and get some user feedback about the user interface and pages.”

Snippets of the React app are in Appendix A.

## Conclusion

Plan and document style architectural analysis take time and a lot of research to make sure the needed information will get in the hands of those who most need it. Learning React and utilizing it in next projects will be great experience because of how it is widely being used today.

## Appendix A

app.js

import React, { Component } from "react";

import "./App.css";

import Home from "./pages/home";

import Page1 from "./pages/page1";

import Page2 from "./pages/page2";

import { BrowserRouter as Router, Switch, Route } from "react-router-dom";

class App extends Component {

render() {

return (

<Router>

<Switch>

<Route exact path="/" component={Home} />

<Route path="/page1" component={Page1} />

<Route path="/page2" component={Page2} />

</Switch>

</Router>

);

}

}

export default App;

//home.js

import React from "react";

import { Link } from "react-router-dom";

const Page1 = () => {

return (

<div className="flags">

<div className="page page1">

<div className="top" />

<div className="center">

<h1 className="words">Hello</h1>

<Link to="/page1">Click to go to Page 1</Link>

</div>

<div className="bottom" />

</div>

<div className="page page2">

<div className="top" />

<div className="center">

<h1 className="words">World</h1>

<Link to="/page2">Click to go to Page 2</Link>

</div>

<div className="bottom" />

</div>

</div>

);

};

export default Page1;

References

<https://medium.com/devtechtoday/why-is-reactjs-gaining-so-much-popularity-6af4c43a3236>

<https://www.c-sharpcorner.com/article/what-and-why-reactjs/>

<https://www.javatpoint.com/pros-and-cons-of-react>

<https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/>

<https://brainhub.eu/blog/pros-and-cons-of-react-js/>

<https://rubygarage.org/blog/pros-and-cons-of-reactjs>

<https://reactjs.org/blog/2013/06/05/why-react.html>

<https://brainhub.eu/blog/10-famous-apps-using-reactjs-nowadays/>

<https://reactjs.org/blog/2020/10/20/react-v17.html>

<https://css-tricks.com/deploying-a-client-side-rendered-create-react-app-to-microsoft-azure/>

<https://medium.com/microsoftazure/deploying-create-react-app-as-a-static-site-on-azure-dd1330b215a5>

<https://www.youtube.com/watch?v=Ny5vJRfQito&feature=share>

<https://www.scaledagileframework.com/spikes/>

## 