

---

# **Technology Investigation Document**

**for**

## **FOES Data Management System**

**Version 1.0**

**Prepared by**

**Yeum Kyu Seok, Lee Chian Hui, Ng Jing Ru, James Wong Siew Hui**

**Curtin University Malaysia**

**7 April 2022**

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Revision History</b>	<b>3</b>
<b>Introduction</b>	<b>1</b>
<b>Frontend</b>	<b>1</b>
Introduction	1
Frontend Technology	1
Laravel Blade	1
ReactJS	1
Angular	2
Comparison	2
Conclusion	2
CSS Framework	3
Bootstrap	3
Tailwind	3
Comparison	3
Conclusion	4
<b>Backend</b>	<b>4</b>
Backend Technology	4
Introduction	4
Strapi	4
Laravel	5
Express	5
Firebase	5
Spring	6
Comparison	6
Conclusion	7
Database	7
Introduction	7
SQLite	7
MySQL	7
PostgreSQL	8
MongoDB	8
Firebase - Cloud Firestore	9
Comparison	9
Conclusion	10
Storage & Deployment	10
<b>References</b>	<b>11</b>

## Revision History

Name	Date	Reason For Changes	Version
All Members	7/4/2022	Initial Draft	1.0

# 1. Introduction

Contributor(s): [Jing Ru Ng](#)

The purpose of this document is to provide the analysis and comparison of the technology stack to choose on a final development technology stack that will be used in the project. This document is designed to demonstrate the research on the strengths and weaknesses of the alternative technology from both the frontend and backend side. As our project is a database management application, this document would be focusing more on the database technology side as compared to the frontend and backend side.

## 2. Frontend

### 2.1 Introduction

Contributor(s): [Jing Ru Ng](#)

Frontend web development is the process of creating a website's graphical user interface through the use of HTML, CSS and JavaScript, so that people can view and interact with it including the text, colours, images, table and buttons. Frontend development combines three programming languages, JavaScript frameworks, and design expertise to produce a website adapted to a certain set of requirements. In the below section, we will be comparing the different technologies and frameworks that are used in the industry and choosing the most appropriate technology for the team.

### 2.2 Frontend Technology

Contributor(s): [James Siew Huei Wong](#)

#### 2.2.1 Laravel Blade

Blade is a templating engine that is included with Laravel upon installation. The advantage of using Laravel's own blade template engine is that it allows vanilla PHP code along with Blade Directives like if statements, loop statements and switch statements (Laravel, n.d.).

Since Laravel uses a model view controller (MVC) architectural pattern, Blade allows the display of data as the backend (controller) can just pass the data taken from the database, and show it in the frontend (view), along with the html and css code.

#### 2.2.2 ReactJS

React is a JavaScript frontend library that is used to build user interfaces. React uses a component based structure. Components are snippets of codes which are reusable and independent. It is somewhat similar to a function, but it returns html instead.

Since React uses the components based structure, it allows the passing of rich data through a XML like syntax called JSX and keeps the state in each controller, keeping it out of the DOM (ReactJS, n.d.)

### 2.2.3 Angular

Developed by Google, Angular is a frontend framework rather than a library of JavaScript. It is often used to build progressive web applications and single page applications, which provides fast and universal compatibility with almost all devices. It has its own directives and components as well. (Angular, n.d.)

Angular is currently being used by multiple companies such as Microsoft, Google and Paypal as their mainstream pipeline (Trio 2020).

### 2.2.4 Comparison

Comparing Laravel Blade and Angular, as they are both a framework, both of them support the MVC architectural pattern, and have their own directives and components. On the other hand, React is a JavaScript library and only focuses on the frontend portion (View in the Model View Controller architectural pattern).

Having its own directives helps to render the data taken from the backend easily, as the user can just call the built-in directives such as “@if” for Laravel Blade and “NgIf” for Angular to use the if condition, and so on.

For React, it uses a component based structure, which is a snippet of code which acts like a function, and can be reused, but it returns html instead. Because of that, it allows the passing of data through JSX, a XML-like syntax and can be rendered in the frontend.

As Laravel Blade and Angular are both a framework, learning it is often difficult as it requires a lot more understanding of the structure of the framework and how it works, as compared to a JavaScript library like React. Not just that, once the user learns React, they can apply React anywhere, be it web app or mobile, and it can even be used between platforms (cross-platform), but Laravel Blade and Angular can only be used on themselves. Hence making React a lot more flexible as compared to the other two in the list.

### 2.2.5 Conclusion

Ultimately, the team chooses Laravel Blade as the frontend technology to use, as our backend also focuses on Laravel instead. Another deciding factor for using Laravel Blade is due to us having two members familiar with Laravel, hence the learning curve for the other members in the team would be slightly easier as compared to a complete beginner.

React and Angular are both not bad of a choice to consider as well, since it is easy to learn for React, and learning a new technology (framework) such as a famous framework like Angular would be nice, but having the risks to learn a brand new technology would be challenging for everyone, so we decided to stay on the easier path to minimize risks in our Capstone Computing Project 2 later in the year.

## 2.3 CSS Framework

Contributor(s): [Jing Ru Ng](#)

### 2.3.1 Bootstrap

Bootstrap is a free and open-source CSS framework for front-end web development that focuses on responsive design and mobile-first. It includes design templates for typography, forms, buttons, navigation, and other interface elements that are based on HTML, CSS, and (optionally) JavaScript (GeeksforGeeks 2022).

Bootstrap provides basic style definitions for all HTML components once it is applied to a project. As a result, texts, tables, and form components look the same across all online browsers. Developers can also use the CSS classes supplied by Bootstrap to further alter the appearance of their contents. According to Bose from BrowserStack (2021), developers don't have to create different projects to customize a site for smaller screen widths when using Bootstrap. They only need to include the required Bootstrap classes, and the design will adjust on its own. It is super easy to learn and maintain consistency across multiple devices and browsers.

### 2.3.2 Tailwind

Tailwind CSS is a utility-first CSS framework that allows users to create custom user interfaces rapidly and efficiently. It is an inline style used to create a beautiful interface without having to write your own CSS code. Tailwind CSS offers customizability and flexibility to transform the appearance and feel of the elements. It is a highly configurable, low-level CSS framework that offers users all the building blocks needed to create personalized design interfaces for different websites (Nair 2021).

Tailwind does not need names for CSS classes and Id's. CSS is a global language, which means that if users modify one attribute in the file, it affects all HTML files that are connected to it. However, users may use utility classes and perform local adjustments with the aid of Tailwind CSS.

### 2.3.3 Comparison

As mentioned by Dubey from GeeksforGeeks (2021), the significant difference between Bootstrap and Tailwind CSS is that Bootstrap comes with a set of pre-styled responsive, mobile-first components that possess a definite UI kit, while Tailwind offers predesigned widgets to build a site from scratch with fast UI development.

Bootstrap websites are famed for their responsiveness and immaculate design, however sites created using Bootstrap follow the generic pattern that makes them look identical. Tailwind CSS uses a set of utility classes to create a neat UI with more flexibility and uniqueness, allowing for more customization on the websites.

Bootstrap is a time-saving framework that saves a lot of time. It has been around for over nine years and, as the most popular CSS Framework, it has a large developer community, forums, tools, and so on. The fact that Bootstrap has a

basic file structure is one of the reasons why it is so popular among web developers and designers. Its files are assembled for simple access, and changing them only requires a basic understanding of HTML, CSS, and JS. Tailwind CSS is newer, and there is still a lot of opportunity for it to expand in terms of its community; nonetheless, it is growing every day, and the number of users, references, tools, and websites associated with it is growing as well (Dubey 2021).

#### 2.3.4 Conclusion

The team has decided to go for Bootstrap since it is a simple framework to learn and due to its popularity, there are numerous tutorials and online forums to assist learners in getting started compared to Tailwind CSS. Although Tailwind CSS has the flexibility of being able to customize and create unique interfaces is great, it is not the primary goal of this project.

### 3. Backend

#### 3.1 Backend Technology

Contributor(s): [Kyu Seok yeum](#)

##### 3.1.1 Introduction

Backend is a part of the system which handles the API calls and the logics to communicate with the database with the frontend. All the logic that will be carried out by the server is implemented as the APIs in the backend application. The CRUD operation, data formatting, various types of I/O, and business processes are implemented through the backend. There are a variety of backend frameworks that the programmers can choose with different features provided by the framework. Each framework varies in the programming language and its paradigm, which will be analyzed and evaluated on each framework in this section.

##### 3.1.2 Strapi

Strapi is an open source headless content management system for front end applications. Strapi helps front end applications accessible to the content via an API without knowledge of implementing a backend application. The system automatically generates APIs based on content models, which makes it easier to expose the data stored in the CMS. This quick and easy way of implementing API helps in saving huge amounts of development time and resources. (Strapi, n.d.)

Advantages :

- Fast and easy to implement API
- Provides flexible and scalable modular architecture

Disadvantages :

- Migrating Existing APIs is difficult
- Frequent updates leads to frequent maintenance by developer
- Strapi is not fully open source

### 3.1.3 Laravel

Laravel is an open source full stack web framework based on PHP which uses MVC architectural pattern based on Symfony. Laravel provides a modular packaging system with its dedicated dependency manager. It helps with building a modern, fast, and scalable product. Laravel also has its own ORM called Eloquent which is used to handle the database efficiently. (Laravel, n.d.)

Advantages :

- Has secure built in access control system
- Provides robust mechanism for handling exceptions and bugs
- Ease of unit testing within the code
- Data migration is simple and easy

Disadvantages :

- The framework is easy to learn, but difficult to master
- Updates are frequent, previous products developed with old versions gets buggy
- May face difficulties with updates

### 3.1.4 Express

Express is the most popular open source web application framework that uses Node.js as a runtime. Express.js provides a robust set of features for web applications. Express emphasizes on high performance, with HTTP helpers like redirection, caching, etc. as Node.js runs on single thread, event based loop, express has advantage on async I/O process for file and network. (ExpressJS, n.d.)

Advantages :

- High performance on I/O using async single thread
- High performance in processing network using event driven loop
- Maximize the efficiency by reducing CPU waiting time
- Light and easy to implement compare to other frameworks

Disadvantages :

- Performs in single threaded model which reduces performance on heavy calculation of process
- Poor readability of code on iteration of event callbacks
- Need care on errors as process kills when an error occurs

### 3.1.5 Firebase

Firebase is a platform supported by Google for mobile and web applications. By using firebase, developers do not have to implement a backend server to carry out the database related API. Firebase provides features like Authentication, Firestore, Realtime database, and Storage. Firestore is a good choice for small mobile and web applications which does not require complicated logic. (Firebase, n.d.)

Advantages :

- Quick and easy to implement, saving development resources.



- Provides secure authentication

Disadvantages :

- Not a open source
- Do not support microservices integration
- Restricted amount of storage

### 3.1.6 Spring

Spring is the largest java based framework that uses POJO format of object instead of EJB. Spring provides features of Inversion of Control (IoC), Aspect Oriented Programming (AOP), and uses containers to control the life cycle of applications by using XML based components. Spring provides secure and flexible libraries that can be used on web development as well as other electronic devices. (Spring, n.d.)

Advantages :

- Provides good portability as program runs on servlet container
- Ease of testing as the framework uses POJO based object
- Provides AOP which is helpful on logging and controlling error
- Provides powerful OOP features with libraries like JPA and Hibernate

Disadvantages :

- The configuration and the setup tends to be complex than others
- Does not support distributed system due to lightweight container

### 3.1.7 Comparison

The technologies for backend and server can be divided into two categories which are either open source, or paid service. The open source project has the advantage of saving the cost of using a framework or a library in a business project. Laravel, Express.js, Spring lie under this category of open source project. On the other hand, Strapi and Firebase are not fully open source as these stacks can be used for free, there are purchases needed when the application grows and reaches a certain point.

Speed of implementing backend applications will be fastest for Firebase, followed by Strapi, Express.js, Laravel and Spring framework.

As for the knowledge needed for implementing the server, Strapi and Firebase require the least amount of knowledge in many fields such as programming, network system, database systems, etc. Amongst the backend frameworks (Laravel, Express.js, Spring), Express.js and Laravel are easier to learn compared to the Spring framework due to complex configuration and system setup.

The performance between Laravel, Express.js and Spring will differ depending on the purpose and the size of the application. Laravel provides an intuitive architectural structure which is most suitable for small and mid size applications, Express.js performs best on heavy I/O operations, and Spring is suitable for large applications with heavy calculations. All the backend frameworks above provide good maintenance features with fast and secure features.



### 3.1.8 Conclusion

To conclude, Laravel will be used in the project. Laravel is a PHP based open source framework which has a built in access control system to help with easy processing of data access and security access control. Laravel also provides easy exception handling, stable security management, Ease of automated testing, separation of business logic from presentation logic, simplified URL routing configuration to help with carrying out the project.

## 3.2 Database

Contributor(s): [Chian Hui Lee](#)

### 3.2.1 Introduction

A database is a collection of related data. A database management system provides an efficient and easy way for the user to store, retrieve and manage the data. There are two types of databases, which are the relational database and the non-relational database. The main difference between the two databases is that relational databases store data in tables while non-relational databases store data in key-value format. In this section, we will discuss the details of the different types of databases and make comparisons between them.

### 3.2.2 SQLite

SQLite is a C-language library that implements a speedy, efficient, high-reliability, fully-featured SQL database engine. According to the official website of SQLite, it is the world's most popular database engine. Its source code is publicly accessible. SQLite is easy to start with since it does not require any setup or administrative action. It is serverless because it does not require a separate server process or system to function. By using SQLite, the user is allowed to work on multiple databases in the same session at the same time, thus providing a flexible environment. It is also a cross-platform database management system capable of running on all platforms, including macOS and Windows (SQLite, n.d.).

SQLite is used to create embedded software for devices such as televisions, cell phones, cameras, and so on. It is capable of handling HTTP requests with low to medium traffic. SQLite has the ability to compress files into smaller archives with less metadata. SQLite is a temporary dataset that is used to process data within an application. SQLite is used by beginners for learning and training because it does not require installation or configuration (Simplilearn 2021).

### 3.2.3 MySQL

MySQL is a well-known relational database management system that has users all over the world. It uses a client-server architecture. MySQL is written in the C and C++ programming languages. MySQL creates a database to store and handle data, with the relationship of each table established. The client can send requests to MySQL by inputting specified SQL statements. The server programme will return the required information and show it on the client's side (Arno et al. 2002).

MySQL software provides a SQL database server that is fast, multi-threaded, multi-user, and resilient. MySQL server is intended for mission-critical applications and heavy-load production systems. MySQL is a relational database that works by storing data in separate tables instead of putting the data in one big table. This architecture adds speed and flexibility. MySQL has tested no memory leaks and can work on many different platforms. It is fully multi-threaded using kernel threads, which means it can easily use multiple CPUs if the CPUs are available. It has very fast B-tree disk tables with index compression. MySQL uses an optimized one-sweep multi-join to achieve fast joins. Moreover, MySQL implements a secure and flexible password system and allows host-based verification. The passwords are secure because all password traffic is encrypted when the client connects to a server. MySQL is the solution for mission-critical, heavy-load database systems.

### 3.2.4 PostgreSQL

PostgreSQL is one of the most advanced open sources of object-relational database systems available. Support for user-defined objects and associated behaviors, such as data types, functions, operators, domains, and indexes, is a core feature of an object-relational database. As a result, PostgreSQL is incredibly adaptable and robust. Complex data structures, for example, can be built, stored, and retrieved.

PostgreSQL provides several powerful features. The database management system offers automatically updatable views. This means that a view resulting from a query will automatically update when a transaction updates the data that serves the view. Other than that, PostgreSQL is extensible. It can support numerous data types. Among them are geometric/GIS, network address types, JSONB, native UUID, and time zone aware timestamps. PostgreSQL provides data integrity by introducing constraints and regulating data added by the user. With PostgreSQL, the user can ignore invalid and orphan records. The performance of PostgreSQL is also reliable. It provides parallelization of reading queries, powerful indexing methods and multi-version concurrency control (Borozenets 2021).

### 3.2.5 MongoDB

MongoDB is a database management system that allows for the rapid development of online applications and internet infrastructure. The data model and persistence mechanisms are designed for high read/write throughput and easy scaling with automatic failover. MongoDB may give unexpectedly good performance whether an application wants just one database node or hundreds of them. If you've had trouble growing relational databases, this could be wonderful news. The most compelling reason to choose MongoDB is its simple data model. MongoDB stores data in documents rather than rows (Verch et al. 2016).

The document format used by MongoDB is based on JSON, a common technique for storing arbitrary data structures. A data model based on documents can describe complex, hierarchical data systems. It is frequently possible to avoid the many joins that are prevalent in relational databases (MongoDB, n.d.).

Document databases are extremely adaptable, allowing for variations in document structure and the storage of partially completed documents. Multiple documents can be embedded in a single document. Fields in a document function similarly to columns in a SQL database, and they, like columns, can be indexed to improve search performance. MongoDB was founded on a scale-out design, which allows many small machines to collaborate to develop fast systems and process massive volumes of data. MongoDB has always prioritized providing developers with a fantastic user experience, which, together with all of its other benefits, has made MongoDB a preference among developers worldwide for a wide range of applications.

### **3.2.6 Firebase - Cloud Firestore**

Cloud Firestore is a versatile, scalable database from Firebase and Google Cloud for mobile, web, and server development. It, like Firebase Realtime Database, keeps your data in-sync across client apps via real-time listeners and provides offline support for mobile and web, allowing you to create responsive apps that run regardless of network latency or Internet connectivity. Cloud Firestore also integrates seamlessly with other Firebase and Google Cloud technologies, such as Cloud Functions (Firebase, n.d.).

Flexible, hierarchical data structures are supported by the Cloud Firestore data model. Data of the users will be stored in documents, and be organized into collections. Documents, in addition to subcollections, can include complicated nested objects. Queries in Cloud Firestore can be used to obtain individual, particular documents or all documents in a collection that match the user's query criteria.

Filtering and sorting can be combined in the query by using numerous, chained filters. The queries are indexed by default, so query performance is related to the size of the result set rather than the size of the data collection. Cloud Firestore, like Realtime Database, uses data synchronization to update data on any connected device. However, it is also intended to efficiently perform simple, one-time fetch queries. Cloud Firestore caches data that the app is actively consuming, allowing the programme to write, read, listen to, and query data even when the device is not connected to the internet. When the device is reconnected, Cloud Firestore synchronizes any local changes to Cloud Firestore. Cloud Firestore leverages the best of Google Cloud's robust infrastructure, including automatic multi-region data replication, high consistency guarantees, atomic batch operations, and real-time transaction support. Cloud Firestore is built to manage the most demanding database workloads from the world's most popular apps (Firebase, n.d.).

### **3.2.7 Comparison**

Relational databases are called SQL databases because their tables can be joined together. SQL can be used to manage relational databases. It helps to achieve complex querying, provides flexibility, and helps to analyze data. The relational database cannot be categorized further.

Meanwhile, the non-relational databases are known as NoSQL databases.

The joint concept does not apply to the non-relational databases. It stores data in key-value format. There are various database types, including document databases, column databases, key-value stored databases, cache databases and graph databases. Document databases are used to store dynamic data. The data will be stored in JavaScript Object Notation (JSON) format (Lithmee 2018).

The main difference between relational databases and non-relational databases is that they use different schemas to store the data. The schema of the relational database is defined, and it describes the structure and types of the structure and records. The schema of the non-relational database is not defined. Users will be able to store the data using any structure they would like to use. Other than that, the ways of accessing data in relational databases and NoSQL databases vary. SQL data can be exported as rows and tables, while NoSQL queries provide a way for users to access data as documents. The data can be viewed from multiple different perspectives (David 2021).

One of the primary distinctions between relational databases and NoSQL systems is that, whereas relational databases normally support only a limited number of transactions, NoSQL systems allow transactions to occur on any row and transaction records. While this may appear to be a minor distinction, it makes a huge difference (Ostezer and Drake 2014).

### **3.2.8 Conclusion**

In conclusion, we will be using MongoDB, a non-relational database as our database. We choose a non-relational database over a relational database because we have more flexibility and the ability to customize our database. Non-relational databases support transactions to occur on any row and transaction records. This provides much more freedom for us to make changes to the database, which improves our availability of customization and effectiveness of data editing. All in all, MongoDB is closer to the needs of our database management system project.

## **3.3 Storage & Deployment**

*Contributor(s):* [James Siew Huei Wong](#)

As this project is requested by Curtin University's FOES faculty, and since the nature of the data is sensitive and the breach of confidentiality of the data would be detrimental, it would be hosted and handled locally by Curtin University (Approval from the client as well).

## 4. References

Contributor(s): [James Siew Huei Wong](#)

Angular. n.d. "The modern web developer's platform." Angular. Accessed April 9, 2022.

<https://angular.io/>.

Arno, Kaj, MySQL AB, David Axmark, and Michael Widenius. 2002. *MySQL Reference*

*Manual: Documentation from the Source*. N.p.: O'Reilly Media, Incorporated.

[https://books.google.com.my/books?hl=en-EN&lr=&id=9c-pkLaNmqoC&oi=fnd&pg=PR5&dq=mysql&ots=G2pCgUR7n9&sig=JzHOuGnUDNM81PLTYtXmKzio4KA&redir\\_esc=y#v=onepage&q=mysql&f=false](https://books.google.com.my/books?hl=en-EN&lr=&id=9c-pkLaNmqoC&oi=fnd&pg=PR5&dq=mysql&ots=G2pCgUR7n9&sig=JzHOuGnUDNM81PLTYtXmKzio4KA&redir_esc=y#v=onepage&q=mysql&f=false).

Borozenets, Michael. 2021. "Why use PostgreSQL as a Database for my Next Project in

2021." Fulcrum Rocks. <https://fulcrum.rocks/blog/why-use-postgresql-database/>.

Bose, Shreya. 2021. "Top 5 CSS Frameworks for Developers and Designers."

BrowserStack. <https://www.browserstack.com/guide/top-css-frameworks>.

David, Matthew. 2021. "What Is the Difference Between NoSQL and Relational Database Systems?" Simplilearn.

<https://www.simplilearn.com/difference-between-nosql-and-relational-database-systems-article>.

Dubey, Sarvesh. 2021. "Tailwind CSS vs Bootstrap." GeeksforGeeks.

<https://www.geeksforgeeks.org/tailwind-css-vs-bootstrap/>.

Ermigiotti, Lorenzo. 2021. "17 JavaScript Frameworks that You Should Know in 2021 - A Comprehensive Guide." Codemotion.

<https://www.codemotion.com/magazine/frontend/javascript/javascript-frameworks-guide/#9-bootstrap>.

ExpressJS. n.d. "Overriding the Express API." Express.js. Accessed April 12, 2022.

<https://expressjs.com/en/guide/overriding-express-api.html>.

Firebase. n.d. "Firebase Documentation." Firebase. Accessed April 12, 2022.

<https://firebase.google.com/docs>.

GeeksforGeeks. 2022. "Bootstrap." GeeksforGeeks.

<https://www.geeksforgeeks.org/bootstrap/?ref=lbp>.

Laravel. n.d. "The PHP Framework For Web Artisans." Laravel. Accessed April 12, 2022.

<https://laravel.com/docs/9.x>.

Lithmee. 2018. "What is the Difference Between Relational and Nonrelational Database."

Pediaa.

<https://pediaa.com/what-is-the-difference-between-relational-and-nonrelational-database/>.

MongoDB. n.d. "Why Use MongoDB And When To Use It?" MongoDB. Accessed April 8,

2022. <https://www.mongodb.com/why-use-mongodb>.

Mourya, Naincy. 2021. "Tailwind CSS vs. Bootstrap: Which Is a Better Framework?"

MakeUseOf.

<https://www.makeuseof.com/tailwind-css-vs-bootstrap-which-is-a-better-framework/>

.

Nair, Hari S. 2021. "Introduction to Tailwind CSS." GeeksforGeeks.

<https://www.geeksforgeeks.org/introduction-to-tailwind-css/>.

Ostezer, and Mark Drake. 2014. "SQLite vs MySQL vs PostgreSQL: A Comparison Of

Relational Database Management Systems." DigitalOcean.

<https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>.

ReactJS. n.d. React – A JavaScript library for building user interfaces. Accessed April 9,

2022. <https://reactjs.org/>.

Simplilearn. 2021. "What is SQLite And When to Use SQLite." Simplilearn.

<https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite>.

Spring. n.d. "Guides." Spring. Accessed April 12, 2022. <https://spring.io/guides>.



SQLite. n.d. "Features Of SQLite." SQLite. Accessed April 8, 2022.

<https://sqlite.org/features.html>.

Strapi. n.d. "Content Architecture: the backbone of your digital projects." Strapi. Accessed April 12, 2022. <https://strapi.io/content-architecture>.

Trio. 2020. "Who Uses Angular in 2022? 12 Global Websites Built With Angular." Trio Dev. <https://trio.dev/blog/companies-use-angular>.

Verch, Shaun, Douglas Garrett, Peter Bakkum, and Kyle Banker. 2016. *MongoDB in Action: Covers MongoDB Version 3.0*. N.p.: Manning.

<https://livebook.manning.com/book/mongodb-in-action-second-edition/chapter-1/>.

W3Schools. n.d. "React Components." W3Schools. Accessed April 9, 2022.

[https://www.w3schools.com/react/react\\_components.asp](https://www.w3schools.com/react/react_components.asp).