



Received: 15.09.2023
Accepted for printing: 15.12.23
Published: 29.12.2023
License: CC BY-SA 4.0

DOI: 10.15584/jetacomps.2023.4.21

PAWEŁ DYMORA ¹, MIROSLAW MAZUREK ²,
MARIUSZ NYCZ ³

Comparison of Angular, React, and Vue Technologies in the Process of Creating Web Applications on the User Interface Side

¹ ORCID: 0000-0002-4473-823X, Ph.D. Eng., Rzeszów University of Technology, Faculty of Electrical, and Computer Engineering, Department of Complex Systems, Poland

² ORCID: 0000-0002-4366-1701, Ph.D. Eng., Rzeszów University of Technology, Faculty of Electrical, and Computer Engineering, Department of Complex Systems, Poland

³ ORCID: 0000-0002-6297-5730, Ph.D. Eng., Rzeszów University of Technology, Faculty of Electrical, and Computer Engineering, Department of Complex Systems, Poland

Abstract

This paper presents a comparative analysis of three programming technologies and their application in building a web application, taking into account the code on the server-side and the user interface side. Three equivalent applications Angular, Vue, and React were analyzed. The needs for web applications are described, the implementation of particular functionalities in all technologies is presented together with the programming of database functionalities. The article is aimed at helping students of computer science faculties to make an informed choice of a framework or library for a specific project, taking into account the differences between these popular solutions.

Keywords: WEB applications, design process

Introduction

A WEB application is an interactive program that runs on a server accessible from any browser. To use a web application you don't need to install anything, but you do need a browser and access to the Internet. Continuous development of web technologies has made the applications less and less scalable, therefore developers found it useful to separate the front-end and back-end of the application (Molina-Ríos, Pedreira-Souto, 2020; Technical Documentation in Software Development: Types and Best Practices, 2; Kaluža, Vukelić, 2018).

The first part is responsible for what the user sees, it makes up the visible interface, while the second part is responsible for the layer that is invisible to the user, it is responsible for the operation of the site from the “inside”. The server part communicates with the interface part and is responsible for the functionalities. With each user action, the browser sends a request to the server which returns information in the form of client layer code. In turn, the browser interprets the code and displays it in a user-friendly form (Kaluža, Vukelić, 2018; Current browser market share, 2021; Dymora, Paszkiewicz, 2020).

The main tools without which the compilation of script code into an actual page would not succeed are browser engines. They are used to render web pages, mainly used by web browsers. It is software that is responsible for processing the content of web pages and their formatting elements and then rendering them in a user-friendly manner. On the user interface side, there are three main types of technologies: HTML, CSS, JavaScript. On the server-side of the application, the database system can be separated – this is the place where data is stored. The programmer defines their structure and describes them. The most popular databases are MySQL, PostgreSQL, Firebird (Current browser market share, 2021; Dymora, Paszkiewicz, 2020; Dymora, Mazurek, Mroczka, 2021; <https://www.postgresql.org/>; <https://firebirdsql.org/>). The programming language used is important. The server-side programmer has a choice of many programming languages, and the most popular are Python, Java, C#. In Java, servlets can be distinguished as a basic element of business applications, they are used both in Java EE applications and those developed in Spring MVC. The most complex applications nowadays also need highly developed tools, so the article focuses on comparing the three most popular solutions: Angular, Vue, React (Dymora, Paszkiewicz, 2020; Flanagan, 2021; <https://www.javascript.com/>; Freeman, 2018).

This paper presents a simplified methodology and description of the application development process. The presented approach shows in detail what stages the entire web application development process consists of and what influences its efficiency. The presented cycle of creation of a new IT product allows one to properly understand the project process, as the knowledge gained is crucial from the business point of view of the work on a given IT project. In particular, novice developers as well as IT students can compare the essential properties of popular web application programming technologies and use the corresponding one in the process of web application development. This is achievable through a rather in-depth comparison of Angular, React, and Vue technologies in the process of creating web applications on the user interface side.

The paper is divided into five sections. The introduction provides a review of the literature and recent trends in technologies used in the development of

user interfaces in web-based applications. Chapter two characterizes the test environment and methods selected for the study. Chapter three presents selected research technologies used and their main characteristics. Chapter four presents a description of the experiments and a comparative analysis of the results obtained. Summary, conclusion, and scope of future research are presented in Chapter five.

Test application design

In order to make an objective comparison of the discussed technologies, a web application “Car Seller” was designed. The main function of the application is to enable the user to buy a car in an online car shop. Each user, to use the store, has to register. After registration, the user can choose from over 1700 cars, a model that interests him. The data about the cars, such as price and condition, are selected according to a proprietary algorithm based data on from an open API. The user can filter the cars by 3 features. After selecting cars or a car, the user is able to add it to his/her cart and then purchase it. The data from the purchase is stored in the Firebase database. The structure of the application is shown in Figure 1.

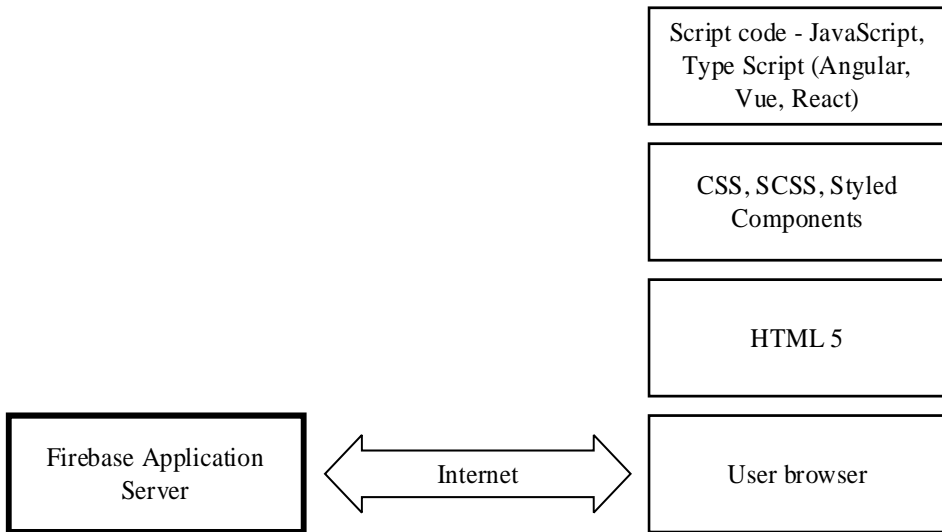


Figure 1. Structure of the application “Car Seller”

Description of selected technologies

Three applications identical in terms of functionality were prepared for testing using different technologies and programming environments. The detailed implementation is shown in Table 1.

Table 1. Summary of implementation details of developed applications and used technologies

	Application #1	Application #2	Application #3
Script Language	TypeScript	JavaScript	JavaScript
Technology	Angular	React	Vue
Page Styles	Pre-processor SCSS	Styled Components Library	CSS
Used environment	PHP Storm	Visual Studio Code	Visual Studio Code
Hypertext Markup Language	HTML 5	HTML 5	HTML 5

JavaScript Language

The JavaScript language is a dynamically typed, high-level programming language. It is most commonly used for web programming where it provides interactivity and event handling. This language has many libraries that make it easier to work with one of the first was the jQuery library. This language has gained enormous popularity, especially among web developers on the user interface side, but this language is also used on the server-side. An environment that helps you run JavaScript code directly on your computer without any browser involvement is Node.js. According to the Tiobe index, the language is ranked as the seventh most popular language (Flanagan, 2021; <https://www.javascript.com/>).

Angular programming technology

Angular is a comprehensive development tool for creating SPA (Single Page Application) applications that are loaded dynamically in the browser. This technology does not need to download the page from the server the user goes through the individual tabs. Angular has been developed by the Google team and community and is an open-source project. Angular as technology provides all the tools to create a SPA, it does not need to download other libraries like HTTP library which other technologies need. Angular is fully written in TypeScript which allows for greater scalability of any project (Freeman, 2018; <https://angular.io/features>).

React programming technology

React is a JavaScript programming language library that is used to create user interfaces for various types of applications and is available to the general public because it is open-source. React was created by one of Facebook employees Jordan Walke. The React library, like Angular, allows you to create interactive SPA-type pages. React by itself does not provide all the tools you need to create SPA-type applications, many additional features need to be downloaded, but when a developer plans to make a smaller application and cares about its smaller size it can be considered a plus. An additional advantage that can be given to React is the documentation in Polish (<https://pl.reactjs.org/>; Chinnathambi, 2019).

Comparative analysis of study results

Depending on the technology, components are written in different ways. In an application written in Angular technology one component is contained in three files: CSS, HTML, TypeScript. When components written in React and Vue technologies are contained in a single file, in React technology these are files with .js extensions, in turn, components written in Vue technology are components with .vue extension (Hanchett i Listwon, 2020). In each application one file is responsible for configuration settings, in each of them it is a package.json file. It contains, among others, information on how to run the application, the version number of the application, or installed dependencies. Additionally, in an application written in Angular technology, there is an angular.json file that has information about the configuration of Angular CLI which makes it easier for the developer to develop the application. An overview of the application structures can be seen in Figure 2.

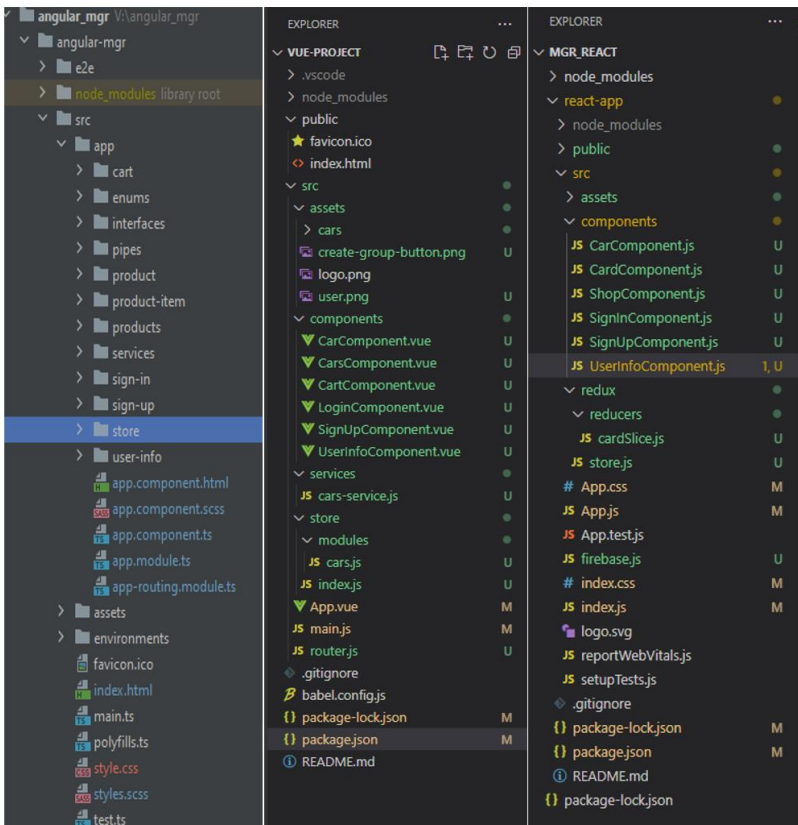


Figure 2. File and directory structure of the application written in three technologies: Angular (left), Vue, React (right)

An identical mechanism for adding cars to the list was used in the three applications. It was implemented differently depending on the application, but the algorithm works the same way. The data to be processed in the application was taken from an open API. They were created in 1997. There are 1728 rows of data in 7 columns in the downloaded data.

The comparison scenario involved listing each component in each application, listing selected parts of the implementation, and then comparing components written in three different technologies and development tools. The application is written in Angular technology used: TypeScript scripting language, SCSS Preprocessor in page styling. It was written in the PHPStorm environment. It used HTML 5 as a hypertext markup language.

The next application written in React technology used JavaScript scripting language, Styled Component library in page styling, Visual Studio Code environment, HTML 5. The last application written used Vue technology and also used JavaScript scripting language, CSS for page styling, and Visual Studio Code and HTML5.

The comparison scenario also included a list of runtimes for each component in the applications, as well as the time period needed to write a given component and the number of lines of code needed to implement a given functionality separately for the code responsible for the hypertext markup language, the code responsible for the scripting language, and the code responsible for view styling. The rendering time studies for a given component will be recalculated thanks to the add-ons: Vue Devtools for an application written in Vue, React Developer Tools for an application written in React, and Angular DevTools for an application written in Angular.

Comparison of login function implementations

The codes responsible for hypertext markup languages differed very little. In React technology, after using the Styled Component library, each component that was styled separately was assigned to JavaScript const variables. Components written in Vue and Angular technologies were tagged with ordinary HTML code tags, which can be seen in Figure 3.

```

<section class="login-box">
  <h1 class="login-box__title">Zaloguj się do platformy</h1>

  <div class="login-box__wrapper">
    <div class="login-box__icon-wrapper">
      
    </div>

    <form class="login-box__form" [formGroup]="userForm">
      <input placeholder="Login" class="login-box__input" type="text" formControlName="login">
      <input placeholder="Hasło" class="login-box__input" type="text" formControlName="password">
      <button class="login-box__button-login" (click)="loginUser()">Zaloguj się</button>
      <span (click)="goToSignUp()" class="login-box__button-create-account">Założ konto</span>
    </form>
  </div>
</section>

```

```

<template>
  <section class="login-box">
    <h1 class="login-box__title">Zaloguj się do platformy</h1>

    <div class="login-box__wrapper">
      <div class="login-box__icon-wrapper">
        
      </div>

      <div class="login-box__form">
        <input v-model="email" placeholder="Login" class="login-box__input" type="text" />

        <input v-model="password" placeholder="Hasło" class="login-box__input" type="text" />
        <button class="login-box__button-login" @click="loginUser()">
          Zaloguj się
        </button>
        <span @click="goToSignUp()" class="login-box__button-create-account">
          Założ konto</span>
      </div>
    </div>
  </section>
</template>

```

```

<div>
  <StyledLoginBox>
    <StyledTitle>Zaloguj się do platformy</StyledTitle>
    <StyledWrapper>
      <StyledIconWrapper>
        <StyledIcon />
      </StyledIconWrapper>
      <StyledForm>
        <StyledInput onChange={event => setLogin(event.target.value)} placeholder="Login"></StyledInput>
        <StyledInput onChange={event => setPassword(event.target.value)} placeholder="Hasło" ></StyledInput>
        <StyledButtonCreateAccount onClick={handleChange}>Zaloguj się</StyledButtonCreateAccount>
        <StyledButtonLogin onClick={handleChangeSignIn}>Założ konto</StyledButtonLogin>
      </StyledForm>
    </StyledWrapper>
  </StyledLoginBox>
</div>

```

Figure 3. HTML code of applications written in Angular, Vue and React

As we can see in Figure 4, most codes appeared in the application written in Vue technology.

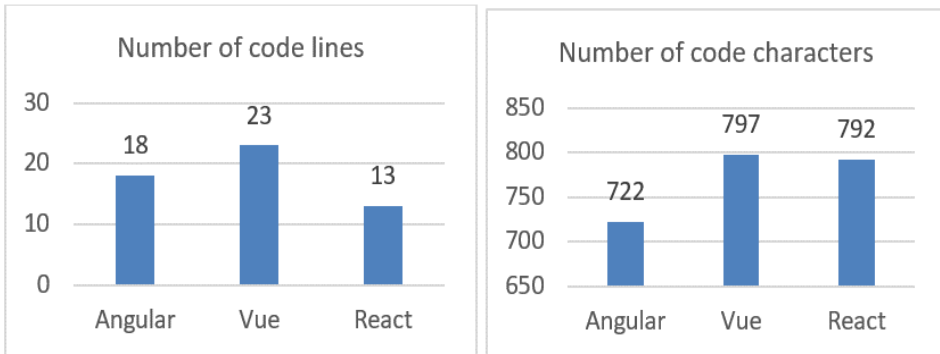


Figure 4. The number of code lines (left) and code characters count (right) responsible for HTML

Cascading Style Sheets Implementation

The technique for styling the view responsible for logging in all three applications differed. In the application written in Vue technology, the styling was contained in .css files. When styling in Angular, the application was supported by SCSS preprocessor. On the other hand, the styling of elements in the React application differed the most from the others. In this case, the styled-components library was used. In this library, each element that was styled had a separate entity and was separately assigned to a variable. Sample style codes are shown in Figure 5.

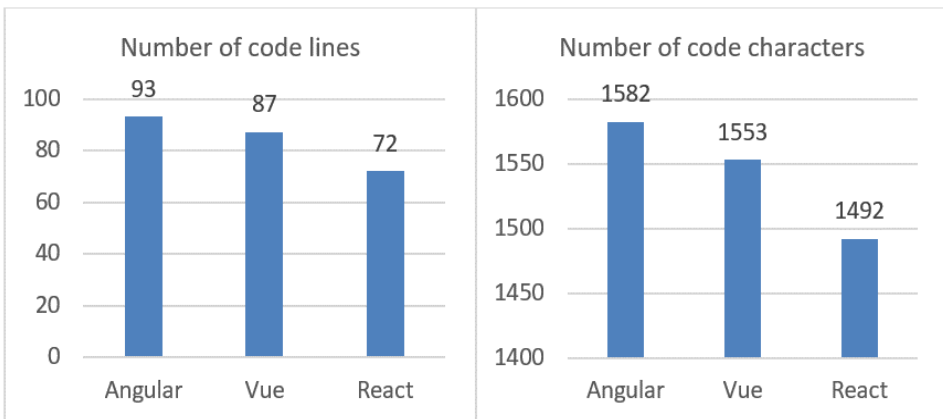


Figure 5. Comparison of the number of code lines (left) and code characters count (right) responsible for styling

Script Language

In the application written in Angular technology, Typescript was the scripting language, while in the Vue and React applications, Javascript was the scripting language. Figure 6 illustrates that most codes appeared in the application written in Angular technology. On the other hand, the script code of the application written in Vue technology had the most lines.

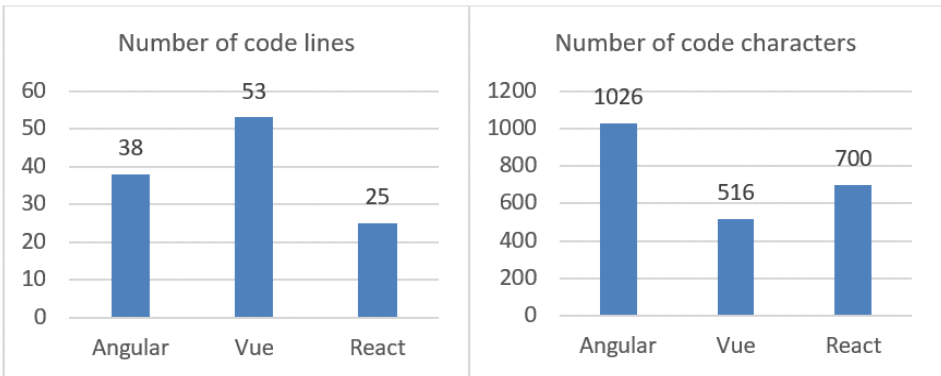


Figure 6. Comparison of the number of code lines (left) and code characters count (right) in the script language

The time required to create a given component

Due to the different types of language, a different amount of time is needed for software development. The calculation of programming time is not reliable, however, it was observed that it did not take long to write each component. The first component written in Angular technology took by far the longest due to the fact that TypeScript language was used. The comparison of workloads is shown in Figure 7.

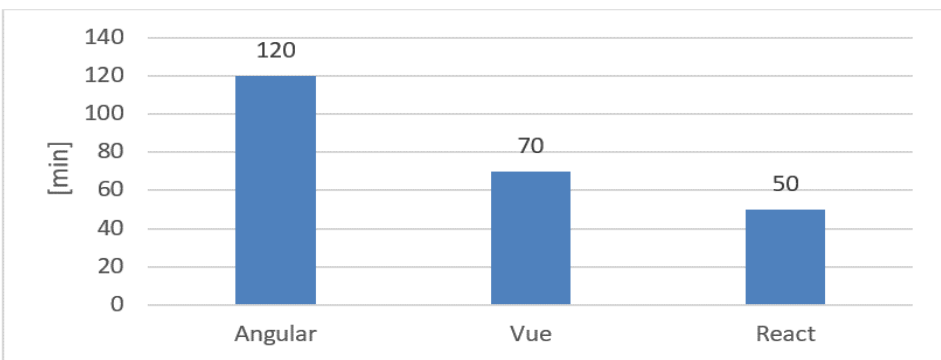


Figure 7. The time required to write the login component in three applications

Comparison of browser login component creation time

The next test analyzed the time it takes to create a component in a browser. Each browser has different rendering times, but the browser used for testing was Chrome, which has the most facilities for a web developer. In Figures 8 and 9, you can see that the component created in Vue technology took the longest time to create.

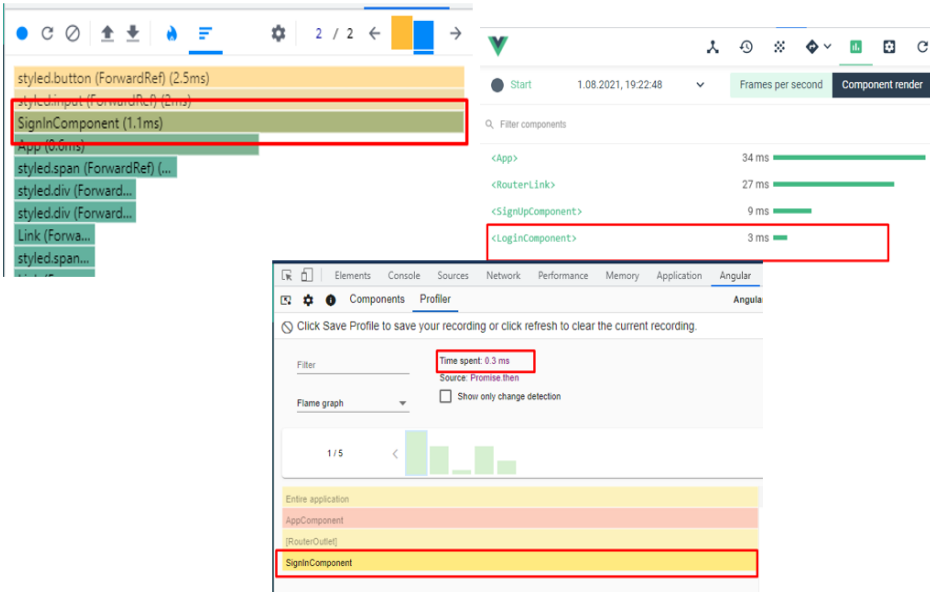


Figure 8. Rendering times of individual components in React, Vue, Angular applications in add-ons installed in Google Chrome

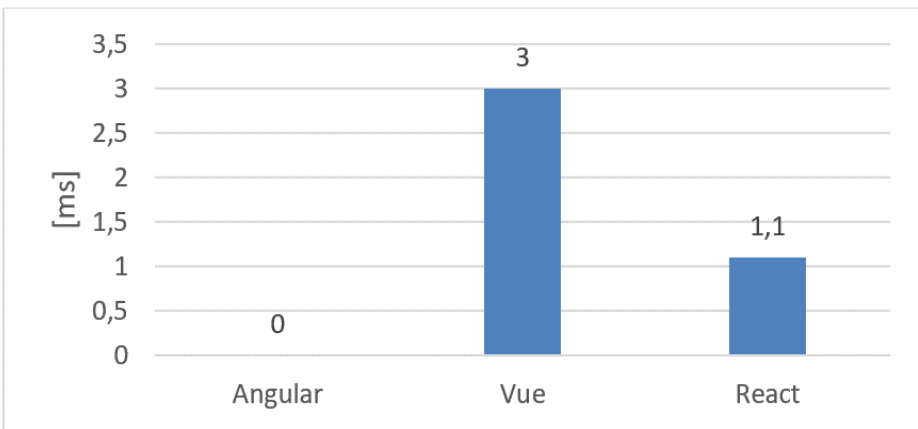


Figure 9. The time required for the browser to generate the components

A comparative analysis of the entire application

The overall results of the tests are shown in Table 2. The time needed to compile each application, as well as the number of lines of code and characters were analyzed in turn. In the time analysis test, the Windows stopwatch was started after running the command responsible for compiling the application (for the Vue application `npm run serve`, for the React application `npm start`, for the Angular application `ng serve`).

The time required to compile the React application was 15 seconds, in Vue similarly 12 seconds. The longest application was compiled in Angular (25 seconds). The total number of lines of code in each application, as well as the total number of characters in each application depending on the HTML code, code responsible for styling and scripting code in each component, showed that Angular is the most demanding. The lowest values were obtained for React. The compiled sums of the number responsible for the script code in applications showed that the number of lines of script code in all cases was close to 600, and the number of script characters, as before, was the highest for Angular.

Table 2. Comparison of key application features

	React	Vue	Angular
Complication time [s]	15	12	25
Number of code lines in HTML	150	213	216
Characters number in HTML	6430	7843	9218
Number of code lines in CSS	585	537	545
Characters number in CSS	10122	10416	10758
Number of code lines in the script language	566	619	600
Characters number in the script language	17421	15487	19075

Conclusion

This paper provides a detailed comparative analysis of the most popular techniques and technologies used in programming web applications on the user interface side. Three applications written specifically for testing in Angular, Vue, and React technologies were used to compare the tools. In addition to the various scripting language technologies, JavaScript and TypeScript were used in the applications. Styling in the applications was implemented in different ways. SCSS preprocessor, ordinary CSS, and styled-components library were used. Applications using different technologies had the same functions. The comparisons showed that most of the markup language code appeared in the application written in Angular technology.

In the case of styling the application, most codes appeared in React technology. Influence on this definitely had to use the library Styled-Component, which makes that each element that we want to style should be assigned to a variable. This solution makes the code definitely more, but it is more scalable and easier to develop. Using SCSS preprocessor in the application written in

Angular technology made the code the least. You can feel that by doing so the code is the least readable, but the syntax of the preprocessor promotes intuitiveness and scalability of the code.

Analysis of the amount of code responsible for the scripting language showed that the most appeared in the application written in React. However, if the .ts files responsible for the service interfaces or “pips” were added to the scripting code of all the components in Angular, by far the most scripting code appears in Angular. This technology favors scalability, for example, the user may be forced to specify the type of a newly defined variable or the type of the returned element from a function. This situation is conducive to the development of projects with greater complexity and on which more developers work.

The analysis of the time needed to implement particular components showed interesting results. The Angular application took the most time, which was due to the fact that it was the first application that was written, a lot of functionality had to be designed, and the complexity of the technology, resulted from the need to use a TypeScript scripting language, which differs from JavaScript. To sum up, the application written in Angular technology turned out to be the most complex and time-consuming, so definitely this type of technology has the highest thresh-old of entry for young developers. The application written in Vue took second place. It took relatively the least time to implement an application written in React. The complexity of this application is also the lowest, as evidenced by the amount of code that is written intuitively.

The functionality of the application was chosen because currently e-commerce applications are by far the most developed in the Polish market. A great advantage of such applications is the ability to implement additional elements such as product details and the implementation of a more complex user page. By far the most needed functionality in applications is the mobile view. The application can be further developed and, most importantly, it has met the user’s expectations.

References

- (2, 12 2021). Retrieved from: <https://distantjob.com/blog/web-application-framework> (15.08.2021).
- Chinnathambi, K. (2019). *React I Redux. Praktyczne tworzenie aplikacji WWW*. Gliwice: Helion.
- Current browser market share* (2021). StatCounter.
- Dymora, P., Mazurek, M., Mroczka, B. (2017). VII Konferencja Naukowa “Symbioza Techniki i Informatyki”. *Code optimization of advanced applications on the example of selected development technologies*.
- Dymora, P., Paszkiewicz, A. (2020). Performance Analysis of Selected Programming Languages in the Context of Supporting Decision-Making Processes for Industry 4.0. *Applied Sciences*, 10(23), 8521. doi:10.3390/app10238521.
- Flanagan, D. (2021). *JavaScript. Przewodnik. Poznaj język mistrzów programowania*. Gliwice: Helion.
- Freeman, A. (2018). *Angular. Profesjonalne techniki programowania*. Gliwice: Helion.

- Hanchett, E., Listwon, B. (2020). *Vue.js w akcji*. Gliwice: Helion.
<https://angular.io/features> (26.09.2021).
<https://firebirdsql.org/> (27.10.2021).
<https://pl.reactjs.org/> (21.10.2021).
<https://www.javascript.com/> (29.10.2021).
<https://www.mysql.com/> (29.10.2021).
<https://www.postgresql.org/> (29.10.2021).
- Kaluža, M., Vukelić, B. (2018). Comparison front-end frameworks for web applications development. *Zbornik Veleučilišta u Rijeci, 1*. doi:10.31784/zvr.6.1.19.
- Molina-Ríos, J., Pedreira-Souto, N. (2020). Comparison of development methodologies in web applications. *Information and Software Technology, 119*. doi:<https://doi.org/10.1016/j.infsof.2019.106238>.