УДК 004.42

І. Г. Майданюк

DEVELOPMENT OF WEB APPLICATIONS WITH JAVASCRIPT

Vinnytsia National Technical University

Анотація

У статті досліджено актуальність розробки веб додатків. Проаналізовано підходи до розробки веб додатків за допомогою мови програмування JavaScript.

Ключові слова: веб додаток, веб розробка, javascript, typescript, повний стек розробки

Abstract

The article examines the relevance of creating web applications. Approaches to web application development using JavaScript programming language are analyzed.

Keywords: web application, web development, javascript, typescript, full stack development

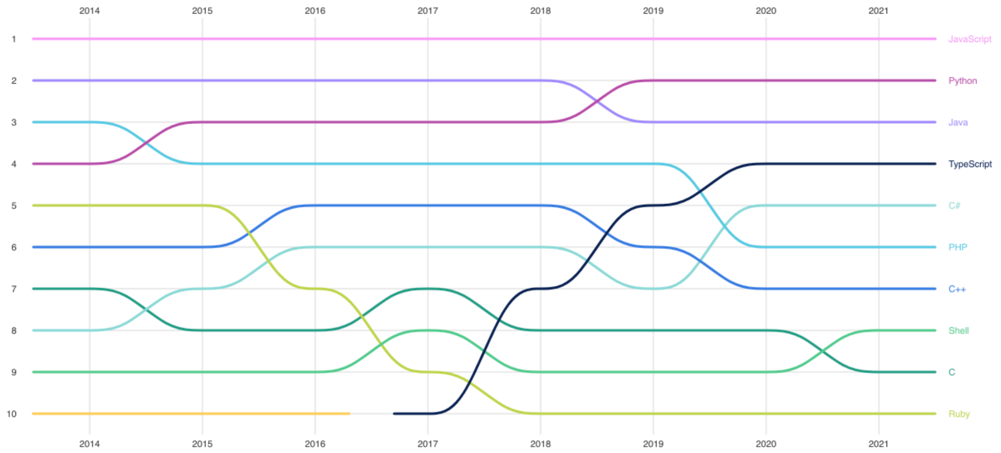
Intro

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps. According to Web.AppStorm editor Jarel Remick, any website component that performs some function for the user qualifies as a Web app [1].

Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops. Many software tools have SaaS implementations, those are based on web technologies stack.

Relevance of JavaScript

JavaScript is one of the most popular programming languages and a dominant web development technology, JavaScript has been around for over 25 years now. TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale. JavaScript and TypeScript are extremely popular programming languages [2].



Picture 1 — JavaScript has maintained its #1 position throughout the years

Being initially bound to front end engineering, the language has seen even more growth in popularity after an ability to use it on the server side was represented. Ever since there has been a heated discussion going on about the use of full stack JavaScript — with MEAN and MERN being its most popular software stacks.

What is full stack JavaScript

It is almost impossible to create a full-fledged software product using only one technology. This usually involves so-called tech stacks — combinations of programming languages, frameworks, libraries, databases, templates, and other tools.

As all software products consist of the client side and the server side, there are usually two separate stacks that cover each of the layers. However, in the case of full stack development, web or mobile applications can be built from A to Z, covering both front end and back end sides of the project with one technology stack. Accordingly, a specialist that has broad skills across all aspects of product engineering is usually called a full stack developer. JavaScript is the universal language that can be used across all software layers, so a person who applies it for both front end and back end programming is called a JavaScript full stack developer [3].

Front end. The choice of possible frameworks for front end JavaScript is quite broad. Besides Angular that was the most popular front end development tool for many years since the introduction of AngularJS back in 2010, numerous newer frameworks and libraries such as React, VueJS, Knockout.js, or Backbone.js can be used, depending on the project specifics.

Back end. Server-side JavaScript programming with Node.js and Express web framework is the most common technology choice, but other Node.js-based tools such as Meteor, Sails.js, Koa, restify, or Keystone.js might be used as well.

Database. As for this part, MongoDB or almost any alternative can be utilized, i.e., MySQL, PostgreSQL, Apache CouchDB, or Apache Cassandra.

MEAN Stack

When talking about full stack JavaScript the first thing that comes to mind is the MEAN stack. It’s an open-source technology bundle that includes MongoDB, Express, Angular and Node.js. Such big companies as Google, Accenture, Sisense, and PayPal are embracing the MEAN approach.

MongoDB is a NoSQL database program that uses a JSON (JavaScript Object Notation) format to store data. Its perfect compatibility with other JS-based parts of the stack ensures great speed and performance. MongoDB is schema-less, which makes it highly flexible, and is built on a horizontal scale-out architecture that allows it to handle big data. All that made it a preferred choice of millions of developers worldwide.

Express.js is a back end web application framework package that runs on top of Node.js. Simply put, it’s a set of tools that manage the workflow between the client side and the data model, ensuring data transferring. Express is used to create APIs, manage HTTP requests, and render basic routing.

Angular is a popular front end framework to create dynamic user interfaces. Its main distinguishing feature is the ability to interact with the web application without reloading the page, reducing site bandwidth and improving performance

Node.js is a back end JS runtime environment that allows the JS code to be executed outside the browser. With its creation, the server-side JS programming finally became possible, enabling a full development cycle using JavaScript only, changing the software world forever. It’s fast (running on the powerful V8 engine), lightweight, extremely efficient, and suitable for building data-intensive, real-time apps, so no surprise it tops the list of most preferred frameworks [4].

Those stack also has a lot of modifications. For example, you may use React or Vue instead of Angular; NestJS instead of Express.js; any SQL database (PostgreSQL, MySQL) instead of MongoDB. You are free to choose appropriate tool for your project, JavaScript stacks are quite flexible to your needs.

The pros of full stack JavaScript development

The fact that companies like Groupon, Airbnb, Netflix, Medium, and PayPal adopted the full stack JavaScript approach to build some of their products speaks for itself. However, small startups seem to enjoy using it as well. This is mostly due to the number of benefits full stack programming offers.

Extensive code reuse. With full stack JavaScript, you save time through code reuse and sharing. Following the “don’t repeat yourself” (DRY) principle, you might be able to reduce the effort by reusing the parts of the code (or sharing libraries, templates, and models) on both back and front end that are very close in terms of logic and implementation.

In other words, you don’t need to think about the JavaScript utility equivalents in Python or Ruby, you just use the same utility on the server and in the browser. Reducing the number of lines of code by up to 40 percent is also a valuable capability when refactoring and maintaining the source code.

High performance and speed. Node.js uses an event-driven, non-blocking IO model that makes it lightweight and fast as compared to other commonly used back end technologies. To prove this, PayPal published a comprehensive report on the results they have seen in the process of migrating from Java to full stack JavaScript. The company was able to make the development almost 2 times faster while reducing the engineering personnel involved. Moreover, they have seen a dramatic improvement in performance, doubling the number of requests completed per second and decreasing the average response time by 35 percent for the same page. This means that the pages are served 200ms faster, which is definitely a noteworthy result.

Common language, better team efficiency with fewer resources. Having all parts of your web application written in JavaScript allows for better understanding of the source code within the team. Therefore, there is no such thing as a gap between front and back end engineering that occurs when two teams are working separately using different technologies.

Moreover, you can now work with only one team instead of two, for back and front end, which should significantly reduce the cost and effort of finding and retaining the right talent. Such a cross-functional team is a great asset when following Agile methodologies.

Free, open source toolset. Most of the full stack JavaScript development tools are free or open source projects. This means you don’t need to bear additional expenses for costly licenses or subscriptions. The tools that are open sourced are updated regularly and evolving fast due to the active community contributions. Instead of relying on a fixed set of technologies, you may use any of more than 1.3 million packages hosted by npm, the largest JavaScript modules registry in the world, recently acquired by Microsoft and integrated with GitHub.

The cons of full stack JavaScript approach

Yet, no technology is perfect. Despite all the benefits the full stack approach offers, there are always some drawbacks to be aware of.

Insufficiency with computation-heavy back end. When it comes to heavy computation and data processing on the server side, Node.js is not the best option. There are lots of far better technologies to handle projects like machine learning, algorithms, or heavy mathematical calculations.

Having a single CPU core and only one thread that processes one request at a time, it might be easily blocked by a single computationally intensive task. While the thread is busy processing the numbers, your application won’t be able to work with other requests, which might result in serious delays.

Yet, there are numerous ways to overcome this limitation. By simply creating child processes or breaking complex tasks into smaller independent microservices that use more suitable technologies and communicate with your back end, you can handle complex computational tasks in Node.js.

Jack of all trades, master of none? It is a common belief that a developer can truly master only one area of knowledge. With every other skill gained the quality of his/her expertise will decline. While the syntax and grammar of JavaScript are mostly the same on the client and server side, there are still many details to consider.

Aside from being proficient in front end development, full stack JavaScript developers need to have an expertise in back end programming, such as HTTP protocol, asynchronous I/O, data storage fundamentals, cookies, etc. That is why some say that there are really no full stack engineers: Every one of them is either front or back end oriented.

Drawbacks of every separate tool in the stack combined. As any technology stack, MEAN/MERN/MEVN combines the weak sides of all 4 of its elements. Most of them are minor technical limitations, which appear under certain circumstances. However, in order to use the stack, it’s important to realize possible bottlenecks of every tool and adjust your development strategy accordingly.

Conclusion

You may have noticed that the aforementioned stacks do not offer any default front end solutions. The reason for that is that the front end part of the system is much more versatile. Developers can use a variety of frameworks for front-end regardless of the chosen stack, each created with their own use cases in mind. Aside from choosing between the existing software packages, you can compose your own stack by looking at the technology used by popular websites and applications.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Web application (Web app) [Електронний ресурс] // Techtarget. – 2019. Режим доступу до ресурсу: https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app

2. GitHub's State of the Octoverse 2021 survey [Електронний ресурс] // GitHub. – 2021. Режим доступу до ресурсу: https://octoverse.github.com/#top-languages-over-the-years

3. Full Stack JavaScript Development Explained [Електронний ресурс] Режим доступу до ресурсу: https://www.altexsoft.com/blog/engineering/mean-mern-javascript-full-stack/

4. MEAN (solution stack) [Електронний ресурс] Режим доступу до ресурсу: https://en.wikipedia.org/wiki/MEAN_(solution_stack)

Майданюк Іван Григорович — студент групи 1КІ-21м, кафедра обчислювальної техніки, ВНТУ, ivan.maidaniuk@gmail.com, Вінниця, Хмельницьке шосе, 95

Maidaniuk Ivan Hryhorovych — student of group 1KI-21м, department of the computer engineering, Vinnytsya National Technical University, ivan.maidaniuk@gmail.com, Vinnytsya, Khmelnytsk highway, 95