Lodz University of Technology
Faculty of Technical Physics, Information Technology and Applied Mathematics

Piotr Kocik
Student no. 217848

JavaScript language for developing language-uniform web applications on the example of e-commerce applications with payments

Bachelor's thesis
in INFORMATION TECHNOLOGY

Supervisor:
dr inż. Jędrzej Mońko

Lodz, May 2017

Abstract

Modern web development landscape changes very rapidly, correspondingly with the growing demand for Internet applications. In recent years, a plethora of technologies and frameworks has been released, what may cause a lot of confusion in developers looking for an efficient, seamless and productive way to create an application. Furthermore, the front-end and back-end division requires separate teams of programmers knowing different, specific languages. However, thanks to Node.js runtime environment, it has become possible to use JavaScript language not just for adding interactivity to the website, but also to create a server that is able to run back end code in this language as well. The objective of this thesis is to create an e-commerce application using a set of JavaScript frameworks to prove that it is possible to quickly and efficiently create a fully-functioning Internet service using single language, therefore eliminating context-switching and requiring developers' proficiency only in one.

Keywords

JavaScript, React, Node.js, Express.js, MongoDB, web development, front-end, back-end, full-stack, payments, Braintree, e-commerce

Thesis domain (Socrates-Erasmus subject area codes)

???

Subject classification

???

Tytuł pracy w języku polskim

???

Contents

Chapter 1

Introduction

Nowadays, the world of programming is developing very rapidly. It is quite a task to develop an application in a way that would be maintainable in the near future and simultaneously require relatively low amount of knowledge in order to simplify the development process. This paper will present that such thing is possible by using an example of an e-commerce application.

1.1. Description of the problem

In the ever-changing landscape of web development, it is hard to keep up with the new technologies that keep popping up constantly. New frameworks appear all the time, JavaScript gets new edition yearly since 2015. Such situation may cause a lot of confusion when it comes to choosing a relatively easy, maintainable and swift way to develop an application. Since the rise of World Wide Web in 1989, the Internet has taken by storm almost entire human civilization. A plethora of human activity domains have moved to the Web, most notably shopping (of which this thesis uses an an mean for proposing a solution), as well as education, administration or even dating. It opened the new possibilities for remote work, which is very significant especially in the world ravaged by SARS-CoV-2 pandemic. People are using Internet to connect with each other, share and discuss ideas, organize into political movements, share their life stories on social media, to name a few possibilities that the Web provides. The extremely rapid expansion the World Wide Web created an enormous amount of demand for web programmers and webmasters. In the 1990s, when the Internet was at the verge of entering the global consciousness, web development workflow was relatively easy and uncomplicated [**modernwebdev**]. However, over the course of the next three decades, there has been an enormous increase in the demand for scalable, advanced and multiplatform applications, as the amount of Internet users has been growing exponentially and mobile devices has come into picture. Accordingly to these trends, the amount of technologies used to develop software has been growing intensively as well.[**chronology**] Naturally, some of the technologies are becoming obsolete and replaced by other ones. This trend concerns even the major technologies that were once the backbone of internet pages, such as Adobe Flash [**eolflash**], whose support would end with the year of 2020, after 24 years of service. This is where problem addressed by this paper arises. In a plethora of web development technologies, frameworks, choosing a right one to start with own project may be confusing and problematic at the very beginning of the process of creation. The workload for learning diverse technologies and programming languages may be time consuming for individuals, as well as troublesome and costly when it comes to enterprise-level development.

1.2. The purpose of the thesis project

This thesis suggests a solution for the aforementioned problem inspired by the JavaScript reaching maturity with ES5 and ES6 in 2015 and 2016, respectively, as well as the creation of Node.js runtime environment that enabled JavaScript code to run outside the web browser and the other frameworks that helped this language to be The Language of the Year 2014. This sort of 'revolution' in the web development landscape allowed JavaScript to be used not just for the front-end User Interface development, but also on the server-side. Therefore, it is possible to develop a web application using just one language, as opposed to the common division between JavaScript on the front-end and another language on the back-end [**evolution**]. More precisely, the solution is about proving that a fully-functional, maintainable web application can be built using just JavaScript and its frameworks - React, Node.js, Express.js, as well as MongoDB database program. Such stack is often reffered to as the MERN stack [**mern**]. Additionally, this thesis will prove that such stack can be a go-to solution for building small to medium web applications.

1.3. Composition of the thesis

This thesis consists of the following chapters:

1. Introduction
2. Overview of the thesis project as a solution
3. Technical details of the project
4. End-user usage of the application
5. Summary

The first chapter introduces to the problem and suggests a solution to it. The second chapter takes a closer look at the thesis project and how it ties to the problem. The following chapters describe technical details of the application, as well as its functionality. The thesis is concluded by a summary and corresponding bibliography.

Chapter 2

Overview of the thesis project as a solution

2.1. E-commerce phenomenon overview

The term 'e-commerce' in general stands for the act of selling or buying goods and services over the Internet, most often using World Wide Web platforms[**wikiecommerce**]. The three most popular platforms for e-commerce are web stores (eg. Amazon), auction portals (eg. Ebay, Allegro) and advertisements portals (eg. OLX, Craigslist). Web stores can be run as a complimentary component to the main stationary business. This system, called "brick-and-mortar" [**growth**] is common for all branches of the trade, raging from electronic stores to bookstores. There are auction portals that provide a platform for sellers to offer their goods as regular products, in exchange for a commission based on the income - for example Polish auction site Allegro. The growth of e-commerce in recent years is enormous; only between 2014 and 2020 the worth of e-commerce sales worldwide went from 1.3 billion dollars to 4.1 billion, according to Shopify[**ecommercestats**]. Moreover, given the current global situation regarding the pandemic of COVID-19, It is almost natural to conclude that the role of e-commerce in the industry services will be bigger[**covid**]. This will lead to the increase in the demand for electronic trade platforms, and therefore for programmers who can swiftly and effectively develop such services. This is where the problem stated in the introduction and the proposed solution come into picture.

2.2. Solution application overview

As mentioned in the previous chapter, the application presenting the solution to the stated problem, will be written using JavaScript across the whole stack (back-end and front-end). Core technologies used in this project are:

- MongoDB
- Express.js
- React
- Node.js

MongoDB database system will be used to store application data. Express.js is a back-end framework that will be used to develop a server running in the Node.js runtime environment, which enables usage of JavaScript outside web browser. User interface will be developed using React framework, enriched with Bootstrap components. Payment processing will be implemented by using Braintree system.

Chapter 3

Technical details of the project

Chapter 4

End-user usage of the application

Chapter 5

Summary

Bibliography