

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Tjaša Heričko

**ANALIZA IN PRIMERJAVA UPORABE PHP IN NODE.JS
PRI RAZVOJU SPLETNIH STRANI**

Diplomsko delo

Maribor, avgust 2017

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Tjaša Heričko

**ANALIZA IN PRIMERJAVA UPORABE PHP IN NODE.JS
PRI RAZVOJU SPLETNIH STRANI**

Diplomsko delo

Maribor, avgust 2017

**ANALIZA IN PRIMERJAVA UPORABE PHP IN NODE.JS PRI RAZVOJU
SPLETNIH STRANI
Diplomsko delo**

Študentka: Tjaša Heričko
Študijski program: Univerzitetni študijski program
Medijske komunikacije
Smer: Vizualna komunikacija
Mentor: doc. dr. Marko Hölbl, univ. dipl. inž. rač. in inf.



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



Številka: E1083604

Datum in kraj: 07. 04. 2017, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Statut UM – UPB 11, Ur. l. RS, št. 44/2015)
izdajam

SKLEP O ZAKLJUČNEM DELU

1. **Tjaši Heričko**, študentki študijskega programa prve stopnje UN MEDIJSKE KOMUNIKACIJE, smer Vizualna komunikacija, se dovoljuje izdelati zaključno delo.
2. Tema zaključnega dela je pretežno s področja Inštituta za medijske komunikacije.
3. **MENTOR:** **doc. dr. Marko Hölbl**
4. **Naslov zaključnega dela:**
ANALIZA IN PRIMERJAVA UPORABE PHP IN NODE.JS PRI RAZVOJU SPLETNIH STRANI
5. **Naslov zaključnega dela v angleškem jeziku:**
ANALYSIS AND COMPARISON OF PHP AND NODE.JS FOR WEB DEVELOPMENT
6. Rok za izdelavo in oddajo zaključnega dela je 30. 09. 2017. Zaključno delo je potrebno izdelati skladno z "Navodili za izdelavo zaključnega dela" in ga v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) oddati v pristojnem referatu članice. Hkrati se odda tudi izjava mentor-ja/-ice (in morebitnega somentor-ja/-ice) o ustreznosti zaključnega dela.

Pravni pouk: Zoper ta sklep je možna pritožba na Senat članice v roku 10 delovnih dni od dneva prejema sklepa.



Dekan:

red. prof. dr. Borut Žalik

Obvestiti:

- kandidatko,
- mentor-ja/-ico,
- odložiti v arhiv.

ZAHVALA

Zahvaljujem se mentorju doc. dr. Marku Hölblu za vodenje in nasvete pri izdelavi diplomskega dela.

Zahvaljujem se tudi družini in prijateljem, ki so me ves čas študija podpirali in spodbujali pri uresničevanju zastavljenih ciljev.

Analiza in primerjava uporabe PHP in Node.js pri razvoju spletnih strani

Ključne besede: spletni razvoj, spletne tehnologije, PHP, JavaScript, Node.js

UDK: 004.774.6(043.2)

Povzetek

V diplomskem delu smo preučili in predstavili možnosti ter načine uporabe tehnologij PHP in Node.js pri razvoju spletnih strani. Na osnovi rezultatov pregleda literature smo oblikovali model za analizo in primerjavo tehnologij, ki je temeljil tudi na izkušnjah, pridobljenih pri razvoju ekvivalentne spletne rešitve z uporabo obeh vrednotenih strežniških tehnologij. Ugotovili smo, da se tehnologiji, razen po zrelosti in razširjenosti, še najbolj razlikujeta glede na razpoložljivost podpore in dostopnost virov ter vidikov, povezanih z zmogljivostjo.

Analysis and comparison of PHP and Node.js for web development

Key words: web development, web technologies, PHP, JavaScript, Node.js

UDK: 004.774.6(043.2)

Abstract

In this diploma thesis possibilities and uses of PHP and Node.js for web development were studied and presented. Based on results of literature overview, a model for analysis and comparison of technologies was formed, where experiences gained from the development of equivalent web solution using both evaluated server-side technologies were also taken into consideration. We found out that technologies, apart from maturity and popularity, differ the most in available support and aspects related to performance.

KAZALO VSEBINE

1	UVOD.....	1
2	STREŽNIŠKA TEHNOLOGIJA PHP.....	3
2.1	Motivacija za izbiro	3
2.2	Začetki in razvoj	3
2.3	Značilnosti	4
3	STREŽNIŠKA TEHNOLOGIJA NODE.JS.....	5
3.1	Motivacija za izbiro	5
3.2	Začetki in razvoj	5
3.3	Značilnosti	6
4	PREGLED SORODNIH DEL	8
4.1	Sorodna dela na področju primerjave tehnologij PHP in Node.js	8
4.2	Sorodna dela na področju primerjave spletnih tehnologij.....	9
5	IZBRANI PRISTOP K PRIMERJAVI TEHNOLOGIJ.....	11
5.1	Model za analizo in primerjavo tehnologij PHP in Node.js.....	11
5.2	K1: razvoj.....	13
5.3	K2: podprte platforme	13
5.4	K3: stroški.....	14
5.5	K4: pomoč in podpora.....	15
5.6	K5: zmogljivost.....	16
5.7	K6: varnost	16
6	RAZVOJ EKVIVALENTNE SPLETNE REŠITVE.....	18
6.1	Namen in funkcionalnosti rešitve	18
6.2	Zasnova in implementacija rešitve	18

6.3	Razvita spletna rešitev	19
7	PRIMERJAVA STREŽNIŠKIH TEHNOLOGIJ PHP IN NODE.JS	22
7.1	K1: razvoj.....	22
7.1.1	RS1.1: razpoložljivost razvojnih okolij	22
7.1.2	RS1.2: sintaksa in vgrajene funkcionalnosti	22
7.1.3	RS1.3: razpoložljivost knjižnic.....	23
7.1.4	RS1.4: krivulja učenja.....	23
7.2	K2: podprte platforme	24
7.2.1	RS2.1: razpoložljivost za operacijske sisteme.....	24
7.2.2	RS2.2: razpoložljivost spletnih strežnikov	24
7.2.3	RS2.3: razpoložljivost podatkovnih baz	25
7.3	K3: stroški.....	26
7.3.1	RS3.1: stroški tehnologije	26
7.3.2	RS3.2: stroški operacijskih sistemov.....	26
7.3.3	RS3.3: stroški spletnih strežnikov	26
7.3.4	RS3.4: stroški podatkovnih baz.....	26
7.4	K4: pomoč in podpora.....	27
7.4.1	RS4.1: kakovost dokumentacije	27
7.4.2	RS4.2: pogostost izdaj verzij	27
7.4.3	RS4.3: čas obstoja tehnologije.....	28
7.4.4	RS4.4: priljubljenost tehnologije med razvijalci	28
7.4.5	RS4.5: trend uporabe na spletnih straneh	29
7.5	K5: zmogljivost	29
7.5.1	RS5.1: zmogljivost preprostih spletnih strani.....	30
7.5.2	RS5.2: zmogljivost pri vhodno-izhodno intenzivnih operacijah	30

7.5.3	RS5.3: zmogljivost pri procesno intenzivnih operacijah.....	31
7.6	K6: varnost	31
7.6.1	RS6.1: zaščita pred SQLi.....	31
7.6.2	RS6.2: zaščita pred XSS	32
7.7	Interpretacija rezultatov primerjave tehnologij PHP in Node.js	32
8	SKLEP	35
VIRI.....		37

KAZALO SLIK

Slika 2.1: Princip delovanja tehnologije PHP [7].....	4
Slika 3.1: Princip delovanja tehnologije Node.js [7].....	6
Slika 3.2: Model procesiranja zahtev s tehnologijo Node.js [17].	7
Slika 6.1: Entitetno-relacijski diagram podatkovne baze spletne rešitve.	19
Slika 6.2: Zaslonski posnetek javnega dela spletne rešitve.	20
Slika 6.3: Zaslonski posnetek zaklenjenega dela spletne rešitve.	20
Slika 6.4: Zaslonski posnetek izseka kode PHP v razvojnem okolju.	21
Slika 6.5: Zaslonski posnetek izseka kode JavaScript v razvojnem okolju.....	21

KAZALO TABEL

Tabela 5.1: Model za analizo in primerjavo tehnologij PHP in Node.js.	12
Tabela 7.1: Razpoložljivost za operacijske sisteme.	24
Tabela 7.2: Razpoložljivost spletnih strežnikov.	25
Tabela 7.3: Razpoložljivost podatkovnih baz.	25
Tabela 7.4: Pogostost izdaj verzij.	28
Tabela 7.5: Število vprašanj v spletnih skupnostih.	28
Tabela 7.6: Trend uporabe na spletnih straneh.	29
Tabela 7.7: Rezultati primerjalne analize PHP in Node.js.	34

KAZALO GRAFOV

Graf 7.1: Zmogljivost preprostih spletnih strani.	30
Graf 7.2: Zmogljivosti pri vhodno-izhodno intenzivnih operacijah.	30
Graf 7.3: Zmogljivost pri procesno intenzivnih operacijah.	31

SEZNAM UPORABLJENIH KRATIC

- AJAX Asynchronous JavaScript and XML (sl. asinhroni JavaScript in XML)
- CSS Cascading Style Sheets (sl. kaskadne stilne predloge)
- HTML Hypertext Markup Language (sl. jezik za označevanje nadbosedila)
- HTTP Hypertext Transfer Protocol (sl. nadbosedilni prenosni protokol)
- MVC Model–View–Controller (sl. model–prikaz–krmilnik)
- NPM Node Package Manager (sl. Node upravljalac paketov)
- PHP PHP: Hypertext Preprocessor (sl. PHP: nadbosedilni predprocesor)
- SQLi SQL injection (sl. vrivanje SQL)
- REST Representational State Transfer (sl. prehod med stanji na osnovi predstavitve)
- URL Uniform Resource Locator (sl. enolični krajevnik vira)
- XSS Cross-Site Scripting (sl. podtikanje skript)

1 UVOD

Razvijalci spletnih strani imajo na razpolago številne tehnologije, zato so soočeni z zahtevnim izzivom – katere tehnologije iz te množice izbrati in uporabiti. Za razvoj strežniškega dela (t. i. backenda) spletnih strani sta na voljo tudi PHP (angl. PHP: Hypertext Preprocessor), najbolj razširjen in uporabljen skriptni jezik na področju razvoja strežniškega dela spletnih strani [1] [2], in Node.js, obetajoča tehnologija, osnovana na strežniškem (t. i. server-side) JavaScriptu [3], ki navkljub svojemu relativno kratkem obstoju pridobiva na popularnosti ter razširjenosti [2]. Da bi razvijalcem olajšali izbiro, smo v diplomskem delu preučili in predstavili možnosti ter načine uporabe tehnologij PHP in Node.js pri razvoju spletnih strani.

Cilji diplomskega dela, ki smo si jih zastavili, so:

- raziskati in spoznati strežniški tehnologiji PHP in Node.js,
- razviti ekvivalentne spletne strani na osnovi tehnologije PHP in na osnovi Node.js,
- primerjati tehnologiji PHP in Node.js ter ugotoviti bistvene razlike in/ali podobnosti glede na oblikovani model za analizo in primerjavo.

Teza diplomskega dela je, da kljub obstoju mnogih tehnologij za razvoj spletnih strani, med katerimi sta tudi PHP in Node.js, ni mogoče v splošnem trditi katera je boljša, ampak je to odvisno od konteksta uporabe, izkušenj razvijalca in želenih funkcionalnosti spletnih strani.

Metode dela, ki smo jih uporabili za doseganje ciljev, so deskriptivna metoda (študija literature), praktičen preizkus tehnologij z razvojem spletnih strani in primerjalna analiza. S pomočjo študije literature smo podrobno preučili in analizirali obe tehnologiji ter oblikovali model za analizo in primerjavo tehnologij za razvoj spletnih strani. Na osnovi izbranih kriterijev smo primerjali značilnosti tehnologij PHP in Node.js. Dodatno smo z omenjenima tehnologijama razvili ekvivalentne spletne strani, kar nam je na osnovi pridobljenih izkušenj omogočilo boljše razumevanje tehnologij za analizo in primerjavo.

Omejitev dobljenih rezultatov analize in primerjave strežniških tehnologij PHP in Node.js je v omejenem naboru kriterijev, ki smo jih izbrali kot relevantne za analizo in primerjavo, pri praktičnem preizkusu pa, da naše ugotovitve temeljijo zgolj na eni – četudi po našem mnenju reprezentativni – študiji primera.

Diplomsko delo je razdeljeno na osem poglavij. V prvem poglavju smo opredelili problem, ki je predmet diplomskega dela, namen, cilje, teze, metodologijo in omejitve. V naslednjih dveh poglavjih smo opisali izbrani strežniški tehnologiji in sicer v drugem poglavju tehnologijo PHP, v tretjem tehnologijo Node.js. Za vsako izmed izbranih tehnologij smo predstavili motivacijo za izbiro, začetke in nadaljnji razvoj ter najpomembnejše značilnosti, vključno s konceptom delovanja. V četrtem poglavju smo predstavili pristope in ugotovitve drugih avtorjev na obravnavanem področju primerjave strežniških tehnologij PHP in Node.js ter povzeli sorodna dela s področja primerjave spletnih tehnologij. Pri tem smo se osredotočili na kriterije in metode primerjave, ki so nam pomagali pri oblikovanju modela za analizo in primerjavo. Nato smo v petem poglavju zasnovali lastni model za analizo in primerjavo tehnologij PHP in Node.js ter definirali kriterije in raziskovalne spremenljivke. V šestem poglavju smo predstavili ekvivalentni spletni rešitvi, razviti s tehnologijo PHP in Node.js, ki sta omogočili kvalitetnejšo analizo in primerjavo. V sedmem poglavju smo tehnologiji med seboj glede na oblikovan model analizirali in primerjali. V osmem in hkrati zadnjem poglavju smo povzeli ugotovitve, jih kritično ovrednotili ter predlagali možnosti nadaljnjega dela.

2 STREŽNIŠKA TEHNOLOGIJA PHP

V tem poglavju smo opisali prvo izmed dveh izbranih tehnologij, in sicer strežniško tehnologijo PHP. Na začetku smo predstavili zakaj smo se odločili za njeno obravnavo, nato smo predstavili še njene začetke in nadaljnji razvoj ter najpomembnejše značilnosti.

2.1 Motivacija za izbiro

PHP ima že vrsto let dominanten položaj med tehnologijami za razvoj strežniškega dela spletnih strani, saj jo dandanes uporablja okrog 82,7 % vseh spletnih strani, ki uporabljajo strežniške tehnologije. Leta 2010 jo je uporabljalo okrog 72,5 % spletnih strani, kar pomeni, da se je delež uporabe skozi leta povečal [2]. Zaradi prevladujočega položaja med strežniškimi tehnologijam in še dodatno naraščajoči uporabi, smo kot prvo preučevano tehnologijo izbrali PHP.

2.2 Začetki in razvoj

Zametki PHP segajo v leto 1994. Rasmus Lerdorf je ustvaril projekt, ki je bil prvotno namenjen sledenju obiskanosti njegove spletne strani z življenjepisom. Prva izdana verzija PHP, takrat poimenovan Personal Home Page Tools (sl. orodja za osebno spletno stran), izhaja iz leta 1995. Omogočala je razvoj zelo preprostih dinamičnih spletnih strani. Nato se je skozi leta PHP precej razvijal, spreminjal in dopolnjeval ter do konca devetdesetih letih prejšnjega stoletja, z verzijo PHP 3.0, dobilo osnovo za PHP kot ga poznamo danes. Med številnimi poimenovanji se je ustalilo poimenovanje PHP kot okrajšava za PHP: Hypertext Preprocessor [1]. Trenutno se v večini uporablja verzija PHP 5.6, v razvoju pa je nova verzija PHP 7.1 [1] [2]. Navkljub skromnim začetkom je postal ena najpomembnejših strežniških tehnologij, saj ga danes uporablja okoli tri četrtine dinamičnih spletnih strani [2], med katerimi so tudi nekatere zelo obiskane spletne strani, kot je na primer največje družbeno omrežje Facebook, ki je celo razvilo prilagojeno implementacijo PHP [4]. Prav tako je PHP osnova največjega blogerskega sistema na spletu – WordPress [5].

2.3 Značilnosti

PHP je splošno namenski skriptni jezik. Njegov glavni cilj je omogočiti spletnim razvijalcem hiter razvoj dinamičnih spletnih strani, čeprav je primeren tudi za druge naloge. Je zelo široko uporabljen, odprtokoden in brezplačen [1] [5].

Datoteka PHP ima končnico ».php« in lahko vsebuje besedilo, HTML (angl. Hypertext Markup Language), CSS (angl. Cascading Style Sheets), JavaScript in kodo PHP. To omogoča, da se lahko PHP enostavno meša s HTML s pomočjo začetne značke `<?php` in končne značke `?>`, med katerima je koda PHP [1] [5].

Deluje tako (glej sliko 2.1), da uporabnik v spletni brskalnik na svojem računalniku vnese naslov URL (ang. Uniform Resource Locator). Spletni brskalnik pošlje zahtevo spletnemu strežniku, kot je na primer Apache. Nato spletni strežnik poveže zahtevek URL z ustrežno skripto PHP. PHP procesor, ki je del spletnega strežnika, nato interpretira skripto ter po potrebi komunicira s podatkovno bazo, kot je na primer MySQL [6]. Tako se koda PHP izvede na spletnem strežniku [1]. Ko je izvršena, se kot rezultat generira navaden HTML dokument, ki je posredovan odjemalcu kot odgovor na poslano zahtevo in ki jo potem spletni brskalnik lahko prikaže uporabniku [5] [6].



Slika 2.1: Princip delovanja tehnologije PHP [7].

3 STREŽNIŠKA TEHNOLOGIJA NODE.JS

V tem poglavju smo predstavili Node.js, drugo izbrano strežniško tehnologijo. Najprej smo opredelili motivacijo za izbiro, nato opisali njene začetke in razvoj ter bistvene značilnosti.

3.1 Motivacija za izbiro

Node.js je relativno mlada strežniška tehnologija, ki hitro pridobiva na priljubljenosti. Po deležu uporabe je že prehitela uveljavljena jezika Perl in Python. W3 Techs tako ocenjuje Node.js kot najhitrejšo rastočo strežniško tehnologijo [2]. Prav tako svetovalna hiša Gartner [8] to tehnologijo uvršča v svoje najnovejše raziskave spletnega računalništva (angl. Hype Cycle for Web Computing), in sicer je trenutno na vrhu napihnenih pričakovanj. Še dodatno Node.js kot JavaScript tehnologija omogoča razvoj celotne spletne strani v JavaScriptu, od odjemalskega do strežniškega dela [9]. Zaradi hitre rasti priljubljenosti ter morebitne zamenljivosti s PHP, smo kot drugo preučevano tehnologijo izbrali perspektivni Node.js.

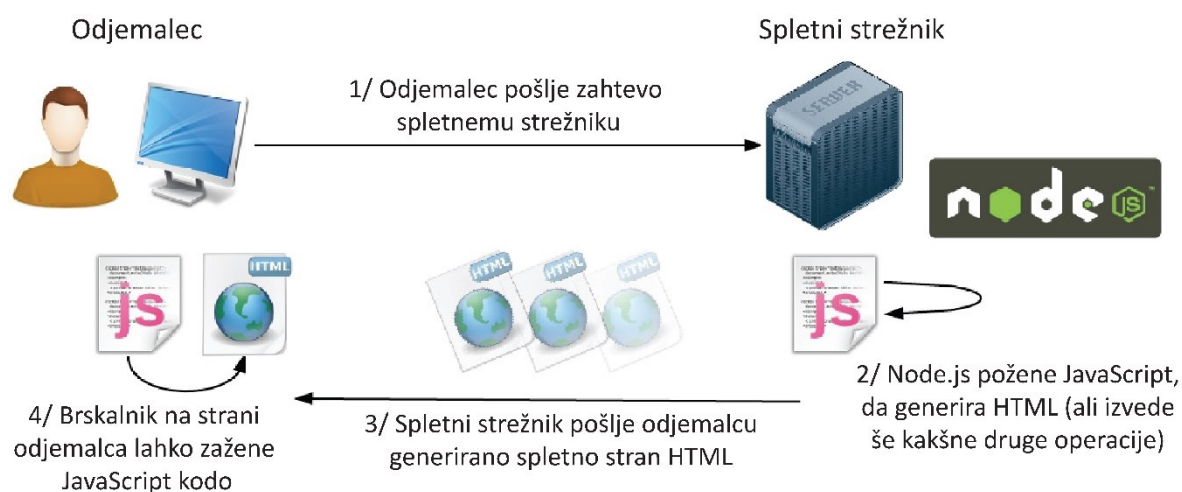
3.2 Začetki in razvoj

Brendan Eich je leta 1995 razvil skriptni jezik JavaScript z namenom upravljanja z dogodki v spletnem brskalniku Netscape Navigator [10]. JavaScript je od nastanka do danes prevzel osrednjo vlogo pri razvoju odjemalskega dela spletnih strani. Vedno bolj popularen postaja tudi za razvoj strežniškega dela, čeprav je bil prvotno namenjen za izvajanje na strani odjemalca. Med njegovimi iznajdbami je tudi obetavna in uveljavljajoča tehnologija za razvoj strežniškega dela spletnih strani v jeziku JavaScript – Node.js [9] [11]. Začetki Node.js segajo v leto 2009, ko je razvijalec Ryan Dahl na konferenci o jeziku JavaScript predstavil svoj eksperimentalni projekt, imenovan Node.js. Njegova ideja je bila omogočiti hiter prenos podatkov do odjemalca, boljšo odzivnost in izrabo virov, ki bi ustrezala današnjim spletnim trendom. Dahl je izkoristil neblokirajočo naravo JavaScripta ter ga uporabil izven spletnega brskalnika na strežniški strani. S pomočjo pogona Google Chrome V8 je izdal prvo verzijo Node.js. Podpora razvijalske skupnosti in nekaterih velikih podjetij so mu omogočili

nadaljnji razvoj [10] [12]. Na njegovo uspešnost kaže tudi to, da danes nekatera velika podjetja, kot so Microsoft, LinkedIn, eBay in YAHOO!, že aktivno uporabljajo Node.js [9].

3.3 Značilnosti

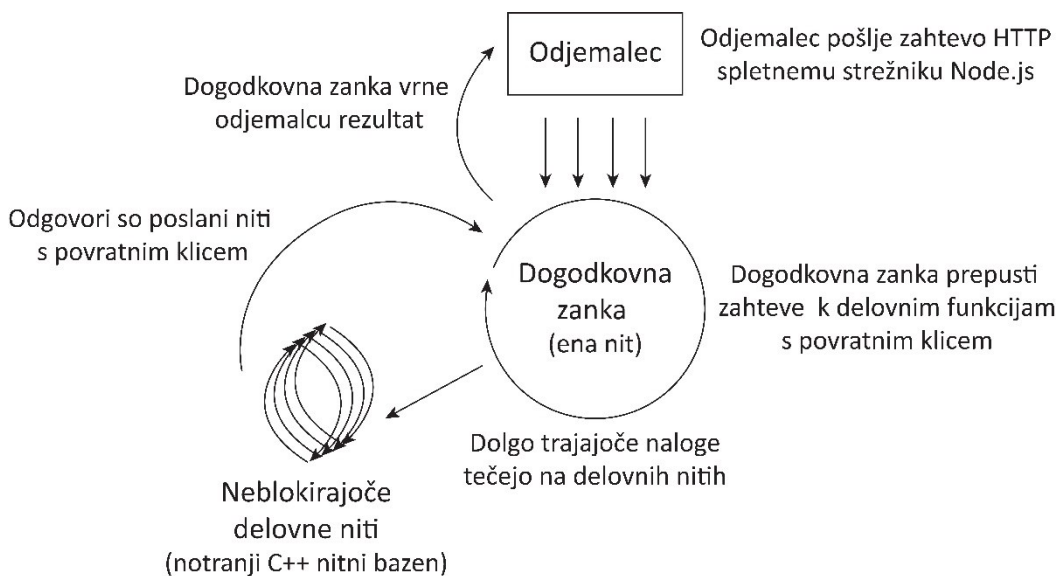
Node.js je relativno mlada odprtokodna in brezplačna JavaScript tehnologija [13] [14]. Ker ponuja razvijanje strežniškega dela v jeziku JavaScript, je JavaScript z njegovim rojstvom ponudil potencialno integracijo od odjemalca do strežnika z uporabo enega jezika (glej sliko 3.1) [9] [4]. Datoteke Node.js imajo končnico ».js«. Vsebujejo naloge, zapisane v jeziku JavaScript, ki so izvedene ob določenih dogodkih [14].



Slika 3.1: Princip delovanja tehnologije Node.js [7].

Node.js predstavlja zanimiv model delovanja strežniškega dela (glej sliko 3.2) [4]. Uporablja dogodkovno voden, enonitni, neblokirajoči (asinhroni) vhodno-izhodni model, ki je enostaven, hiter in učinkovit [13] [15]. Ta model zamenjuje tradicionalno večnitno delovanje z dogodkovno vodeno enonitno obdelavo zahtevkov [4]. Večnitni modeli so namreč relativno neučinkoviti in težki za uporabo [15]. Ker Node.js upravlja zahteve v sklopu ene niti, zmanjša in se izogne dodatni obremenitvi centralne procesne enote in pomnilnika [11]. Namesto, da se ob vsaki novi zahtevi ustvari dodatna nit, je ena nit odgovorna za obdelavo vseh zahtevkov. Ko pride nova zahteva, ta sproži dogodek. Vsak

dogodek uporablja povratni klic, ki se izvede, ko je akcija, na primer povpraševanje iz baze, končana. Medtem pa se lahko nemoteno izvajajo drugi procesi [16].



Slika 3.2: Model procesiranja zahtev s tehnologijo Node.js [17].

NPM (angl. Node Package Manager) je upravljalca paketov za Node.js. Namesti se sočasno z nameščanjem same tehnologije Node.js [14]. Vključuje tisoče brezplačnih paketov za prenos in uporabo, ki so upravljani v odprtokodni skupnosti [11]. Paketi vključujejo vse potrebno za module. Node.js ima namreč module, ki predstavljajo skupek funkcij, ki se lahko vključijo za povečanje funkcionalnosti, podobno kot knjižice v jeziku JavaScript. Nekateri moduli so privzeto vključeni in jih ni potrebno dodatno namestiti. Takšen je modul HTTP (angl. Hypertext Transfer Protocol), ki omogoča, da Node.js prenese podatke preko protokola HTTP in po potrebi ustvari tudi HTTP strežnik. Vključeni so na primer tudi modul za datotečni sistem, modul za dogodke in URL modul. Drugi moduli so izbirni in se jih namesti po potrebi. Lahko se ustvarijo tudi lastni moduli za poljubno vključitev v spletno stran [14].

4 PREGLED SORODNIH DEL

V tem poglavju smo opravili pregled že znanih dosežkov drugih avtorjev na obravnavanem področju primerjave tehnologij PHP in Node.js. Nato smo pregledali literaturo s področja primerjave spletnih tehnologij in predstavili različne pristope k primerjavi, ki so omogočili oblikovanje modela za analizo in primerjavo tehnologij PHP in Node.js.

4.1 Sorodna dela na področju primerjave tehnologij PHP in Node.js

Članke smo iskali s pomočjo iskalnika UM:NIK, ki obsega množico priznanih digitalnih knjižnic, kot sta IEEEExplorer in SpringerLink, ter mnoge druge vire. Iskali smo glede na iskalni niz PHP AND Node.js OR primerjava (angl. comparison). Še dodatno smo pregledali članke baze Google Učenjak. Dobljene rezultate glede na vnesene ključne besede smo pregledali ter ocenili njihovo uporabnost. Izluščili smo tiste članke, ki so obravnavali primerjavo tehnologij PHP in Node.js. Slednji so predstavljene v nadaljevanju.

K. Lei, Y. Ma in Z. Tan [13] so primerjali zmogljivost tehnologij Node.js, Python-Web in PHP. Tehnologije so primerjali na podlagi primerjalnih testov zmogljivosti (sistemskih testov) s pomočjo programske opreme ApacheBench in testnih scenarijev (test realističnega obnašanja uporabnikov) s pomočjo programske opreme LoadRunner. Rezultati so pokazali, da je Node.js boljši pri intenzivnih vhodno-izhodnih operacijah kot PHP. Zmogljivost PHP je boljša pri majhnem številu uporabnikov, medtem ko se izkaže za nestabilno pri naraščanju uporabnikov. Node.js je bolj stabilen, predvsem ko je število uporabnikov večje. Ugotavljajo, da imata tako Node.js kakor PHP težave z računsko intenzivnimi operacijami, čeprav Node.js tudi v tem primeru v primerjavi s PHP deluje bolje. Iz pridobljenih rezultatov sklepajo, da je Node.js bolj primeren za intenzivne vhodno-izhodne in realnočasovne operacije, kakor za računске, saj te ne izkoriščajo prednosti Node.js. Medtem je PHP primeren za spletne strani manjšega in srednjega obsega.

I. Chaniotis, K.-I. Kyriakou in N. Tselika [9] so v svoji raziskavi vrednotili zmogljivosti strežniških tehnologij s serijami stresnih testov. PHP se je ponovno izkazal kot neučinkovit pri velikem številu zahtev in uporabnikov, medtem ko je bil Node.js v takšnih primerih veliko boljši. Node.js je bil boljši pri računskih operacijah in boljše upravljal z viri, centralno procesno enoto in pomnilnikom. Node.js, še slabše pa PHP, se je izkazal pomanjkljiv pri statičnih datotekah.

4.2 Sorodna dela na področju primerjave spletnih tehnologij

Literaturo smo ponovno iskali s pomočjo iskalnika UM:NIK. Iskalni niz smo tokrat zgradili na podlagi sledečih terminov v angleškem jeziku: primerjava, kriteriji, evalvacija, primerjalni model, primerjalne matrike, primerjalna analiza, primerjalna študija, spletne tehnologije, PHP, Node.js in logičnih operatorje AND ter OR. Ponovo smo iskanje izvedli tudi v bazi Google Učenjak, nato pa še z iskalnikom Google. Dobljeno literaturo smo pregledali. V nadaljevanju je na kratko zbrana najpomembnejša literatura, ki se je nanašala na preučevano področje pristopov k analizi in primerjavi spletnih tehnologij.

S. Ahuja in R. Clark [18] sta primerjala spletne strežniške tehnologije iz stališča razvijalcev na podlagi časa potrebnega za razvoj, razpoložljivosti ogrodiv, vzdrževalnosti, podpore, prenosljivosti, nadgradljivost ter stroškov.

K. Samkari in A. Joukhadar [19] sta predlagala kriterije za primerjavo spletnih tehnologij in ogrodiv, in sicer kriterije s stališča končnih uporabnikov, razvijalcev in oblikovalcev. Predlagani kriteriji, ki so pomembni za razvijalce, so zrelost tehnologije, nadgradljivost, prilagodljivost, abstrakcija, ločenost med izgledom in vsebino, ločenost med vsebino in delovnim tokom, zmogljivost, zanesljivost, zmožnosti, podpora več kanalom in varnost.

M. Raible [20] je analiziral spletna ogrodja glede na produktivnost razvijalca, dojetanje razvijalca, krivuljo učenja, zdravje projekta, razpoložljivost razvijalcev, trendov na področju služb, predlog za uporabniški vmesnik, komponente, AJAX (angl. Asynchronous JavaScript

and XML) podporo, dodatke, nadgradljivost, testiranje, internacionalizacijo in lokalizacijo, validacijo, več jezikovno podporo, kakovost dokumentacije, količino izdanih knjig, REST (angl. Representational State Transfer) podporo, mobilno podporo in stopnjo tveganja.

B. Kohan [21] je primerjal PHP in ASP.NET na podlagi nadgradljivosti in preprostosti vzdrževanja, zmogljivosti in hitrosti, stroškov, podpore, potrebnega časa za razvoj, razvojnega okolja, platformске neodvisnosti, popularnosti ter uporabnosti.

A. Ranjan, R. Kumar in J. Dhar [22] so primerjali PHP, JSP in ASP.NET na področju stroškov, platform, zunanjih gostovanj, varnosti, nadgradljivosti, namestitvenih prilagodljivosti, ogrodij in zmogljivosti.

M. Sharma [23] je primerjal PHP in ASP.NET na osnovi časa, potrebnega za razvoj, stroškov in hitrosti, podprtih podatkovnih baz in neodvisnosti platform, razumljivosti ter oblikovanju in varnosti.

L. Titchkosky, M. Arlitt in C. Williamson [24] so primerjali PHP s Perl in Javo na podlagi zmogljivosti. S. Trent, M. Tatsubori, T. Suzumura, A. Tozawa in T. Onodera [25] so primerjali zmogljivosti PHP in JSP. K. Bounnady, K. Phanthavong, S. Pathoumvanh in K. Sihalath [26], kakor tudi M. Jailia, A. Kumar, M. Agarwal in I. Sinha [27] pa zmogljivost PHP in ASP.NET.

5 IZBRANI PRISTOP K PRIMERJAVI TEHNOLOGIJ

V tem poglavju smo glede na rezultate analize sorodnih del iz prejšnjega poglavja oblikovali model za analizo in primerjavo strežniških tehnologij PHP ter Node.js. V podpoglavjih so natančneje predstavljeni in opredeljeni kriteriji ter raziskovalne spremenljivke in njihovo vrednotenje.

5.1 Model za analizo in primerjavo tehnologij PHP in Node.js

Za ustrezno in celovito primerjavo tehnologij bi bilo potrebno upoštevati veliko dejavnikov. V prejšnjem poglavju smo ugotovili, da obstajajo mnogi pristopi in kriteriji za primerjavo tehnologij. Zelo malo člankov je analiziralo in primerjalo Node.js, pa še ti le na podlagi zmogljivosti. Torej, brez nekaterih ostalih faktorjev, ki so lahko ključni pri razvoju.

Za namene primerjave spletnih strežniških tehnologij PHP in Node.js smo zasnovali model z lastnim izborom kriterijev primerjave, s pomočjo katerih smo lahko tehnologiji analizirali in primerjali. Kriterije smo izbrali glede na pomanjkljivosti sorodnih del primerjave PHP in Node.js z dopolnilom iz pregleda literature, ki se nanaša na primerjavo spletnih tehnologij. Izpustili smo nekatere sicer pomembne kriterije, kot je na primer podpora testiranju, ker bi bila za njih potrebna podrobnejša analiza. Za nekatere izmed izbranih kriterijev, kot sta zmogljivost in varnost, smo izpostavili samo določene vidike in bi bila v nadaljnjih delih smiselna podrobnejša analiza.

Predlagani kriteriji modela za analizo in primerjavo tehnologij PHP in Node.js so:

- K1: razvoj,
- K2: podprte platforme,
- K3: stroški,
- K4: pomoč in podpora,
- K5: zmogljivost,
- K6: varnost.

Vsak izmed kriterijev modela je sestavljen iz raziskovalnih spremenljivk, s pomočjo katerih je možna analiza in ovrednotenje posameznega kriterija. V tabeli 5.1 je predstavljen model za analizo in primerjavo vključno z vsemi kriteriji in njihovimi spremenljivkami.

Tabela 5.1: Model za analizo in primerjavo tehnologij PHP in Node.js.

Kriterij		Raziskovalna spremenljivka	
Št.	Naziv	Št.	Naziv
K1	Razvoj	RS1.1	Razpoložljivost razvojnih okolij
		RS1.2	Sintaksa in vgrajene funkcionalnosti
		RS1.3	Razpoložljivost knjižnic
		RS1.4	Krivulja učenja
K2	Podprte platforme	RS2.1	Razpoložljivost za operacijske sisteme
		RS2.2	Razpoložljivost spletnih strežnikov
		RS2.3	Razpoložljivost podatkovnih baz
K3	Stroški	RS3.1	Stroški tehnologije
		RS3.2	Stroški operacijskih sistemov
		RS3.3	Stroški spletnih strežnikov
		RS3.4	Stroški podatkovnih baz
K4	Pomoč in podpora	RS4.1	Kakovost dokumentacije
		RS4.2	Pogostost izdaj verzij
		RS4.3	Čas obstoja tehnologije
		RS4.4	Priljubljenost tehnologije med razvijalci
		RS4.5	Trend uporabe na spletnih straneh
K5	Zmogljivost	RS5.1	Zmogljivost preprostih spletnih strani
		RS5.2	Zmogljivost pri vhodno-izhodno intenzivnih operacijah
		RS5.3	Zmogljivost pri procesno intenzivnih operacijah
K6	Varnost	RS6.1	Zaščita pred SQLi
		RS6.2	Zaščita pred XSS

5.2 K1: razvoj

Pri razvoju spletnih strani z določeno tehnologijo so za razvijalca zelo pomembna razpoložljiva razvojna okolja, v katerih lahko razvija spletne strani, sintaksa in vgrajene funkcionalnosti, dodatne knjižnice, ki omogočajo razširitev vgrajenih funkcionalnosti, ter krivulja učenja. Kriterij razvoja tako vključuje:

- RS1.1: razpoložljivost razvojnih okolij,
- RS1.2: sintaksa in vgrajene funkcionalnosti,
- RS1.3: razpoložljivost knjižnic,
- RS1.4: krivulja učenja.

RS1.1 ocenjuje podporo v integriranem razvojnem okolju NetBeans, ki smo ga uporabili pri razvoju ekvivalentnih spletnih strani, ter pet najbolj popularnih razvojnih okolij za vsako izmed tehnologij. RS1.2 zanima sintaksa in vgrajene funkcionalnosti, RS1.3 pa razpoložljivost knjižnic, ki dopolnjujejo vgrajene funkcionalnosti. RS1.4 ocenjuje težavnost učenja ter razumljivost in posledično tudi produktivnost.

Rezultate smo pridobili tako na osnovi pregleda dokumentacij in spletnih strani kakor tudi lastnih izkušenj, pridobljenih pri razvoju ekvivalentnih spletnih rešitev v obeh tehnologijah.

5.3 K2: podprte platforme

Na hitrost in preprostost razvoja spletne strani s pomočjo določene tehnologije vplivajo ne samo značilnosti tehnologije, ampak tudi (ne)odvisnost od tipa platforme, spletnih strežnikov ter podatkovnih baz, ki podpirajo določeno tehnologijo. Zato smo v kriterij podprtih platform vključili naslednje raziskovalne spremenljivke:

- RS2.1: razpoložljivost za operacijske sisteme,
- RS2.2: razpoložljivost spletnih strežnikov,
- RS2.3: razpoložljivost podatkovnih baz.

RS2.1 meri podprtost za tri najpopularnejše operacijske sisteme na trgu, ki so Microsoft Windows, macOS in Linux, ki imajo skupaj 97,7 % tržni delež [28]. RS2.2 ocenjuje podprtost najbolj uporabljenih spletnih strežnikov, ki so Apache, nginx in Microsoft Internet Information Services (IIS) s skupnim 94,7 % deležem na trgu [29] in možnost samostojnega strežnika. RS2.3 meri podprtost relacijskih in/ali nerelacijskih podatkovnih baz ter pripadajočih desetih najpopularnejših sistemov za upravljanje s podatkovnimi bazami – Oracle, MySQL, Microsoft SQL Server, MongoDB, PostgreSQL, DB2, Cassandra, Microsoft Access, SQLite in Redis [30].

Do rezultatov smo prišli z dokumentacijo in informacijami, dostopnimi na spletnih straneh ter v literaturi.

5.4 K3: stroški

Glede na to, da sam razvoj ni odvisen samo od tehnologije, ampak tudi od platform, spletnih strežnikov in podatkovnih baz, je med stroške razvoja spletnih strani z uporabo določene tehnologije potrebno upoštevati tudi slednje. Zato smo oblikovali naslednje raziskovalne spremenljivke kriterija stroškov:

- RS3.1: stroški tehnologije,
- RS3.2: stroški operacijskih sistemov,
- RS3.3: stroški spletnih strežnikov,
- RS3.4: stroški podatkovnih baz.

RS3.1 meri stroške tehnologije, spremenljivke od RS3.2 do RS3.4 pa stroške platform (operacijskih sistemov, spletnih strežnikov in podatkovnih baz), ki so bili ugotovljeni skozi prejšnji kriterij.

Rezultate smo pridobili iz podatkov na uradnih spletnih straneh tehnologij, operacijskih sistemov, spletnih strežnikov in podatkovnih baz ter drugih spletnih straneh.

5.5 K4: pomoč in podpora

Pri razvoju spletnih strani v določeni tehnologiji je zelo pomembna pomoč in podpora, ki jo lahko nudi uradna dokumentacija in skupnost razvijalcev. Ker je težko natančno ugotoviti kolikšna in kakšna je podpora, smo oblikovali naslednje raziskovalne spremenljivke:

- RS4.1: kakovost dokumentacije,
- RS4.2: pogostost izdaj verzij,
- RS4.3: čas obstoja tehnologije,
- RS4.4: priljubljenost tehnologije med razvijalci,
- RS4.5: trend uporabe na spletnih straneh.

RS4.1 meri kakšno podporo nudi dokumentacija – ali ponuja informacije o namestitvi, osnovni sintaksi, funkcionalnostih, vsebuje primere uporabe (preproste in/ali kompleksne) in ali je na voljo v več jezikih. Upoštevali smo samo uradno dokumentacijo in ne razpoložljivosti in kakovosti drugih virov, kot so izdane knjige ter vodiči. RS4.2 meri pogostost izdaj verzij, in sicer v celotnem času obstoja tehnologije ter v ekvivalentnem obdobju od 1. 1. 2016 do 31. 12. 2016, ki nakazuje na uradno podporo. RS4.3 ocenjuje kako dolgo tehnologija že obstaja. Dlje kot obstaja, več je lahko pomoči in podpore. RS4.4 ocenjuje popularnost, priljubljenost in interes med razvijalci spletnih strani ter s tem podporo razvijalcev. Večja kot je skupnost razvijalcev, večja je pomoč in podpora na raznih forumih, blogih in portalih. RS4.5 meri še trend uporabe tehnologij na spletnih straneh danes in pred leti.

Do rezultatov RS4.1 smo prišli z analizo dokumentacije, objavljene na uradnih spletnih straneh in lastnih izkušnj, pridobljenih pri razvoju ekvivalentnih spletnih rešitev, do RS4.2 s podatki, dostopnimi na uradni spletni strani, do RS4.3 s podatki iz literature. RS4.4 smo pridobili na podlagi statističnih podatkov dveh priljubljenih spletnih skupnosti razvijalcev – Stack Overflow in Code Project. Stack Overflow je največja in najbolj zaupanja vredna spletna skupnost razvijalcev za učenje in deljenje znanja. Nastala je leta 2008 [31]. Priljubljenost na portalu Stack Overflow smo merili na podlagi števila zastavljenih vprašanj,

označenih z imenom tehnologije PHP in tehnologije Node.js. Code Project je skupnost razvijalcev in oblikovalcev, prav tako namenjena učenju in deljenju znanja. Nastala je leta 1999 [32]. Priljubljenost na portalu Code Project smo merili na podlagi števila vseh zadetkov ob iskanju tehnologije PHP in tehnologije Node.js na portalu. Vsi zadetki vključujejo članke, tehnične bloge, namige in trike, vprašanja in odgovore, sporočila ter novice. Vrednotili smo tudi na podlagi izkušenj razvoja ekvivalentnih spletnih rešitev. RS4.5 smo pridobili na podlagi podatkov uporabe strežniških tehnologij skozi čas spletne strani W3 Techs.

5.6 K5: zmogljivost

Pomembna lastnost vsake tehnologije je njena zmogljivost, zato smo tudi zmogljivost vključili med kriterije analize in primerjave. Za analizo zmogljivosti smo predpostavili raziskovalne spremenljivke na treh ravneh:

- RS5.1: zmogljivost preprostih spletnih strani,
- RS5.2: zmogljivost pri vhodno-izhodno intenzivnih operacijah,
- RS5.3: zmogljivost pri procesno intenzivnih operacijah.

RS5.1 meri zmogljivost pri preprosti spletni strani in prikazuje splošne razlike med tehnologijami glede obnašanja pri različnem številu sočasnih uporabnikov. RS5.2 ocenjuje zmogljivost pri vhodno-izhodno intenzivnih operacijah, kakršne so na primer poizvedbe iz podatkovne baze. RS5.3 meri zmogljivost pri operacijah intenzivnih za procesor, kakršne so na primer računsko zahtevne naloge.

Rezultate od RS5.1 do RS5.3 smo pridobili s praktičnimi preizkusi zmogljivosti na osnovi ekvivalentnih primerov spletnih strani.

5.7 K6: varnost

Izredno pomembna lastnost vsake spletne strani je varnost, zato je pomembno, da tehnologija zagotavlja rešitve za preprečevanje napadov, bodisi so te neposredno

vključene, bodisi posredno vključene s pomočjo knjižnic. Z njimi smo vrednotili varnost tehnologij ob predvidevanju, da razvijalec te morebitne varnostne rešitve uporablja. Za namene ocenitve varnosti obravnavanih tehnologij smo preučili rešitve dveh zelo pogostih in nevarnih spletnih napadov, ki lahko povzročita veliko škode, in sicer vrivanje SQL ali SQLi (angl. SQL injection) in podtikanje skript ali XSS (angl. Cross-Site Scripting) [33]. Raziskovalne spremenljivke kriterija varnosti tako so:

- RS6.1: zaščita pred SQLi,
- RS6.2: zaščita pred XSS.

SQLi je vrsta spletnega napada, ki se izvaja z vrivanjem SQL z namenom izvajanja spremenjenih SQL poizvedb. Napad je lahko uspešno izveden nad spletno stranjo, podprto s podatkovno bazo SQL, in sicer lahko doseže izvajanje nenamernih SQL poizvedb ter s tem dostop do (občutljivih) podatkov brez ustrezne avtorizacije, spreminjanje podatkov in izvajanje administratorskih operacij nad podatkovno bazo [33] [34]. Pri XSS napadu se vsili zlonamerna skripta sicer zaupanja vredni spletni strani. Napad lahko povzroči spreminjanje podatkov na spletni strani in prestrezanje podatkov uporabnikov spletne strani [33] [35].

RS6.1 in RS6.2 smo merili na podlagi ponujenih rešitev preprečevanja napadov. Ocenjevali smo, če tehnologija neposredno ponuja varnostne rešitve, jih ponuja preko knjižnic ali ne nudi zaščite pred navedenimi napadi.

Rezultate smo pridobili s pomočjo dokumentacije in informacij, dostopnih na spletnih straneh.

6 RAZVOJ EKVIVALENTNE SPLETNE REŠITVE

V tem poglavju smo opisali namen, funkcionalnosti, zasnovo in implementacijo ekvivalentne spletne rešitve, razvite tako s pomočjo tehnologije PHP kot Node.js za potrebe analize in primerjave.

6.1 Namen in funkcionalnosti rešitve

Razvili smo spletno rešitev, ki je namenjena pregledovanju in upravljanju podatkov o izvedenih projektih, kot so naziv in podrobnejši opis projekta, lokacija in datum izvedbe, število udeležencev ter predstavitve sodelujočih, ki so projekt zasnovali in izvedli.

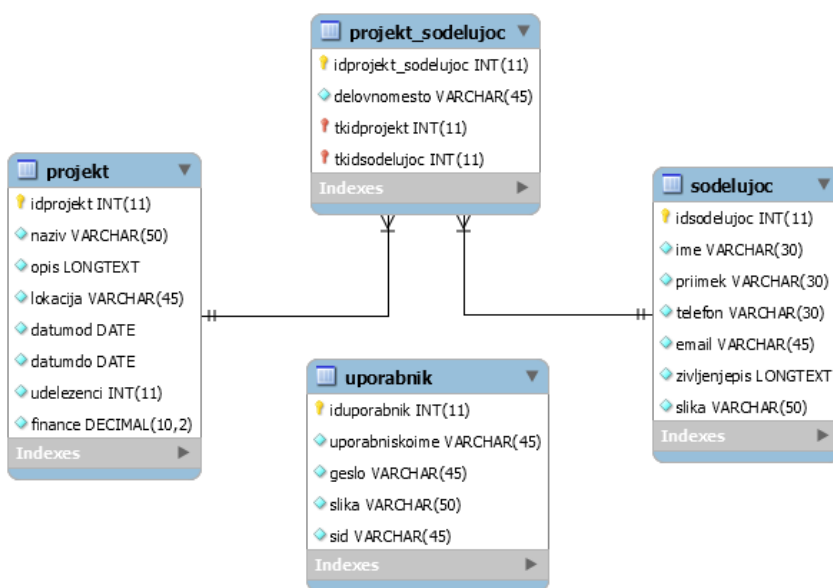
Spletna rešitev je razdeljena na javni in zaklenjen del. Glavne funkcionalnosti javnega dela, za katerega ni potrebna prijava, so pregled osnovnih podatkov o projektih, pregled osnovnih podatkov o sodelujočih in registracija. Registrirani uporabniki se lahko prijavijo in imajo tako na voljo dodatne funkcionalnosti, ki jim omogočajo dostop do podrobnejših podatkov o projektih in o sodelujočih, iskanje projektov glede na ključne besede in prav tako iskanje sodelujočih glede na ključne besede, vnos novih projektov in sodelujočih, izvoz podatkov o določenem projektu oz. generiranje poročil v obliki PDF, dostop do statistike opravljenih projektov ter spreminjanje in prilagajanje uporabniškega profila.

6.2 Zasnova in implementacija rešitve

Za razvoj spletne rešitve s tehnologijo PHP smo uporabili PHP 7.1.7, spletni strežnik Apache 2.4.27 in podatkovno bazo MySQL 5.7.18. Razvijali smo v integriranem razvojnem okolju NetBeans IDE 8.2. Za razvoj s tehnologijo Node.js smo uporabili Node.js v8.2.1 in podatkovno bazo MySQL 5.7.18. Tudi s to tehnologijo smo razvijali v integriranem razvojnem okolju Netbeans IDE 8.2 z vtičnikom za Node.js. Pri obeh rešitvah smo za razvoj izgleda spletnih strani uporabili HTML5 in CSS3. Pri razvoju smo za potrebe implementacije določenih funkcionalnosti vključili knjižnice, in sicer pri razvoju s PHP knjižnico FPDF, ki nam

je omogočila izvoz podatkov o določenem projektu v obliki PDF, pri razvoju z Node.js pa smo vključili večje število dodatnih knjižnic oz. modulov, med katerimi so najpomembnejši modul mysql, ki nam je omogočil delo s podatkovno bazo MySQL, modul multer, namenjen nalaganju datotek (slik), in modul pdfkit, s pomočjo katerega smo lahko razvili generiranje poročil o določenem projektu v obliki PDF. Pri razvoju smo uporabili ogrodje Express ter Handlebars.

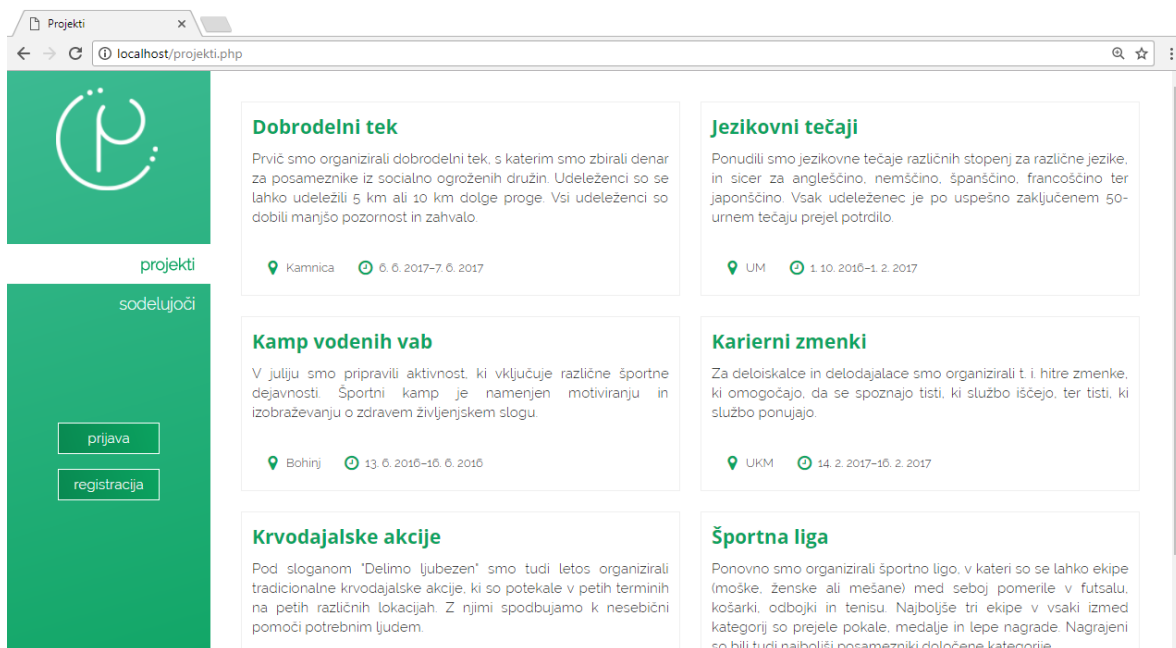
Najprej smo rešitev razvili s pomočjo tehnologije PHP, nato še s pomočjo tehnologije Node.js. Pouporabili smo HTML in CSS. Pri obeh rešitvah smo dostopali do podatkov iste podatkovne baze, ki ima štiri tabele, in sicer projekt, sodelujoc, vmesno tabelo projekt_sodelujoc ter tabelo uporabnik. Na sliki 6.1 je njen entitetno-relacijski diagram.



Slika 6.1: Entitetno-relacijski diagram podatkovne baze spletne rešitve.

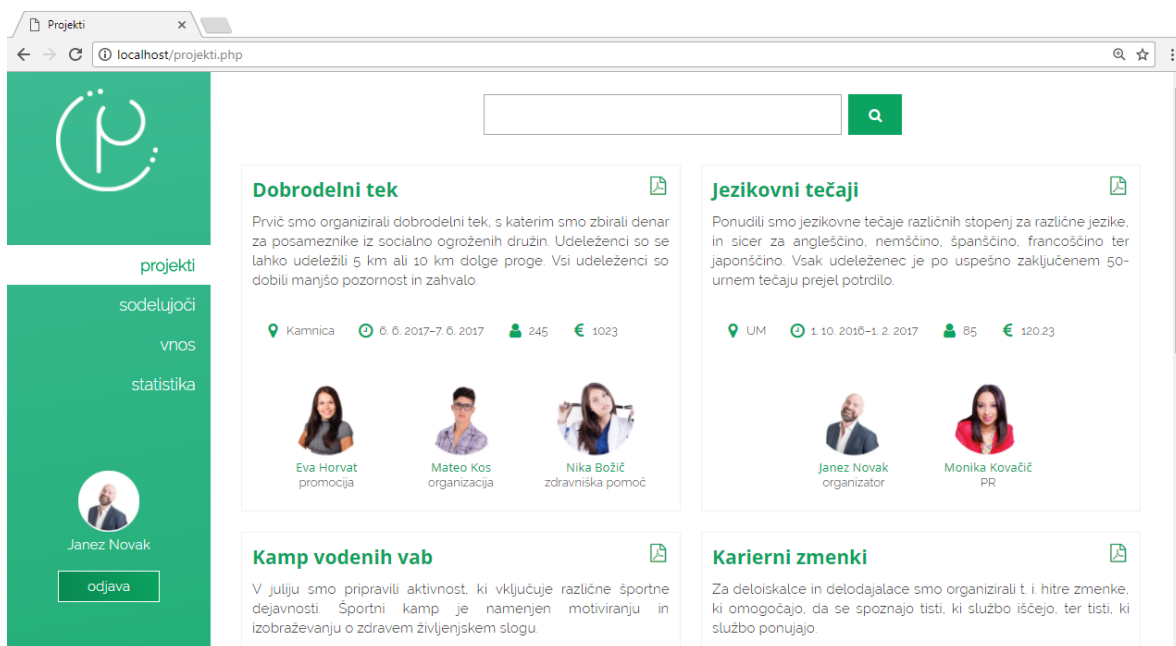
6.3 Razvita spletna rešitev

Ker smo za obe rešitvi uporabili enako grafično zasnovo, se njun izgled ne razlikuje, smo pa zaslonske posnetke spletne rešitve, ki so predstavljeni v nadaljevanju, zajeli v rešitvi, razviti s PHP. Slika 6.2 prikazuje javni del rešitve, ki neprijavljenemu uporabniku omogoča pregledovanje izvedenih projektov in njihovih osnovnih podatkov.



Slika 6.2: Zaslonski posnetek javnega dela spletne rešitve.

Na sliki 6.3. je prikazan zaklenjen del te spletne strani, ki je dostopen registriranim in prijavljenim uporabnikom. Prikazuje več podatkov o projektih in sodelujočih, ki so na projektu sodelovali, ter omogoča brskanje med projekti s pomočjo iskalnega niza in izvoz podatkov določenega projekta v obliki PDF.



Slika 6.3: Zaslonski posnetek zaklenjenega dela spletne rešitve.

Na sliki 6.4 in 6.5 sta predstavljena izseka kode PHP (glej sliko 6.4) in JavaScript (glej sliko 6.5) v razvojnem okolju NetBeans za ekvivalentni spletni strani, in sicer za pregledovanje sodelujočih. Že na prvi pogled so jasne bistvene razlike v sintaksi, načinu uporabe, kompleksnosti, spodbujanju uporabe vzorca MVC (angl. Model–View–Controller) itd.

```

50 <div class="flex-item-vsebina">
51 <?php if ($user == true) { ??
52 <form method="post" action="sodelujoci.php" id="brskanje_po_projektih">
53 <input type="text" name="iskalni_niz" size="40">
54 <button type="submit" name="submit" value="Išči"><i class="fa fa-search" id="isci" aria-hidden="true"></i></button>
55 </form>
56 <br/>
57 <?php } ??
58 <div class="flex-container-izpis">
59 <?php
60     if (isset($_POST['iskalni_niz']) && isset($_POST['submit'])) {
61         $iskalni_niz = preg_replace ("#[^0-9a-z]#", "", $_POST['iskalni_niz']);
62         $stmt = $db->prepare("SELECT * FROM sodelujoc WHERE ime LIKE '%$iskalni_niz%' OR priimek LIKE '%$iskalni_niz%' ORDER BY ime");
63     }
64     else {
65         $stmt = $db->prepare("SELECT * FROM sodelujoc ORDER BY ime");
66     }
67     $stmt->execute();
68     $stmt->store_result();
69     $stmt->bind_result($idsodelujoc, $ime, $priimek, $telefon, $semail, $ziviljenjepis, $slikas);
70     while($stmt->fetch()) {
71     ??
72     <div class="flex-item-izpis">
73     <div class="flex-container-sodelujoci-podatki">
74     <div class="flex-item-sodelujoci-podatki">
75     
76     </div>
77     <div class="flex-item-sodelujoci-podatki">
78     <h2><?php echo $ime . ' ' . $priimek ??</h2>
79     <?php if ($user == true) { ??
80     <p><i class="fa fa-phone fa-fw" aria-hidden="true"></i>&nbsp;&nbsp;&nbsp;<?php echo $telefon ??</p>
81     <p><i class="fa fa-envelope-o fa-fw" aria-hidden="true"></i>&nbsp;&nbsp;&nbsp;<?php echo $semail ??</p>
82     <?php } ??
83     </div>
84     </div>
85     <p><?php echo $ziviljenjepis ??</p>
86     </div>
87     <?php
88     }

```

Slika 6.4: Zaslonski posnetek izseka kode PHP v razvojnem okolju.

```

167 router->post('/sodelujoci', function(req, res, next) {
168     if (req.session.uporabnik) {
169         var uporabnik = req.session.uporabnik;
170         var pot = req.session.pot;
171     }
172     else {
173         var uporabnik = null;
174         var pot = null;
175     }
176     db.query ("SELECT * FROM sodelujoc WHERE ime like ? OR priimek like ? ORDER BY ime", ['%' + req.body.iskalni_niz + '%', '%' + req.body.iskalni_niz + '%']);
177     if (err) throw err;
178     res.render('sodelujoci', {result: result, title: 'Sodelujoči', uporabnik: uporabnik, pot: pot });
179 });

```

```

36 {{#if uporabnik}}
37 <form method="post" action="sodelujoci" id="brskanje_po_projektih">
38 <input type="text" name="iskalni_niz" size="40">
39 <button type="submit" name="submit" value="Išči"><i class="fa fa-search" id="isci" aria-hidden="true"></i></button>
40 </form>
41 <br/>
42 {{/if}}
43 <div class="flex-container-izpis">
44     {{#each result}}
45     <div class="flex-item-izpis">
46     <div class="flex-container-sodelujoci-podatki">
47     <div class="flex-item-sodelujoci-podatki">
48     
49     </div>
50     <div class="flex-item-sodelujoci-podatki">
51     <h2>{{this.ime}} {{this.priimek}}</h2>
52     {{#if ../uporabnik}}
53     <p><i class="fa fa-phone fa-fw" aria-hidden="true"></i>&nbsp;&nbsp;&nbsp;{{this.telefon}}</p>
54     <p><i class="fa fa-envelope-o fa-fw" aria-hidden="true"></i>&nbsp;&nbsp;&nbsp;{{this.email}}</p>
55     {{/if}}
56     </div>
57     </div>
58     <p>{{this.ziviljenjepis}}</p>
59     </div>

```

Slika 6.5: Zaslonski posnetek izseka kode JavaScript v razvojnem okolju.

7 PRIMERJAVA STREŽNIŠKIH TEHNOLOGIJ PHP IN NODE.JS

V tem poglavju smo glede na zastavljen model za analizo in primerjavo strežniških tehnologij opravili primerjalno analizo PHP in Node.js. Tehnologiji smo najprej posamično analizirali glede na zastavljene kriterije in pripadajoče raziskovalne spremenljivke. Na koncu poglavja smo vse rezultate sistematično zbrali, jih interpretirali ter povzeli ugotovitve analize in primerjave.

7.1 K1: razvoj

7.1.1 RS1.1: razpoložljivost razvojnih okolij

Pri razvoju ekvivalentnih spletnih strani v tehnologiji PHP in tehnologiji Node.js smo za obe uporabljali integrirano razvojno okolje NetBeans (pri tehnologiji Node.js s pomočjo vtičnika za to tehnologijo), za katero lahko rečemo, da zagotavlja primerljivo podporo za obe tehnologiji. Sicer pa so najbolj popularna razvojna okolja za PHP NetBeans, PHPStorm, Eclipse, Aptana Studio in Zend Studio [36] [37] ter za Node.js WebStorm, Koding, Eclipse, Komodo IDE in IntelliJ Idea [38] [39].

7.1.2 RS1.2: sintaksa in vgrajene funkcionalnosti

Jezik PHP omogoča različne pristope in načine razvoja ter ne vsiljuje določenega stila programiranja. Posledično lahko razvijemo enostavno nekaj vrstično rešitev, v kateri se prepletata logika in prikaz, ali pa uporabljamo funkcije in objekte ter razvijemo dobro strukturirano objektno-orientirano MVC PHP aplikacijo. Sintaksa PHP se je med verzijami spreminjala, prav tako imena funkcij, zato lahko naletimo na določene nedoslednosti pri poimenovanju funkcij. Pri tem pa PHP nudi ogromno vgrajenih funkcionalnosti. Na drugi strani ima Node.js z jezikom JavaScript manjši nabor osnovnih konstruktorov. Ključna prednost se kaže v dejstvu, da lahko isti jezik uporabljamo tako za programiranje logike, ki se izvaja na strežniku in odjemalcu. Izpostaviti velja še spodbujanje uporabe arhitekture MVC.

7.1.3 RS1.3: razpoložljivost knjižnic

PHP ima zaradi dolgega obstoja na voljo ogromno različnih knjižnic. A vendar jih je potrebno iskati sam, brez pomoči kakšnega orodja. Zato je njihovo iskanje in vključevanje v primerjavi z Node.js težje in zamudnejše. Zaradi zrelosti PHP se velikokrat pojavljajo problemi z zastarelostjo knjižnic. Kot smo ugotovili pri predhodni raziskovalni spremenljivki, Node.js ne omogoča veliko vgrajenih funkcionalnosti. A vendar ima odlično podporo dodatnih knjižic, imenovanih moduli, ki jih je enostavno vključiti v razvoj z Node.js s pomočjo upravljalca paketov za Node.js – NPM. Trenutno je na voljo več kot 475.000 modulov [40], ki znatno prispevajo k produktivnosti razvijalca. NPM omogoča hitro iskanje, namestitve in uporabo ter je privzeto nameščen s tehnologijo Node.js, kar še dodatno poenostavi celoten proces.

7.1.4 RS1.4: krivulja učenja

Glede na to, da smo imeli predhodne izkušnje tako z jezikom PHP kot jezikom JavaScript, nobena izmed tehnologij ni predstavljala večjega izziva. A vendar ocenjujemo, da je razvoj z Node.js zahtevnejši, še posebej za novince. Koda PHP se lahko enostavno meša s HTML, kar sicer ni dobra praksa, saj je takšno kodo težko vzdrževati, ampak omogoča, da lahko novinci hitro razvijejo uporabno kodo. Tudi strukture so v PHP enostavnejše in bolj razumljive, zato je PHP konceptualno preprostejši in enostavnejši za učenje. Node.js pa uporablja drugačen pristop, ki zahteva razumevanje dokaj kompleksnih funkcij že takoj na začetku. Node.js prinaša jezik JavaScript iz odjemalske strani še na strežniško, kar nekoliko olajša vstopno oviro, a vendar prehod med uporabo JavaScripta za odjemalski del na strežniški del ni povsem enostaven, zato zahteva bistveno več truda, četudi je razvijalcu delo z jezikom JavaScript že znano. Večja kompleksnost Node.js omogoča tudi večjo kontrolo. Zato ocenjujemo, da je krivulja učenja pri PHP strma, saj lahko razvijalec v kratkem času bistveno obogati svoje znanje in razumevanje, medtem ko je pri Node.js počasi naraščajoča, saj je učenje počasnejše in je potreben daljši čas za pridobivanje znanja o razvoju s to tehnologijo.

7.2 K2: podprte platforme

7.2.1 RS2.1: razpoložljivost za operacijske sisteme

PHP je lahko uporabljen na številnih operacijskih sistemih, med temi so tudi trije trenutno najpopularnejši. To so Microsoft Windows, macOS in Linux. Podpirajo ga sicer tudi drugi operacijski sistemi, med katerimi so Unix (HP-UX, Solaris, OpenBSD) ter RISC OS [1] [5]. Tudi Node.js je lahko uporabljen na vseh večjih operacijskih sistemih – Microsoft Windows, macOS in Linux. Uporablja se lahko tudi z manj popularnimi operacijskimi sistemi, kot je Unix (AIX, Solaris, FreeBSD, SmartOS) [14] [15]. V tabeli 7.1 so predstavljeni rezultati razpoložljivosti za operacijske sisteme.

Tabela 7.1: Razpoložljivost za operacijske sisteme.

	Windows	macOS	Linux
PHP	Da	Da	Da
Node.js	Da	Da	Da

7.2.2 RS2.2: razpoložljivost spletnih strežnikov

PHP je združljiv z vsemi danes najbolj uporabljenimi spletnimi strežniki, ki so Apache, nginx in Microsoft Internet Information Services (IIS), še dodatno ga podpirajo tudi drugi spletni strežniki, kot je lighttpd [1] [5]. PHP ima možnost vgrajenega strežnika, ampak je ta namenjen samo kot podpora v času razvoja, za testiranje in demonstracije v kontroliranih okoljih [1]. Node.js lahko z vgrajenim HTTP modulom ustvari svoj samostojen strežnik ali pa se izvaja na kakšnem drugem strežniku, kot so popularni Apache, nginx in Microsoft Internet Information Services (IIS) [4] [16]. V tabeli 7.2. so povzeti rezultati razpoložljivosti spletnih strežnikov.

Tabela 7.2: Razpoložljivost spletnih strežnikov.

	Apache	nginx	IIS	Samostojen strežnik
PHP	Da	Da	Da	Ne
Node.js	Da	Da	Da	Da

7.2.3 RS2.3: razpoložljivost podatkovnih baz

PHP ima na voljo širok nabor podatkovnih baz. Med njimi so relacijske in nerelacijske, kakor najpopularnejše Oracle, MySQL, Microsoft SQL Server, MongoDB, PostgreSQL, DB2, Cassandra, SQLite in Redis [6] [30]. Čeprav se Node.js tipično povezuje z uporabo NoSQL podatkovnih rešitev, pa ga lahko uporabimo z različnimi podatkovnimi bazami, tako relacijskimi kakor nerelacijskimi. Med njimi so tudi najpopularnejše podatkovne baze, kot so Oracle, MySQL, Microsoft SQL Server, MongoDB, PostgreSQL, DB2, Cassandra, SQLite in Redis [16] [30]. V tabeli 7.3. so predstavljeni rezultati razpoložljivosti podatkovnih baz za obe tehnologiji.

Tabela 7.3: Razpoložljivost podatkovnih baz.

	Oracle	MySQL	Microsoft SQL Server	MongoDB	PostgreSQL	DB2	Cassandra	Microsoft Access	Redis
PHP	Da	Da	Da	Da	Da	Da	Da	Ne	Da
Node.js	Da	Da	Da	Da	Da	Da	Da	Ne	Da

7.3 K3: stroški

7.3.1 RS3.1: stroški tehnologije

PHP je odprtokoden in brezplačen [1], kar omogoča prosto razširjanje in spreminjanje ter svojevrstne implementacije. Node.js je prav tako odprtokoden in brezplačen [15].

7.3.2 RS3.2: stroški operacijskih sistemov

Najpomembnejši operacijski sistemi, na katerih se lahko izvaja tako PHP kakor Node.js, so glede na rezultate prejšnjega kriterija Microsoft Windows, macOS in Linux. Microsoft Windows je plačljiv [41], macOS je brezplačen za računalnike Mac [42], medtem ko je Linux brezplačen za vse [43].

7.3.3 RS3.3: stroški spletnih strežnikov

PHP in Node.js se lahko izvajata na najpomembnejših spletnih strežnikih. Apache je povsem brezplačen za uporabo [44]. Nginx ima na voljo odprtokodno verzijo, ki je brezplačna, in verzijo nginx plus, ki je plačljiva [45]. IIS se lahko uporablja brez plačila na operacijskih sistemih Windows [46]. Prav tako je HTTP modul, ki v Node.js omogoča ustvarjanje samostojnega strežnika, dostopen brez stroškov plačila, saj je vključen v Node.js, ki je brezplačen [14].

7.3.4 RS3.4: stroški podatkovnih baz

Tako PHP kakor Node.js podpirajo številne najpopularnejše relacijske in nerelacijske podatkovne baze, med katerimi so MySQL, MongoDB, PostgreSQL, Cassandra, SQLite in Redis brezplačne ter Oracle, Microsoft SQL Server in DB2 plačljive z brezplačnimi verzijami [30].

7.4 K4: pomoč in podpora

7.4.1 RS4.1: kakovost dokumentacije

Obsežna dokumentacija PHP [1] vključuje predstavitev PHP, ki podaja osnovne informacije o PHP in njegovem delovanju, navodila in napotke za prve preproste primere za začetnike, informacije o namestitvi in konfiguraciji z različnimi strežniki ter platformami, natančne podatke o tipih spremenljivk, operatorjih, izrazih, strukturah, funkcijah, razredih in objektih, upravljanju z napakami, podprtih protokolih ter ostali sintaksi, kar vključuje teoretičen opis, podkrepjen s praktičnimi primeri, napotke za varnost ter navodila o naprednejših funkcionalnostih. Dokumentacija dodatno omogoča še dodajanje opomb razvijalcev. Dostopna je v desetih jezikih (angleščini, brazilščini/portugalščini, kitajščini, francoščini, nemščini, japonsščini, romunščini, ruščini, španščini in turščini). Dokumentacija torej ponuja informacije o namestitvi, osnovni sintaksi, funkcionalnostih, vsebuje preproste primere uporabe in je na voljo v več jezikih. Node.js nudi dokumentacijo [15], ki obsega tri sklope. Prvi sklop vključuje kratka navodila za razvoj in vzpostavitev preproste spletne strani in izčrpne informacije o pomembnih konceptih ter izključno privzeto vključenih modulih. Informacije so podane tako na teoretični kakor praktični ravni. Drugi sklop je sekcija ES6, ki opisuje katere zmožnosti so omogočene privzeto. Tretji sklop predstavljajo članki, ki opisujejo funkcionalnosti in zmožnosti Node.js s pomočjo kompleksnih rešitev. Dokumentacija Node.js torej nudi informacije o funkcionalnostih in vsebuje primere uporabe – preproste ter kompleksne. Prav tako nam je bila pri razvoju ekvivalentnih spletnih rešitev veliko bolj v pomoč obširna dokumentacija PHP kakor dokumentacija Node.js.

7.4.2 RS4.2: pogostost izdaj verzij

PHP je bil glede na znane podatke uradne spletne strani PHP [1] od leta 2000 naprej, od verzije PHP 4.0, izdan 239-krat. V obdobju enega leta od 1. 1. 2016 do 31. 12. 2016 je bilo izdanih 33 verzij. Node.js je bil od leta 2011 naprej, od Node.js v0.1.14, glede na znane

podatke o izdajah verzij uradne spletne strani Node.js [15], izdan kar 371-krat. V enoletnem obdobju od 1. 1. 2016 do 31. 12. 2016 je bilo izdanih 68 verzij. Rezultati zbranih podatkov so povzeti v tabeli 7.4.

Tabela 7.4: Pogostost izdaj verzij.

	Vse izdaje	V letu 2016
PHP	239	33
Node.js	371	68

7.4.3 RS4.3: čas obstoja tehnologije

Začetki PHP segajo v leto 1995, ko je bila izdana prva verzija PHP 1.0 [1]. Medtem pa začetki Node.js segajo šele v leto 2009 [12].

7.4.4 RS4.4: priljubljenost tehnologije med razvijalci

PHP ima v spletni skupnosti razvijalcev Stack Overflow postavljenih 1.101.785 vprašanj, ki so označeni z označbo PHP [47], 183.444 vprašanj je z označbo Node.js [48]. Na portalu Code Project je 33.761 zadetkov za PHP [49] in 1.825 za Node.js [50]. Rezultati so prikazani v tabeli 7.5. Tudi na podlagi izkušenj, pridobljeni pri razvoju ekvivalentnih spletnih rešitev ocenjujemo, da je PHP veliko bolj priljubljen med razvijalci in ima večjo skupnost razvijalcev.

Tabela 7.5: Število vprašanj v spletnih skupnostih.

	Stack Overflow	Code Project
PHP	1.101.785	183.444
Node.js	33.761	1.825

7.4.5 RS4.5: trend uporabe na spletnih straneh

Analize podjetja W3 Techs [2] o uporabi strežniških tehnologij že vrsto let kažejo, da je PHP najbolj uporabljena tehnologija, namenjena razvoju strežniškega dela spletnih strani. Podatki statistike z dne 1. julija 2017 kažejo, da PHP uporablja 82,7 % vseh spletnih strani, za katere je znana tehnologija, ki je uporabljena za strežniški del. Kakor je razvidno iz tabele 7.6, se delež uporabe PHP skozi zadnja leta povečuje. Analiza hkrati prikazuje porast uporabe Node.js iz popolne neuporabe med konkurenco, pri čemer Node.js hitro pridobiva na popularnosti. Podatki statistike z dne 1. julija 2017 kažejo, da Node.js uporablja 0,4 % vseh spletnih strani, za katere je znana tehnologija, ki je uporabljena za strežniški del.

Tabela 7.6: Trend uporabe na spletnih straneh.

	2011	2012	2013	2014	2015	2016	2017	Julij 2017
PHP	75,3 %	77,3 %	78,7 %	81,6 %	82,0 %	81,7 %	82,4 %	82,7 %
Node.js	/	< 0,1 %	< 0,1 %	0,1 %	0,1 %	0,2 %	0,3 %	0,4 %

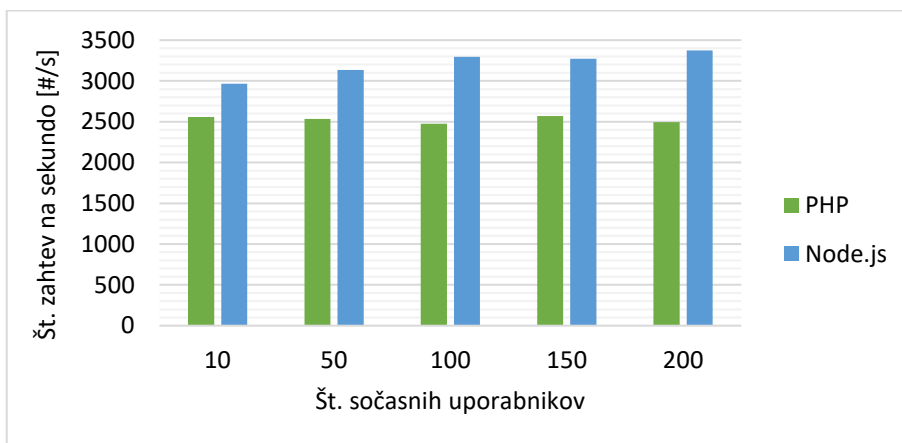
7.5 K5: zmogljivost

Za analiziranje in primerjanje zmogljivosti tehnologij smo oblikovali tri različne testne scenarije, s pomočjo katerih smo evalvirali en faktor naenkrat. Vsak testni scenarij smo preizkusili pod bremenom 5000 zahtev s 10, 50, 100, 150 in 200 sočasnimi uporabniki. V vseh testih smo merili število zahtev na sekundo. Vsak posamezen test smo izvedli v desetih ponovitvah, izločili največjo in najmanjšo vrednost ter izračunali aritmetično sredino preostalih podatkov.

Pri testih s PHP smo uporabili PHP 7.1.7, Apache 2.4.27 in MySQL 5.7.18. Pri testih z Node.js pa Node.js v8.2.1 z modulom mysql in MySQL 5.7.18. Testirali smo na računalniku z Windows 10, 64-bitnim operacijskim sistemom, procesorjem Intel i5 in 4GB pomnilnikom RAM. Za pridobitev rezultatov smo uporabili ApacheBench 2.3.

7.5.1 RS5.1: zmogljivost preprostih spletnih strani

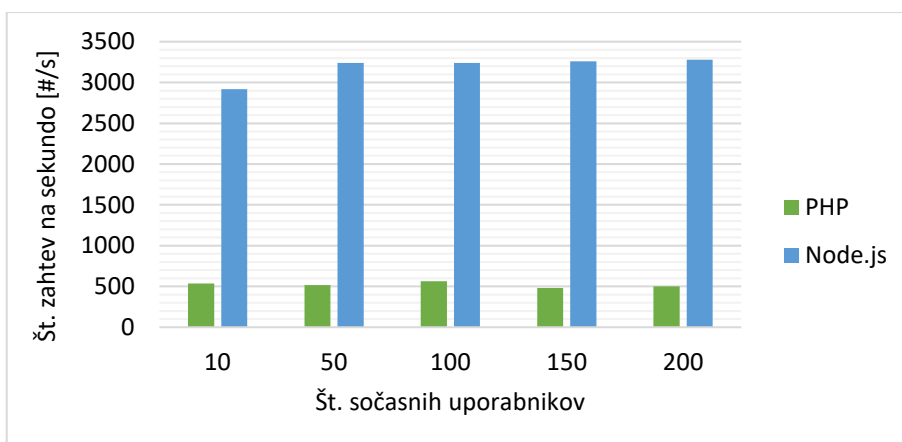
Za testiranje zmogljivosti preproste spletne strani smo uporabili spletno stran, ki le izpiše besedilo. Rezultati testov kažejo (glej graf 7.1), da se Node.js boljše odziva na povečanje števila sočasnih uporabnikov. Kljub temu razlike med tehnologijama niso zelo velike.



Graf 7.1: Zmogljivost preprostih spletnih strani.

7.5.2 RS5.2: zmogljivost pri vhodno-izhodno intenzivnih operacijah

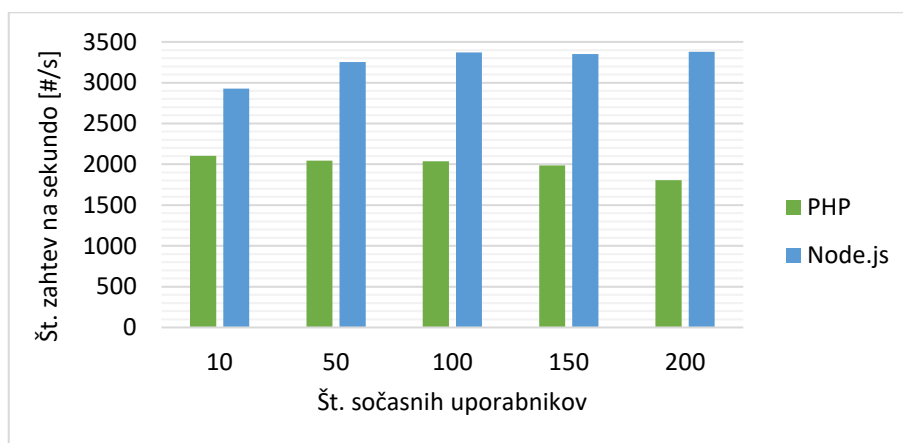
Za vrednotenje zmogljivosti pri vhodno-izhodno intenzivnih operacijah smo testirali spletno stran s povezavo na podatkovno bazo, ki izvede poizvedbo. Rezultati testiranja (glej graf 7.2) so pokazali bistvene razlike med PHP in Node.js, in sicer je zmogljivost Node.js boljša pri vhodno-izhodno intenzivnih operacijah ter ostaja visoka tudi pri povečanju uporabnikov.



Graf 7.2: Zmogljivosti pri vhodno-izhodno intenzivnih operacijah.

7.5.3 RS5.3: zmogljivost pri procesno intenzivnih operacijah

Zmogljivost pri procesno intenzivnih operacijah smo ocenili na podlagi računsko zahteve naloge, in sicer računanje določenega člena Fibonaccijevega zaporedja. Ugotovili smo, da se obe tehnologiji s težavo soočata s takšnimi nalogami, a vendar se je tudi tukaj Node.js izkazal bolje. Graf 7.3 prikazuje zmogljivost obeh tehnologij pri računanju 15. člena Fibonaccijevega zaporedja.



Graf 7.3: Zmogljivost pri procesno intenzivnih operacijah.

7.6 K6: varnost

7.6.1 RS6.1: zaščita pred SQLi

PHP se lahko pred napadi tipa SQLi ščiti z uporabo pripravljenih stavkov (angl. Prepared Statement) [1] [51]. Druga možnost obrambe je z uporabo ustreznih validacijskih funkcij, ki preverjajo skladnost s podatkovnim tipom, kot so *is_numeric()*, *ctype_digit()* [1]. Node.js se lahko pred napadom tipa SQLi varuje s pomočjo modula *mysql*, ki ga je potrebno eksplicitno namestiti. Ta ponuja možnost ubežnih funkcij *mysql.escape()*, *pool.escape()* in *connection.escape()*. Možna je tudi rešitev z uporabo znaka *?*, ki je kasneje povezan z vrednostjo in pravzaprav interno uporabi *connection.escape()* funkcijo. Še ena možnost je uporaba pripravljenih poizvedb (angl. Preparing Queries) s pomočjo funkcije *mysql.format()* [52].

7.6.2 RS6.2: zaščita pred XSS

PHP se lahko pred napadi tipa XSS zaščiti s funkcijo `htmlspecialchars()`. Pri nekaterih XSS napadih je ta neuspešna in je potrebna uporaba varnih kodiranih knjižnic, ki avtomatsko prepoznajo podatke, ki jih želimo filtrirati, kot so `HTMLPurifier`, `OWASP AntiSamy` in `OWASP ESAPI` [51]. `Node.js` se pred XSS napadi lahko varuje z različnimi dodatnimi moduli, ki filtrirajo vnose uporabnikov. Takšni moduli, ki jih je potrebno dodatno namestiti, so `xss` [53], `xss-filter` [54] in `x-xss-protection` [55].

7.7 Interpretacija rezultatov primerjave tehnologij PHP in Node.js

Za PHP, najbolj razširjen in uporabljen skriptni jezik na področju razvoja strežniškega dela spletnih strani, je značilna zrelost in bogat nabor funkcij. Je enostaven in omogoča hitro vpeljavo tudi programiranja manj večjih razvijalcev. Ne spodbuja dobre strukturiranosti kode, jo pa omogoča. Na drugi strani `Node.js` prinaša jezik `JavaScript` tudi na strežniško stran, njegova popularnost pa je povezana s sodobno zasnovo z manjšim naborom osnovnih konstruktov ter odlično podporo aktualnim konceptom. `Node.js` ima sicer manjši nabor vgrajenih funkcionalnosti, a vendar dobre možnosti dodatnih modulov, ki jih je s pomočjo `NPM` enostavno namestiti, vključiti in uporabiti. Tudi PHP ima na razpolago številne knjižnice, ki lahko razširijo že tako široke osnovne funkcionalnosti, vendar je njihovo iskanje, nameščanje in uporaba težja. K rasti popularnosti `Node.js` in zanimanja zanj zagotovo prispeva tudi dogodkovna vodenost ter asinhrono, neblokirajoče izvajanje operacij. A vendar je težji za učenje. Obe tehnologiji sta podprti v številnih integriranih razvojnih okoljih.

Po osnovnih značilnostih podprtosti platform in stroškov razvoja sta PHP in `Node.js` zelo podobna in primerljiva. Oba sta združljiva z najpopularnejšimi operacijskimi sistemi, spletnimi strežniki in istimi najpopularnejšimi podatkovnimi bazami. Edina razlika med njima je, da lahko `Node.js` vzpostavi samostojen spletni strežnik v realnem okolju. Obe

tehnologiji sta brezplačni in odprtokodni. Zaradi zelo široke podpore najpopularnejših platform imata obe na voljo tako plačljive kakor brezplačne verzije omenjenih storitev.

Glede na pomoč in podporo, ki je na voljo za dani tehnologiji, smo ugotovili, da ima PHP bistveno obširnejšo dokumentacijo kakor Node.js, še posebej za začetnike, medtem ko Node.js vključuje tudi kompleksnejše rešitve, namenjene naprednejšim razvijalcem. A vendar ima PHP, kljub 22 letim obstoja, manj izdanih verzij in popravkov kakor Node.js, ki obstaja šele 8 let. V lanskem koledarskem letu je izšlo več kot 2-krat več verzij Node.js kot PHP. Obe tehnologiji se tako sicer aktivno razvijata, a vendar se Node.js razvija hitreje. Še vedno pa ima PHP večje zanimanje in popularnost med razvijalci ter več izkušenih razvijalcev, kar kažejo podatki pomoči skupnosti razvijalcev spletnega foruma Stack Overflow in Code Project ter statistika uporabljanostih tehnologij na spletnih straneh.

Rezultati glede zmogljivosti tehnologij so skladni oz. primerljivi z ugotovitvami drugih raziskav [9] [13] ter so tudi pričakovani. Zaradi svojih značilnosti je PHP, čeprav je dokaj učinkovit na strežniški strani, primeren za spletne strani mlajšega in srednjega obsega. Node.js je primeren za podatkovno zahtevne, hitro odzivne, realnočasovne aplikacije in aplikacije z več uporabnikov, medtem ko ni primeren za aplikacije, ki so procesorsko intenzivne, saj te ne izkoriščajo njegovih prednosti.

Tako PHP kakor Node.js nudita rešitve za najpogostejše varnostne grožnje, med katerimi sta SQLi in XSS. Medtem ko PHP zagotavlja varnostne rešitve pred napadi, ki so večinoma neposredno vključene v samo tehnologijo, jih Node.js zagotavlja samo posredno preko modulov, ki jih je potrebno dodatno (brezplačno) namestiti.

Zaradi boljše preglednosti smo vsa spoznanja o tehnologijah predstavili v tabeli 7.7, kjer smo povzeli rezultate naše analize in tehnologiji sistematično med seboj primerjali.

Tabela 7.7: Rezultati primerjalne analize PHP in Node.js.

Kriterij	Raziskovalna spremenljivka	Tehnologija	
Št.	Št.	PHP	Node.js
K1	RS1.1	NetBeans, PHPStorm, Eclipse, Aptana Studio, Zend Studio	WebStorm, Koding, Eclipse, Komodo IDE, IntelliJ Idea
	RS1.2	Enostavnost, a nedoslednost, bogat nabor funkcij	Sodobno zasnovan jezik, malo konstruktov, spodbuja vzorec MVC
	RS1.3	Množica knjižic	Množica modulov (NPM)
	RS1.4	Strma krivulja učenja	Položna krivulja učenja
K2	RS2.1	Windows, macOS, Linux	Windows, macOS, Linux
	RS2.2	Apache, nginx, IIS	Apache, nginx, IIS, Node.js
	RS2.3	Oracle, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, DB2, Cassandra, Redis	Oracle, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, DB2, Cassandra, Redis
K3	RS3.1	Brezplačno	Brezplačno
	RS3.2	Plačljivi in brezplačni	Plačljivi in brezplačni
	RS3.3	Plačljivi in brezplačni	Plačljivi in brezplačni
	RS3.4	Plačljive in brezplačne	Plačljive in brezplačne
K4	RS4.1	Bolj obsežna dokumentacija	Manj obsežna dokumentacija
	RS4.2	Manj izdanih verzij	Več izdanih verzij
	RS4.3	Daljše obdobje obstoja	Krajše obdobje obstoja
	RS4.4	Več zanimanja na forumih	Manj zanimanja na forumih
	RS4.5	Večji delež uporabe	Manjši delež uporabe
K5	RS5.1	Dobra zmogljivost	Dobra zmogljivost
	RS5.2	Slaba zmogljivost, predvsem pri povečanju števila uporabnikov	Odlična zmogljivost, predvsem pri povečanju števila uporabnikov
	RS5.3	Slabša zmogljivost	Boljša zmogljivost
K6	RS6.1	Neposredno	Posredno
	RS6.2	Neposredno in posredno	Posredno

8 SKLEP

V diplomskem delu smo preučili in primerjali dve spletni tehnologiji za razvoj strežniških rešitev, in sicer že uveljavljen in zelo razširjen PHP, za katerega je značilna predvsem enostavnost in možnost mešanja PHP in HTML kode, ter vse bolj popularno in po mnenju mnogih sodobnejšo tehnologijo Node.js. Ključna značilnost slednje je, da prinaša uporabo jezika JavaScript tudi na strežniško stran ter temelji na dogodkovno vodenem, asinhronem, neblokirajočem izvajanju operacij.

Na podlagi opravljene analize in primerjave smo potrdili tezo diplomskega dela, in sicer v splošnem ni mogoče trditi katera tehnologija izmed PHP in Node.js je boljša, ampak je to odvisno od mnogih vidikov. Vsaka izmed obravnavanih tehnologij ima svoje prednosti, zato je potrebno in smiselno v konkretnem primeru izbire primerne tehnologije ustrezno utežiti posamezne kriterije ter po potrebi vključiti še dodatne vidike in kriterije. Postavljene vrednosti uteži bi zagotovo bile odvisne od značilnosti razvijalcev in organizacije, pa tudi konkretnega projekta, saj je pomembno, ali želimo razviti preprosto predstavitevno stran ali pa zahtevnejšo spletno aplikacijo, na primer za celovito upravljanje rezervacijskega sistema hotelskih storitev. Glede na rezultate modela za analizo in primerjavo tehnologij PHP in Node.js se je namreč izkazalo, da sta si tehnologiji po osnovnih značilnostih zelo podobni, razlikujeta se le v tem, da ima Node.js možnost vzpostavitve samostojnega strežnika. Tudi pri varnostnih problematikah nudita podobne rešitve za preprečitev spletnih napadov. PHP je starejši, a preprostejši in enostavnejši za učenje, hkrati pa ima v primerjavi z modernejšim Node.js večjo podporo dokumentacije in razvijalske skupnosti. Glavna razlika med tehnologijama se kaže v zmogljivostih, in sicer je PHP bolj primeren za strani, povezane z upravljanjem vsebin in prikazom podatkov, medtem ko je Node.js primernejša izbira za razvoj kompleksnejših rešitev z množico sočasnih uporabnikov ter zahtevnejšimi obdelavami.

Smiselna bi bila nadaljnja primerjava in razširitev modela za analizo in primerjavo še na podlagi drugih kriterijev in raziskovalnih spremenljivk. Pri kriteriju razvoja bi lahko na

primer raziskali še ogrodja, ki olajšajo in pohitijo razvoj ter omogočajo dodatne funkcionalnosti. Lahko bi raziskali kompleksnost in količino kode pri naprednejših rešitvah itd. Dodali bi lahko tudi druge kriterije, kot so testiranje, nadgradljivost in vzdrževalnost.

VIRI

- [1] PHP. Dostopno na: <http://php.net/> [20. 6. 2017].
- [2] W3 Techs. Usage of server-side programming languages for websites. Dostopno na: https://w3techs.com/technologies/history_overview/programming_language/ms/y [25. 6. 2017].
- [3] Tilkov, S., Vinoski, S. Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing*, 14, (2010), 6, str. 80–83. Dostopno na: <http://ieeexplore.ieee.org/document/5617064/> [28. 6. 2017].
- [4] Schutt, K., Osman, B. Cloud software development platforms: A comparative overview. V: Song, Y.-T. (ur.), *SERA. 2016 IEEE/ACIS 14th International Conference on Software Engineering Research, Management and Applications*, Balitmore, 8.–10. junij 2016. Balitmore: IEEE Computer Society, 2016, str. 3–13. Dostopno na: <http://ieeexplore.ieee.org/document/7516122/> [20. 6. 2017].
- [5] W3Schools. PHP Tutorial. Dostopno na: <https://www.w3schools.com/php/default.asp> [29. 6. 2017].
- [6] Sklar, D. *Learning PHP: A Gentle Introduction to the Web's Most Popular Language*. Sebastopol: O'Reilly Media, 2016.
- [7] OpenClassrooms. Node.js: what is it for exactly? Dostopno na: <https://openclassrooms.com/courses/ultra-fast-applications-using-node-js/node-js-what-is-it-for-exactly> [26. 7. 2017].
- [8] Revang, M. Gartner Hype Cycle for Web Computing, 2016. Dostopno na: <https://www.gartner.com/document/3398518> [29. 6. 2017].
- [9] Chaniotis, I., Kyriakou, K.-I., Tselikas, N. Is Node.js a viable option for building modern web applications? A performance evaluation study. *Computing*, 97, (2015), 10, str. 1023–1044. Dostopno na: <https://link.springer.com/article/10.1007%2Fs00607-014-0394-9> [22. 6. 2017].
- [10] Howard, D. *Node.js for PHP Developers*, prva izdaja. Sebastopol: O'Reilly Media, 2012.

- [11] Ogasawara, T. Workload characterization of server-side JavaScript. V: *//SWC 2014*. 2014 IEEE International Symposium on Workload Characterization, Raleigh, 26.–28. oktober 2014. Raleigh: IEEE Computer Society, 2014, str. 13–21. Dostopno na: <http://ieeexplore.ieee.org/document/6983035/> [20. 6. 2017].
- [12] Haviv, A., Mejia, A., Onodi, R. *Web Application Development with MEAN*. Birmingham: Packt Publishing, 2016.
- [13] Lei, K., Ma, Y., Tan, Z. Performance Comparison and Evaluation of Web Development Technologies in PHP, Python and Node.js. V: X. Liu, D. E. Baz, C.-H. Hsu, K. Kang, W. Chen (ur.), *CSE 2014*. 2014 IEEE 17th International Conference on Computational Science & Engineering, Chengdu, 19.–21. december 2014. Chengdu: IEEE Computer Society, 2014, str. 661–668. Dostopno na: <http://ieeexplore.ieee.org/document/7023652/> [19. 6. 2017].
- [14] W3Schools. Node.js Tutorial. Dostopno na: <https://www.w3schools.com/nodejs/default.asp> [28. 6. 2017].
- [15] Node.js. Dostopno na: <https://nodejs.org/> [20. 6. 2017].
- [16] Bangare, S. L., Gupta, S., Dalal M., Inamdar, A. Using Node.Js to Build High Speed and Scalable Backend Database Server. V: *NCPCI-2016*. National Conference and Project Competition for Students in Computer Engineering and Information Technology, Pune, 19. marec 2016. Pune: International Journal of Research in Advent Technology, 2016, str. 61–64. Dostopno na: https://www.researchgate.net/publication/301788361_Using_NodeJs_to_Build_High_Speed_and_Scalable_Backend_Database_Server [20. 6. 2017].
- [17] Stack Overflow. How the single threaded non blocking IO model works in Node.js. Dostopno na: <https://stackoverflow.com/questions/14795145/how-the-single-threaded-non-blocking-io-model-works-in-node-js> [26. 7. 2017].
- [18] Ahuja, S., Clark, R. Comparison of Web Services Technologies from a Developer's Perspective. V: Selvaraj, H., Srimani, K. (ur.), *ITCC 2005*. International Conference on Information Technology: Coding and Computing, Las Vegas, 4.–6. april 2005. Los Alamitos: IEEE Computer Society, 2005, vol. 2, str. 791–791. Dostopno na: <http://ieeexplore.ieee.org/document/1425251/> [15. 7. 2017].

- [19] Samkari, K., Joukhadar, A. Comparison Matrix for Web HCI V: *ICTTA 2008*. 3rd International Conference on Information and Communication Technologies: From Theory to Applications. Damascus, 7.–11. april 2008. Damascus: IEEE Computer Society, 2016, str. 1–5. Dostopno na: <http://ieeexplore.ieee.org/document/4530335/> [15. 7. 2017].
- [20] Raible, M. How I Calculated Ratings for My JVM Web Frameworks Comparison, 2010. [https://raibledesigns.com/rd/entry/how i calculated ratings for](https://raibledesigns.com/rd/entry/how_i_calculated_ratings_for) [15. 7. 2017].
- [21] Kohan, B. PHP vs ASP.net Comparison, 2010. Dostopno na: <http://www.comentum.com/php-vs-asp.net-comparison.html> [14. 7. 2017].
- [22] Ranjan, A., Kumar, R., Dhar, J. A Comparative Study between Dynamic Web Scripting Languages. V: Kannan, R., Andres, F. (ur.), *Data Engineering and Management: Lecture Notes in Computer Science*. Second International Conference ICDEM 2010, Tiruchirappalli, 29.–31. julij 2010. Berlin: Springer, 2012, str. 288–295. Dostopno na: https://link.springer.com/chapter/10.1007%2F978-3-642-27872-3_43 [14. 7. 2017].
- [23] Sharma, M. Web Development Technology-PHP. How It Is Related To Web Development Technology ASP.NET. *International Journal of Scientific & Technology Research*, 4, (2015), 1, str. 23–24. Dostopno na: <https://doaj.org/article/28a3ca4f9a66496aa6dc8d49951c054f> [14. 7. 2017].
- [24] Titchkosky, L., Arlitt, M., Williamson, C. A performance comparison of dynamic web technologies. *ACM SIGMETRICS Performance Evaluation Review*, 31, (2003), str. 2–11. Dostopno na: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.93.5875> [15. 7. 2017].
- [25] Trent, S., Tatsubori, M., Suzumura, T., Tozawa, A., Onodera, T. Performance Comparison of PHP and JSP as Server-Side Scripting Languages. V: Issarny, V., Schantz, R. (ur.), *Middleware 2008: Lecture Notes in Computer Science*. ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, 1.–5. december 2008. Berlin: Springer, 2008, str. 164–182. Dostopno na: https://link.springer.com/chapter/10.1007/978-3-540-89856-6_9 [15. 7. 2017].

- [26] Bounnady, K., Phanthavong, K., Pathoumvanh, S., Sihalath, K. Comparison the processing speed between PHP and ASP.NET. V: *ECTI-CON*. 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Chiang Mai, 28. junij – 1. julij 2016. Chiang Mai: IEEE Computer Society, 2016, str. 1–5. Dostopno na: <http://ieeexplore.ieee.org/document/7561484/> [16. 7. 2017].
- [27] Jailia, M., Kumar, A., Agarwal, M., Sinha, I. Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework. V: *ICTBIG 2016*. International Conference on ICT in Business Industry & Government, Indore, 18.–19. november 2016. Indore: IEEE Computer Society, 2016, str. 1–5. Dostopno na: <http://ieeexplore.ieee.org/document/7892651/> [16. 7. 2017].
- [28] StatCounter GlobalStats. Desktop Operating System Market Share Worldwide. Dostopno na: <http://gs.statcounter.com/os-market-share/desktop/worldwide/#daily-20170601-20170601-bar> [19. 7. 2017].
- [29] W3 Techs. Usage of web servers for websites. Dostopno na: https://w3techs.com/technologies/overview/web_server/all [19. 7. 2017].
- [30] DB-Engines. Ranking. Dostopno na: <https://db-engines.com/en/ranking> [19. 7. 2017].
- [31] Stack Overflow. About. Dostopno na: <https://stackoverflow.com/company> [17. 7. 2017].
- [32] Code Project. What is Code Project? Dostopno na: <https://www.codeproject.com/info/guide.aspx> [17. 7. 2017].
- [33] OWASP Foundation. Top 10 Application Security Risks 2017. Dostopno na: https://www.owasp.org/index.php/Top_10_2017-Top_10 [18. 7. 2017].
- [34] OWASP Foundation. SQL Injection. Dostopno na: https://www.owasp.org/index.php/SQL_Injection [18. 7. 2017].
- [35] OWASP Foundation. Cross-Site Scripting (XSS). Dostopno na: [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)) [18. 7. 2017].

- [36] Khan, A. Top Code Editors And IDE For PHP Development Of 2017. Dostopno na: <https://www.cloudways.com/blog/top-ide-and-code-editors-php-development/> [24. 7. 2017].
- [37] Noeticforce. PHP IDE: The10 Best IDEs for PHP Programmers. Dostopno na: <http://noeticforce.com/best-php-ide-for-programmers-windows-and-mac> [24. 7. 2017].
- [38] FromDev. 15+ Best Node.js IDE's. Dostopno na: <http://www.fromdev.com/2016/11/best-nodejs-ides.html> [24. 7. 2017].
- [39] Mishra, S. Top 10 Best IDEs For Node.js Development. Dostopno na: <http://techcresendo.com/2016/02/02/top-10-best-ides-for-node-js-development/> [24. 7. 2017].
- [40] NPM. Dostopno na: <https://www.npmjs.com/> [21. 7. 2017].
- [41] Microsoft. Windows. Dostopno na: <https://www.microsoft.com/sl-si/windows/get-windows-10> [20. 7. 2017].
- [42] Apple. Mac. Dostopno na: <https://support.apple.com/explore/new-to-mac> [20. 7. 2017].
- [43] The Linux Foundation. What is Linux? Dostopno na: <https://www.linux.com/what-is-linux> [20. 7. 2017].
- [44] Apache Software Foundation. Dostopno na: <https://www.apache.org/free/> [20. 7. 2017].
- [45] Nginx. Dostopno na: <https://www.nginx.com/products/pricing/> [20. 7. 2017].
- [46] Microsoft. Web Server. Dostopno na: <https://www.microsoft.com/web/platform/server.aspx> [20. 7. 2017].
- [47] Stack Overflow. Tagged Questions: PHP. Dostopno na: <https://stackoverflow.com/questions/tagged/php> [23. 7. 2017].
- [48] Stack Overflow. Tagged Questions: Node.js. Dostopno na: <https://stackoverflow.com/questions/tagged/node.js> [23. 7. 2017].
- [49] Code Project. PHP. Dostopno na: <https://www.codeproject.com/search.aspx?q=php> [23. 7. 2017].

- [50] Code Project. Node.js. Dostopno na: <https://www.codeproject.com/search.aspx?q=node.js> [23. 7. 2017].
- [51] OWASP Foundation. PHP Security Cheat Sheet. Dostopno na: https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet [21. 7. 2017].
- [52] NPM. Modul mysql. Dostopno na: <https://www.npmjs.com/package/mysql> [21. 7. 2017].
- [53] NPM. Modul xss. Dostopno na: <https://www.npmjs.com/package/xss> [23. 7. 2017].
- [54] NPM. Modul xss-filter. Dostopno na: <https://www.npmjs.com/package/xss-filters> [23. 7. 2017].
- [55] NPM. Modul x-xss-protection. Dostopno na: <https://www.npmjs.com/package/x-xss-protection> [23. 7. 2017].



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko

IZJAVA O AVTORSTVU IN ISTOVETNOSTI TISKANE IN ELEKTRONSKE OBLIKE ZAKLJUČNEGA DELA

Ime in priimek študent-a/-ke: TJAŠA HERIČKO

Študijski program: MEDIJSKE KOMUNIKACIJE

Naslov zaključnega dela: ANALIZA IN PRIMERJANA UPORABE PHP IN
NODE.JS PRI RAZVOJU spletnih strani

Mentor: MARKO HÖBL

Somentor: /

Podpisan-i/-a študent/-ka TJAŠA HERIČKO

- izjavljam, da je zaključno delo rezultat mojega samostojnega dela, ki sem ga izdelal/-a ob pomoči mentor-ja/-ice oz. somentor-ja/-ice;
- izjavljam, da sem pridobil/-a vsa potrebna soglasja za uporabo podatkov in avtorskih del v zaključnem delu in jih v zaključnem delu jasno in ustrezno označil/-a;
- na Univerzo v Mariboru neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico ponuditi zaključno delo javnosti na svetovnem spletu preko DKUM; sem seznanjen/-a, da bodo dela deponirana/objavljena v DKUM dostopna široki javnosti pod pogoji licence Creative Commons BY-NC-ND, kar vključuje tudi avtomatizirano indeksiranje preko spleta in obdelavo besedil za potrebe tekstovnega in podatkovnega rudarjenja in ekstrakcije znanja iz vsebin; uporabnikom se dovoli reproduciranje brez predelave avtorskega dela, distribuiranje, dajanje v najem in priobčitev javnosti samega izvirnega avtorskega dela, in sicer pod pogojem, da navedejo avtorja in da ne gre za komercialno uporabo;
- dovoljujem objavo svojih osebnih podatkov, ki so navedeni v zaključnem delu in tej izjavi, skupaj z objavo zaključnega dela;
- izjavljam, da je tiskana oblika zaključnega dela istovetna elektronski obliki zaključnega dela, ki sem jo oddal/-a za objavo v DKUM.

Uveljavljam permissivnejšo obliko licence Creative Commons: / (navedite obliko)

Začasna nedostopnost:

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, zaščite poslovnih skrivnosti, varnosti ljudi in narave, varstva industrijske lastnine ali tajnosti podatkov naročnika:

/ (naziv in naslov naročnika/institucije) ne sme biti javno dostopno do / (datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela). To se nanaša na tiskano in elektronsko obliko zaključnega dela.

Temporary unavailability:

Temporary unavailability:

To ensure competition priority, protection of trade secrets, safety of people and nature, protection of industrial property or secrecy of customer's information, the thesis _____ (institution/company name and address) must not be accessible to the public till _____ (delay date of thesis availability to the public must not exceed the period of 3 years after thesis defense). This applies to printed and electronic thesis forms.

Datum in kraj:

MARIBOR, 1. 8. 2017

Podpis študent-a/-ke:



Podpis mentor-ja/-ice: _____
(samo v primeru, če delo ne me biti javno dostopno)

Ime in priimek ter podpis odgovorne osebe naročnika in žig:

(samo v primeru, če delo ne sme biti javno dostopno)



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



IZJAVA O OBJAVI OSEBNIH PODATKOV

Ime in priimek diplomant-a/ magistrant-/-ke: TJAŠA HERIČKO

ID številka: 1002204402

Študijski program: MEDIJSKE KOMUNIKACIJE

Naslov zaključnega dela: ANALIZA IN PRIMERJAVA VPORABE PHP IN
NODE.JS PRI RAZVOJU SPLETNIH STRANI

Mentor/-ica: MARKO HÖBL

Somentor/-ica: /

Podpisan-i/-a izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto zaključka študija, naslov zaključnega dela) na spletnih straneh Univerze v Mariboru in v publikacijah Univerze v Mariboru.

Datum in kraj:

MARIBOR, 1. 8. 2017

Podpis diplomant-a/magistrant-a/-ke:

Heričko



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



IZJAVA O USTREZNOSTI ZAKLJUČNEGA DELA

Podpisani mentor/-ica : MARKO HÖBL
(ime in priimek mentor-ja/-ice)

In somentor/-ica (eden ali več, če obstajajo): _____
(ime in priimek somentor-ja/-ice)

Izjavlja-m/-va/-mo, da je študent/-ka

Ime in priimek: TJAŠA HERIČKO, ID številka: 1002204402

vpisna številka: EA083604, na študijskem programu:

MEDISKE KOMUNIKACIJE

Izdelal/-a zaključno delo z naslovom:

ANALIZA IN PRIMERJAVA UPORABE PHP IN NODE.JS PRI RAZVOJU
SPLETNIH STRAN

(naslov zaključnega dela v slovenskem jeziku)

v skladu z odobreno temo zaključnega dela, navodili o pripravi zaključnih del in mojimi (najinimi/našimi) navodili.

Preveril/-a/-i in pregledal/-a/-i sem/sva/smo poročilo o preverjanju podobnosti vsebin z drugimi deli (priloga) in potrjujem/potrjujeva/potrjujemo, da je zaključno delo ustrezno.

Datum in kraj: 8.8.2017,
MARIBOR

Podpis mentor-ja/-ice: 

Datum in kraj:

Podpis somentor-ja/-ice (če obstaja):

Priloga:

- Poročilo o preverjanju podobnosti vsebin z drugimi deli.