# Technical Report 1:
# Software Analysis

**Teo Wei Han 30242711**

**Ong Jeng Yue 29637996**

**Cheong Chee Feng 30222397**

**Stuart Boey Chie Sheng 29519985**

**Ahmed Mohamed H A Nassar 28901797**

FIT3170

Semester 2, 2020

Monash University Malaysia

October 23, 2020

# Abstract

The purpose of this technical report is to aid the process of decision making when choosing technologies to create a web-based tutor allocation system by discovering possible resolutions to a few problem areas that have been identified in our project. This report will cover four areas of the project. Namely the frontend framework, frontend testing frameworks, hosting services and CI/CD pipeline.

The frontend is perhaps the part of the systems that almost all users will interact with, making it important that the framework chosen is well-suited to the task. The React framework is very popular and has all the helpful and necessary tools for us to easily create a fast and responsive user interface. Some members of the team also have experience with these frameworks which is an added bonus.

Frontend testing frameworks are important to ensure that what is deployed to the user is working properly. Therefore, we recommend JEST which is easy to set up and use. It also has all the testing tools needed to ensure that the testing can be conducted quickly and efficiently.

To display our system on the web, we will need to use a hosting service. Due to the system being newly created and rolled out without significant financial or infrastructure support, we recommend the use of cloud hosting. Therefore, we propose the use of Amazon Web Services(AWS), which besides being recommended by the client, is the leading cloud service provider. With the team having no experience with hosting, AWS also provides plenty of resources for the team to learn from.

Continuous Integration/Continuous Development (CI/CD) is necessary due to the size of the project and the number of people working on it. Having integration and development automated would also significantly boost the productivity of the team. We recommend the GitLab CI/CD for this as we are currently using GitLab as our version control system as well. GitLab's CI/CD is also not lacking in any way compared to other CI/CD and most of the team has had experience working with it.

# Table of contents

**Introduction** **1**

**1. Frontend** **2**
1.1. Frameworks 2
1.1.1. Angular JS 2
1.1.2. Angular 2
1.1.3. React 3
1.1.4. Vue 4
1.2 Limitations 4
1.3 Recommendations 5

**2. Frontend Testing** **5**
2.1. Frameworks 5
2.1.1 MochaJS 5
2.1.2 JEST 5
2.2. Recommendations 6

**3. Hosting services** **6**
3.1. Types of hosting services 6
3.1.1. Shared Hosting 6
3.1.2. Dedicated Hosting 6
3.1.3. Managed Hosting 7
3.1.4. Colocation Hosting 7
3.1.5. Cloud Hosting 7
3.2. Hosting service providers 7
3.2.1. Heroku 7
3.2.2. Amazon Web Services (AWS) 7
3.2.3. Microsoft Azure 8
3.3. Limitation of analysis 8
3.4. Recommendations 8

**4. CI/ CD pipeline** **9**
4.1 Problem Statement 9
4.2 Tools 9
4.2.1 GitLab 9
4.2.2 Jenkins 10
4.2.3 CircleCI 10
4.3 Limitation of analysis 10
4.4 Recommendation 11

**Conclusion** **11**

**References** **12**

# Introduction

In this technical report, team members are tasked to research problem areas that exist in our tutor allocation system. Team members will make recommendations of suitable solutions based on their research for each problem area to assist the system architects' decision making.

There are four sections to this report. The first and second section are both related to the frontend of the project, the first section will explore the different types of frontend frameworks that we can take advantage of in our project and the second section will look into frontend testing frameworks that can reduce the amount of defects present in our presentation layer.

Third section will investigate the existing hosting services that we can use to host the web application that we will be developing. This is a crucial part for a web application because it will be related to the performance, accessibility, reliability and maintainability of the web application. Next, Continuous Integration/Continuous Development(CI/CD) tools are researched to ease the process of testing and deployment during the development of the tutor allocation system.

# 1. Frontend

Front-end is the user interface of a software which is what users can see (Concepta, 2019). For web applications, these often involve using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS). To help in creating the frontend, we will be using a framework.

## 1.1. Frameworks

A framework is a platform “where common code with generic functionality can be selectively specialized or overridden by developers or users”(Techopedia, 2018). The frameworks are in the form of libraries, where a well-defined application program interface (API) is reusable anywhere within the software under development(Techopedia, 2018). By relying on these frameworks, it helps us to expedite the process of building up the front-end.

When picking our framework, we have to take into account the speed, scalability, ease-of-use and popularity of the framework. Speed and scalability are required as the application would be running for the hundreds of Monash University FIT faculty staff. This would mean the application would have to take on large amounts of tasks and complete it quickly. Ease-of-use and popularity was requested by the client who wanted hiring for future maintaining and scaling to be more convenient.

With many frameworks out there to support development of the frontend, we have evaluated the following:

### 1.1.1. Angular JS

First released in 2009 by Google, AngularJS is a JavaScript-based open-source front-end web framework. Angular JS has declined in popularity since then and has been replaced by Angular which is mentioned next. Since Angular JS is being phased out by the Angular team and will only be supported until the end of 2021(AngularJS, 2020), it is not being considered for this project.

### 1.1.2. Angular

A complete rewrite of Angular JS and released in 2016, Angular is a TypeScript-based open-source web application framework created by the Angular Team at Google and a community of individuals and corporations.

In StackOverflow’s 2019 Developer Survey(StackOverflow, 2019), Angular was the second most-used framework by professional developers with 32.4% of the total, making it a very popular choice. Being created by Google means we can be assured experienced professionals are responsible for all the updates, support, and further development. Compared to another popular choice, React, Angular is a full fledged web design framework. Thus, providing greater consistency throughout the entire application. Performance-wise, since Angular mostly relies on enhanced HTML, it renders a web page faster as HTML is a

web markup language (ProgramAce, 2019). In contrast, JavaScript, used predominantly by React and Vue, would slow down the loading speed of the page.

Angular is a complete framework with all the tools a modern JavaScript project needs. On the other hand, this means Angular has the steepest learning curve which has to be taken into account as most of the team is unfamiliar with Angular. It is also a notoriously hard framework to pick up. Angular also uses Typescript which allows static checking, making certain issues easier to debug. However, since at least half of the team has not used Typescript, this means it also adds to the learning curve for some team members.

### 1.1.3. React

ReactJS, or more commonly known as React, is an open source front-end JavaScript library mainly used to build user interfaces or UI components mostly for web or mobile applications (ReactJS, n.d). The developers of ReactJS claim that using the library makes it seamless and pretty straightforward to create an interactive UI, and emphasizes on the synchronization between the back-end data and their representation in the front-end. In a MVC architecture, ReactJS, being the front-end framework, acts as the view elements, and its extensive library components allow developers to build either a simple or complex UI to meet their requirements (Coder Academy, 2016).

React is based on the concept of reusable components, which simply translates to blocks of codes that are modular enough to be reused throughout the project with no to little modification or additional plug-ins. When it comes to the underlying foundations, ReactJS uses JSX, Javascript XML, which is a syntax that combines Javascript and HTML as the name suggests. Some of the successful websites and web applications that were developed were using ReactJS, amongst these websites are Adroll, AirBnb, and Asana.

According to Geeks for Geeks, an organization that focuses on development languages, ReactJS is the number one most popular JavaScript Frameworks for web development (geeksforgeeks, 2020). The fact that ReactJS is built on reusable components aid in a smooth and overall faster development (Willoughby, 2017). Additionally, ReactJS avoids the trouble caused by bottlenecks that exist due to the document object model where a small change at the upper layer can cause ripples to the interface by using Virtual DOM. On the other hand, one of React weaknesses is the lack of proper documentation, and that is due to the fact that React technologies are updating and accelerating too fast (DDI Development, 2017). This leads to another point of weakness, which is the acceleration rate of React development. With such a high acceleration rate, developers might find themselves discouraged from relearning new techniques regularly (Java T Point, n.d.).

Having said that, due to the self-evaluation sheet filled by the team members, ReactJS seems a valid choice of framework for the front-end part of the project due to the lack of sufficient experience to use other more demanding frameworks. With reusable components, familary language, and spikes, this framework can be adapted quickly and effectively by all the team members. However, some effort needs to be put into learning more about this framework as it is constantly evolving.

### 1.1.4. Vue

Vue is a progressive framework that is used, like most frameworks, to build user interfaces. Vue is primarily designed to be incrementally adoptable throughout the development stage of a project (VueJS, n.d.). Similar to most frameworks, Vue plays the role of a View in an MVC software architecture, and is said to be capable of easily integrating with other projects or existing libraries. Additionally, Vue is capable of giving power to single page applications with the help of modern tooling and supporting libraries developed by the communities and available on github, accessible to the public.

What differentiates Vue from other existing frameworks such as React and Angular, is a concept called “high decoupling”, which simply means that a change in a piece of code does not necessitate a change in another. While other frameworks exhibit this concept, Vue focuses on it even more. Because of that, it is easy to keep adding additional modules to the project without having to refactor change existing code. Additionally, Vue’s core library is vast enough to cover most small visual components, and thus, Vue’s users can reduce the dependencies of the system to multiple external libraries (Mamani, 2019). Vue has been adapted by multiple companies, amongst them are very successful ones such as Xiaomi, Alibaba, and Gitlab (Mamani, 2019).

According to  Geeks for Geeks, Vue.js is ranked as the third most popular framework that uses JavaScript for web development (geeksforgeeks, 2020). Top features of this framework include declarative code (Thiessen, 2019). Declarative code or programming is a style of building the program expressed in the logic of the computation without describing the control flow. Simply put, it means instructing a program on what it needs to be done instead of how it is done (Pererio, n.d.). Aside that, Vue.js is praised for its simplicity, integration capabilities, user-friendliness, and the good documentation it comes with (Sidorenko, 2019). On the other hand, Vue presents a disadvantage when it comes to supporting larger projects as it is a relatively new framework without much financial support. Another disadvantage would be that there are limited resources. Aside from the core library, you may not be able to find as many resources and plugins as those that can be found with other frameworks (Altexsoft, 2019).

To summarise, Vue.js is definitely a capable framework and is fairly easy to use mainly due to its core library supply and modularity. However, VueJS is aimed for smaller projects, and might impose a challenge for larger projects. Thus, the decision to use Vue or an alternative software partly depends on the size of the project.

## 1.2 Limitations

We were unable to test out all of the aforementioned frameworks due to time constraints. Thus, we only conducted a spike for React. Another limitation was that we are unable to research the existing framework that Monash employs as we have yet been able to contact the people in charge to find out the framework that they use.

## 1.3 Recommendations

Having studied the options and alternatives, the team is recommended to use ReactJS as the framework for this project as it aligns with the project scope and requirements. Aside from the learning opportunities adopting ReactJS presents, the modularity and resources of this framework provides an opportunity to develop an interactive user interface tailored to the needs of the client, enabled by availability of different usable components that are integratable easily. Furthermore, the reusability of these components allows the project to expand and extend as it needs. Additionally, the familiar language of JavaScript and HTML makes it convenient to develop a web-based application interface, which is exactly what the client is looking for. Another excellent feature that would aid the development phase is the presence of Virtual DOM in React, minimizing issues arising from changes in the document object model and prevent a ripple down effect that would hold back the progression of the project.Beyond that, adopts ReactJS is the perfect opportunity for the students to expand their knowledge and increasing their skills. Having said that, ReactJS is more than sufficient to cover all the project current and foreseeable requirements with almost a tutorial or online learning resources available for all features of the system being developed.

# 2. Frontend Testing

## 2.1. Frameworks

A testing framework is a collection of tools and processes working together to support automated testing of any application. There are a few testing frameworks for JavaScript but the two most popular ones are MochaJS and JEST (stateofjs, 2018), which we will take a look at below.

### 2.1.1 MochaJS

MochaJS is a JavaScript test framework for Node.js programs, featuring browser support, asynchronous testing for both frontend and backend, test coverage reports, and use of any assertion library. It has robust documentation support and has proven to be a well-established framework over the past few years(Unadkat, 2019). “Hosted on Github, Mocha is recognized for its flexibility, and as a result, it has proven to be one of the most depended upon libraries among JavaScript developers.” (Unadkat, 2019).

A cost that comes along with this flexibility is that more configuration to set up is needed(Nedrich, 2017). Assertion libraries, mocking frameworks and others need to be chosen and installed manually.

### 2.1.2 JEST

JEST is a JavaScript testing framework maintained by Facebook. It provides a “zero-configuration” testing experience and is compatible with most projects such as NodeJS, React, Angular, VueJS, TypeScript and others(JEST, 2020). With a clear and convenient user interface and a lot of resources online, JEST is an increasingly popular

testing framework(Unadkat, 2019). It is fast, safe, has code coverage, easy mocking and great exceptions(JEST, 2020).

A weakness of JEST is caused by it being newer and less widely used. This has also led to less tooling and library support available compared to more mature libraries such as MochaJS.

## 2.2. Recommendations

Both testing frameworks are fast, small and have most of the testing requirements needed by our systems. However, JEST would be more suitable for our project since it has minimal configuration and setup, allowing it to be implemented and used more quickly by our team. This also prevents extra time spent on picking our own libraries for certain things like assertions and code coverage that we will definitely use in the Tutor Allocation Systems. JEST is also a highly preferred testing framework for applications based on React (Unadkat, 2019) which is what the client recommended.

# 3. Hosting services

Web hosting is a service that grants us the ability to display a website on the internet, making it available to everyone (Web Host). To be able to host a complete working website, hosting service that supports the hosting of a frontend, backend and database are required.

## 3.1. Types of hosting services

There is a wide variety of web hosting services (Hosting Types) available at the time of writing this report, to better understand the differences between them, a more detailed description of types of hosting services is listed below.

### 3.1.1. Shared Hosting

Shared hosting is the most common type of hosting because of its relatively low cost. The term “shared” means that the website we will host will be sharing the same server as other websites that are hosted as well. Thus, the performance of our website is highly influenced by more popular websites running on the same server.

Additionally, there will be no access to server configurations since a shared web hosting is managed by a system administrator. However, if our project is relatively small and does not incur a lot of traffic, shared hosting could still be considered as an option.

### 3.1.2. Dedicated Hosting

Dedicated hosting means renting a physical server that is located in a hosting company’s data center and have full control of the computer. This means that hosting websites with a lot of traffic will not be an issue as it is not sharing the same server with any other websites. The downside to this that it is the renters responsibility to provide technical support. This could mean hiring a team of in-house specialists. Taking our project into consideration, this option of hosting is not viable.

### 3.1.3. Managed Hosting

Managed Hosting is very similar to dedicated hosting. The difference being that the renters are not responsible for maintaining the computer anymore as the hosting company manages it. However, this means that the renters do not have full control over the server. As it is quite similar to dedicated hosting, this option of hosting is also not a viable option.

### 3.1.4. Colocation Hosting

Colocation is the most expensive type of hosting. With this option of hosting, we do not share or rent computers, we instead choose the hosting companies's data centers as server locations. This means that we have full control over settings, updates, software, data storage and more. Considering the scale of our project, this is also not a viable option.

### 3.1.5. Cloud Hosting

The latest type of hosting is also known as cloud hosting. It uses a system of clustered servers, which also means that the responsibility of hosting is spread across multiple connect servers. There is an increase in popularity for cloud servers over the years. This type of hosting is the best for startups.

## 3.2. Hosting service providers

Our project will most likely be a dynamic website (Dynamic website) as it should be quite functional, has a database design and requires web programming. Given the scale of our project, cloud hosting will be the best option to go with. Presented below are a few of the options that have been explored.

### 3.2.1. Heroku

Heroku is a cloud platform that lets companies build, deliver, monitor and scale apps. Heroku's hosting service supports several programming languages such as Python, Node.js, Ruby, Java, Scala, Clojure, and Go. Heroku's service will eliminate the pain of needing to install software, maintaining it, monitoring the software for required updates and setting up deployments etc.

Heroku also has Postgres support available (Heroku Postgres). Postgres is one of the world's most popular relational database systems. Heroku Postgres will enable us to maximize our data instead of spending time on database setup and maintenance. We can also rest easy knowing that Heroku Postgres is being protected by behind-the-scenes efforts. There are also many resources available on deploying a full stack onto Heroku (PERN Deployment).

However Heroku also comes with its own disadvantage, which is that we will not have full control over our project's infrastructure. As for the pricing of Heroku's services, the price ranges from free up to enterprise level pricing.

### 3.2.2. Amazon Web Services (AWS)

Amazon Web service is Amazon's subsidiary. It is a cloud service platform that provides computing resources over the internet to host your web applications. AWS in nature is

flexible, reliable, and secure. It has a number of useful features such as the Amazon S3 (a special object storage to store, collect, and analyze data), AWS Relational Database Service (a service that makes it easy to set up, operate, and scale relational databases in the cloud), Autoscaling (a service that let developers automatically adjust server capabilities to correspond to the current demand), or Elastic Load Balancer (a service that distributes incoming application traffic across multiple cloned servers) that will ultimately ease the process of application development.

The AWS cloud system is based in 16 geographic regions with 44 Availability Zones in total (Regions and Zones).

The pricing for Amazon's service is flexible in that, you pay for what you use. However, there are free options such as the Amazon EC2 which is available for free for 12 months upon registration with a run time of up to 750 hours.

### 3.2.3. Microsoft Azure

Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools, and frameworks.

Virtual Machine service by Azure can be used to host the entirety of the web application that the team develops.(Microsoft, 2020) Other than that, Azure also has the Web App service that allows for deployment of scalable web application backends, it has support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. For database hosting, Azure can support both SQL and noSQL with services such as Azure Database for PostgreSQL, Azure Database for MySQL and Azure Cosmos DB.(Microsoft, 2020)

Azure offers the pay as you go model as well as up front payment for reserving virtual machine instances for one or three years which can reduce resource costs by up to 72% from pay-as-you-go prices. Pay as you go virtual machines are more well suited for applications with short-term, spiky, or unpredictable workloads that cannot be interrupted, which should be more suitable for our system as the allocation of tutor activities will be carried out in a certain period of time during the year.(Microsoft, 2020)

## 3.3. Limitation of analysis

As of writing this report, the team does not consist of anyone who has any experience in hosting on any of the cloud platforms above. Thus, our understanding and decisions made were based strictly on the readings and reviews that are available to us. Spikes can be created for testing purposes, but the extent of issues and problems that can be detected is minimal as these issues would normally only arise when a project scales in size.

## 3.4. Recommendations

After going through the three hosting services listed above, it is known that Amazon Web Services and Microsoft Azure are comparable in terms of service provided, flexibility and

pricing.(Netapp, 2020) The recommended hosting service will be Amazon Web Service, other than the client having preference towards aws, it is also because of maintainability in the long run, aws was the leading cloud service provider in Q2 2020, accounting for 31% share in the worldwide cloud infrastructure services market.(Canalys, 2020) This means that aws should have more support in terms of learning resources for the team members. Besides, AWS' pay for what you use pricing is more suitable for our project that will not have steady traffic throughout the year, only a period of time before the start of a semester.

# 4. CI/ CD pipeline

CI/ CD is one of agile methodology's best practices that is implemented by most of the devops teams. It ensures the software development teams to deliver code changes more frequently while focusing on meeting the code quality, requirements and security as the deployment steps are automated. It helps developers to automate things that are required to do manually and increase the efficiency in detecting small problems early before they grow into major disasters.

Continuous Integration (CI) is a set of practices where the development teams make small changes and check in the code. It has developers to commit their code to a shared repository and each commit will trigger an automated workflow that will notify the developers of any issues integrating their changes. A better collaboration and software quality can be achieved as the developer will be more likely to commit code changes frequently.

Continuous Development or Deployment (CD) comes after continuous integration ends. It is the automation that pushes application to delivery environments so that it can be quickly demonstrated to customers.

## 4.1 Problem Statement

This topic is about choosing the right CI/CD tools that are suitable for building our system. There are no such tools that are able to work well in any circumstances, so an ideal way is to choose the best CI/CD tool that is able to meet varied needs while building our project. The chosen CI/CD tool must be able to deliver work collaboratively such as automated build, automated testing and automated provisioning for deployment in order to speed up the deployment. CI/CD tools with communication features such as sending notifications will be useful especially if something went wrong or it is ready for testing or to be deployed. Some tools with good features are not free and need payment.

## 4.2 Tools

Developer teams usually work with multiple environments other than implementation, such as development and testing environments. So with CI/CD tools, the environment-specific parameter that must be packaged with each delivery can be stored.

### 4.2.1 GitLab

Gitlab is a web-based DevOps tool for handling different aspects of software development life cycles. The tool provides a Git repository management, issue tracking, code reviews and continuous integration and deployment pipeline features. With the CI/CD feature, GitLab

allows each commit or push to automated build, run tests and deploy code. GitLab CI/CD has high availability deployments as the installation and configuration for GitLab CI/CD can be done with ease. It is free and has a self-hosted CI tool built into GitLab. This tool also consists of milestone settings which are good for tracking issues, improving on a series of issues and draw requests in a repository. Project milestones can be easily assigned to any issue or combine requests in that project. The auto-scaling CI runner in GitLab CI can easily be managed especially for a parallel testing environment. It is also being widely used for numerous open-source projects due to its good issue tracking and issue shuffling features. It allows parallel test pull requests and branches and the test outputs will display on GitHub UI. GitLab CI/CD enables code reviews and merge requests done easily.

### 4.2.2 Jenkins

Jenkins is an open-source automation server where the central build and continuous integration process takes place. The program is a Java-based self-containing program with packages for Windows, macOS, etc. Jenkins supports the construction, deployment and automation of software development projects by providing hundreds of plugins. Jenkins can easily be installed in any OS platform. It can be combined with Docker to increase consistency and performances for automating jobs. Jenkins offered 1500+ plugins compared to other CI/CD tools. It makes customization easier and more flexible since those plugins are ranging from language-specified development tools to build tools. Jenkins plugin integration also available for different DevOps testing tools. The setup and configuration process are easy and only require some steps for installation. The documentation available for Jenkins also provided helps in configuring. Jenkins also supports parallel testing, so it can be easily integrated with communication tools to get notifications if the build is a success or fail.

### 4.2.3 CircleCI

CircleCI is a cloud-based CI/CD tool that automates installation and delivery procedures, from code building, testing to deployment. As a cloud-based tool, it required lower maintenance of a constant local server host. The cloud-based server also ensures faster deployment of applications. CircleCI can integrate with GitHub, GitHub Enterprise and Bitbucket to create builds when new code lines are committed. It has an automated testing feature which automatically runs the pipeline in a clean container or virtual machine which allows testing for every commit. Development team will be notified with Slack integration if a pipeline fails so issues can be fixed as soon as possible. Automated deployment allows passing pipelines to be deployed to various environments so the product can be shown faster to the client.

## 4.3 Limitation of analysis

Our development teams are currently in the implementation phase so we are unable to do testing and deployment using suggested CI/CD tools to find out the best tool that could be used in our project. Some CI/CD tools provide a lot of plugins which are written by third parties, which vary in quality. Building a software based on third-party plugins is not a good way to ensure availability or stability.

### 4.4 Recommendation

It is best to recommend using GitLab CI/CD tools for CI/CD pipeline since our project is using GitLab as version control and able to control the git repository with total control over branches and other aspects to keep the code safe from risk. WIth the total control over branches, conflict issues can also easily be handled. Every single project has a tracker to track the problem and carry out review for improvement. Moansh also have its own FIT GitLab server so developers can have permission to maintain the system in the future.

## Conclusion

Overall, the recommendations for each of the area of concerns are:

| **Area of concerns** | **Recommendation** |
|---|---|
| Frontend | ReactJS |
| Frontend Testing Frameworks | JEST |
| Hosting Services | Amazon Web Services (AWS) |
| CI/ CD pipeline | GitLab CI/CD |

All four recommended tools for each of the areas of concern are popular and well established options for their respective domains. This is in line with the request by the owner to use popular technologies that will be easy to hire for in the future. All the recommendations are also the most well-suited to the personnel and project that we have at hand given what we know now.

Currently, we are unable to conduct a spike to test the integration of these technologies together as we do not have sufficient time. Although we do not foresee any issue when integrating all these technologies together, care must be taken to do this as early as possible in case the technologies are unable to work well together.

## Individual Contribution:

Work breakdown and time tracking:
https://drive.google.com/drive/u/0/folders/1JWjWNjnDFbh7HICQWRrVSMGcgwoldZEa

# References


Altexsoft. (2019). The Good and the Bad of Vue.js Framework Programming. Retrieved from https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/

AngularJS. (2020). *Version Support Status*. Retrieved from: https://docs.angularjs.org/misc/version-support-status

Canalys. (2020). Global cloud services market Q2 2020. Retrieved from: https://canalys.com/newsroom/worldwide-cloud-infrastructure-services-Q2-2020

CircleCI (2020) CircleCi website homepage. Retrieved from https://circleci.com/

Concepta. (2019). *What Is the Difference Between Front-End and Back-End Development?* Retrieved from: https://www.conceptatech.com/blog/difference-front-end-back-end-development

DDI Development. (2017). Pros and Cons of ReactJS Web App Development. Retrieved from: https://ddi-dev.com/blog/programming/pros-and-cons-reactjs-web-app-development/

Dynamic website. *Static vs Dynamic webpage.* Retrieved from: https://www.hughesandco.ca/blog/the-difference-between-dynamic-and-static-websites

Dzone (2020) CircleCI vs Jenkins: Choosing The right CI/CD Tool. Retrieved from https://dzone.com/articles/circleci-vs-jenkins-choosing-the-right-ci-cd-tool

Dzone (2020) Jenkins vs GitLab CI: Battle of CI/CD Tools. Retrieved from https://dzone.com/articles/jenkins-vs-gitlab-ci-battle-of-cicd-tools

Geeks for Geeks. (2020). Top 10 Most Popular JavaScript Frameworks for Web Development - GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/top-10-most-popular-javascript-frameworks-for-web-development/

GitLab (2020) Continuous Integration with GitLab. Retrieved from https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/

Heroku Postgres. *Heroku Postgres.* Retrieved from: https://www.heroku.com/postgres

Hosting Types. *How to Choose Proper WHS*. Retrieved from: https://yalantis.com/blog/types-of-hosting-solutions/

InfoWorld (2020) What is CI/CD? Continuous integration and continuous delivery explained. Retrieved from https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html

Jash Unadkat. (2019). *Top 5 Javascript Testing Frameworks*. Retrieved from: https://www.browserstack.com/guide/top-javascript-testing-frameworks

Java T Point. Pros and Cons of ReactJS - javatpoint. Retrieved from https://www.javatpoint.com/pros-and-cons-of-react

Jenkins (2020) Jenkins website homepage. Retrieved from https://www.jenkins.io/

JEST. (2020). *JEST website homepage*. Retrieved from: https://jestjs.io/

Katalon (2020) Best 14 CI/CD Tools You Must Know. Retrieved from https://www.katalon.com/resources-center/blog/ci-cd-tools/

Mamani, M. (2019). What is Vue.js and How do we Use It?. Retrieved from https://www.avantica.com/blog/what-is-vue.js-and-how-do-we-use-it

Matt Nedrich. (2017). *React Testing – Jest or Mocha?* Retrieved from: https://spin.atomicobject.com/2017/05/02/react-testing-jest-vs-mocha/

Microsoft. (2020). Linux virtual machines in Azure. Retrieved from: https://docs.microsoft.com/en-gb/azure/virtual-machines/linux/overview

Microsoft. (2020). App Service overview. Retrieved from: https://docs.microsoft.com/en-gb/azure/app-service/overview

Microsoft. (2020). Linux Virtual Machines Pricing. Retrieved from: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/

Netapp. (2020). Azure vs AWS Pricing: A Quick Comparison. Retrieved from: https://cloud.netapp.com/blog/azure-vs-aws-pricing-comparing-apples-to-apples-azure-aws-cvo-blg

PEREIRO, F. Declarative Programming: Is It A Real Thing?. Retrieved from https://www.toptal.com/software/declarative-programming

PERN Deployment. *PERN Stack and its Deployment.* Retrieved from: https://medium.com/tech-iiitg/pern-stack-and-its-deployment-488e01b209a

ProgramAce. (2019). *React vs Angular: What to Choose for Enterprise Application*. Retrieved from: https://program-ace.com/blog/react-vs-angular/

ReactJS. React – A JavaScript library for building user interfaces. Retrieved from https://reactjs.org

Region and Zones. *AWS available region and zones.* Retrieved from: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Sidorenko, I. (2019). What Are The Pros And Cons Of Using Vue.js. Retrieved from https://towardsdatascience.com/what-are-the-pros-and-cons-of-using-vue-js-3689d00d87b0

StackOverflow. (2019). *Developer Survey Results 2019*. Retrieved from: https://insights.stackoverflow.com/survey/2019#technology

Stateofjs. (2018). *Testing - Overview*. Retrieved from: https://2018.stateofjs.com/testing/overview/

Techopedia. (2018). *Software Framework*. Retrieved from: https://www.techopedia.com/definition/14384/software-framework

Thiessen, M. (2019). The Most Important Feature in Vue - Michael Thiessen. Retrieved from https://michaelnthiessen.com/most-important-feature-vue/

Threat Stack (2020) Tips for Choosing the Right CI/CD Tools. Retrieved from https://www.threatstack.com/blog/tips-for-choosing-the-right-ci-cd-tools

Top 32 Sites Built With ReactJS. (2016). Retrieved from https://medium.com/@coderacademy/32-sites-built-with-reactjs-172e3a4bed81

VueJS. Vue.js. Retrieved from https://vuejs.org

Web Host. *What is a web host*? Retrieved from: https://yalantis.com/blog/types-of-hosting-solutions/

Willoughby, J. (2017). The Top 5 Benefits of React that Make Life Better. Retrieved from https://www.telerik.com/blogs/5-benefits-of-reactjs-to-brighten-a-cloudy-day