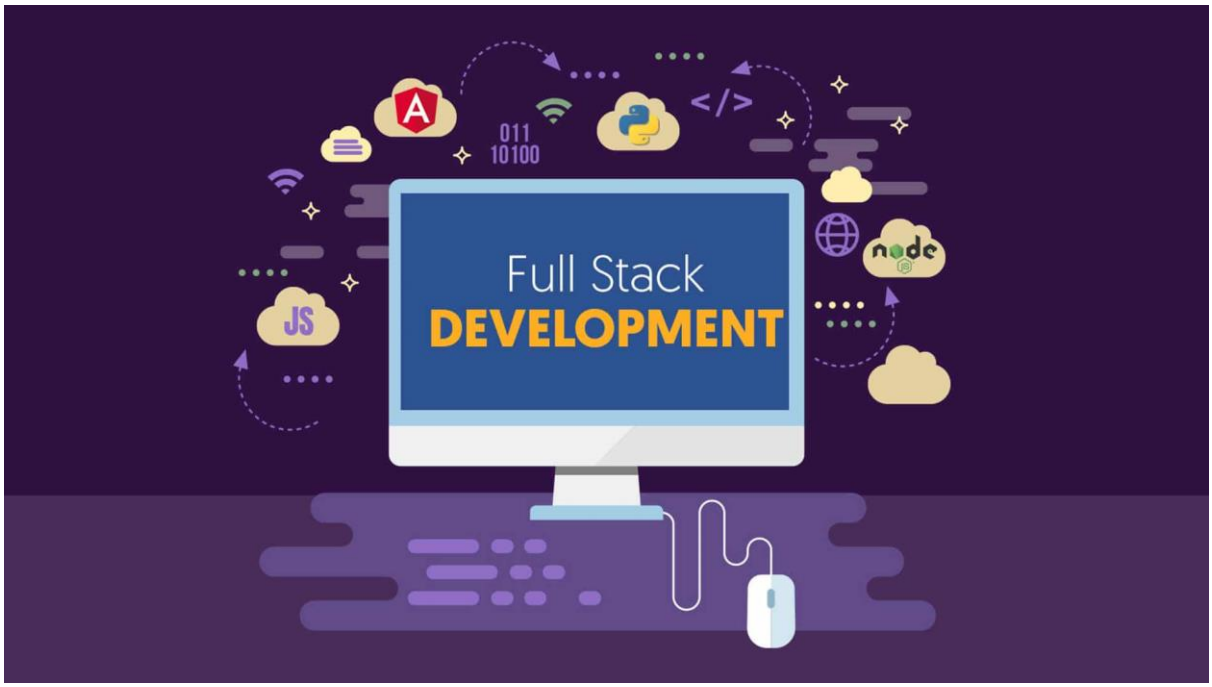
Full Stack Research



Carlo van Kessel
S-DB-IP3 and S-DB-GP3
S3-DB03
Hans van Heumen, Marc van Grootel
31-10-2022
V5

Version control

Version	Changes
1	Initial version.
2	Added front image. Added Frontend research.
3	Added PWA research.
4	Added Database research.
5	Added Backend research.

Contents

1. Frontend

In this research I want to learn what framework I want to use for my frontend.

For the frontend you have many frameworks the most popular are these three:

1. Angular
2. Vue
3. React

1.1 What do these frameworks have in common?

All these frameworks are based on JavaScript. They just render data, but with a framework It will be more structured. They all have pros and cons and this I will be explaining:

1.2 Different frameworks:

1.2.1 Angular

Angular is made by Google and was first released in 2010 and is the oldest of these 3. It is a TypeScript-based framework. It has got a second release in 2016 where they made Angular 2. Angular has a lot of pros such as: it uses TypeScript which means it will spot common mistakes when typing code. While the smaller JS frameworks do not have this feature and requires developers to check code more often. But Angular also has it cons such as: it is complex and verbose. This means it is not easy to learn and start on it. The learning curve is way bigger than other JS frameworks.

1.2.2 Vue

Vue is an up-and-coming framework and is the youngest of the 3. Vue is created by an ex-Google employee and is the only one of these that does not have a large company backing it. Vue is mostly used as an MVC, but it can be used as an CBA like React. A pro of Vue is that it is small and has virtual DOM rendering. Vue's learning curve is a lot smaller then Angular and React. For Vue you can start easy with just the knowledge of HTML, CSS, and JavaScript. There are also some cons such as a language barrier where a lot of forms about Vue are in Chinese. Or that Vue is still a lot smaller project then React or Angular and does not have that large company backing them.

1.2.3 React

React is made by Facebook now known as Meta and released in 2013. Most of Meta products like Facebook, Instagram and WhatsApp use React. React has a good supply of documentation and is easier to learn than Angular. React is very popular and therefore you have a lot of tools and support that u can use in your React project. A con of React is that is changing quick, it is constantly evolving. This means that de documentation is sometimes a bit lacking for new parts of React. As I mentioned React is changing quick and developers are kind of exhausted by the relearning process.

1.3 Conclusion:

In conclusion all these 3 frameworks have their pros and cons. For me Vue is attracting me because it is a smaller and rising framework. I can make a PWA with it and it is fast and tiny. The learning curve for Vue is smaller than for React or Angular. I have worked with HTML, CSS, and JavaScript so I would like to know how a JavaScript framework works.

1.4 Sources:

Angular:
https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/

Vue:
https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/

React:
https://massivepixel.io/blog/react-advantages-disadvantages/
https://www.javatpoint.com/pros-and-cons-of-react

2. Backend

For my backend I want to learn something new. Last semester I used C# with entity framework, and this was interesting to learn but I want to see something different. So, for this semester I want to learn JAVA. I know you have some frameworks like Spring Boot and Quarkus.

2.1 Different frameworks:

2.1.1 Quarkus

Quarkus is a JAVA framework with a fast boot time. The framework is mostly developed for serverless or cloud-native environments. Quarkus has a fast hot reload as it can see changes made to files and resources. The documentation is also good in comparison to different frameworks, but in the forms, there will not always be an answer to your question.

2.1.2 Spring Boot

Spring boot is an open-source JAVA framework. Spring Boot has an amazing community support and documentation. There are a lot of production-ready integrations for Spring. Spring is developed to make microservices and I mostly used for enterprise applications. The development of Spring is easy, and it helps you with a lot. But Spring also has a longer startup time and uses more memory.

2.2 Conclusion:

Both Spring and Quarkus have their pros and cons, but since we got a presentation in class of Quarkus I was mostly attracted to Quarkus. Quarkus is more innovative but has less community support. But because of the presentation and its better startup time I have made the decision to start with Quarkus and learn JAVA for the first time.

2.3 Sources:

https://litslink.com/blog/slug-quarkus-vs-spring-boot-the-difference
https://www.baeldung.com/spring-boot-vs-quarkus
https://rollbar.com/blog/quarkus-vs-spring-boot/#

3. PWA

3.1 What is a PWA?

Progressive Web App is a website which can be used as an app. If you are going to make an interactive website where you can do things like, make an account, and add things or delete things. PWA is interesting, you can implement PWA easy and therefore you can make users download your 'website'. What this means is the laptop or phone sees that the website is compatible as an app, and you can download it and use it as an app instead of a website. The biggest advantage is that you don't have to download the app from the app store or play store on your phone

3.2 What makes a PWA?

A PWA needs to have a service worker running and it needs to have a manifest. In this manifest there are some simple information about your website. You can simply make a PWA with just an index.html, manifest.json, logo and service-worker.js files.

3.3 For help with Vue:

https://hnpwa.com/

3.4 Sources:

https://www.youtube.com/watch?v=sFsRylCQblw
https://web.dev/learn/pwa/

4. Database

4.1 What is a database?

A database is a place where data is stored. It can be viewed as a box where you can put something in and out. You can get, update, delete and put data in a database. In code language this is called CRUD (create, read, update, and delete). You can only access a database if you have the right credentials therefore not everyone can manipulate a database. Easily explained it is like a vault which only you and the application can make changes to.

4.2 Different databases

In the database world there are a few options you can choose from: SQL, NoSQL and many more. All the databases have the same meaning: store data. But how you store it is different:

4.2.1 NoSQL

NoSQL is a form of storing data without relational tables. What this means it stores data, but the data does not have a relation to different data. Some of the benefits of NoSQL are flexible data models, fast queries, and horizontal scaling. What flexible data models means is that you can easily make changes to the database which gives the user and the developer a better experience. Fast queries means that in SQL queries tend to get big if you have multiple tables that needs to join. In NoSQL you store data that is accessed together which means that the queries are typically smaller.

4.2.2 SQL

SQL is a form of storing data in relational tables. This means it stores data in a boxed way, so tables have relations between each other. This can be easier to use if the data needs to be boxed and is kind of fixed. But this means it is a not easy to update and to scale. In SQL you have a few management systems like MySQL and PostgreSQL

4.2.3 MySQL

MySQL is one of the most famous management systems. The data is stored in tables which is easy to perform CRUD operations on. It is reliable and can be used for small and big applications. But it has some downsides because it is a little older it can only support standard data types and is more used for simple applications with write and reading.

4.2.4 PostgreSQL

PostgreSQL is a newer management system. PostgreSQL is a powerful and open-source system. It provides good performance with low maintenance. Because it is newer it has more features, and it provides advanced security features.

4.3 Conclusion:

For my own project I want to choose a SQL management system. This is because I have structured data and it has relations to one and another. For the management system MySQL and PostgreSQL both have good sides and bad sides, but I think because PostgreSQL is newer and has more functionalities, I am going to choose PostgreSQL.

4.4 Sources:

https://www.mongodb.com/nosql-explained/nosql-vs-sql#what-are-the-benefits-of-nosql-databases
https://www.geeksforgeeks.org/difference-between-mysql-and-postgresql/