

VIŠJA STROKOVNA ŠOLA ACADEMIA

MARIBOR

**PRIMERJAVA UPORABNE VREDNOSTI
OGRODJA VUE
IN KNJIŽNICE REACTJS**

Kandidat: Rok Krajnc

Vrsta študija: študent izrednega študija

Študijski program: Informatika

Mentor predavatelj: mag. Ervin Schaff

Mentor v podjetju: Petra Munda inž. inf.

Lektor: Nika Vremšak, univ. dipl. slov. in univ. dipl. komp.

Maribor, 2024

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Podpisani Rok Krajnc sem avtor diplomskega dela z naslovom *Primerjava uporabne vrednosti ogrodja Vue in knjižnice ReactJS*, ki sem ga napisal pod mentorstvom mag. Ervina Schaffa.

S svojim podpisom zagotavljam, da:

- je predloženo delo izključno rezultat mojega dela,
- sem poskrbel, da so dela in mnenja drugih avtorjev, ki jih uporabljam v predloženi nalogi, navedena oz. citirana skladno s pravili Višje strokovne šole Academia Maribor,
- se zavedam, da je plagiatorstvo – predstavljanje tujih del oz. misli kot moje lastne kaznivo po Zakonu o avtorski in sorodnih pravicah (Uradni list RS, št. 16/07 – uradno prečiščeno besedilo, 68/08, 110/13, 56/15 in 63/16 – ZKUASP); prekršek pa podleže tudi ukrepom Višje strokovne šole Academia Maribor skladno z njenimi pravili,
- skladno z 32.a členom ZASP dovoljujem Višji strokovni šoli Academia Maribor objavo diplomskega dela na spletnem portalu šole.

Maribor, februar 2024

Podpis študenta:

ZAHVALA

Za številne nasvete, komentarje, stalno razpoložljivost ter vložen čas, trud in pomoč pri izdelavi diplomskega dela se iskreno zahvaljujem mentorju na šoli mag. Ervinu Schaffu in mentorici v podjetju Petri Munda, inž. inf..

Hvala tudi vsem ostalim na Akademii, ki so s svojim časom, delom in dobro voljo na kakršenkoli način prispevali k izdelavi diplomskega dela.

Posebno zahvalo namenjam Urški in Reneju, ki sta vedno našla besedo vzpodbude in me tekom celotnega postopka bodrila, ohranjala osredotočenega na delo in se z mano poveselila, kadar sem zaključil kakšen večji sklop ali razvozlal posebej zapleten problem.

Za podporo in včasih prepotrebno vzpodbudo se zahvaljujem tudi ostalim članom družine.

POVZETEK

Obstaja veliko tehnologij za razvoj *front-end* dela spletnih aplikacij. Diplomsko delo se ukvarja s knjižnico ReactJS in ogrodjem Vue z namenom ugotovitve, kateri ima višjo uporabno vrednost, kar bi podjetjem, kolegom razvijalcem in meni olajšalo izbiro med njima. Narava takšne raziskave zahteva izdelavo dveh primerljivih spletnih aplikacij, vsake z drugim orodjem.

Najprej sem namestil in pripravil integrirano razvojno okolje Visual Studio Code, v katerem sem aplikaciji lahko ustvaril. Po tem sem si zamislil koncept aplikacij in narisal skico ter spisal načrt delovanja, v katerem je opredeljeno, kaj morata aplikaciji vsebovati in na kakšen način bosta delovali. Naslednji korak je bila izdelava samih aplikacij. V nadaljevanju sem primerjal, katerega orodja se je moč hitreje naučiti glede na dostop do učnega gradiva in spletnih tečajev ter na njuno skupnost in kompleksnost. Izkazalo se je, da se razvijalec lahko hitreje nauči uporabljati knjižnico ReactJS. Sledila je raziskava tržnega položaja, kjer sem na podlagi primerjav razširjenosti in priljubljenosti, možnosti zaposlitve ter skupnosti ugotovil, da je ReactJS v boljšem položaju. Potem sem začel meritve obremenjenosti sistemov, za kar sem meril porabo pomnilnika in obremenjenost centralne procesne enote. Za tem sem izvedel meritve hitrosti in odzivnosti. Aplikaciji sta lokalni na mojem računalniku, zato sem bil gostitelj in hkrati odjemalec, kar je zahtevalo, da sem za meritve uporabil različna orodja (Windows Performance Monitor in Google Chrome devTools). S poglobljeno analizo sem ugotovil, da ogrodje Vue porabi več sredstev in tako bolj obremeni sistem.

Meritve hitrosti so bile namenjene ugotavljanju, katera aplikacija se hitreje naloži do različnih faz uporabnosti. Izkazalo se je, da pri osnovni količini podatkov aplikacija, izdelana z ReactJS, potrebuje dlje do faze interaktivnosti, medtem ko pri povečani količini podatkov potrebuje manj časa kot aplikacija, izdelana z Vue. Na podlagi meritev obremenjenosti sem tudi ugotovil, da ima ReactJS aplikacija večji potencial za rast, v kolikor se narava aplikacije ne spremeni, ampak se spremeni zgolj količina podatkov, kar je hkrati privedlo do spoznanja, da ReactJS aplikacija učinkovitejše dela z *Big Data*. Na podlagi vseh meritev in ugotovitev sem spoznal, da ima knjižnica ReactJS večjo uporabno vrednost kot ogrodje Vue.

Ključne besede: razvoj spletnih aplikacij, front-end, ReactJS, Vue, primerjava.

ABSTRACT

Comparison of the useful value of the ReactJS library and the Vue framework

There are numerous technologies available for developing the front-end of web applications. This thesis focuses on the ReactJS library and the Vue framework with the aim of determining which one provides higher utility. Such a comparison is valuable for businesses, fellow developers, and myself for choosing among them. The nature of this research necessitates the creation of two comparable web applications, each built with a different tool. The initial step involved setting up and preparing the Visual Studio Code integrated development environment, where I could create the applications. Subsequently, I conceptualized the applications, drafted sketches, and formulated an operational plan defining the content and functionality of the applications. The next stage encompassed the actual development of the applications. Following the development, I compared the ease of learning each tool based on the availability of learning materials and online courses, as well as their communities and complexity. I found out that developers can learn using the ReactJS library more faster. After this came a market research segment where, through comparisons of popularity, employment opportunities, and community support, ReactJS emerged as the stronger contender. I then proceeded with system load measurements, gauging memory consumption and CPU load. Additionally, I conducted speed and responsiveness measurements. Since the applications are local on my computer, I served as both the host and the client, this necessitated the use of different tools for measurements (Windows Performance Monitor and Google Chrome DevTools). My analysis revealed that the Vue framework consumes more resources, thereby placing a heavier burden on the system. With the speed measurements I tried to find out which application loaded faster into various phases of usability. It was found that, at the baseline data volume, the ReactJS-built application takes longer to reach the interactivity phase, while at an increased data volume, it requires less time than the one in Vue. Based on the measurements of the CPU load and the memory consumption, I established that the ReactJS application has greater potential for growth if the nature of the application remains unchanged and it only grows in data size. This realization led to the conclusion that the ReactJS application operates more efficiently with Big Data. In summary, considering my findings, I conclude that the ReactJS library provides higher utility compared to the Vue framework. This conclusion is valuable for informed decision-making in selecting the appropriate technology for front-end web development.

Key Words: Development of Web Applications, Front-end, ReactJS, Vue, Comparison.

KAZALO VSEBINE

1	UVOD	15
1.1	OPIS PODROČJA IN OPREDELITEV PROBLEMA	15
1.2	NAMEN, CILJI IN OSNOVNE TRDITVE	16
1.3	PREDPOSTAVKE IN OMEJITVE	17
1.4	UPORABLJENE RAZISKOVALNE METODE	17
1.4.1	<i>Pregled literature</i>	17
1.4.2	<i>Meritve</i>	18
1.4.3	<i>Primerjalna metoda</i>	18
2	BIG DATA	19
2.1	DEFINICIJA	19
2.2	O BIG DATA	19
2.3	OPIS KRITERIJEV BIG DATA	20
2.3.1	<i>Obsežnost:</i>	20
2.3.2	<i>Hitrost:</i>	20
2.3.3	<i>Raznolikost:</i>	21
2.4	ŽIVLJENJSKI CIKEL PODATKOV BIG DATA.....	21
3	IZDELAVA SPLETNE APLIKACIJE	23
3.1	NAMESTITEV IN PRIPRAVA INTEGRIRANEGA RAZVOJNEGA OKOLJA.....	23
3.2	SKICA IN NAČRT DELOVANJA SPLETNE APLIKACIJE	23
3.2.1	<i>Skica</i>	23
3.2.2	<i>Načrt delovanja</i>	24
3.3	IZDELAVA SESTAVNIH DELOV SPLETNE APLIKACIJE.....	24
3.3.1	<i>ReactJS</i>	24
3.3.2	<i>Vue</i>	29
4	PRIMERJAVA ZAHTEVNOSTI UČENJA UPORABE ORODJA	31
4.1	DOSTOP DO UČNEGA GRADIVA	31
4.1.1	<i>Tečaji na spletu</i>	31
4.1.2	<i>Uradna dokumentacija</i>	35
4.1.3	<i>Knjige in E-knjige</i>	37
4.2	SKUPNOST	39
4.2.1	<i>Stack Overflow, Reddit</i>	39
4.2.2	<i>Facebook</i>	41
4.3	KOMPLEKSNOŠT	44
4.3.1	<i>Sintaksa</i>	44

4.3.2	Virtual DOM (virtualni objektni model dokumenta)	44
4.3.3	Struktura komponent.....	45
4.4	UGOTOVITVE	46
5	TRŽNI POLOŽAJ.....	47
5.1	RAZŠIRJENOST IN PRILJUBLJENOST	47
5.1.1	Spletne strani izdelane z Vue in z ReactJS.....	47
5.1.2	Priljubljenost med razvijalci.....	49
5.1.3	Priljubljenost med večjimi podjetji.....	49
5.2	MOŽNOST ZAPOSLOITVE	50
5.2.1	Glassdoor	51
5.2.2	Indeed	52
5.2.3	LinkedIn	54
5.2.3.1	Slovenija	54
5.2.3.2	Evropska unija.....	56
5.2.3.3	Združene države Amerike	58
5.2.4	Zavod Republike Slovenije za zaposlovanje	60
5.2.5	Pregled	61
5.3	SKUPNOST	61
5.4	UGOTOVITVE	62
6	PRIMERJAVA SPLETNIH APLIKACIJ IZDELANIH V VUE IN REACTJS	63
6.1	OBREMENJENOST SISTEMOV GOSTITELJA IN ODJEMALCA	63
6.1.1	Gostitelj	63
6.1.1.1	Priprava orodja in postopek merjenja.....	63
6.1.1.2	Meritve	64
6.1.1.2.1	Osnovna količina podatkov (1x).....	64
6.1.1.2.1.1	Pomnilnik	64
6.1.1.2.1.2	Primerjava	64
6.1.1.2.1.3	Centralna Procesna Enota.....	65
6.1.1.2.1.4	Primerjava	65
6.1.1.2.2	Povečana količina podatkov (10x)	66
6.1.1.2.2.1	Pomnilnik	66
6.1.1.2.2.2	Primerjava	66
6.1.1.2.2.3	Centralna Procesna Enota.....	67
6.1.1.2.2.4	Primerjava	67
6.1.2	Odjemalec.....	68
6.1.2.1	Priprava orodja in postopek merjenja.....	68
6.1.2.2	Meritve	69
6.1.2.2.1	Osnovna količina podatkov (1x).....	69

6.1.2.2.1.1	Pomnilnik in CPE med nalaganjem Vue	69
6.1.2.2.1.2	Pomnilnik in CPE med nalaganjem ReactJS	69
6.1.2.2.1.3	Pomnilnik in CPE v mirovanju	70
6.1.2.2.1.4	Primerjava	70
6.1.2.2.2	<i>Povečana količina podatkov (10x)</i>	71
6.1.2.2.2.1	Pomnilnik in CPE med nalaganjem Vue	71
6.1.2.2.2.2	Pomnilnik in CPE med nalaganjem ReactJS	71
6.1.2.2.2.3	Pomnilnik in CPE v mirovanju	72
6.1.2.2.2.4	Primerjava	72
6.2	HITROST IN ODZIVNI ČAS.....	72
6.2.1	<i>Priprava orodja in postopek merjenja</i>	73
6.2.2	<i>Meritve</i>	73
6.2.3	<i>Osnovna količina podatkov (1x)</i>	73
6.2.3.1	Vue.....	73
6.2.3.2	ReactJS.....	74
6.2.3.3	Primerjava	74
6.2.4	<i>Povečana količina podatkov (10x)</i>	74
6.2.4.1	Vue.....	74
6.2.4.2	ReactJS.....	75
6.2.4.3	Primerjava	75
6.3	ANALIZA MERITEV	76
6.3.1	<i>Gostitelj</i>	76
6.3.1.1	Pomnilnik.....	76
6.3.1.2	CPE	76
6.3.2	<i>Odjemalec</i>	77
6.3.2.1	Pomnilnik.....	77
6.3.2.2	CPE	77
6.3.2.3	Hitrost in odzivni čas	77
6.4	POTENCIAL ZA RAST PROJEKTA	77
6.5	BIG DATA.....	78
7	SKLEP	79
8	VIRI, LITERATURA.....	81

KAZALO SLIK

SLIKA 1: SKICA	23
SLIKA 2: VMESNIK CHARACTERDATATYPE	25
SLIKA 3: LAYOUT.TSX	25
SLIKA 4: CHARACTER.TSX	26
SLIKA 5: CHARACTERTABLE.TSX	27
SLIKA 6: RICK AND MORTY.JSON	28
SLIKA 7: NAMESTITEV VUE	29
SLIKA 8: SPLETNI TEČAJI UDEMY VUE	31
SLIKA 9: SPLETNI TEČAJI UDEMY REACTJS	32
SLIKA 10: SPLETNI TEČAJI COURSEVA VUE	33
SLIKA 11: SPLETNI TEČAJI COURSEVA REACTJS	34
SLIKA 12: URADNA DOKUMENTACIJA VUE 1	35
SLIKA 13: URADNA DOKUMENTACIJA VUE 2	36
SLIKA 14: URADNA DOKUMENTACIJA REACTJS 1	36
SLIKA 15: URADNA DOKUMENTACIJA REACTJS 2	37
SLIKA 16: COBISS VUE	38
SLIKA 17: COBISS REACTJS	38
SLIKA 18: STACK OVERFLOW VUE	39
SLIKA 19: STACK OVERFLOW REACTJS	40
SLIKA 20: REDDIT VUE	40
SLIKA 21: REDDIT REACTJS	41
SLIKA 22: FACEBOOK VUE	42
SLIKA 23: FACEBOOK REACTJS	43
SLIKA 24: ŠTEVILO SPLETNIH STRANI IZDELANIH Z VUE	47
SLIKA 25: ŠTEVILO SPLETNIH STRANI IZDELANIH Z REACTJS	48
SLIKA 26: GLASSDOOR - MOŽNOST ZAPOSLOTVE VUE	51
SLIKA 27: GLASSDOOR - MOŽNOST ZAPOSLOTVE REACTJS	51
SLIKA 28: INDEED - MOŽNOST ZAPOSLOTVE VUE	52
SLIKA 29: INDEED - MOŽNOST ZAPOSLOTVE REACTJS	53
SLIKA 30: LINKEDIN SLOVENIJA - MOŽNOST ZAPOSLOTVE VUE	54
SLIKA 31: LINKEDIN SLOVENIJA - MOŽNOST ZAPOSLOTVE REACTJS	55
SLIKA 32: LINKEDIN EVROPSKA UNIJA - MOŽNOST ZAPOSLOTVE VUE	56
SLIKA 33: LINKEDIN EVROPSKA UNIJA - MOŽNOST ZAPOSLOTVE REACTJS	57
SLIKA 34: LINKEDIN ZDA - MOŽNOST ZAPOSLOTVE VUE	58
SLIKA 35: LINKEDIN ZDA - MOŽNOST ZAPOSLOTVE REACTJS	59
SLIKA 36: ZAVOD ZA ZAPOSLOVANJE SLO - MOŽNOST ZAPOSLOTVE VUE	60

SLIKA 37: ZAVOD ZA ZAPOSLOVANJE SLO - MOŽNOST ZAPOSLOTITVE REACTJS.....	60
SLIKA 38: OBREMENJENOST POMNILNIKA IN CPE PRI ODJEMALCU – VUE	69
SLIKA 39: OBREMENJENOST POMNILNIKA IN CPE PRI ODJEMALCU – REACTJS.....	69
SLIKA 40: OBREMENJENOST POMNILNIKA IN CPE PRI ODJEMALCU V MIROVANJU	70
SLIKA 41: OBREMENJENOST POMNILNIKA IN CPE PRI ODJEMALCU - VUE (10X KOLIČINA PODATKOV)	71
SLIKA 42: OBREMENJENOST POMNILNIKA IN CPE PRI ODJEMALCU - REACTJS (10X KOLIČINA PODATKOV)	71
SLIKA 43: OBREMENJENOST POMNILNIKA IN CPE PRI ODJEMALCU V MIROVANJU (10X KOLIČINA PODATKOV)	72
SLIKA 44: ČAS NALAGANJA VUE.....	73
SLIKA 45: ČAS NALAGANJA REACTJS	74
SLIKA 46: ČAS NALAGANJA VUE (10X KOLIČINA PODATKOV).....	75
SLIKA 47: ČAS NALAGANJA REACTJS (10X KOLIČINA PODATKOV)	75

KAZALO TABEL

TABELA 1: UGOTOVITVE ZAHTEVNOSTI UČENJA UPORABE ORODJA	46
TABELA 2: PREGLED SKUPNEGA ŠTEVILA AKTIVNIH SPLETNIH MEST ORODIJ	48
TABELA 3: PRILJUBLJENOST ORODIJ MED RAZVIJALCI RAZLIČNEGA NIVOJA ZNANJA	49
TABELA 4: UPORABA ORODIJ PRI VEČJIH PODJETJIH	49
TABELA 5: PREGLED MOŽNOSTI ZAPOSLOTVE	61
TABELA 6: PRIMERJAVA OBREMENJENOSTI POMNILNIKA PRI GOSTITELJU	64
TABELA 7: PRIMERJAVA OBREMENJENOSTI CPE PRI GOSTITELJU.....	65
TABELA 8: PRIMERJAVA OBREMENJENOSTI POMNILNIKA PRI GOSTITELJU (10X KOLIČINA PODATKOV).....	66
TABELA 9: PRIMERJAVA OBREMENJENOSTI CPE PRI GOSTITELJU (10X KOLIČINA PODATKOV)	67
TABELA 10: PRIMERJAVA OBREMENJENOSTI POMNILNIKA IN CPE PRI ODJEMALCU V MIROVANJU	70
TABELA 11: PRIMERJAVA OBREMENJENOSTI POMNILNIKA IN CPE PRI ODJEMALCU V MIROVANJU (10X KOLIČINA PODATKOV)	72
TABELA 12: PRIMERJAVA ČASA DO FAZ UPORABNOSTI	74
TABELA 13: PRIMERJAVA ČASA DO FAZ UPORABNOSTI (10X KOLIČINA PODATKOV)	75

KAZALO GRAFOV

GRAF 1: OBREMENJENOST POMNILNIKA PRI GOSTITELJU	64
GRAF 2: OBREMENJENOST CPE PRI GOSTITELJU	65
GRAF 3: OBREMENJENOST POMNILNIKA PRI GOSTITELJU (10X KOLIČINA PODATKOV)	66
GRAF 4 : OBREMENJENOST CPE PRI GOSTITELJU (10X KOLIČINA PODATKOV)	67

RAZLAGA STROKOVNIH IZRAZOV IN KRATIC

PID (angl. Process Identifier) je številka, ki jo uporablja večina operacijskih sistemov, kot so Unix, macOS in Windows, da edinstveno identificirajo aktivni proces.

CPE ali CPU (angl. Central Processing Unit) oz. centralna procesna enota je osrednja komponenta računalnika, odgovorna za izvajanje navodil programske opreme.

IDE (angl. Integrated Development Environment) oz. integrirano razvojno okolje je programska oprema, ki razvijalcem omogoča pisanje, preizkušanje in odpravljanje napak programske opreme v enem okolju.

DOM (angl. Document Object Model) oz. objektni model dokumenta je vmesnik, ki naredi reprezentacijo html ali xml dokumenta, v katerem so vsi elementi (node) organizirani v drevesno strukturo.

V-DOM je virtualna reprezentacija DOM-a.

Node se nanaša na posamezen objekt ali element v strukturi drevesa, kot je DOM. Vsak node predstavlja del dokumenta, na primer element, atribut ali besedilo, in ga je mogoče programsko manipulirati.

V-Node je enak kot node, le da se nanaša na posamezen objekt ali element v strukturi virtualnega drevesa, kot je V-DOM.

NPM (angl. Node Package Manager) je privzeti upravitelj paketov za Node.js, ki omogoča preprosto namestitvev in upravljanje zunanjih modulov.

JSON (angl. JavaScript Object Notation) je lahka oblika zapisa podatkov, ki se pogosto uporablja za prenos podatkov med strežniki in spletnimi aplikacijami.

Typescript je odprtokodni jezik za programiranje, ki je zasnovan na JavaScriptu, vendar vključuje dodatne funkcije, kot so tipi podatkov in razširjena sintaksa.

Prop je kratko ime za lastnost (angl. Property) v programiranju, zlasti v kontekstu komponentnega programa, kjer predstavlja vrednost ali podatek, ki se prenaša med komponentami.

Pre-procesor je program, ki predprocesira (obdeluje pred glavnim procesom) izvorno kodo, običajno s ciljem optimizacije ali izvajanja določenih sprememb pred končno fazo prevajanja ali interpretacije.

Front-end v programiranju se nanaša na vidik razvoja programske opreme, ki je odgovoren za uporabniški vmesnik in interakcijo z uporabnikom. Front-end zajema vse, kar uporabnik vidi.

Back-end del programske opreme, ki deluje na strežniku in je odgovoren za obdelavo podatkov, poslovno logiko ter komunikacijo z bazami podatkov.

Vue ali VueJS je ogrodje za izdelavo front-end dela spletnih aplikacij.

ReactJS je knjižnica za izdelavo front-end dela spletnih aplikacij.

JSX je razširitev JavaScript sintakse, ki omogoča vstavljanje HTML podobne kode neposredno v JavaScript. Pogosto se uporablja v ogrodjih, kot sta ReactJS in Vue za olajšanje deklarativnega opisa uporabniških vmesnikov.

FCP (angl. First Contentful Paint) označuje čas, ko se prvi vizualni element pojavi na zaslonu med nalaganjem spletne strani.

LCP (angl. Largest Contentful Paint) označuje čas, ko se na zaslonu prikaže največji vizualni element med nalaganjem spletne strani.

TTI (angl. Time to Interactivity) označuje čas, ko spletna stran postane interaktivna in odzivna na uporabnikove akcije.

1 UVOD

1.1 Opis področja in opredelitev problema

Diplomsko delo se ukvarja s področjem razvoja spletnih aplikacij. Živimo v dobi digitalizacije in interneta, v kateri so spletne aplikacije nepogrešljiv del družbe. Spletne aplikacije so v osnovi sestavljene iz dveh delov, *front-end* dela in *back-end* dela. Ta raziskava se osredotoča na *front-end* del. Trenutno obstaja veliko število različnih pristopov in tehnologij, kot so ogrodja in knjižnice, ki nam omogočajo učinkovitejši razvoj *front-end* dela spletnih aplikacij in to število se vsako leto večja. Velika izbira tehnologij je dvorezni meč, po eni strani ima to prednost, da lahko podjetja in razvijalci spletnih aplikacij izberejo tehnologijo, ki je najbolj prilagojena za njihove projekte, po drugi strani pa sta potrebna čas in trud, da se razvijalec določene tehnologije nauči uporabljati, kar pomeni, da mora smiselno in previdno izbrati, v katero se bo podal, saj se ne more naučiti vseh. Zaradi tega razloga morajo biti tudi podjetja previdna, katero tehnologijo izberejo za svoje projekte, saj ni produktivno izbrati na primer ogrodja, ki je za dani projekt optimalno, a ga uporablja izredno malo število razvijalcev, ali pa za vsak projekt izbrati drugo tehnologijo in s tem močno obremeniti razvijalce v podjetju. Po drugi strani pa se poraja vprašanje, ali je smiselno storiti nasprotno in izbrati samo eno tehnologijo ter vse projekte, ne glede na njihovo naravo in potrebe razvijati v njej. Ta dilema je posebej velika med podjetji z občutno različnimi projekti in med razvijalci, ki šele začenjajo svojo pot in področja še ne poznajo dobro. Problem, s katerim se torej ukvarja to diplomsko delo, je, katero ogrodje je podjetjem bolj uporabno. Rešitev tega problema pa tudi odgovori na vprašanje, katere tehnologije se je razvijalcu bolj smiselno in uporabno naučiti. V raziskavi želim tudi ugotoviti, katera tehnologija je učinkovitejša za *Big Data*. Tovrstne tehnologije so zelo obširne, kot povedano, jih je zamudno in zahtevno osvojiti do takšne mere, da nam postanejo uporabne, zato sem se v tej raziskavi omejil zgolj na dve, ki sta med seboj primerljivi: knjižnica ReactJS in ogrodje Vue. Za ti dve orodji sem se odločil, ker sta zaradi svoje učinkovitosti med razvijalci najpogostejše uporabljene. Namen obeh orodij je enak, svojo funkcijo pa opravljata na različne načine, zato se za posamični projekt praviloma uporablja zgolj eno ali drugo in ne obe hkrati. To pomeni, da sta tako podjetje kot razvijalec prisiljena izbrati, katero uporabiti.

1.2 Namen, cilji in osnovne trditve

Namen tega diplomskega dela je objektivna primerjava uporabne vrednosti dveh priljubljenih konkurenčnih tehnologij za razvijanje spletnih aplikacij na različnih področjih s ciljem, da podjetjem, kolegom razvijalcem in sebi olajšam izbiro med izbranimi tehnologijama za razvijanje *front-end* dela spletnih aplikacij. Naslednji cilj, ki sem si ga zastavil, pa je, da postanem tekom raziskovanja in razvijanja izdelka bolj učinkovit v uporabi JavaScript-a, HTML-ja, CSS-ja, knjižnice ReactJS in ogrodja Vue.

V raziskovalnem delu iščem odgovore na naslednja vprašanja:

- Ali se razvijalec hitreje nauči uporabljati ogrodje Vue ali knjižnico ReactJS?
- Katero orodje ima večjo skupnost?
- Kako se orodji razlikujeta po intuitivnosti in kompleksnosti sintakse?
- Ali so na spletu dostopni primerljivi učni pripomočki in gradiva za ogrodje Vue in knjižnico ReactJS?
- Ali spletna aplikacija, izdelana z ogrodjem Vue, porabi manj sredstev kot primerljiva spletna aplikacija, izdelana s knjižnico ReactJS?
- Ali je tržni položaj ogrodja Vue boljši kot tržni položaj knjižnice ReactJS?
- Ali je uporaba podatkov *Big Data* učinkovitejša ob uporabi ogrodja Vue ali ob uporabi knjižnice ReactJS?

Raziskava si prizadeva potrditi ali ovreči naslednje hipoteze:

- Razvijalec se hitreje nauči delati z ogrdjem Vue kot s knjižnico ReactJS (H1).
- Spletna aplikacija, izdelana z ogrodjem Vue za delovanje potrebuje manj sredstev kot primerljiva spletna aplikacija, izdelana s knjižnico ReactJS (H2).
- Knjižnica ReactJS ima boljši tržni položaj kot ogrodje Vue (H3).
- Uporaba podatkov *Big Data* je učinkovitejša ob uporabi knjižnice ReactJS kot ob uporabi ogrodja Vue (H4).

1.3 *Predpostavke in omejitve*

Knjižnice ReactJS in ogrodja Vue pred začetkom izdelave diplomskega dela nisem poznal, ker ju nisem nikoli uporabil. Posledično sem bil omejen glede tega, do kakšne mere sem obe tehnologiji v danem času lahko usvojil. To se na izdelkih izraža tako, da nimata obsežne funkcionalnosti, interaktivnosti, animacij in podobnih naprednih procesov. Prva predpostavka je, da mora moje razumevanje orodij dosegati zadosten nivo za izdelavo razmeroma preprostih spletnih aplikacij s knjižnico ReactJS in z ogrodjem Vue. Naslednja predpostavka je, da morata obe aplikaciji kljub razlikam v sintaksi, strukturi komponent in pristopu biti dovolj podobni, da so mogoče objektivne in smiselne primerjave med njuno učinkovitostjo. To nas pripelje do naslednje omejitve, ki je stabilnost omrežja in sistema. Za natančne meritve učinkovitosti, hitrosti in odzivnosti je potrebno stremeti k temu, da se zagotovijo enaki ali vsaj čim bolj podobni pogoji za obe aplikaciji, kar je v mojem primeru bilo le delno mogoče. Sredstva, ki so mi bila na voljo za izvajanje meritev, so bila neprofesionalna, posledično je prihajalo v delih sistema, predvsem pa v omrežju, do nihanj, kar je vplivalo na natančnost rezultatov meritev. Naslednja omejitev je, da je isti računalnik hkrati gostitelj in odjemalec, kar pomeni, da je za meritve potrebno uporabiti različna orodja. Zadnja predpostavka pa je, da se uporabna vrednost orodja oceni na podlagi obremenjenosti pomnilnika in CPE med uporabo aplikacije, časa nalaganja aplikacije, tržnega položaja orodja, potenciala za rast projekta in sposobnosti učinkovitosti orodja pri upravljanju z *Big Data*. Faktorjev, ki vplivajo na uporabno vrednost orodja, je več, med drugim tudi obremenjenost omrežja med uporabo aplikacije, ampak se je bilo v izogib presega obširnosti diplomskega dela potrebno omejiti na najpomembnejše.

1.4 *Uporabljene raziskovalne metode*

Uporabili smo dva raziskovalna pristopa: pregled literature in primerjalno metodo.

1.4.1 *Pregled literature*

Zadosten nivo znanja in razumevanja knjižnice ReactJS in ogrodja Vue za izdelavo zadanih spletnih aplikacij sem dosegel primarno zahvaljujoč uradni dokumentaciji obeh orodij na njihovih spletnih straneh ((React: Quick Start, 2023) in (Vue.js: Introduction, 2023)). Uporabe orodij za meritve sem se prav tako naučil iz uradnih dokumentacij na njihovih spletnih straneh: Google Chrome devTools (Google Chrome devTools , 2023) in (Windows Performance Monitor,

2023). Podatke o *Big Data* sem črpal primarno s spletne strani Understanding Big Data Processing: 2024's Ultimate Guide. (Understanding Big Data Processing: 2024's Ultimate Guide , 2023). Ti viri so bili ključnega pomena za izdelavo diplomskega dela.

1.4.2 Meritve

Za objektivno primerjavo uporabne vrednosti orodij je bilo na aplikacijah, ki sem ju z njuno pomočjo izdelal, potrebno izvesti meritve več parametrov. Meril sem obremenjenost pomnilnika in centralne procesne enote tako pri gostitelju kot pri odjemalcu. Meril sem tudi hitrost nalaganja. Meritve sem ponovil še pri povečani količini podatkov.

1.4.3 Primerjalna metoda

Za primerjavo uporabne vrednosti med knjižnico ReactJS in ogrodjem Vue je bila primerjalna metoda nepogrešljiva. V raziskovalnem delu sem namreč primerjal razne faktorje, ki vplivajo na to, katero orodje se razvijalec hitreje nauči uporabljati. Primerjal sem tudi tržni položaj obeh orodij. Nato sem v analizi primerjal podatke obremenjenosti sistemov gostitelja in odjemalca ter hitrosti nalaganja, ki sem jih pridobil z zgoraj omenjenimi meritvami, kar je privedlo do ugotovitev o učinkovitosti orodij.

2 BIG DATA

2.1 Definicija

Izbral sem definicijo, ki jo nudi Gartnerjev IT slovar *Glossary: Big Data*. Omenjeni slovar vsebuje definicije in opise različnih konceptov in izrazov s področja informacijske tehnologije. Ta vir zagotavlja priznano svetovalno in raziskovalno podjetje na področju informacijske tehnologije in poslovanja Gartner, Inc.

»Big data so informacijska sredstva visokega obsega, visoke hitrosti in/ali visoke raznolikosti, ki zahtevajo stroškovno učinkovite, inovativne oblike obdelave informacij, ki omogočajo izboljšano razumevanje, sprejemanje odločitev in avtomatizacijo procesov.« (Gartner Glossary: Big Data, 2023).

2.2 O Big Data

Prvi dokumentiran pojav izraza »Big Data« je v članku znanstvenikov Nase iz leta 1997, kjer opisujejo težavo pri vizualizaciji velikih sklopov podatkov. Leta 2001 je analitik Doug Laney definiral tri kriterije (obsežnost, hitrost in raznolikost), po katerih še danes sklope podatkov kategoriziramo v *Big Data*. Zaradi digitalizacije tako rekoč vsega v našem vsakdanjem življenju, hkrati pa zaradi naraščajočega števila spletnih uporabnikov, se generira eksponentno več podatkov, ki morajo biti beleženi, kar prispeva k rasti *Big Data*. V knjigi *Big Data: Concepts, Technology, and Architecture* avtorji navajajo, da je bilo 80 % vseh podatkov, ki so danes na voljo, generiranih v zadnjih nekaj letih, ter da dnevno ustvarimo 2,5 kvintilijona ((2.500.000.000.000.000.000 ali 2.5×10^{18}) bajtov podatkov. Knjiga je izšla leta 2021, tako da je razumno domnevati, da je to število zdaj še nekoliko večje. Masivne količine podatkov generira več milijonov sporočil, vnosov, slik in odzivov na družbenih omrežjih, kot je Facebook, ali komunikacijskih aplikacijah, kot je WhatsApp. Poleg tega se vsako minuto na razne spletne platforme, kot sta Youtube in Tiktok, naloži več ur video posnetkov. Analize takšnih količin podatkov so dragocene raznim organizacijam, saj jim dajejo uporabne informacije za boljše poslovanje. Med drugim omogočajo izdelavo potrošniških profilov uporabnikov, ugotavljajo dejavnike in okoliščine, pod katerimi je verjetnost, da bo uporabnik kupil določen izdelek ali storitev, večja, olajšajo sledenje tržnim spremenljivkam, trendom in podobno. To organizacijam posledično omogoča prilagajanje izdelka ali storitve, ciljano oglaševanje na podlagi na primer podatkov o preteklih nakupih in zgodovini brskanja ter

podobne strategije za povečanje konkurenčnosti na trgu. Podatki morajo biti obdelani v določenem času oz. z določeno hitrostjo, da so še relevantni in da še nudijo uporabne vpoglede, zato je prišlo ob vedno večji kompleksnosti in količini zajetih podatkov do težav pri njihovi učinkoviti analizi s tradicionalnimi sistemi za upravljanje z bazami podatkov. Prišlo je do bodisi zmanjšanja zmogljivosti in povečanja zakasnitev ali pa dragega nadgrajevanja strojne opreme z novimi pomnilniškimi enotami. Z razvojem *Big Data* tehnologij so se te omejitve odpravile. Omenjene tehnologije omogočajo zajemanje, shranjevanje, obdelavo in analizo podatkov v distribuiranem okolju. Primeri *Big Data* tehnologij so Hadoop, Hadoop Distributed File System (HDFS) in Map Reduce. (Balusamy, Abirami, R. N., Kadry, S., & Gandom, A. H., 2021)

2.3 Opis kriterijev *Big Data*

V poglavju 2.2 sem omenil Laney-evo definicijo treh kriterijev: obsežnost (angl. Volume), hitrost (angl. Velocity) in raznolikost (angl. Variety). V začetku 20. stoletja se je o *Big Data* govorilo le v smislu teh treh kriterijev, sčasoma pa sta bila dodana še dva V-ja: vrednost (angl. Value) in verodostojnost (angl. Veracity), kar je sedaj poznano kot »pet V-jev«. Najpomembnejši so prvi trije, zato jih bom v nadaljevanju opisal. (TechTarget, 2023)

2.3.1 Obsežnost

Obseg dnevno generiranih podatkov, ki morajo biti beleženi in procesirani, je tako velik, da presega zmogljivosti konvencionalnih sistemov za obdelavo podatkov. Do takšnega obsega podatkov pride v večji meri zaradi visoke stopnje digitalizacije, vedno večjega števila spletnih uporabnikov in priljubljenosti družbenih omrežij in spletnih platform, ki nudijo filme, serije in glasbo. (Balusamy, Abirami, R. N., Kadry, S., & Gandom, A. H., 2021)

2.3.2 Hitrost

V knjigi *Big Data: Principles and Best Practices of Scalable Real Time Data Systems* avtorja razpravljata, da se podatki v današnjem svetu generirajo in zbirajo z velikimi hitrostmi, saj družbena omrežja, senzorji in spletni dnevniki uporabljajo podatke v realnem času. Tovrstni

hitri priliv podatkov zahteva specializirana orodja in postopke za učinkovito obdelavo. (Marz N. & Warren, J., 2015)

2.3.3 Raznolikost

Poznamo tri formate, v katerih se pojavljajo podatki:

- Strukturirani (angl. Structured)

Ta izraz zajema podatke, ki so organizirani znotraj tabel, na primer podatki o članih knjižničnega informacijskega sistema, inventarni listi in podobno. To vrsto podatkov lahko procesirajo tradicionalni sistemi za upravljanje podatkovnih baz.

- Delno strukturirani (angl. Semi-Structured)

Ta vrsta podatkov je kombinacija strukturiranih in nestrukturiranih podatkov. Primer tega je XML, saj nima tabelnega modela podatkov, ampak uporablja oznake za organiziranje polj znotraj podatkov.

- Nestrukturirani (angl. Unstructured)

V tem pojmu so zajeti vsi podatki, ki nimajo določene strukture. Primeri tega so spletne strani, video posnetki, elektronske pošte, slike, podatki iz družbenih omrežij in raznih senzorjev in tako naprej. (Balusamy, Abirami, R. N., Kadry, S., & Gandom, A. H., 2021)

2.4 Življenjski cikel podatkov Big Data

Življenjski cikel *Big Data* podatkov se lahko kategorizira na naslednje segmente:

- Generiranje *Big Data* (angl. Big Data Generation)

Big Data podatki se generirajo iz raznoraznih virov, na primer zgodovine brskanja, družbenih omrežij, spletnega bančništva, elektronskih sporočil, geoloških in meteoroloških zapisov, telefonije, zdravstva, GPS senzorjev in drugih.

- Dopolnjevanje podatkov (angl. Data Aggregation)

Ta korak zajema zbiranje golih podatkov in njihovo prenašanje na spominske platforme.

- Predprocesiranje podatkov (angl. Data Preprocessing)

Ta korak golim podatkom v štirih korakih (integracija podatkov, čiščenje podatkov, zmanjšanje podatkov, transformacija podatkov) odpravi napake in nekonsistence ter jih pretvori v razumljiv format.

- Podatkovna analitika (angl. Big Data Analytics)

V tem koraku pride končni uporabnik s kombinacijo *Big Data* tehnologij in analitičnih orodij do uporabnih spoznanj, vpogledov in informacij, na podlagi katerih bo lahko sprejel odločitve.

- Vizualizacija *Big Data* (angl. Big Data Visualisation)

Zadnji korak življenjskega cikla *Big Data* podatkov je vizualni prikaz analiz, s katerim se končnim uporabnikom na pregleden in učinkovit način predstavijo nove informacije. V ta namen se uporabljajo vizualizacijska orodja in razni grafikoni (črtni, palični, tortni, ploščinski). (Balusamy, Abirami, R. N., Kadry, S., & Gandom, A. H., 2021)

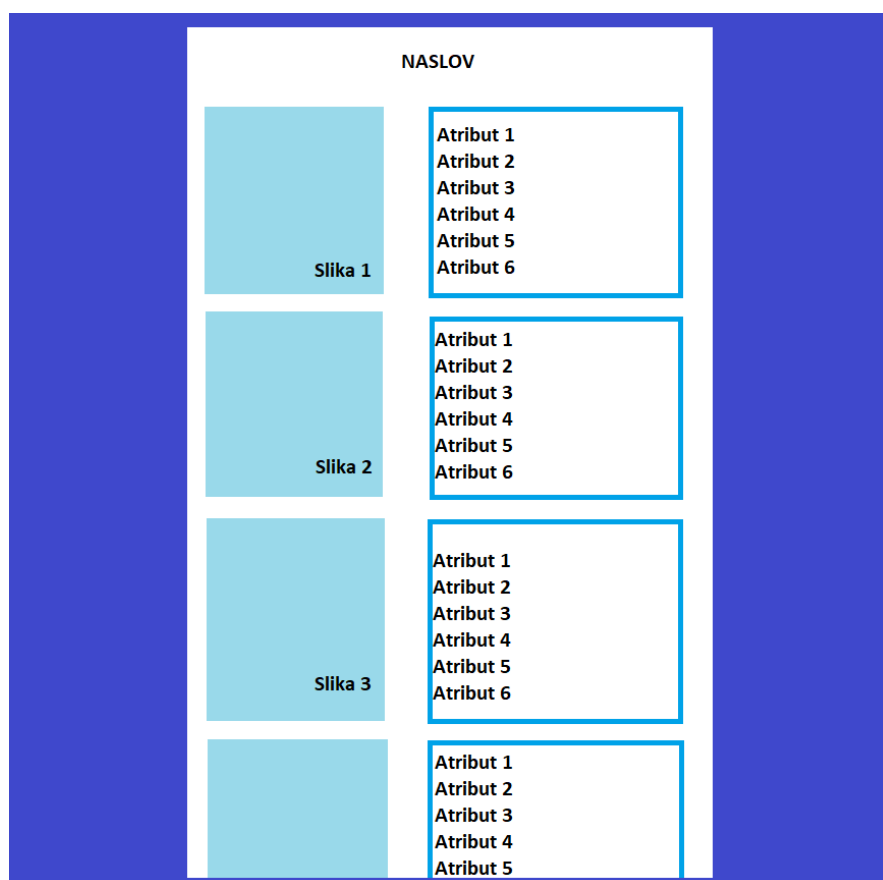
3 IZDELAVA SPLETNE APLIKACIJE

3.1 Namestitev in priprava integriranega razvojnega okolja

Najprej sem prenesel namestitveno datoteko za Visual Studio Code in z njo namestil razvojno okolje (angl. IDE). Nato sem s spletne strani <https://nodejs.org/en/download> prenesel namestitveno datoteko za node.js in namestil Node in NPM, saj sta pogoj za razvoj ReactJS in Vue spletnih aplikacij. Med tem ko Node omogoča izvajanje JavaScript kode v operacijskem sistemu, NPM skrbi za upravljanje odvisnosti projekta (angl. Dependencies). (Visual Studio Code, 2023)

3.2 Skica in načrt delovanja spletne aplikacije

3.2.1 Skica



Slika 1: Skica.

Vir: Lasten

3.2.2 Načrt delovanja

Spletna aplikacija mora prikazati večjo količino podatkov. Izbral sem zbirko podatkov o likih iz animirane serije *Rick and Morty*, saj vsak vnos vsebuje različne atribute in eno sliko. Vsi podatki razen slike se bodo nahajali lokalno na računalniku v JSON datoteki, pridobljeni s spletne strani *The Rick and Morty API*, na čemer bosta tekli aplikaciji. (The Rick and Morty API, 2023)

Slike bodo klicane s spletnega mesta GitHub. Zaradi morebitnih nihanj v omrežju ali pa na strežniku, na katerem so slike, lahko pride do nenatančnih meritev delovanja same aplikacije. Za namene ugotavljanja te problematike bom izdelal še eno verzijo aplikacije z desetkratnikom količine podatkov. Aplikaciji ne bosta imeli drugih funkcionalnosti. (GitHub, 2023)

3.3 Izdelava sestavnih delov spletne aplikacije

3.3.1 ReactJS

Za implementacijo spletne aplikacije s pomočjo knjižnice ReactJS sem uporabil orodno verigo (angl. Tool-Chain) 'create-react-app.'

Spletno aplikacijo Rick & Morty sem ustvaril s pogonom naslednjega ukaza v terminalu:

```
npx create-react-app rick-and-morty-react --template typescript
```

Na tej točki mi je terminal ponudil namestitev orodne verige »create-react-app«, ki je potrebna za pogon prejšnjega ukaza, kar sem potrdil s tipkama »y« in »enter«.

To mi je generiralo osnovno strukturo spletne aplikacije. Kot naslednje sem v Visual Studio Code odprl direktorij, ki vsebuje novo aplikacijo, ter pognal ukaz za namestitev odvisnosti:

```
npm install
```

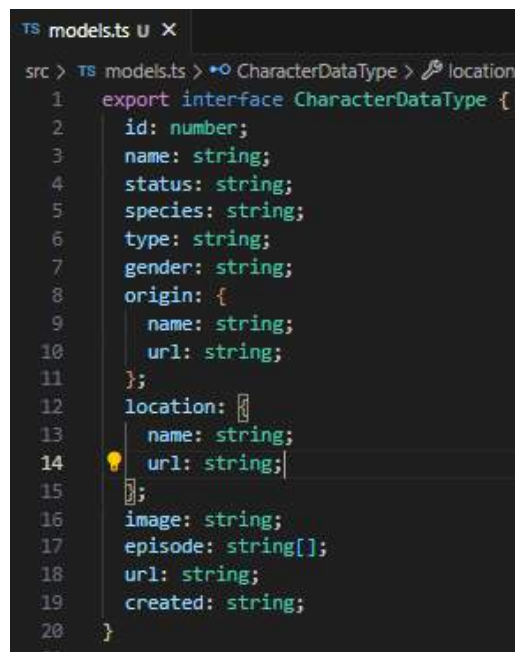
Spletno aplikacijo sem pognal v lokalnem okolju z ukazom:

```
npm start
```

Za lažje delo s CSS sem v projekt namestil tudi knjižnico sass (super awesome CSS) z naslednjim ukazom:

```
npm i sass --save-dev
```

V direktoriju aplikacije sem dodal JSON datoteko, ki vsebuje Rick & Morty like ter pričel izdelovati komponente za prikaz teh likov. Ker uporabljam typescript, sem v novi datoteki models.ts definiral typescript vmesnik (interface) CharacterDataType, ki določa strukturo objektov likov iz JSON datoteke:



```
TS models.ts U X
src > TS models.ts > CharacterDataType > location
1 export interface CharacterDataType {
2   id: number;
3   name: string;
4   status: string;
5   species: string;
6   type: string;
7   gender: string;
8   origin: {
9     name: string;
10    url: string;
11  };
12  location: {
13    name: string;
14    url: string;
15  };
16  image: string;
17  episode: string[];
18  url: string;
19  created: string;
20 }
```

Slika 2: Vmesnik CharacterDataType.

Vir: Lasten

Glavna komponenta je Layout.tsx:



```
TS Layout.tsx X
src > components > Layout > Layout.tsx
1 import classes from "../Layout.module.scss";
2 import characters from "../../data/rickandmorty.json";
3 import Character from "../Character/Character";
4
5 const Layout = () => {
6   const realCharacters = characters.concat(characters).concat(characters);
7   return (
8     <div className={classes.container}>
9       <div className={classes.innerContainer}>
10        <div className={classes.innermostContainer}>
11          <h1>Rick & Morty heroes</h1>
12          <div className={classes.content}>
13            {realCharacters.map(character => (
14              <Character character={character} />
15            ))}
16          </div>
17        </div>
18      </div>
19    </div>
20  );
21 }
22
23 export default Layout;
```

Slika 3: Layout.tsx.

Vir: Lasten

Ta komponenta je odgovorna za pridobitev podatkov, njihov prikaz ter postavitve prikaznih elementov. Kot naslednje sem implementiral komponento Character, ki se prikaže za vsak lik Rick & Morty. Ta komponenta je sestavljena iz slike in tabele podatkov.

```
TS Character.tsx X
src > components > Character > TS Character.tsx > ...
1  import classes from "../Character.module.scss";
2  import { CharacterDataType } from "../../models";
3  import CharacterTable from "../CharacterTable/CharacterTable";
4
5  interface CharacterProps {
6    character: CharacterDataType;
7  }
8
9  const Character = ({ character }: CharacterProps) => {
10   return (
11     <div className={classes.container}>
12       <img src={character.image} alt={character.name} />
13       <div className={classes.content}>
14         <div className={classes.header}>
15           <div>{character.name}</div>
16         </div>
17         <CharacterTable character={character} />
18       </div>
19     </div>
20   );
21 };
22
23 export default Character;
24
```

Slika 4: Character.tsx.

Vir: Lasten

Komponenta za tabelo je imenovana CharacterTable, ki služi prikazu podatkov Rick & Morty lika v obliki tabele. Kot prop vzame objekt tipa CharacterProps, ki opredeli, kakšne vrste podatkov komponenta sprejme.

```

TS CharacterTable.tsx X
src > components > CharacterTable > TS CharacterTable.tsx > ...
1  import { CharacterDataType } from "../../models";
2  import classes from "../CharacterTable.module.scss";
3
4  interface CharacterTableProps {
5    character: CharacterDataType;
6  }
7
8  const CharacterTable = ({ character }: CharacterTableProps) => {
9    return (
10     <table className={classes.table}>
11       <tbody>
12         { /* status */ }
13         <tr>
14           <th>Status</th>
15           <td>{character.status}</td>
16         </tr>
17         { /* species */ }
18         <tr>
19           <th>Species</th>
20           <td>{character.species}</td>
21         </tr>
22         { /* type */ }
23         <tr>
24           <th>Type</th>
25           <td>{character.type || "Unknown"}</td>
26         </tr>
27         { /* gender */ }
28         <tr>
29           <th>Gender</th>
30           <td>{character.gender}</td>
31         </tr>
32         { /* origin */ }
33         <tr>
34           <th>Origin</th>
35           <td>{character.origin.name}</td>
36         </tr>
37         { /* location */ }
38         <tr>
39           <th>Location</th>
40           <td>{character.location.name}</td>
41         </tr>
42       </tbody>
43     </table>
44   );
45 };
46
47 export default CharacterTable;

```

Slika 5: CharacterTable.tsx.

Vir: Lasten

Podatke o likih aplikacija pridobi iz JSON datotke. Za boljšo primerjavo med aplikacijama ReactJS in Vue sem za obe aplikaciji uporabil enako JSON datoteko.

```

() rickandmorty.json X
sec > data > () rickandmorty.json > {} 48
427   "gender": "Male",
428   "origin": {
429     "name": "unknown",
430     "url": ""
431   },
432   "location": {
433     "name": "Citadel of Ricks",
434     "url": "https://rickandmortyapi.com/api/location/3"
435   },
436   "image": "https://raw.githubusercontent.com/Laboratoria/rick-and-morty-images/master/images/15.jpeg",
437   "episode": ["https://rickandmortyapi.com/api/episode/10"],
438   "url": "https://rickandmortyapi.com/api/character/15",
439   "created": "2017-11-04T20:56:13.215Z"
440 },
441 {
442   "id": 16,
443   "name": "Anish Cyborg",
444   "status": "Dead",
445   "species": "Alien",
446   "type": "Parasite, Cyborg",
447   "gender": "Male",
448   "origin": {
449     "name": "unknown",
450     "url": ""
451   },
452   "location": {
453     "name": "Earth (Replacement Dimension)",
454     "url": "https://rickandmortyapi.com/api/location/20"
455   },
456   "image": "https://raw.githubusercontent.com/Laboratoria/rick-and-morty-images/master/images/16.jpeg",
457   "episode": ["https://rickandmortyapi.com/api/episode/15"],
458   "url": "https://rickandmortyapi.com/api/character/16",
459   "created": "2017-11-04T21:12:45.235Z"
460 },
461 {
462   "id": 17,
463   "name": "Annie",
464   "status": "Alive",
465   "species": "Human",
466   "type": "",
467   "gender": "Female",
468   "origin": {
469     "name": "Earth (C-137)",
470     "url": "https://rickandmortyapi.com/api/location/1"
471   },
472   "location": {
473     "name": "Anatomy Park",
474     "url": "https://rickandmortyapi.com/api/location/5"
475   },
476   "image": "https://raw.githubusercontent.com/Laboratoria/rick-and-morty-images/master/images/17.jpeg",
477   "episode": ["https://rickandmortyapi.com/api/episode/3"],
478   "url": "https://rickandmortyapi.com/api/character/17",
479   "created": "2017-11-04T22:21:24.081Z"
480 },
481 {
482   "id": 18,
483   "name": "Antenna Party",
484   "status": "Alive",
485   "species": "Human",

```

Slika 6: Rick and Morty.json.

Vir: Lasten

3.3.2 Vue

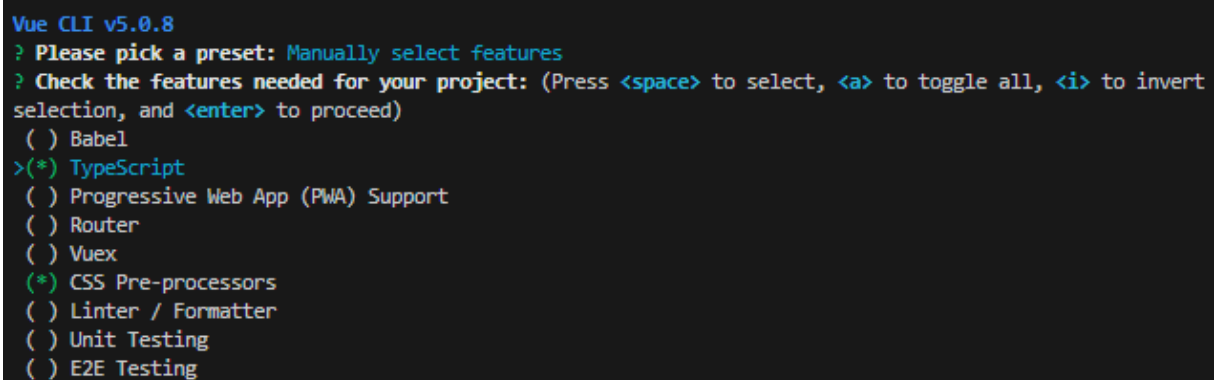
Za implementacijo spletne aplikacije Rick & Morty z ogrodjem Vue sem v prvem koraku moral namestiti Vue na svoj računalnik s pomočjo naslednjega ukaza:

```
npm i -g @vue/cli
```

Nato sem v terminalu navigiral v direktorij, v katerem sem želel ustvariti Vue aplikacijo, ter pognal ukaz:

```
vue create rick-and-morty-vue
```

Terminal mi je ponudil izbor med Vue 2, Vue 3 in ročnim izborom. Izbral sem ročni izbor in potrdil s tipko »enter«. Terminal mi je ponudil naslednje stvari:



```
Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
  ☐ Babel
>(*) TypeScript
  ☐ Progressive Web App (PWA) Support
  ☐ Router
  ☐ Vuex
  (*) CSS Pre-processors
  ☐ Linter / Formatter
  ☐ Unit Testing
  ☐ E2E Testing
```

Slika 7: Namestitev Vue.

Vir: Lasten

Obkljukal sem TypeScript in CSS Pre-processors, odkljukal pa Linter in Babel ter potrdil izbor s tipko »enter«. Terminal mi je ponudil izbor Vue verzije 2.x ali 3.x. Izbral sem verzijo 3.x in potrdil s tipko »enter«. V nadaljevanju mi je terminal ponudil različne CSS pre-procesorje, med katerimi sem izbral Sass/SCSS, saj to uporabljamo tudi v ReactJS aplikaciji. Za postavitev konfiguracijskih datotek sem izbral privzeto lokacijo.

To je povzročilo prenos vseh potrebnih datotek na moj računalnik. Nato sem v Visual Studio Code odprl direktorij, ki vsebuje novo aplikacijo, odprl nov terminal ter pognal ukaz za namestitev odvisnosti:

```
npm install
```

Aplikacijo sem pognal z ukazom:

```
npm run serve
```

V nadaljevanju sem namestil knjižnico, imenovano `@vitejs/plugin-vue-jsx`, ki omogoči enako sintakso kot ReactJS, da sem lažje prenesel komponente iz React aplikacije v Vue. Ukaz, ki sem ga pognal, je:

```
npm i @vitejs/plugin-vue-jsx
```

Za lažje delo s CSS sem tako kot v ReactJS aplikaciji namestil knjižnico `sass` z naslednjim ukazom:

```
npm i sass --save-dev
```

Kot v spletni aplikaciji v ReactJS, sem tudi tukaj dodal JSON datoteko z Rick & Morty liki ter ustvaril `models.ts` datoteko, ki vsebuje definicijo typescript vmesnika `CharacterDataType`. Naslednji korak je bilo kopiranje direktorija `components` iz ReactJS projekta v Vue projekt. Da sem lahko uvozil JSON datoteko v komponento `Layout.tsx`, sem moral v datoteki `tsconfig.json` dodati vrstico:

```
"resolveJsonModule": true,
```

Zaradi JSX knjižnice sem komponente lahko praktično kopiral. Edini razliki sta bili, da se deklaracija `css` razreda v React JSX imenuje »`className`«, med tem ko se v Vue imenuje »`class`«, ter da je pri uvozu vmesnika iz `models.ts` datoteke v Vue aplikaciji potrebno uporabiti ključno besedo »`type`«, med tem ko pri ReactJS to ni potrebno.

Da sem lahko uspešno uvozil datoteke z razširitvijo `scss`, sem v korenski mapi ustvaril datoteko `declaration.d.ts`, in ji dodal naslednjo vsebino:

```
declare module '*.scss';
```

Nato sem v datoteki `tsconfig.json` uredil uvoz te datoteke. S tem sem lahko uspešno uvozil datoteke `scss` v moje `tsx` komponente.

Zadnji korak je bil, da sem kopiral datoteko `base.css` iz moje ReactJS aplikacije, saj je v njej definirana pisava za prikaz. S tem je bil razvoj spletne aplikacije Rick & Morty s pomočjo ogrodja Vue zaključen.

4 PRIMERJAVA ZAHTEVNOSTI UČENJA UPORABE ORODJA

4.1 Dostop do učnega gradiva

4.1.1 Tečaji na spletu

Spletni tečaji so priljubljen način učenja programskih jezikov, saj nudijo interaktivno okolje, v katerem se učenec na primerih nauči postopno zahtevnejših konceptov. Na trgu so plačljivi in prosto dostopni tečaji. Primerjal bom ponudbo dveh priljubljenih ponudnikov teh storitev. Najprej sledi primerjava na spletni strani Udemy, ki nudi omenjene tečaje. ((Udemy: Results for vue, 2023) (Udemy: Results for react, 2023))

The screenshot displays the Udemy search results for the keyword "vue". At the top, it indicates "1,664 results for 'vue'". Below this, there are filter options for "Filter" and "Sort by Most Relevant". The results are listed in a grid format, showing course titles, ratings, durations, and prices. The first three courses are:

- Vue - The Complete Guide (incl. Router & Composition API)** by Maximilian Schwarzmüller. Rating: 4.7 stars (50,844). 37 total hours, 100 lectures. Price: €79.99.
- Complete Vue Developer 2023: Zero to Mastery (Pinia, Vite)** by Daniel Flanagan, Luis Hernandez. Rating: 4.7 stars (2,524). 32 total hours, 97 lectures. Price: €89.99.
- Vue Masterclass (Covers Vue 2 and 3)** by Boris Polibin. Rating: 4.7 stars (770). 45 total hours, 486 lectures. Price: €79.99.

Below the course listings, there is a banner for "Top companies trust Udemy Business to build in-demand career skills." featuring logos for Nasdaq, IBM, box, NetApp, and eventbrite. At the bottom, another course is partially visible: "Vue JS 3: Composition API (with Pinia, Firebase 9 & Vite)" by Danny Connolly, with a rating of 4.7 stars (3,462).

Slika 8: Spletni tečaji Udemy Vue.

Vir: (Udemy: Results for vue, 2023)

9,776 results for "react"

Filter

Sort by Most Relevant

9,776 results

Ratings

4.5 & up (3,773)

4.0 & up (5,803)

3.5 & up (6,306)

3.0 & up (8,170)

Language

English (3,002)

Spanish (166)

Turkish (146)

Portuguese (100)

Show more

Video Duration

Features

Topic

Level

Subtitles



The Ultimate React Course 2023: React, Redux & More

Master modern React from beginner to advanced! Context API, React Query, Redux Toolkit, Tailwind, advanced patterns

James Schmalzer

4.9 (5,616)

27 total hours - 411 lectures - All Levels

Enroll now



React - The Complete Guide 2023 (incl. React Router & Redux)

Done in and learn React.js from scratch! Learn React, Hooks, Redux, React Router, Next.js, Best Practices and why "should"

Academind by Maximilian Schwarzmüller, Maximilian Schwarzmüller

4.8 (196,123)

55.3 total hours - 808 lectures - All Levels

Enroll now



The Complete 2023 Web Development Bootcamp

Become a Full-Stack Web Developer with just ONE course: HTML, CSS, Javascript, Node, React, MongoDB, Web3 and DApps

Dr. Angela Yu

4.7 (120,006)

82 total hours - 598 lectures - All Levels

Enroll now

Top companies trust Udemy Business to build in-demand career skills.









Complete React Developer in 2023 (w/ Redux, Hooks, GraphQL)

Updated! Become a Senior React Developer. Build a massive E-commerce app with Redux, Hooks, GraphQL, Stripe, Firebase

Andrei Neagoie, Piusa Shang

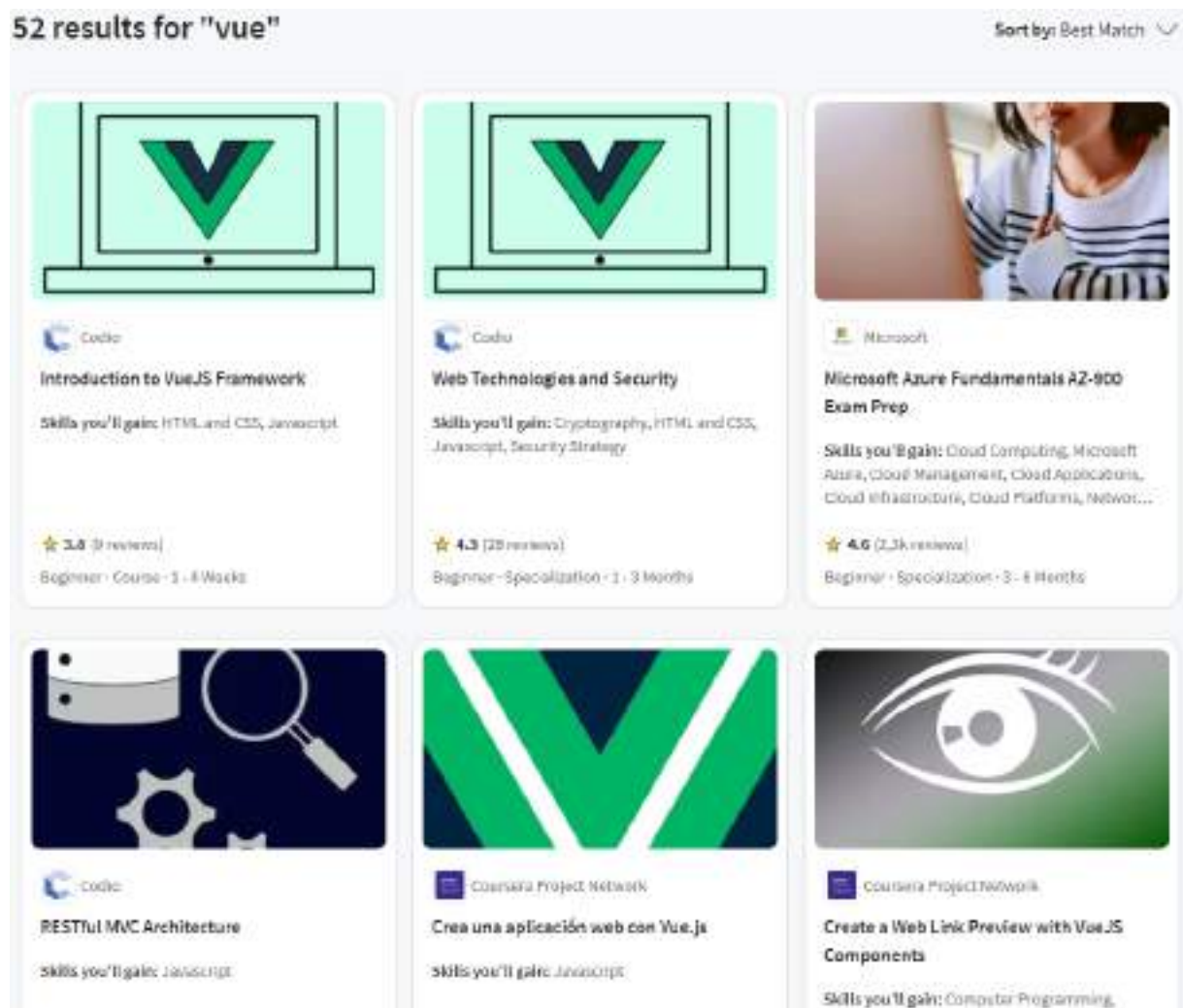
4.7 (10,000)

Slika 9: Spletni tečajji Udemy ReactJS.

Vir: (Udemy: Results for react, 2023)

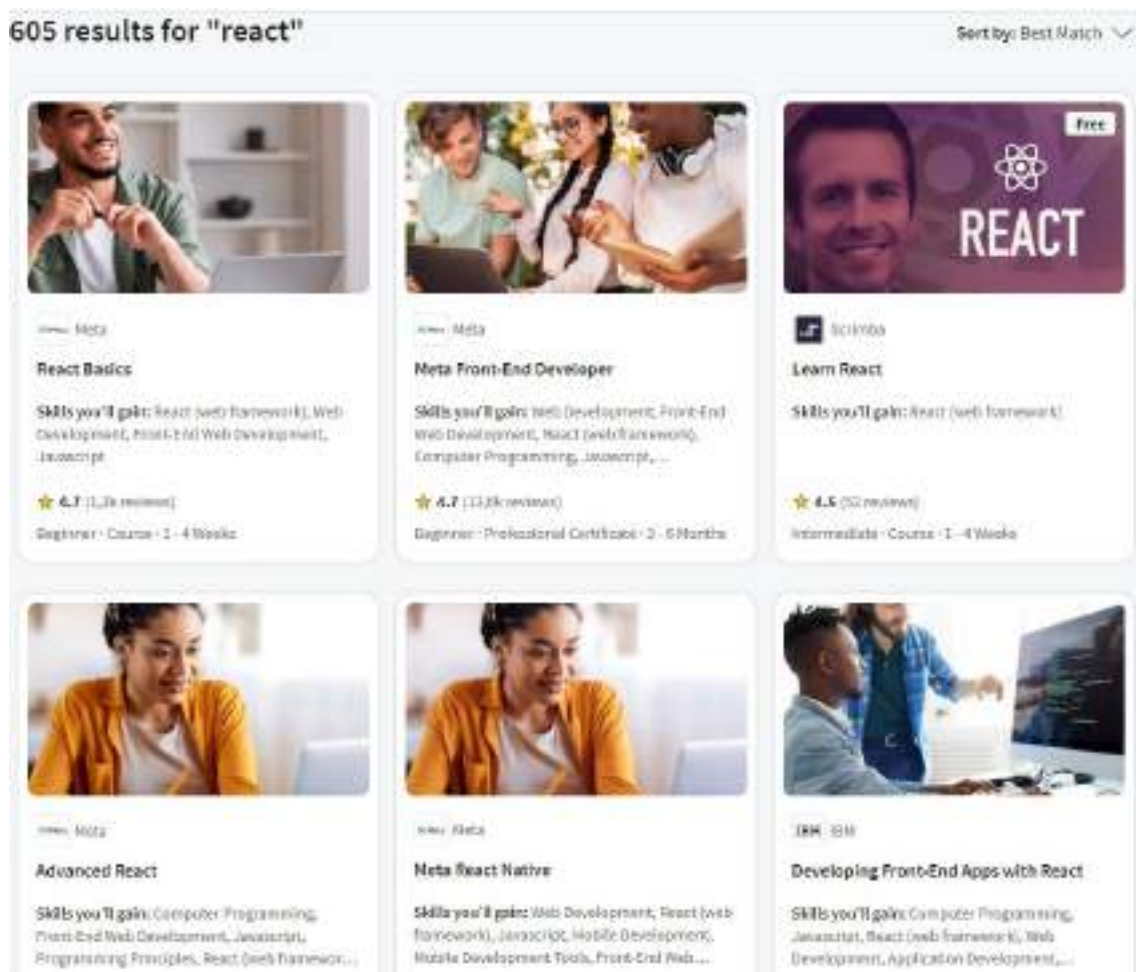
32

Sledijo podatki s spletne strani Coursera. ((Coursera: vue, 2023) (Coursera: react, 2023))



Slika 10: Spletni tečaji Coursera Vue.

Vir: (Coursera: vue, 2023)



Slika 11: Spletni tečajji Coursera ReactJS.

Vir: (Coursera: react, 2023)

Udemy nudi 1664 tečajev, ki vsebujejo ključno besedo Vue, od tega 632 takšnih, ki so s strani uporabnikov ocenjeni s 4,5/5 in višje, s ključno besedo React pa 9776, od tega 3772 takšnih, ki so ocenjeni s 4,5/5 ali višje.

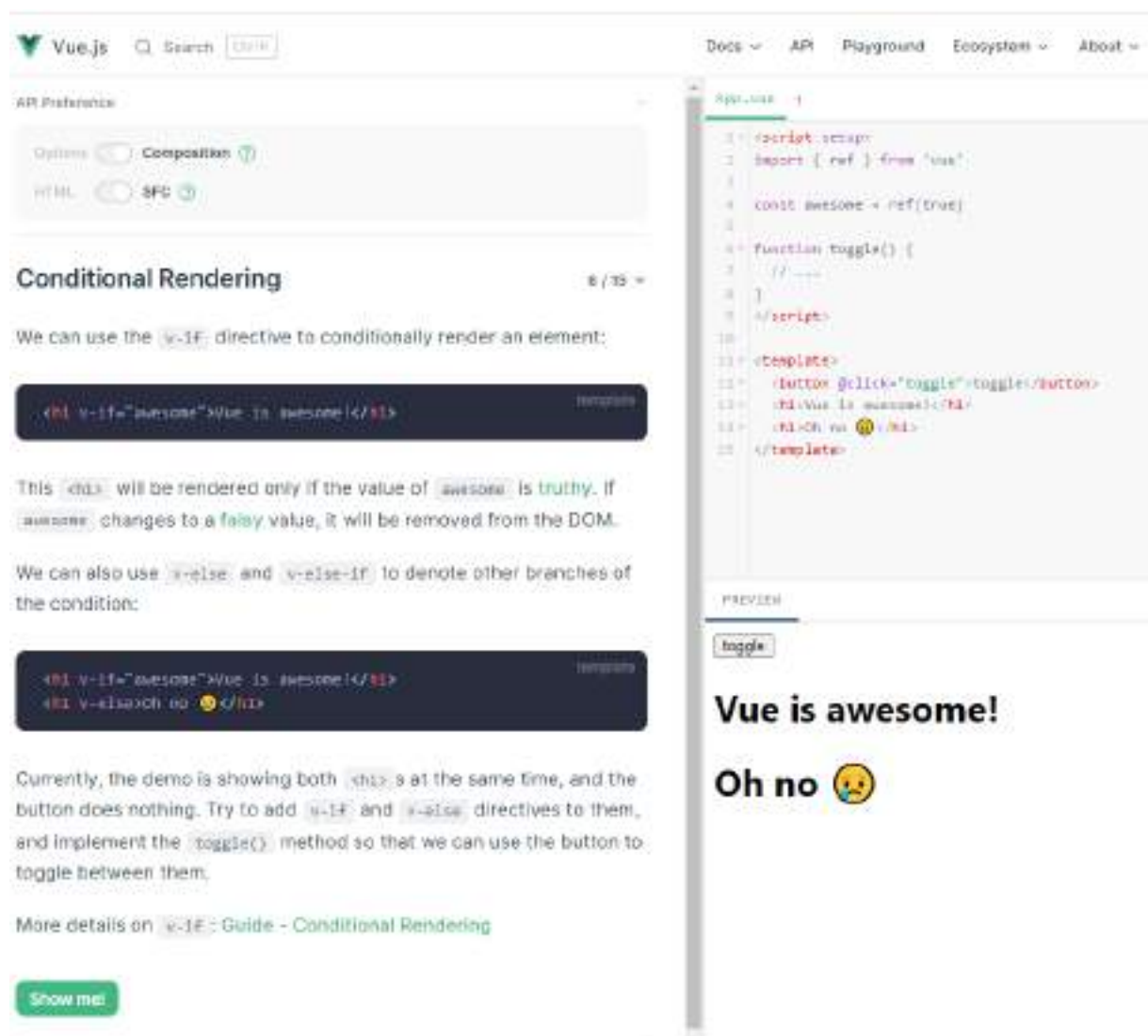
Coursera nudi 52 tečajev s ključno besedo Vue in 605 s ključno besedo react, stran pa ne omogoča prikaza števila tečajev, ki so jih uporabniki visoko ocenili.

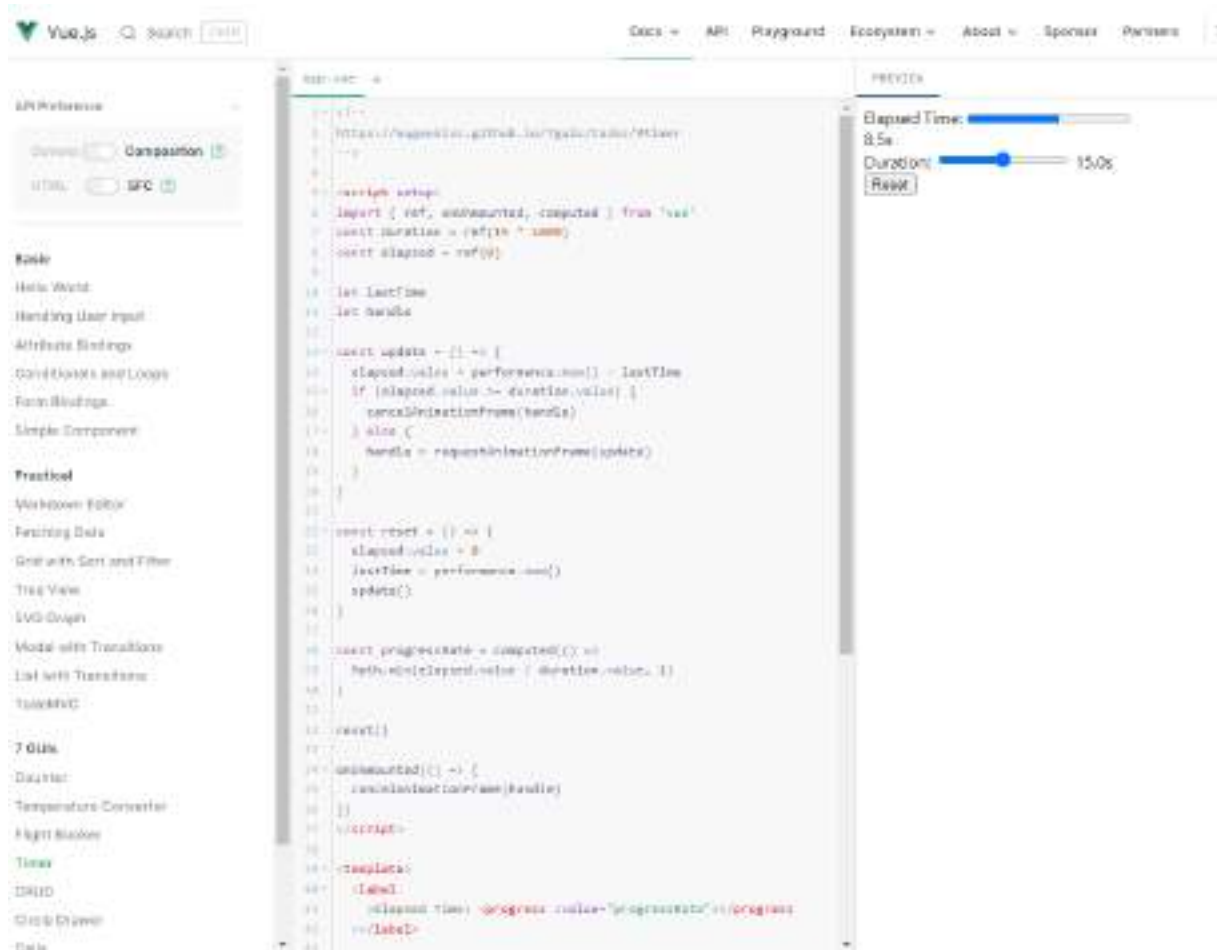
Razlika med ponudnikoma je, da so tečajji na Udemy plačljivi, na Coursera pa prosto dostopni. V obeh primerih je na voljo več tečajev za ReactJS kot za Vue. Pri plačljivih tečajih to ni pomembno, saj ima uporabnik kljub temu na voljo dovolj obsežen nabor tečajev za Vue. Število prosto dostopnih tečajev za Vue pa je na Coursera zelo omejeno. Morebitni razlog za to so dobro urejeni delno prosto dostopni tečajji za Vue na Vueschool in Vuemastery. Obe orodji imata obsežen nabor visoko ocenjenih tečajev na spletu, največ jih pa ima ReactJS. (Udemy:

Results for vue, 2023) (Udemy: Results for react, 2023) (Coursera: vue, 2023) (Coursera: react, 2023))

4.1.2 Uradna dokumentacija

Ponudniki obeh orodij nudijo uradno dokumentacijo na svojih spletnih straneh, ki vsebuje razlage, koncepte in primere.





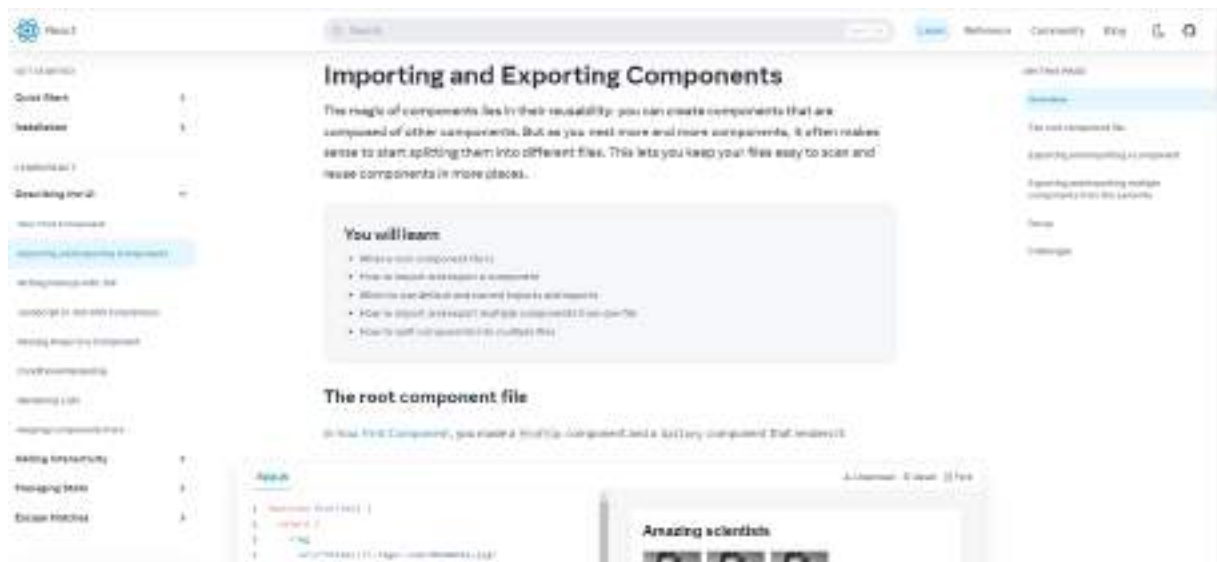
Slika 13: Uradna dokumentacija Vue 2.

Vir: (Vue.js: Introduction, 2023)



Slika 14: Uradna dokumentacija ReactJS 1.

Vir: (React: Quick Start, 2023)



Slika 15: Uradna dokumentacija ReactJS 2.

Vir: (React: Quick Start, 2023)

Spletni mesti obeh ponudnikov imata kvalitetno, pregledno in sistematsko urejeno interaktivno dokumentacijo, ki uporabniku na prijazen način približa razne aspekte orodja. ((Vue.js: Introduction, 2023) (React: Quick Start, 2023))

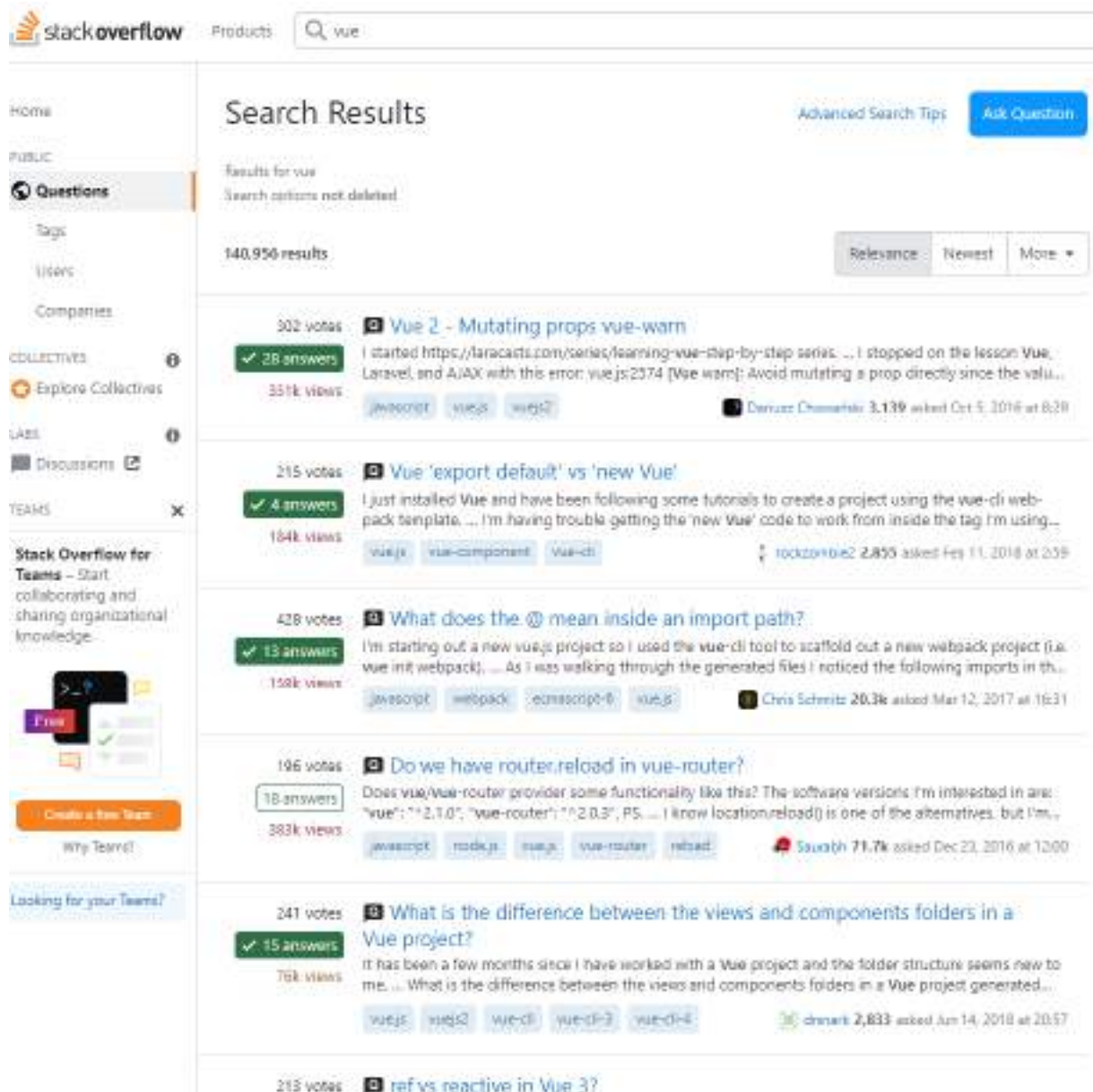
4.1.3 Knjige in E-knjige

Za dostop do učnega gradiva v knjižni obliki sem na mreži informacijskih knjižnic Cobiss primerjal zadetke s ključno besedo »React« in ključno besedo »Vue« z aktivnimi filtri »vrsta gradiva: knjiga; predmet 1: računalniška omrežja, internet, umetna inteligenca; predmet 2: računalništvo, računalniška znanost.«

Podobno kot pri tečajih na spletu, je tudi pri iskanju knjižnega gradiva občutno več zadetkov na ključno besedo »React« (243) kot na »Vue« (121). ((Cobiss: Vue, 2023) (Cobiss: React, 2023))

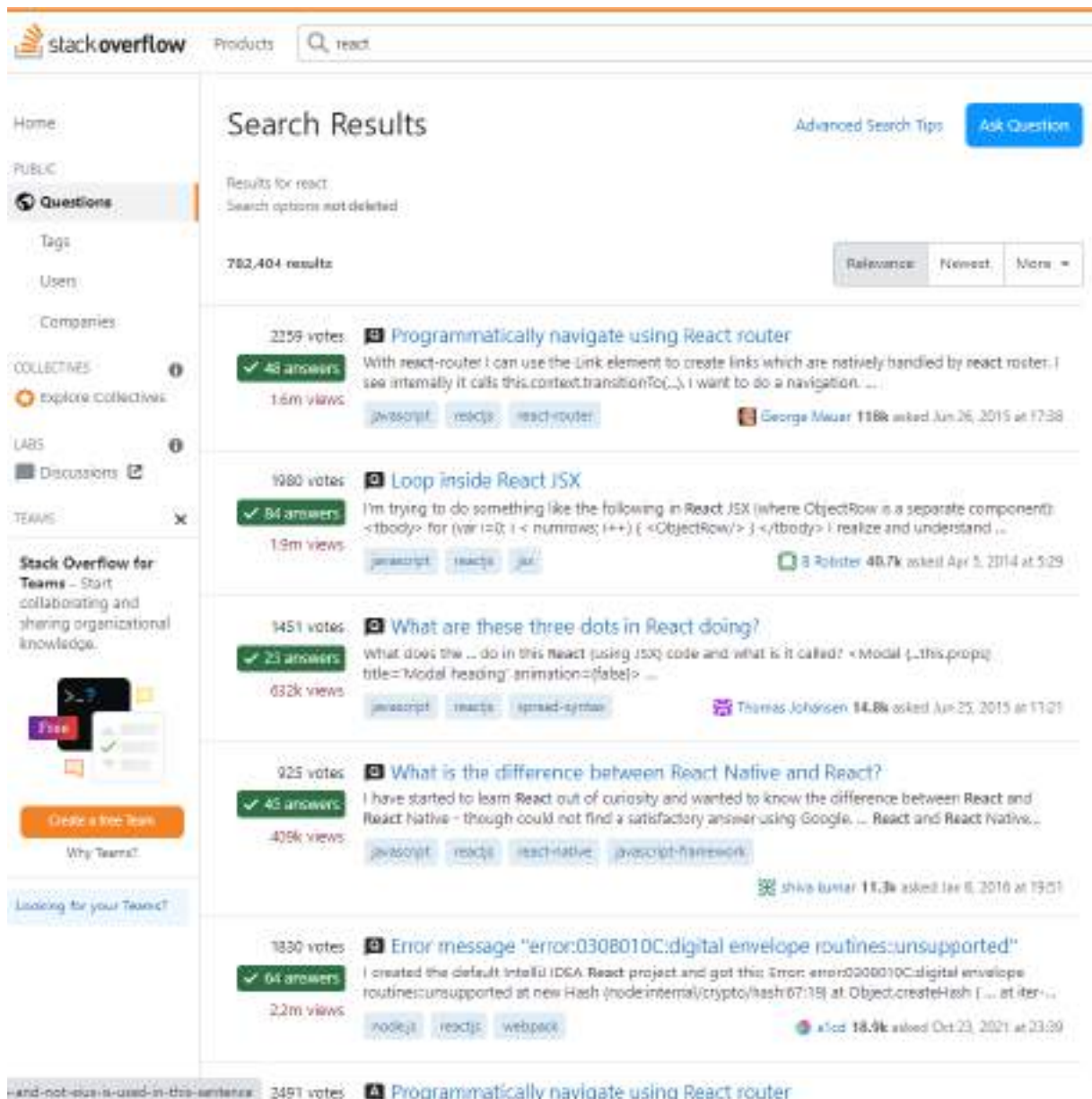
4.2 Skupnost

4.2.1 Stack Overflow, Reddit



Slika 18: Stack Overflow Vue.

Vir: (Stack Overflow: vue, 2023)



Slika 19: Stack Overflow ReactJS.

Vir: (Stack Overflow: react, 2023)



Slika 20: Reddit Vue.

Vir: (Reddit: vue, 2023)



Slika 21: Reddit ReactJS.

Vir: (Reddit: react, 2023)

Stack Overflow in Reddit sta dve od najbolj priljubljenih spletnih strani, kamor se programerji zatekajo po informacije, postavljajo vprašanja in drugim uporabnikom odgovarjajo na njihova.

Na Stack Overflow je zadetkov pri iskanju ključne besede »Vue« 140.956, pri iskanju »React« pa kar 782.404. Največji dve skupnosti za Vue na spletni strani Reddit imata skupaj okrog 108.900 uporabnikov, največji dve skupnosti za React na Redditu pa imata skupaj okrog 421.000 uporabnikov.

Na spletnih platformah Stack Overflow in Reddit je skupnost za ReactJS občutno večja kot skupnost za Vue. ((Stack Overflow: vue, 2023) (Stack Overflow: react, 2023) (Reddit: vue, 2023) (Reddit: react, 2023))

4.2.2 Facebook

Facebook je priljubljeno družbeno omrežje, kjer lahko uporabniki med drugim ustvarjajo interesne skupine. Primerjal bom število članov skupin, posvečenim Vue in ReactJS med prvimi osmimi zadetki.

	Vue.js Developers Javno · 70 tis. članov · 10+ objav na dan This group is for discussion about Vue.js development and welcomes both experienced Vue developers and those who are just getting started. Follow the...	Pridruži se
	Vue.js For Hire Zasebna · 17 tis. članov · 2 objavi na teden The Official Vue.js group for those looking for/offering Vue.js jobs or projects.	Pridruži se
	Javascript, React, Angular, Vue Developers Zasebna · 9,3 tis. članov · 10 objav na teden Purpose of creating this group is to create a community to learn all web development technologies for free and solve eachother problems at one place.	Pridruži se
	Vue.js Philippines (Official) Javno · 5,6 tis. članov · 2 objavi na teden Vue.js Philippines is a community of Vue.js developers from across the country. The aim of the group is to build a create a self-sustaining community and create...	Pridruži se
	Vue.js Frontend Developers Javno · 18 tis. članov · 10+ objav na dan	Pridruži se
	Vue.js Developers Javno · 15 tis. članov · 3 objave na teden	Pridruži se
	Vue.js 3 Developers Community Javno · 7,2 tis. članov · 4 objave na teden A community for developers using the Vue 3 framework	Pridruži se
	Laravel + Vue.js Developers Javno · 44 tis. članov · 2 objavi na teden	Pridruži se

Slika 22: Facebook Vue.

Vir: (Facebook: vue, 2023)



React.JS developers -2023

Javno · 106 tis. članov · 10+ objav na dan

In computing, React is a JavaScript library for building user interfaces. It is maintained by Facebook, Instagram and a community of individual developers...

 1 prijatelj je član

Pridruži se



React JS Developers

Zasebna · 274 tis. članov · 10 objav na dan

Pridruži se



React.js Developers

Javno · 213 tis. članov · 9 objav na dan

Pridruži se



ReactJs for beginners

Javno · 49 tis. članov · 10 objav na teden

Pridruži se



React Developers

Javno · 13 tis. članov · 10 objav na teden

Pridruži se



ReactJS Developer

Zasebna · 24 tis. članov · 2 objavi na teden

React is a UI library developed at Facebook. It is a Javascript library for building user interfaces. It is used by Facebook and Instagram. JSX is the extension for th...

Pridruži se



Nodejs, React, Angular Developers

Javno · 14 tis. članov · 7 objav na teden

This group is for sharing professional knowledge in Angular, React, jQuery, JavaScript.

Pridruži se



React Native Developers

Zasebna · 22 tis. članov · 7 objav na dan

As we know , react Native is an open-source mobile application framework created by Facebook. Let us more explore in this group experts . Official website...

Pridruži se

Slika 23: Facebook ReactJS.

Vir: (Facebook: react, 2023)

Prvih osem skupin, prikazanih pri iskanju ključne besede »Vue« dne 17. 10. 2023, vsebuje okrog 186.100 članov. Pri iskanju ključne besede »React« pa kar okrog 715.000. (Facebook: vue, 2023) (Facebook: react, 2023))

4.3 Kompleksnost

4.3.1 Sintaksa

Vue uporablja predlogovno sintakso, ki temelji na html. ReactJS uporablja JSX. To je razširitev javascripta, ki omogoči pisanje javascripta in html-ja skupaj. Tukaj je potrebno dodati, da lahko s pravo knjižnico tudi v Vue uporabljamo JSX.

4.3.2 Virtual DOM (virtualni objektni model dokumenta)

Dom je vmesnik, ki naredi reprezentacijo html ali xml dokumenta, v katerem so vsi elementi (node) organizirani v drevesno strukturo – podobno kot pri družinskem drevesu. To omogoča javascriptu dostop do html in css spletne strani tako, da lahko celotno vsebino dokumenta urejamo z javascriptom. Vsaka sprememba uporabniškega vmesnika povzroči, da se DOM posodobi in tako naprej reprezentira aktualno različico uporabniškega vmesnika. DOM se uporablja pri vseh spletnih straneh.

Težava pogostega manipuliranja DOM-a je, da vpliva na učinkovitost. Po vsaki manipulaciji DOM-a morajo biti vsi posodobljeni elementi in njihovi »otroci« na novo prikazani oz. izrisani (angl. Re-Rendered), da posodobijo uporabniški vmesnik aplikacije. Ravno to »re-render-anje« uporabniškega vmesnika upočasni sistem.

Rešitev za to težavo se je pojavila leta 2013, ko je knjižnica Riot.js (predhodnik ReactJS) kot prva začela uporabljati Virtual DOM, kar je znatno izboljšalo učinkovitost. Virtual DOM je virtualna reprezentacija DOM-a. Vsakič, ko pride do spremembe uporabniškega vmesnika, se namesto DOM-a posodobi Virtual DOM. Za tem se primerja novi Virtual DOM s prejšnjo različico Virtual-DOM-a in tako ugotovi, na katerih elementih (v-node) v drevesni strukturi je prišlo do sprememb. Ta postopek se imenuje diffing. Za tem Virtual DOM izračuna najboljšo metodo za posodobitev DOM-a tako, da pride do najmanjšega možnega števila operacij na

DOM. To drastično zmanjša število elementov, ki morajo biti ob vsaki spremembi »re-renderani«, kar poveča učinkovitost in hitrost.

Kasneje v letu 2013 (29. maj) je izšel ReactJS in po svojem predhodniku podedoval Virtual-DOM. Ogradje Vue, ki je izšlo februarja 2014 tudi uporablja Virtual DOM, se pa v pristopu razlikuje od ReactJS.

Vue uporablja tako imenovan »Compiler-informed virtual DOM«, ki je praviloma hitrejši in učinkovitejši od tradicionalnega virtual DOM-a. Prevajalnik (angl. Compiler) predvideva strukturo virtual DOM-a in generira kodo, ki izkorišča predvideno tako, da omogoča bližnjice, kjer je to mogoče. S tem se izogne temu, da bi diffing algoritm primerjal vse v-node-e (element v virtual DOM-u, tako, kot je node element v DOM-u) v drevesni strukturi. Primer: Kadar sistem ve, da se določen v-node ne bo nikoli spremenil, ga algoritem zaobide in ne »re-rendera«.

ReactJS pa uporablja tradicionalni virtual DOM, kjer diffing algoritm ne more predvidevati strukture prihajajočega virtual DOM-a. Posledično mora primerjati vsak sleherni v-node v drevesu in v novem virtual DOM-u ustvariti nove v-node tudi za elemente, ki se ne spremenijo.

Oba pristopa imata svoje prednosti. Med tem ko je virtual DOM od Vue učinkovitejši in hitrejši, je ReactJS-ov preprostejši in lažji za implementacijo. Od potreb posamičnega projekta je odvisno, ali je višja prioriteta učinkovitost ali preprostost. ((Legacy React, 2023) (Vue.js, 2023))

4.3.3 Struktura komponent

Vue uporablja ugnезdeno (angl. Nested) strukturo globalnih in lokalnih komponent, ReactJS pa uporablja ravno (angl. Flat) strukturo komponent s funkcionalnimi in razrednimi komponentami. Pri ogróđu Vue so komponente ugnезdene ena v drugo. To pomeni, da je podrejeno komponento mogoče uporabiti samo znotraj obsega njene nadrejene komponente.

Vue uporablja tudi globalne komponente, ki jih je po registraciji mogoče uporabiti kjerkoli v aplikaciji. Lokalne komponente pa so registrirane znotraj obsega druge komponente in jih je mogoče uporabiti samo znotraj te komponente.

Ravna (flat) struktura komponent pri ReactJS pomeni, da lahko tudi podrejene komponente uporabimo kjerkoli v aplikaciji, saj niso ugnезdene. ReactJS uporablja funkcionalne in razredne

komponente. Funkcionalne komponente so brez stanja, kar pomeni, da ne morejo shranjevati podatkov ali z uporabo kode spreminjati svojega videza. Razredne komponente pa za razliko od funkcionalnih lahko vzdržujejo stanje, torej shranjujejo podatke in spreminjajo svoj videz s kodo. ((Vue.js: Components Basics, 2023) (React: Thinking in React, 2023))

4.4 Ugotovitve

Tabela 1: Ugotovitve zahtevnosti učenja uporabe orodja.

Platforma	Vue	ReactJS
Udemy	1664 tečajev	9776 tečajev
Coursera	52 tečajev	605 tečajev
Cobiss	121 knjig/e-knjig	243 knjig/e-knjig
Stack Overflow	140956 objav	782404 objav
Reddit	108900 objav	421000 objav
Facebook	186100 uporabnikov	715000 uporabnikov
Virtual DOM	DA	DA

Vir: ((Udemy: Results for vue, 2023) (Udemy: Results for react, 2023) (Coursera: vue, 2023) (Coursera: react, 2023) (Cobiss: Vue, 2023) (Cobiss: React, 2023) (Stack Overflow: vue, 2023) (Stack Overflow: react, 2023) (Reddit: vue, 2023) (Reddit: react, 2023) (Facebook: vue, 2023) (Facebook: react, 2023) (Legacy React, 2023) (Vue.js, 2023)

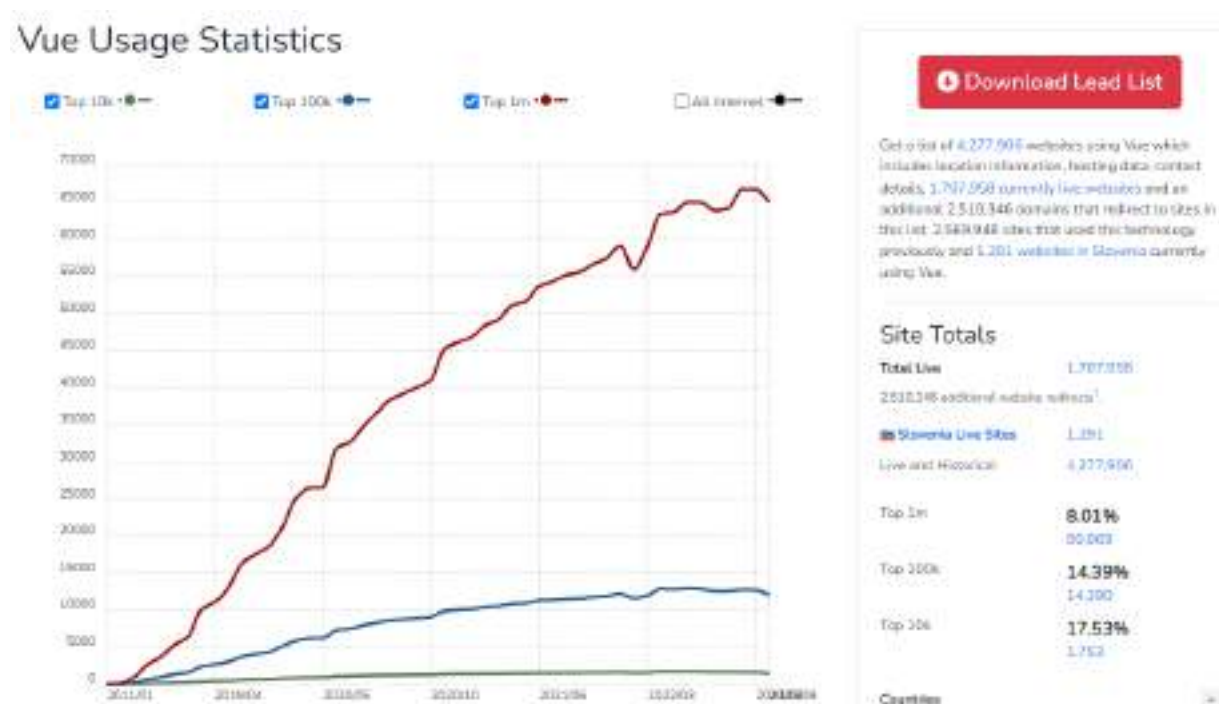
Zahtevnost učenja uporabe orodij je odvisna od več dejavnikov. Najprej sem raziskal dostop do plačljivega in prosto dostopnega učnega gradiva, potem sem raziskal skupnosti obeh orodij na priljubljenih platformah in nato primerjal kompleksnost. Ugotavljam, da obstaja mnogo več gradiva, objav, tečajev in večja skupnost za knjižnico ReactJS kot za ogrodje Vue. Kljub temu je na voljo dovolj vsega naštetega tudi za Vue za primerljivo učno uspešnost. Kompleksnost obeh orodij je podobna, je pa nekaj razlik, ki vplivajo na zahtevnost njunega učenja. Vue uporablja preprostejšo sintakso, ki temelji na html, med tem ko ReactJS uporablja JSX. Začetnik (angl. Junior) front-end razvijalec se bo lažje in hitreje naučil uporabe ogrodja Vue kot knjižnice ReactJS, le-ta ima namreč bolj zapleteno sintakso in zahteva poglobljeno razumevanje Javascript konceptov.

5 TRŽNI POLOŽAJ

Tržni položaj je status ali položaj storitve, izdelka, tehnologije ali podjetja v določeni panogi ali na določenem trgu. Položaj obeh orodij bom ocenil na podlagi pogostosti uporabe pri izdelavi aktivnih spletnih mest, priljubljenosti med razvijalci, možnosti za zaposlitev, skupnosti in na podlagi tega, katera večja podjetja uporabljajo katero orodje.

5.1 Razširjenost in priljubljenost

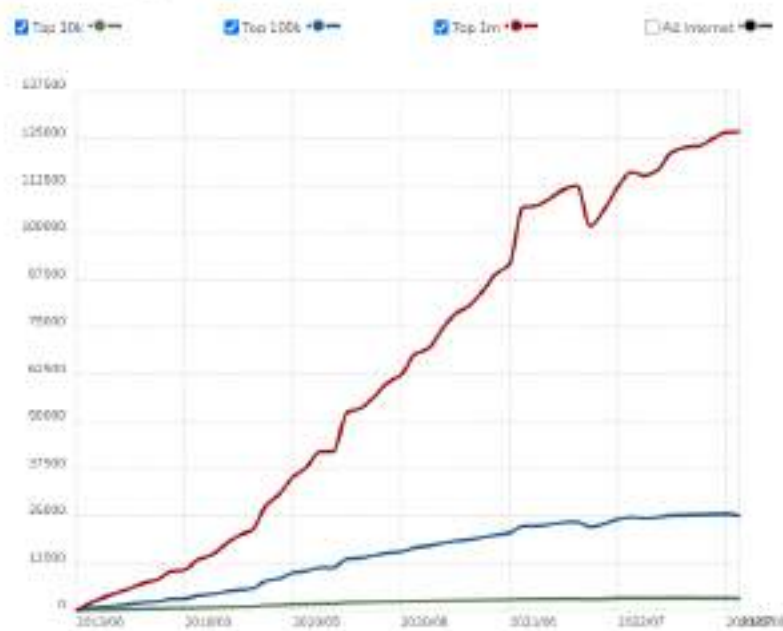
5.1.1 Spletne strani izdelane z Vue in z ReactJS



Slika 24: Število spletnih strani, izdelanih z Vue.

Vir: (Built With: Vue Usage Statistics, 2023)

React Usage Statistics



Slika 25: Število spletnih strani, izdelanih z ReactJS.

Vir: (Built With: React Usage Statistics, 2023)

Spletna stran trends.builtwith.com nudi vpogled v to, koliko spletnih strani je izdelanih s pomočjo katerega orodja. ((Built With: Vue Usage Statistics, 2023) (Built With: React Usage Statistics, 2023))

Tabela 2: Pregled skupnega števila aktivnih spletnih mest orodij.

	Vue	ReactJS
Število aktivnih spletnih mest v Sloveniji	1291	5200
Skupno število aktivnih spletnih mest	1707958	13441975
Število aktivnih spletnih mest, ki so med 10.000 najbolj obiskanimi spletnimi mesti na svetu	1753 (17,53 %)	3810 (38,10 %)

Vir: ((Built With: Vue Usage Statistics, 2023) (Built With: React Usage Statistics, 2023))

Iz podatkov je razvidno, da je tako v Sloveniji kot tudi v tujini za izdelavo spletnih mest knjižnica ReactJS uporabljena pogosteje kot ogrodje Vue.

5.1.2 Priljubljenost med razvijalci

Stack Overflow je leta 2023 naredil spletno anketo o priljubljenosti različnih tehnologij, ki je vsebovala področje »Spletna ogrodja in tehnologije«. Anketirani so med drugim bili vprašani, katero orodje uporabljajo za razvoj. Anketiranih je bilo okrog 90.000 uporabnikov. (Stack Overflow: 2023 Developer Survey, 2023).

Tabela 3: Priljubljenost orodij med razvijalci različnega nivoja znanja.

	Vue	ReactJS
Začetniki	9,69 %	36,66 %
Profesionalni razvijalci	17,64 %	42,87 %
Ostali	12,61 %	30,13 %
Skupno	16,38 % (14742 uporabnikov)	40,58 % (36522 uporabnikov)

Vir: (Stack Overflow: 2023 Developer Survey, 2023)

V tabeli sem prikazal le podatke za Vue in ReactJS, ker drugih ne potrebujem. Razvidno je, da razvijalci na vseh nivojih znanja pogosteje uporabljajo knjižnico ReactJS kot ogrodje Vue.

5.1.3 Priljubljenost med večjimi podjetji

Tabela 4: Uporaba orodij pri večjih podjetjih.

	Vue	ReactJS
1	Facebook	Dropbox
2	Netflix	Instagram

3	Nintendo	Airbnb
4	Adobe	Discord
5	Alibaba	Walmart
6	BMW	Wix
7	EuroNews	Pinterest
8	WizzAir	Skype
9	Upwork	Gyroscope
10	Grammarly	Myntra

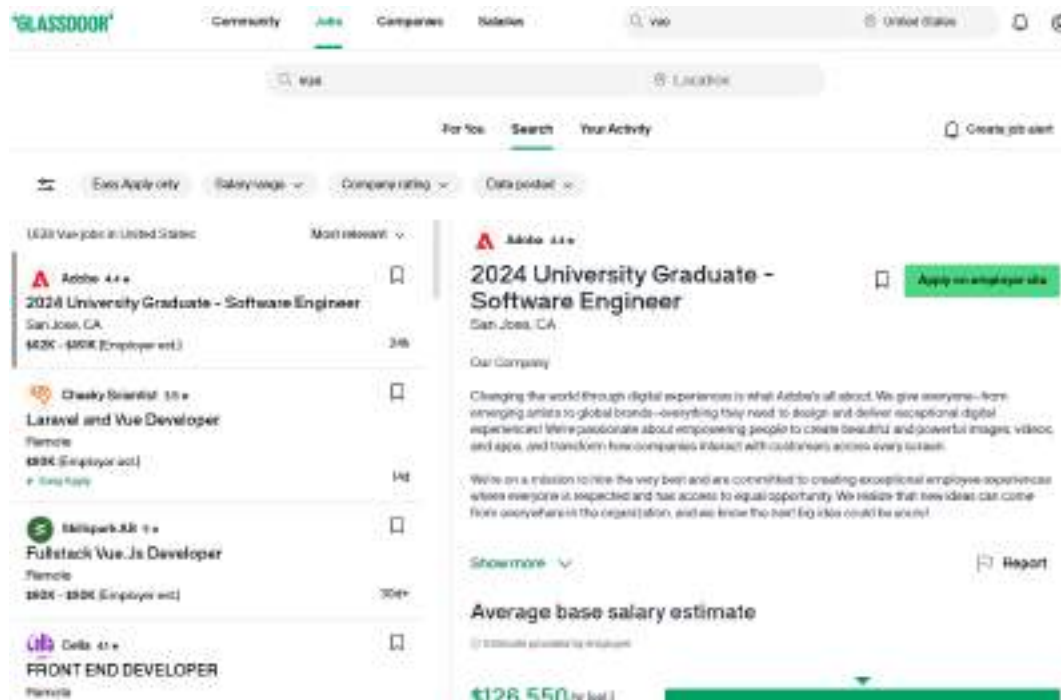
Vir: ((TatvaSoft: Top 10 Global Companies using Vue js, 2023) (TatvaSoft: Top 10 Companies that use React, 2023))

Med večjimi podjetji sta zastopani obe orodji. ((TatvaSoft: Top 10 Global Companies using Vue js, 2023) (TatvaSoft: Top 10 Companies that use React, 2023))

5.2 ***Možnost zaposlitve***

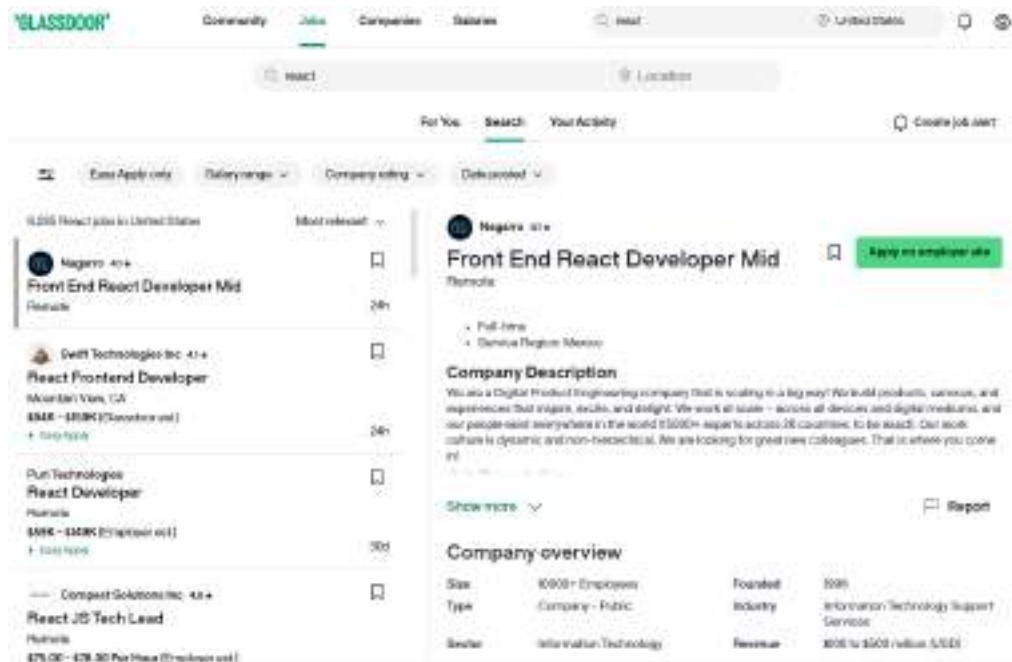
V tem poglavju bom primerjal povpraševanje delodajalcev po razvijalcih, ki uporabljajo Vue in razvijalcih, ki uporabljajo ReactJS.

5.2.1 Glassdoor



Slika 26: Glassdoor – možnost zaposlitve Vue.

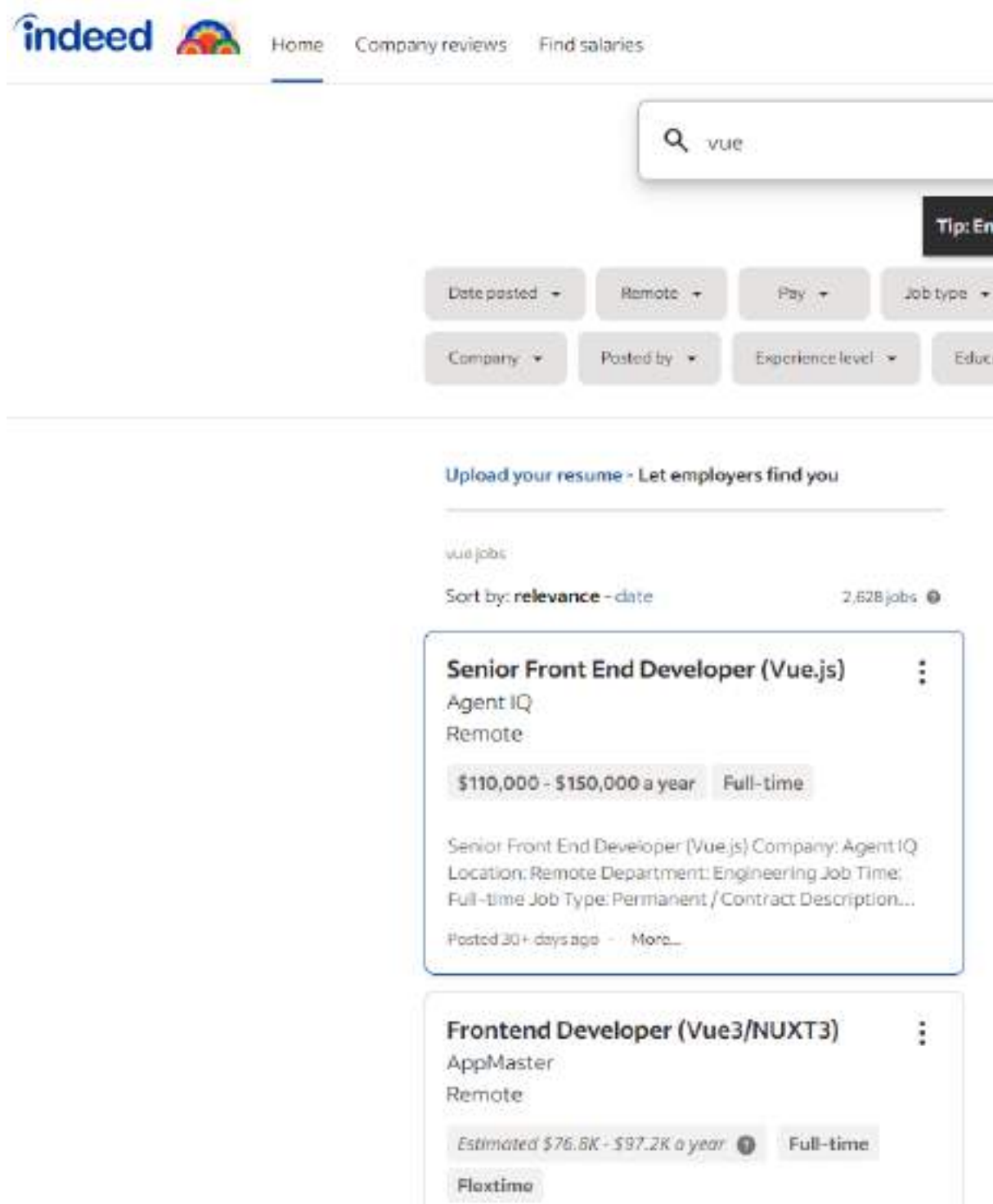
Vir: (Glassdoor: All »vue« Results, 2023)



Slika 27: Glassdoor – možnost zaposlitve ReactJS.

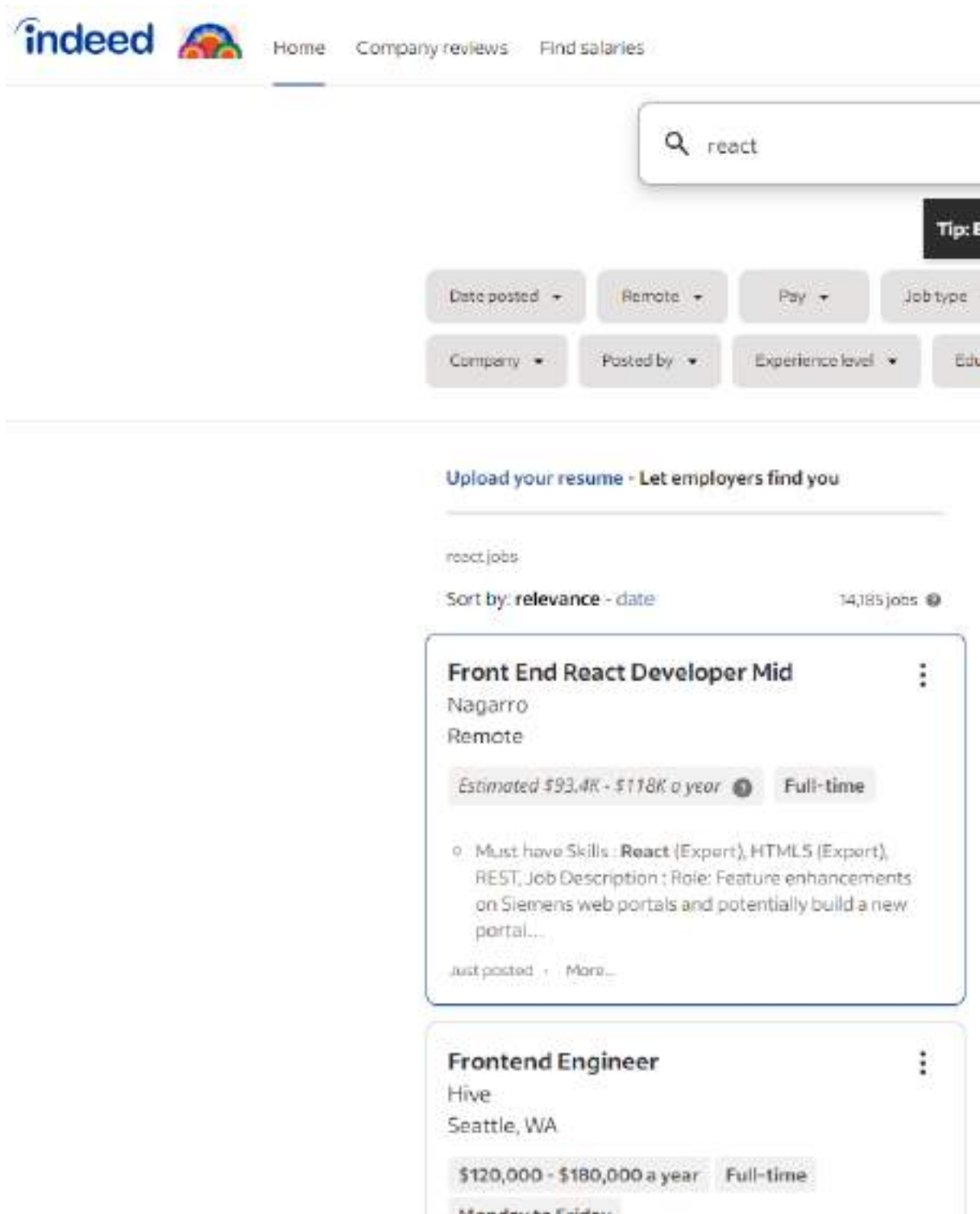
Vir: (Glassdoor: All »react« Results, 2023)

5.2.2 Indeed



Slika 28: Indeed – možnost zaposlitve Vue.

Vir: (Indeed: vue, 2023)

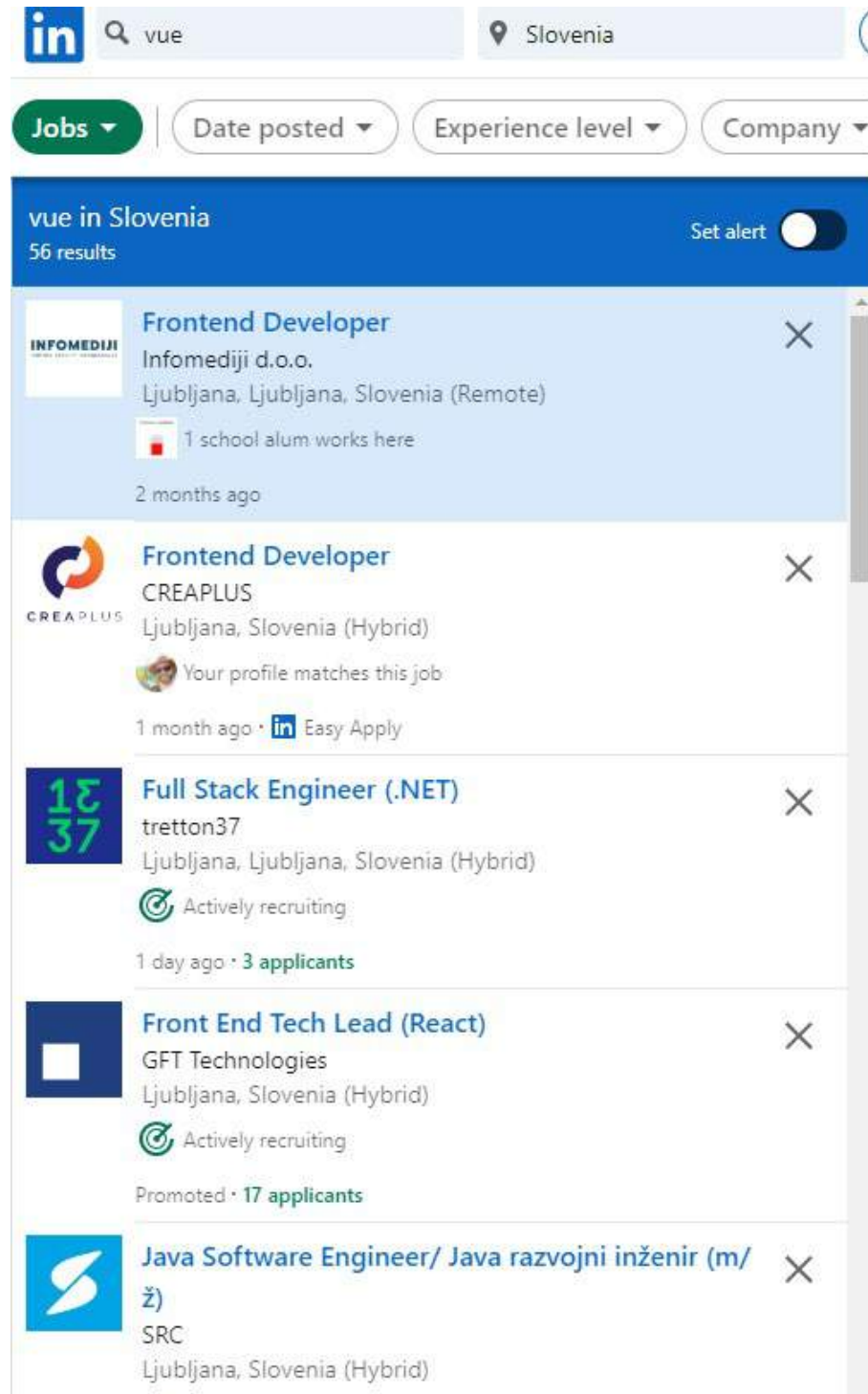


Slika 29: Indeed – možnost zaposlitve ReactJS.

Vir: (Indeed: react, 2023)

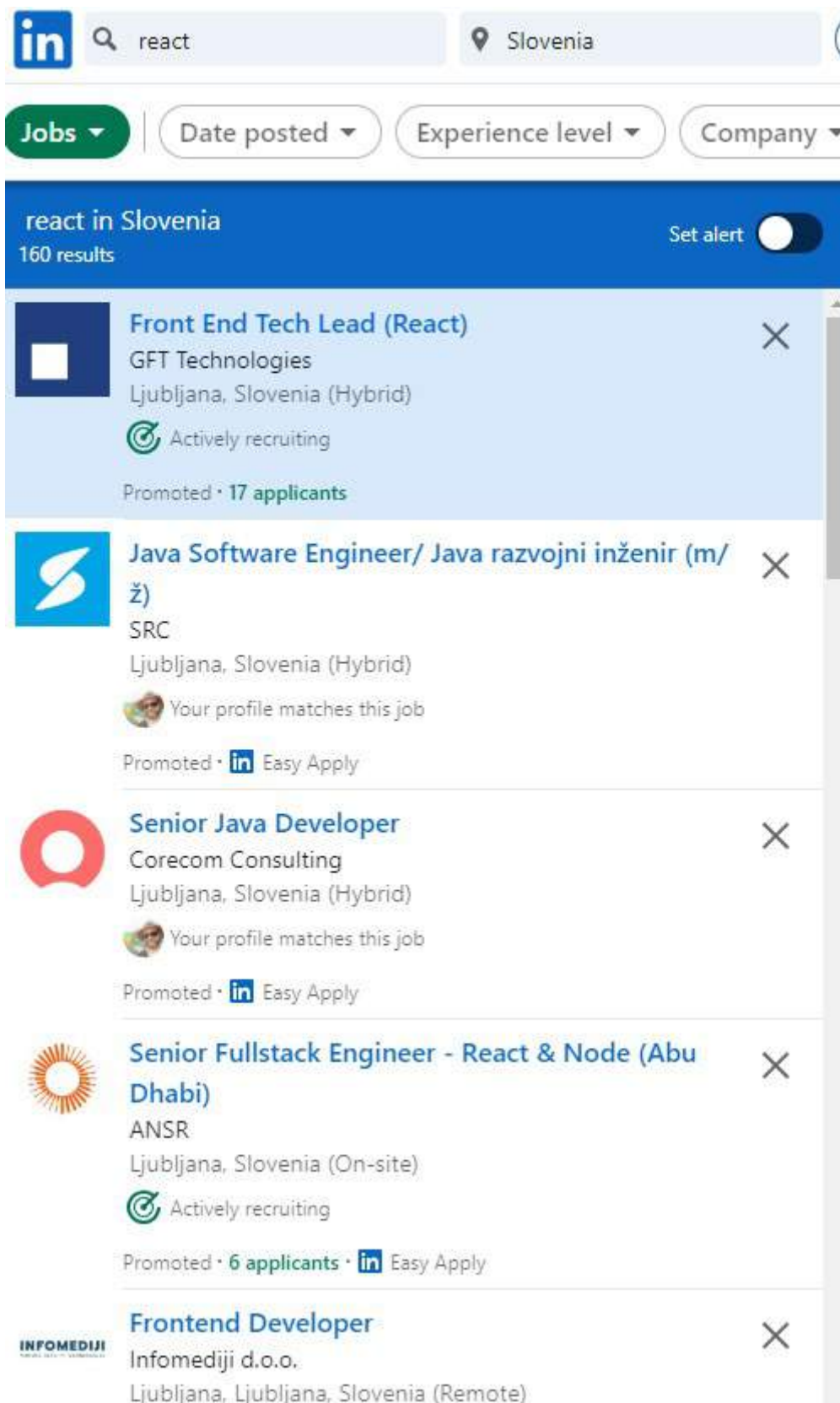
5.2.3 LinkedIn

5.2.3.1 Slovenija



Slika 30: LinkedIn Slovenija – možnost zaposlitve Vue.

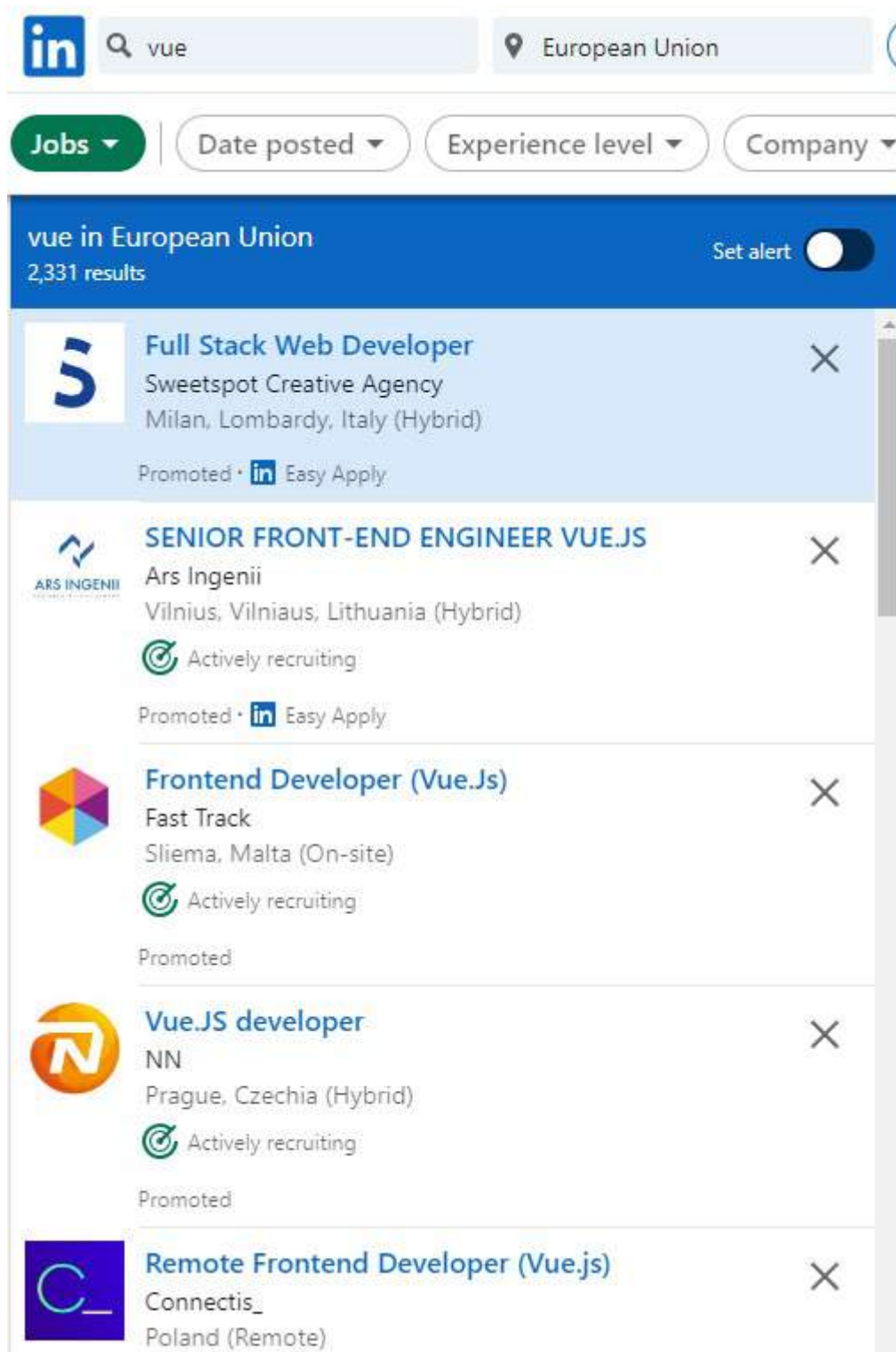
Vir: (LinkedIn: vue js, Slovenia, Jobs, 2023)



Slika 31: LinkedIn Slovenija – možnost zaposlitve ReactJS.

Vir: (LinkedIn: react, Slovenia, Jobs, 2023)

5.2.3.2 Evropska unija



Slika 32: LinkedIn Evropska unija – možnost zaposlitve Vue.

Vir: (LinkedIn: vue js, European Union, Jobs, 2023)

in react European Union

Jobs ▾ Date posted ▾ Experience level ▾ Company ▾

react in European Union 2,350 results Set alert ☐



Front End Tech Lead (React)
GFT Technologies
Ljubljana, Slovenia (Hybrid)
Actively recruiting
Promoted • 17 applicants





Front-endontwikkelaar
StarApple
Rijswijk, South Holland, Netherlands (Hybrid)
Actively recruiting
Promoted • Easy Apply





Junior React Native Engineer
Picnic Technologies
Amsterdam, North Holland, Netherlands (On-site)
Actively recruiting
Promoted





Internship - Node JS / React JS Developer
SES Satellites
Betzdorf, Grevenmacher, Luxembourg (Hybrid)
Actively recruiting
Promoted





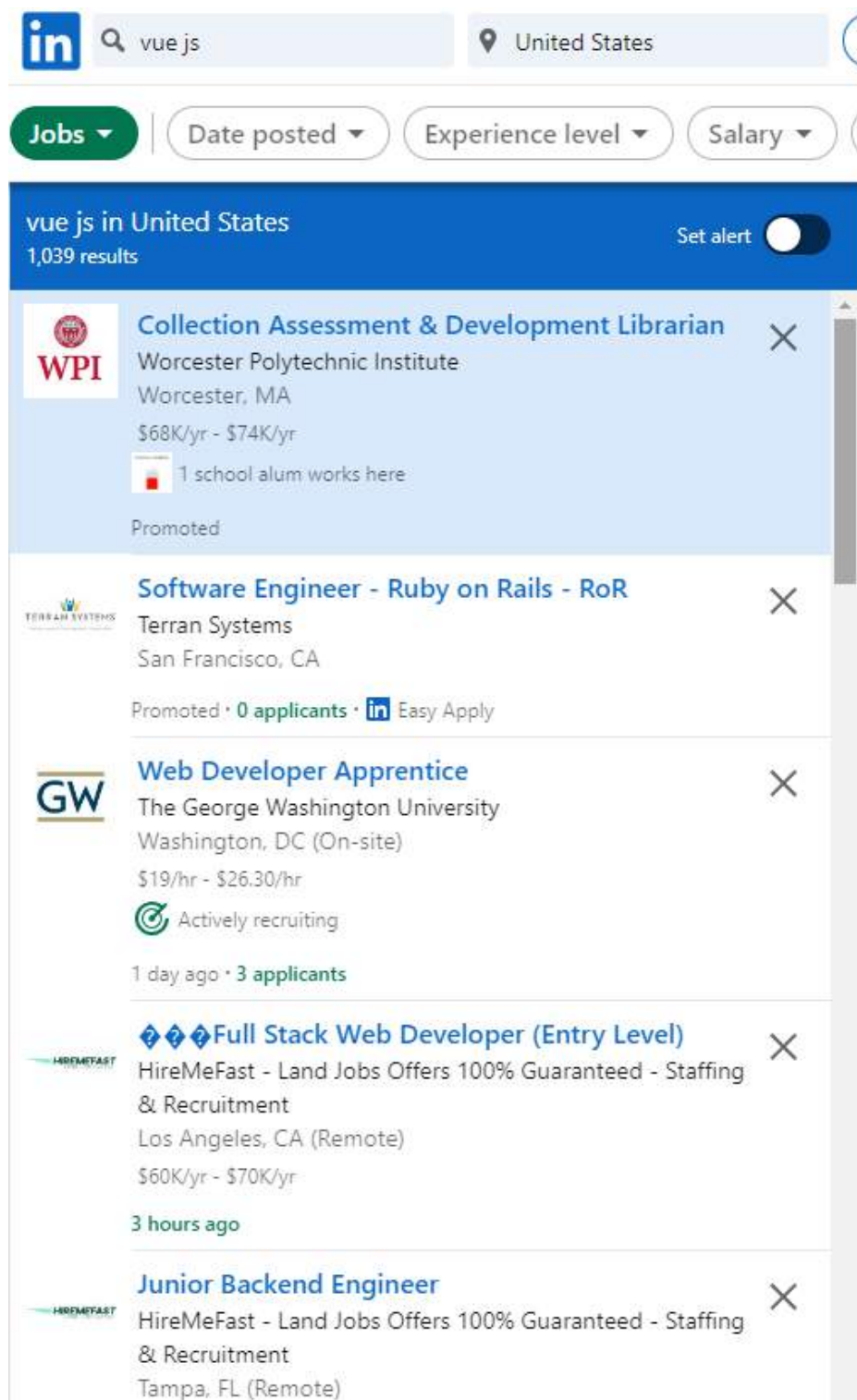
Junior Frontend Developer (React) - Student (m/f/d)
b2match
Zagreb, Zagreb, Croatia (Hybrid)



Slika 33: LinkedIn Evropska unija – možnost zaposlitve ReactJS.

Vir: (LinkedIn: react, European Union, Jobs, 2023)

5.2.3.3 Združene države Amerike



Slika 34: LinkedIn ZDA – možnost zaposlitve Vue.

Vir: (LinkedIn: vue.js, United States, Jobs, 2023)

in react United States

Jobs | Date posted | Experience level | Salary

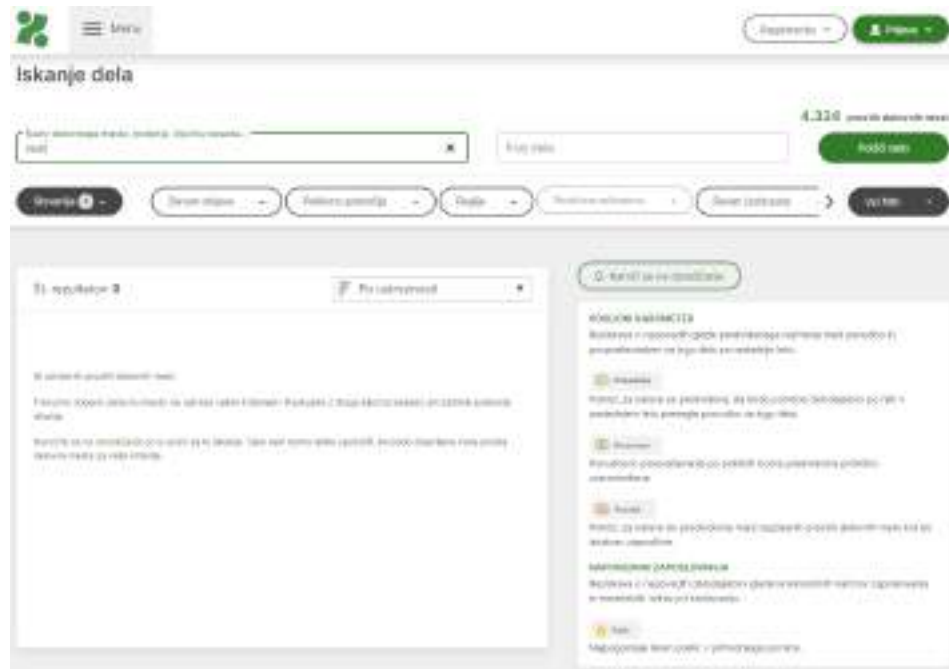
react in United States 1,552 results Set alert

- Junior React Developer**
HireMeFast - Land Jobs Offers 100% Guaranteed - Staffing & Recruitment
Los Angeles, CA (Remote)
\$67K/yr - \$75K/yr
3 hours ago
- React/React Native Engineer**
RBA, Inc.
Minnesota, United States (Remote)
Actively recruiting
1 hour ago · Easy Apply
- ReactJs Developer**
Diverse Lynx
Hartford, CT (Hybrid)
Actively recruiting
6 hours ago
- Full Stack Engineer (Node.JS, React.JS and Vue.JS)- ALT-012**
Five Cubes
Mountain View, CA (On-site)
\$55/hr - \$65/hr
6 hours ago
- React Web Developer - 100% REMOTE!**
Perceptive Recruiting, LLC
United States (Remote)

Slika 35: LinkedIn ZDA – možnost zaposlitve ReactJS.

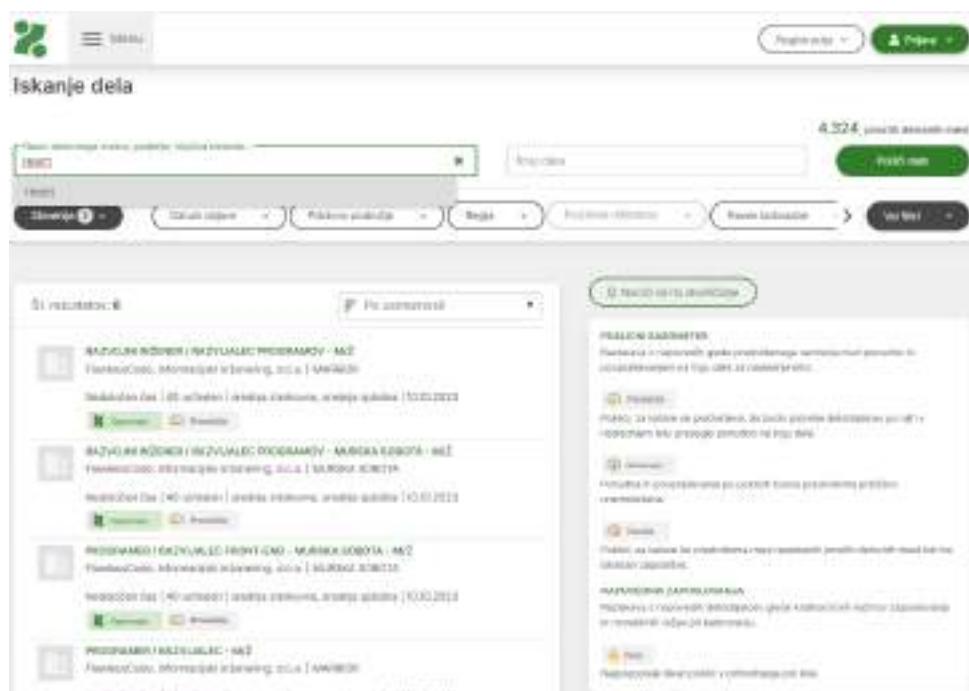
Vir: (LinkedIn: react, United States, Jobs, 2023)

5.2.4 Zavod Republike Slovenije za zaposlovanje



Slika 36: Zavod za zaposlovanje SLO – možnost zaposlitve Vue.

Vir: (Iskalci zaposlitve, dobrodošli, 2023)



Slika 37: Zavod za zaposlovanje SLO – možnost zaposlitve ReactJS.

Vir: (Iskalci zaposlitve, dobrodošli, 2023)

5.2.5 Pregled

Tabela 5: Pregled možnosti zaposlitve.

	Vue	ReactJS
Glassdoor (Združene Države Amerike)	1638	8285
Indeed (globalno)	2628	14185
LinkedIn (Slovenija)	56	160
LinkedIn (Evropska Unija)	2331	2350
LinkedIn (Združene Države Amerike)	1039	1552
Zavod Republike Slovenije za zaposlovanje (Slovenija)	0	6

Vir: ((Glassdoor: All »vue« Results, 2023) (Glassdoor: All »react« Results, 2023) (Indeed: vue, 2023) (Indeed: react, 2023) (LinkedIn: vue js, Slovenia, Jobs, 2023) (LinkedIn: react, Slovenia, Jobs, 2023) (LinkedIn: vue js, European Union, Jobs, 2023) (LinkedIn: react, European Union, Jobs, 2023) (LinkedIn: vue js, United States, Jobs, 2023) (LinkedIn: react, United States, Jobs, 2023) (Iskalci zaposlitve, dobrodošli, 2023))

Ugotavljam, da je tako med domačimi kot med tujimi delodajalci potreba po razvijalcih, ki uporabljajo knjižnico ReactJS veliko večja kot po razvijalcih, ki uporabljajo ogrodje Vue. Presenečajo me podatki za Evropsko Unijo in Združene Države Amerike na LinkedIn-u, saj indicirajo, da je število ponudb za delo pri obeh orodjih podobno, kljub temu da je na drugih platformah razlika v prid ReactJS veliko bolj izrazita.

5.3 Skupnost

Skupnost je pomemben dejavnik pri določanju tržnega položaja. Na tem področju sem orodji primerjal že v četrtem poglavju, natančneje v podnaslovu 4.2 Skupnost. Ugotovil sem, da ima ReactJS tako na družbenem omrežju Facebook kot tudi na priljubljenih platformah Reddit in Stack Overflow občutno večjo skupnost kot Vue.

5.4 Ugotovitve

Analiza pridobljenih podatkov kaže na to, da je bil ob času pisanja ReactJS uporabljen v izdelavi več aktivnih spletnih mest kot Vue. Razvidno je tudi, da je ReactJS med razvijalci spletnih aplikacij vseh nivojev znanja bolj priljubljen kot Vue, ima večjo skupnost in razvijalcu nudi boljšo možnost zaposlitve tako v tujini kot v Sloveniji. Obe orodji sta zastopani s strani večjih podjetij, na tem področju nisem ugotovil razlik, ki bi bile predmetne za mojo raziskavo. Zaradi naštetih dejavnikov ima ReactJS pred Vue konkurenčno prednost. Posledično ugotavljam, da je tržni položaj ReactJS-a močnejši.

6 PRIMERJAVA SPLETNIH APLIKACIJ, IZDELANIH V VUE IN REACTJS

6.1 *Obremenjenost sistemov gostitelja in odjemalca*

Obremenjenost sistemov je eden od najpomembnejših faktorjev pri določanju uporabne vrednosti orodij. Meritve sem naredil na aplikaciji, izdelani s pomočjo Vue in na aplikaciji, izdelani s pomočjo ReactJS, nato pa sem v obeh aplikacijah število podatkov povečal na 1000 % in meritve ponovil. Namen povečanja podatkov je, da ugotovim, ali se aplikaciji obnašata drugače pri različnih količinah podatkov in katero orodje je bolj primerno za obdelavo podatkov *Big Data*.

6.1.1 *Gostitelj*

Obremenjenost sistema gostitelja sem izmeril v brskalniku Google Chrome z orodjem Windows Performance Monitor. Parametri, ki sem jih meril, so centralna procesna enota oz. CPE (angl. Central Processing Unit – CPU), pomnilnik (angl. Memory) in omrežje. Vse meritve v tem podpoglavju sem izvedel po enakem principu: najprej sem aplikacijo odprl v brskalniku, nato sem začel dve minuti merjenja. Deset sekund po začetku sem zavihek, na katerem je bila odprta aplikacija, ročno osvežil, nato pa počakal, da se je merjenje po preteku dveh minut samodejno zaključilo.

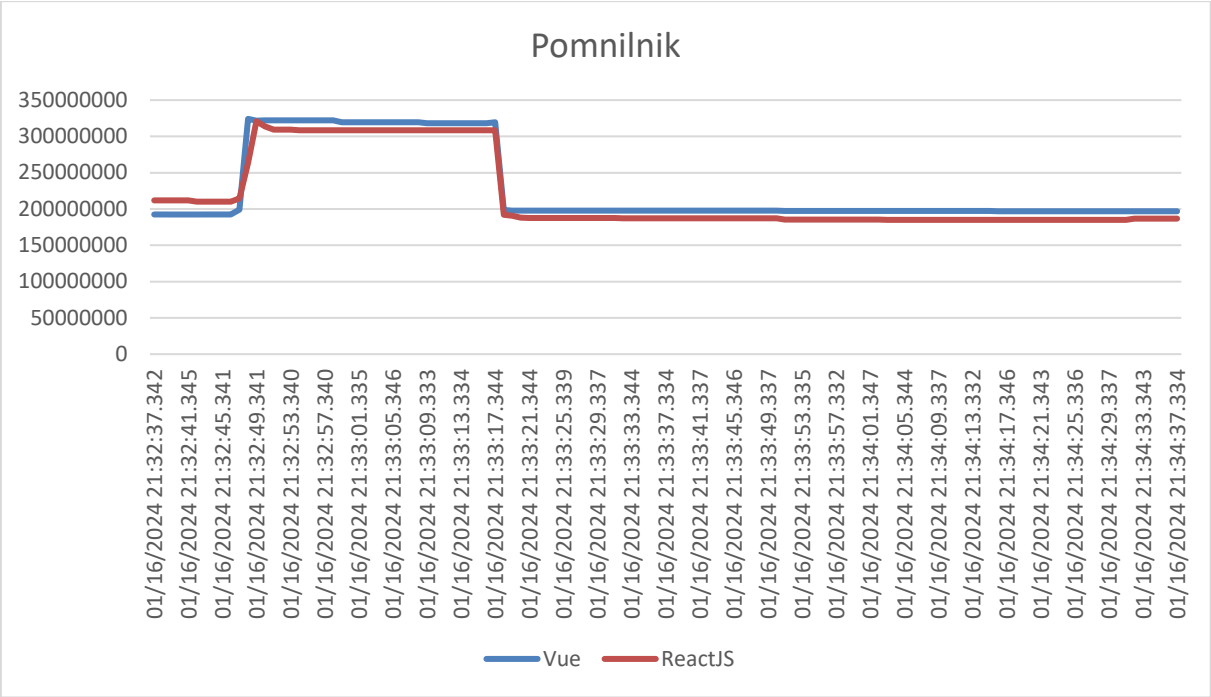
6.1.1.1 *Priprava orodja in postopek merjenja*

V orodju Windows Performance Monitor sem najprej ustvaril štiri uporabniško določene zbirnike podatkov (angl. User Defined Data Collector Set), saj je bilo potrebno izmeriti dva različna parametra v dveh različnih aplikacijah. V lastnostih sem pri vseh definiriral interval meritev na eno sekundo, čas merjenja na dve minuti in obliko izhodnega zapisa v dnevnik podatkov (angl. Data Log) iz binarne na »Ločeno z vejico« za lažji uvoz v Excel, s pomočjo katerega sem nato izrisal grafe. Vsakemu zbirniku podatkov sem dodal en merilnik (angl. Counter), kateremu sem določil parameter (CPE ali pomnilnik) in proces, katerega mora meriti, v tem primeru zavihek brskalnika Google Chrome, na katerem sem imel odprto aplikacijo (izdelano s pomočjo Vue ali ReactJS).

6.1.1.2 Meritve

6.1.1.2.1 Osnovna količina podatkov (1x)

6.1.1.2.1.1 Pomnilnik



Graf 1: Obremenjenost pomnilnika pri gostitelju.

Vir: Lasten

6.1.1.2.1.2 Primerjava

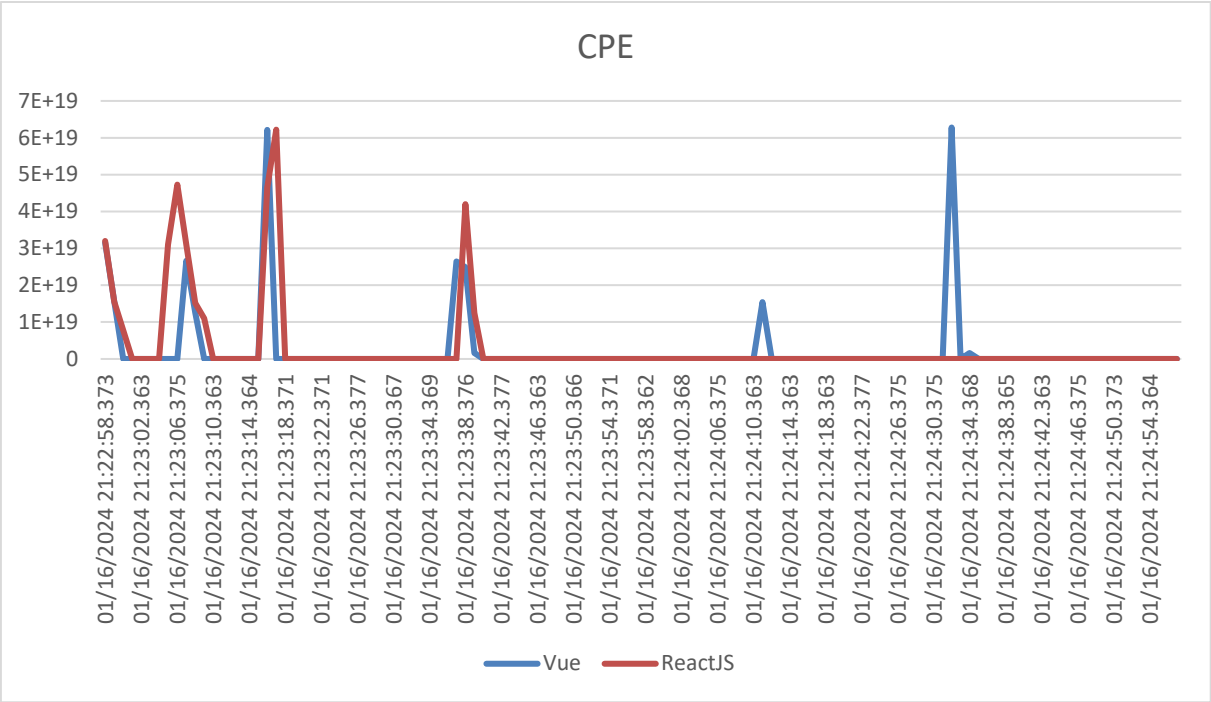
Vrednosti so v bajtih.

Tabela 6: Primerjava obremenjenosti pomnilnika pri gostitelju.

	Vue	ReactJS
Minimalna obremenitev	192389120 B	184995840 B
Maksimalna obremenitev	324132864 B	320827392 B
Povprečna obremenitev	227406331,8 B	218572139,9 B

Vir: Lasten

6.1.1.2.1.3 Centralna Procesna Enota



Graf 2: Obremenjenost CPE pri gostitelju.

Vir: Lasten

6.1.1.2.1.4 Primerjava

Vrednosti so v CPE ciklih.

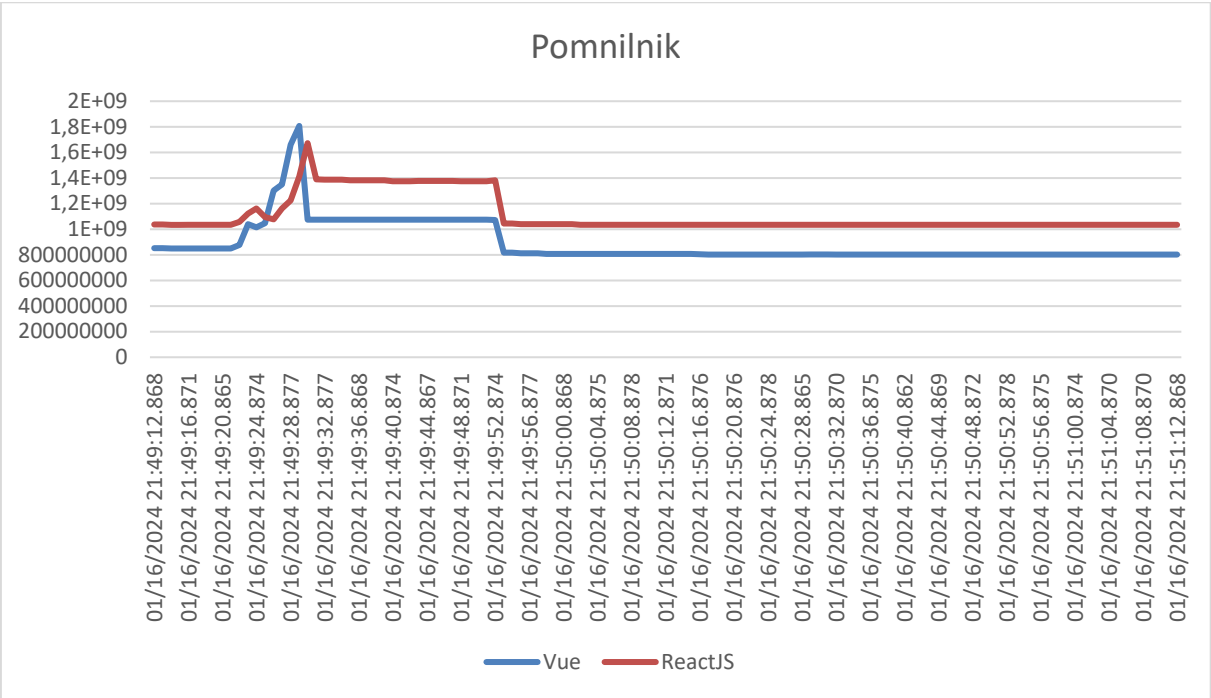
Tabela 7: Primerjava obremenjenosti CPE pri gostitelju.

	Vue	ReactJS
Minimalna obremenitev	0,00 CPE ciklov	0,00 CPE ciklov
Maksimalna obremenitev	62786305553323100000,00 CPE ciklov	62203890818536500000,00 CPE ciklov
Povprečna obremenitev	2342790004933610000,00 CPE ciklov	2949727074755800000,00 CPE ciklov

Vir: Lasten

6.1.1.2.2 Povečana količina podatkov (10x)

6.1.1.2.2.1 Pomnilnik



Graf 3: Obremenjenost pomnilnika pri gostitelju (10x količina podatkov).

Vir: Lasten

6.1.1.2.2.2 Primerjava

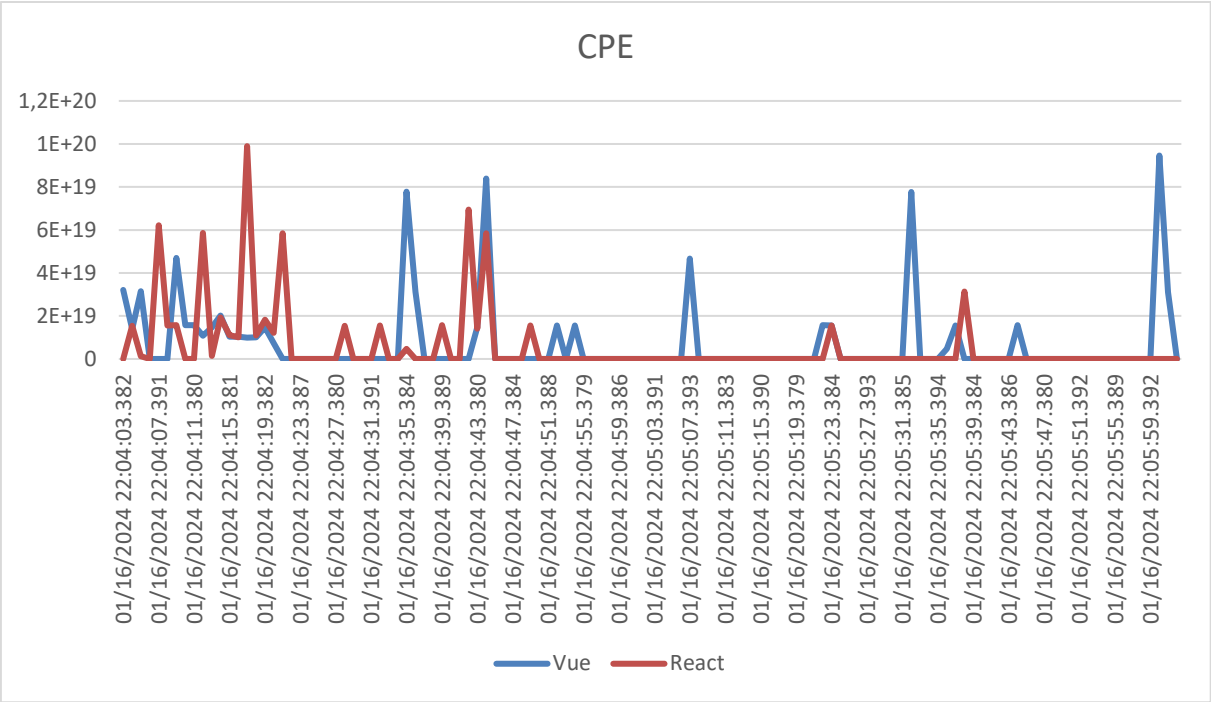
Vrednosti so v bajtih.

Tabela 8: Primerjava obremenjenosti pomnilnika pri gostitelju (10x količina podatkov).

	Vue	ReactJS
Minimalna obremenitev	802869248 B	1034006528 B
Maksimalna obremenitev	1807007744 B	1672425472 B
Povprečna obremenitev	889800043,9 B	1111689334 B

Vir: Lasten

6.1.1.2.2.3 Centralna Procesna Enota



Grat 4 : Obremenjenost CPE pri gostitelju (10x količina podatkov).

Vir: Lasten

6.1.1.2.2.4 Primerjava

Vrednosti so v CPE ciklih.

Tabela 9: Primerjava obremenjenosti CPE pri gostitelju (10x količina podatkov).

	Vue	ReactJS
Minimalna obremenitev	0,00 CPE ciklov	0,00 CPE ciklov
Maksimalna obremenitev	94599560815862900000,00 CPE ciklov	98965928575629600000,00 CPE ciklov
Povprečna obremenitev	6836706715642420000,00 CPE ciklov	5546749590367540000,00 CPE ciklov

Vir: Lasten

6.1.2 Odjemalec

Obremenjenost sistema odjemalca sem izmeril v brskalniku Google Chrome z uporabo integriranih orodij Chrome Task Manager in Chrome DevTools.

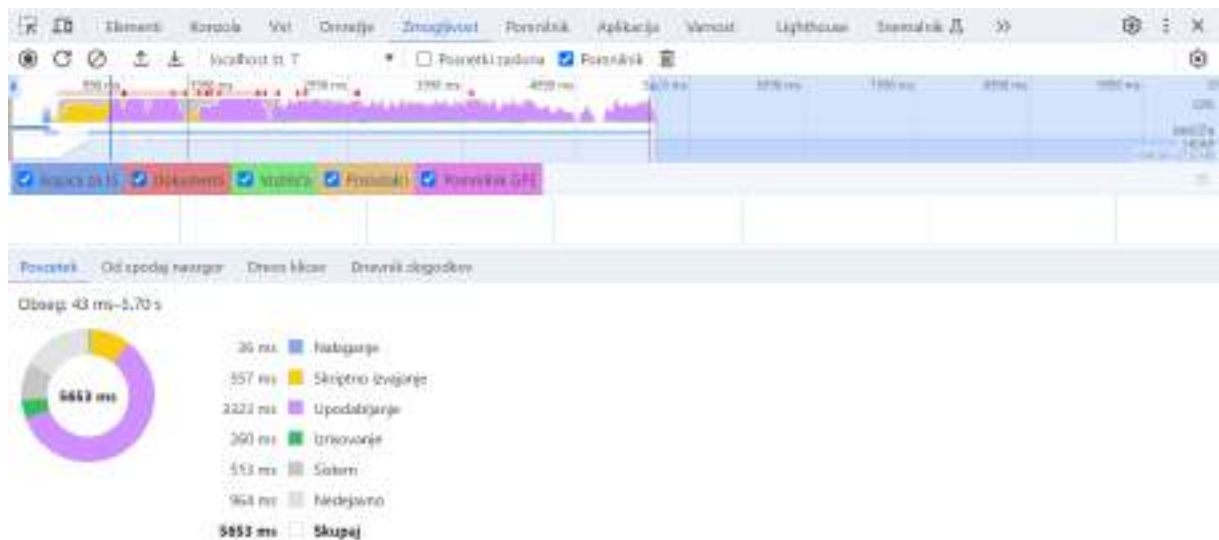
6.1.2.1 Priprava orodja in postopek merjenja

V brskalniku Google Chrome sem zagnal eno od svojih aplikacij, nato sem z bližnjico »ctrl+shift+i« odprl Chrome DevTools. V orodju sem odprl zavihek »Omrežje« in obkljukal »Onemogoči predpomnilnik« zato, da vsakič, ko se stran osveži, v celoti naloži vso vsebino. Nato sem odprl zavihek »Zmogljivost« in v njem obkljukal polje »Pomnilnik«. Z bližnjico »ctrl+shift+e« sem sprožil profiliranje in osvežitev aplikacije. Funkcija deluje tako, da hkrati začne merjenje parametrov in osveževanje aplikacije, ko se ta v celoti naloži, pa se merjenje samodejno zaključi in izpišejo se meritve. Parametri, ki so s to metodo merjeni, so pomnilnik, CPE in čas nalaganja po sklopih. O času nalaganja več v poglavju 6.2. Po tem sem v brskalniku Google Chrome odprl obe aplikaciji in z bližnjico »shift+esc« zagnal Google Chrome Upravitelj Opravil (angl. Google Chrome Task Manager), s pomočjo katerega sem zabeležil vrednosti pomnilnika in CPE obeh aplikacij v mirovanju oz. že v naloženem stanju. Postopek sem ponovil še pri povečani količini podatkov in nato vse ponovil še z drugo aplikacijo.

6.1.2.2 Meritve

6.1.2.2.1 Osnovna količina podatkov (1x)

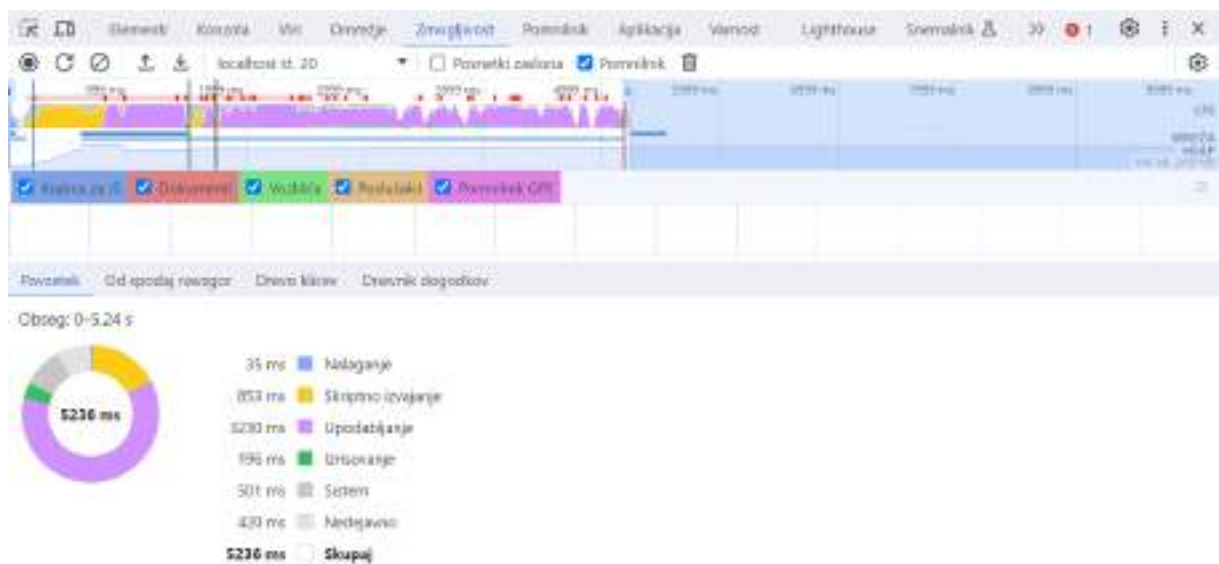
6.1.2.2.1.1 Pomnilnik in CPE med nalaganjem Vue



Slika 38: Obremenjenost pomnilnika in CPE pri odjemalcu – Vue.

Vir: Lasten

6.1.2.2.1.2 Pomnilnik in CPE med nalaganjem ReactJS



Slika 39: Obremenjenost pomnilnika in CPE pri odjemalcu – ReactJS.

Vir: Lasten

6.1.2.2.1.3 Pomnilnik in CPE v mirovanju

Opravo	Količina uporabljenega pomnilnika	CPE	Omrežje
 Zavihek: rick-and-morty-vue	197.556 KB	0.0	0
 Zavihek: React App	240.076 KB	0.0	0

Slika 40: Obremenjenost pomnilnika in CPE pri odjemalcu v mirovanju.

Vir: Lasten

6.1.2.2.1.4 Primerjava

Vrednosti pomnilnika so v kilobajtih, CPE pa v milisekundah (čas, ki ga procesor potrebuje, da naloži aplikacijo).

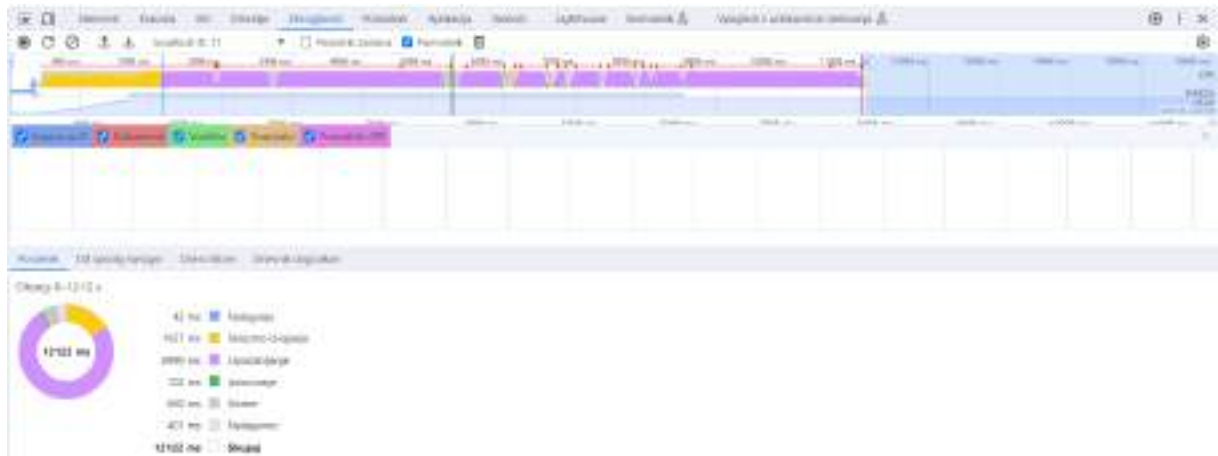
Tabela 10: Primerjava obremenjenosti pomnilnika in CPE pri odjemalcu v mirovanju.

	Vue	ReactJS
Pomnilnik	646 KB – 27600 KB	646 KB – 24000 KB
CPE	5653 ms	5236 ms
Pomnilnik v mirovanju	197,556 KB	240,076 KB

Vir: Lasten

6.1.2.2.2 Povečana količina podatkov (10x)

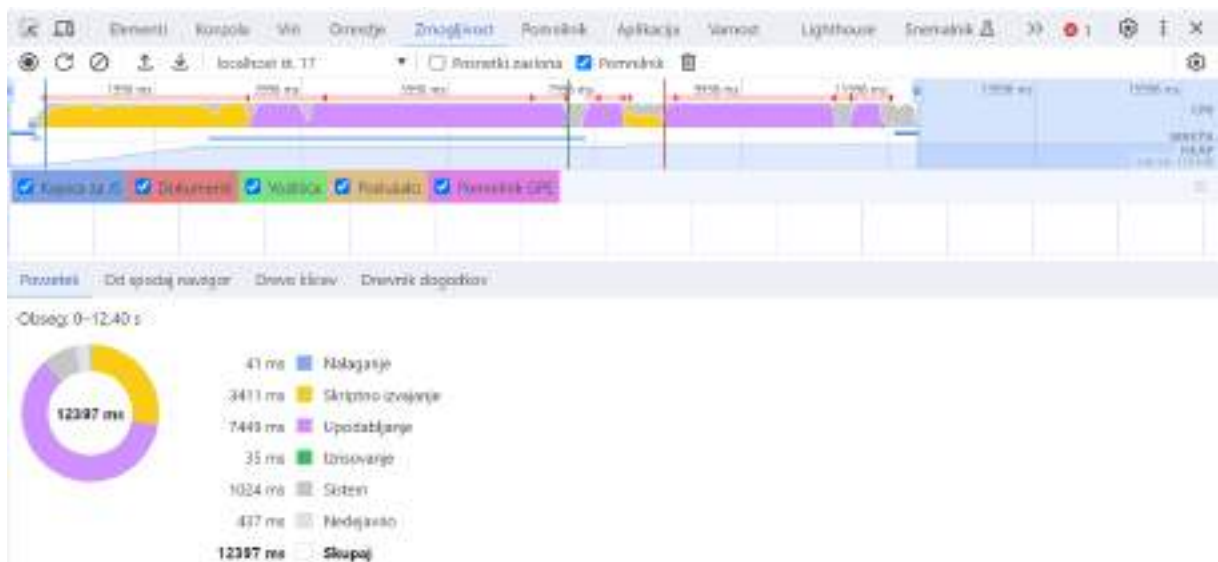
6.1.2.2.2.1 Pomnilnik in CPE med nalaganjem Vue



Slika 41: Obremenjenost pomnilnika in CPE pri odjemalcu – Vue (10x količina podatkov).

Vir: Lasten

6.1.2.2.2.2 Pomnilnik in CPE med nalaganjem ReactJS



Slika 42: Obremenjenost pomnilnika in CPE pri odjemalcu – ReactJS (10x količina podatkov).

Vir: Lasten

6.1.2.2.3 Pomnilnik in CPE v mirovanju

Opravilo ▼	Količina uporabljenega pomnilnika	CPE	Omrežje
•  Zavihek: rick-and-morty-vue	824.088 KB	0.0	0
•  Zavihek: React App	701.380 KB	0.0	0

Slika 43: Obremenjenost pomnilnika in CPE pri odjemalcu v mirovanju (10x količina podatkov).

Vir: Lasten

6.1.2.2.4 Primerjava

Vrednosti pomnilnika so v kilobajtih, CPE pa v milisekundah (čas, ki ga procesor potrebuje, da naloži aplikacijo).

Tabela 11: Primerjava obremenjenosti pomnilnika in CPE pri odjemalcu v mirovanju (10x količina podatkov).

	Vue	ReactJS
Pomnilnik	646 KB – 144000 KB	646 KB – 119000 KB
CPE	12122 ms	12397 ms
Pomnilnik v mirovanju	824,088 KB	701,380 KB

Vir: Lasten

6.2 Hitrost in odzivni čas

Aicontentify, podjetje, ki se ukvarja z optimizacijo spletnih iskalnikov (angl. Search Engine Optimization – SEO) in napredno umetno inteligenco, na svoji spletni strani navaja širok nabor negativnih posledic počasnega nalaganja spletnih aplikacij. Med drugim to, da predolgo nalaganje strani vpliva na uporabnikovo percepcijo znamke, v smislu, da ljudje povezujejo dolgo nalaganje s slabo kvaliteto storitve, nizko kvaliteto produktov ali zastarelo tehnologijo. Sklicuje se tudi na študije, ki govorijo o tem, da že samo nekajsekundna zamuda pri nalaganju znatno vpliva na število uporabnikov, ki zapustijo spletno aplikacijo, brez da bi izvedli

kakršnokoli dejanje na njej. Razlog za to naj bi bil, da uporabniki zelo verjetno izgubijo interes ob predolgem čakanju. Razvidno je, da sta hitrost in odzivni čas ključna faktorja pri ugotavljanju uporabne vrednosti aplikacije.

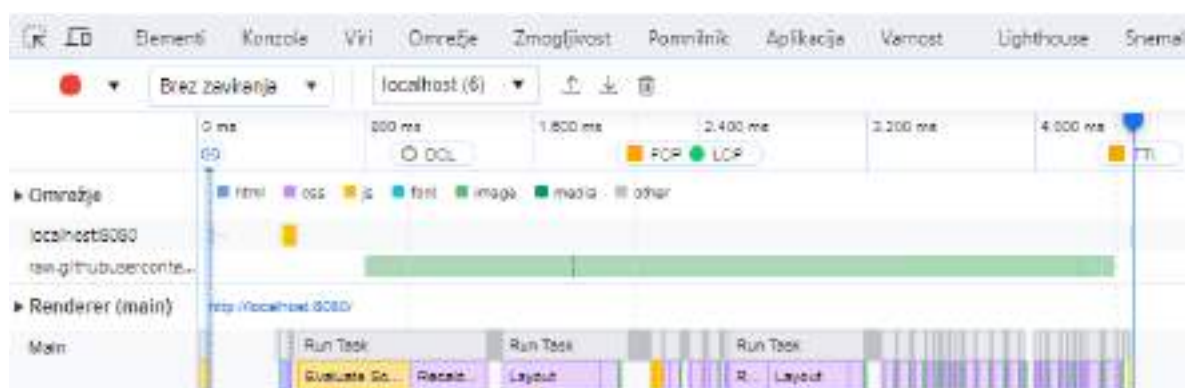
6.2.1 Priprava orodja in postopek merjenja

V brskalniku Google Chrome sem odprl eno od svojih aplikacij, nato sem z bližnjico »ctrl+shift+i« odprl Google Chrome DevTools. V tem orodju sem odprl zavihek »Omrežje« in v nastavitvah zavihka obkljukal polje »Onemogoči predpomnilnik«. Po tem sem v orodju odprl zavihek »Vpogledi v učinkovitost delovanja« in kliknil na »Merjenje nalaganja strani«. Ta funkcija med drugim meri, kako dolgo aplikacija potrebuje, da se naloži do raznih faz uporabnosti. Faze so FCP, LCP in TTI. Postopek sem ponovil še pri povečani količini podatkov in nato vse ponovil še z drugo aplikacijo. Vrednosti meritev so v sekundah.

6.2.2 Meritve

6.2.3 Osnovna količina podatkov (1x)

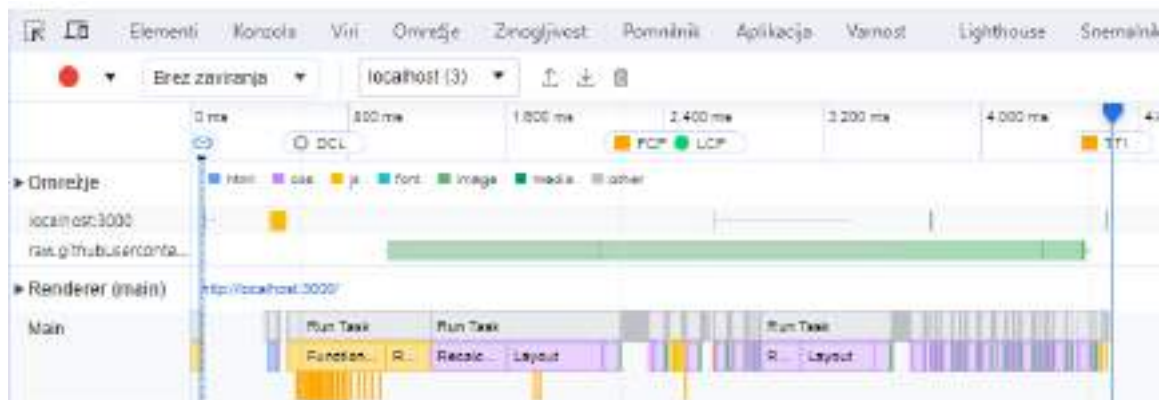
6.2.3.1 Vue



Slika 44: Čas nalaganja Vue.

Vir: Lasten

6.2.3.2 ReactJS



Slika 45: Čas nalaganja ReactJS.

Vir: Lasten

6.2.3.3 Primerjava

Tabela 12: Primerjava časa do faz uporabnosti.

	Vue	ReactJS
FCP	2,02 s	2,12 s
LCP	2,02 s	2,37 s
TTI	4,33 s	4,49 s

Vir: Lasten

6.2.4 Povečana količina podatkov (10x)

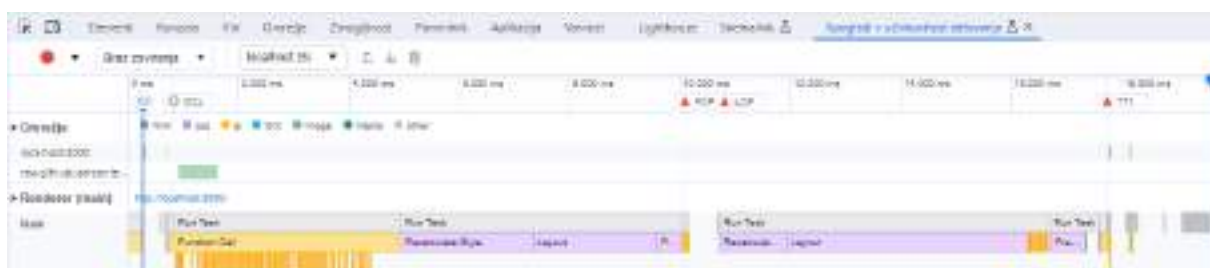
6.2.4.1 Vue



Slika 46: Čas nalaganja Vue (10x količina podatkov).

Vir: Lasten

6.2.4.2 ReactJS



Slika 47: Čas nalaganja ReactJS (10x količina podatkov).

Vir: Lasten

6.2.4.3 Primerjava

Tabela 13: Primerjava časa do faz uporabnosti (10x količina podatkov).

	Vue	ReactJS
FCP	9,83 s	9,81 s
LCP	9,83 s	9,81 s
TTI	17,68 s	17,46 s

Vir: Lasten

6.3 Analiza meritev

Za merjenje obremenjenosti sistemov sem poskušal zagotoviti enake pogoje za obe aplikaciji, vendar pa je potrebno izpostaviti, da sem pri večkratnem izvajanju istih meritev zaznal manjša odstopanja. To pripisujem nihanju v omrežju in v nekaterih delih sistema, ki ga uporabljam. Zagotovitev laboratorijskih pogojev, v katerih bi večkratne meritve izbranega parametra pod istimi pogoji proizvedle enake rezultate presega sredstva, ki so mi na voljo. Med izvajanjem meritev sem ugotovil, da so nekatere razlike v obremenjenosti sistemov razmeroma majhne, a ne zanemarljive. Razumno je domnevati, da bi pri povečani količini podatkov in dodani interaktivnosti in funkcionalnosti aplikacij razlike bile večje. Takšna raziskava bi bila preobširna za diplomsko delo, lahko pa bi nekdo drug nadaljeval, kjer sem jaz končal in raziskavo razširil v to smer. Na tej točki bi dodal, da je bil eden od mojih ciljev tudi primerjava obremenjenosti omrežja gostitelja in odjemalca pri uporabi obeh aplikacij, kar je naslednja tema, ki bi jo lahko v razširitvi mojega dela raziskal nekdo drug. Ta del sem namreč kasneje umaknil iz dokumenta, saj se je izkazalo, da bi tudi s tem presegel zastavljeno obširnost diplomskega dela.

6.3.1 Gostitelj

6.3.1.1 Pomnilnik

Aplikacija, ki sem jo ustvaril s pomočjo ogrodja Vue, gostiteljev pomnilnik bolj obremeni (tako maksimalna obremenitev kot minimalna in povprečna obremenitev) kot aplikacija, ki sem jo ustvaril s pomočjo knjižnice ReactJS. Povprečna obremenitev je pri Vue aplikaciji za 3,88 % večja kot pri ReactJS aplikaciji. Pri desetkratniku količine podatkov povzroči Vue aplikacija višjo maksimalno obremenjenost pomnilnika, med tem ko ReactJS aplikacija povzroči višjo minimalno obremenjenost in kar za 19,94 % višjo povprečno obremenjenost pomnilnika kot Vue aplikacija.

6.3.1.2 CPE

Obe aplikaciji imata enako vrednost minimalne obremenitve. Višjo maksimalno število CPE ciklov povzroči Vue aplikacija, kljub temu pa povzroči ReactJS aplikacija občutno višjo povprečno število CPE ciklov, in to kar 20,59 % višjo. Pri desetkratniku količine podatkov imata obe aplikaciji ponovno enako vrednost minimalne obremenitve, pride pa do zanimive

spremembe pri ostalih vrednostih. Pri desetkratniku količine podatkov namreč Vue aplikacija potrebuje višjo maksimalno število CPE ciklov, med tem ko ReactJS aplikacija potrebuje kar za 18,82 % nižje povprečno število CPE ciklov.

6.3.2 Odjemalec

6.3.2.1 Pomnilnik

Pri odjemalcu Vue aplikacija med nalaganjem uporabi 13,04 % več pomnilnika kot ReactJS aplikacija, v mirovanju pa ReactJS aplikacija uporabi 17,7 % več pomnilnika kot aplikacija, ustvarjena s pomočjo Vue. Pri desetkratniku količine podatkov aplikacija, ustvarjena z ogrođjem Vue med nalaganjem uporabi 17,36 % več pomnilnika kot aplikacija, ustvarjena s knjižnico ReactJS. V mirovanju pa ravno ReactJS aplikacija uporablja 14,93 % več pomnilnika.

6.3.2.2 CPE

Procesor za nalaganje aplikacije, ustvarjene s pomočjo knjižnice ReactJS potrebuje 7,4 % manj časa kot za nalaganje Vue aplikacije. Pri desetkratniku količine podatkov pa procesor potrebuje 2,18 % več časa za ReactJS aplikacijo.

6.3.2.3 Hitrost in odzivni čas

Aplikacija, ustvarjena s knjižnico ReactJS, potrebuje 4,72 % dlje do FCP, 14,75 % dlje do LCP in 3,96 % dlje do TTI v primerjavi z Vue aplikacijo. Pri desetkratniku količine podatkov je razlika v času do FCP med aplikacijama manj kot četrtna procenta, enako velja za čas do LCP med aplikacijama. TTI pa doseže ReactJS aplikacija 1,12 % hitreje.

6.4 Potencial za rast projekta

Aplikaciji sta si zelo podobni in kljub temu, da ima vsaka svoje prednosti in slabosti, ugotavljam, da je težko določiti, katera ima večji potencial za rast. Pomemben faktor, ki vpliva na to, je namreč tudi, v kakšno smer želimo, da projekt raste, kar pogojuje zahteve takšnega projekta. V primeru, da bi narava aplikacij ostala enaka, kot je zdaj in bi rasla samo v velikosti, potem je pomnilnik bolj pomemben kot CPE, saj lahko aplikacije, ki zgolj prikazujejo večje

količine slik in podatkov, nimajo pa kaj dosti drugih funkcionalnosti in interaktivnosti, porabljajo veliko spomina. V tem primeru bi bila aplikacija, izdelana z ogrodjem ReactJS, bolj primerna, saj je sodeč po moji analizi meritev obremenjenosti sistemov ter hitrosti in odzivnega časa iz poglavja 6.3 razvidno, da pri povečani količini podatkov v obeh aplikacijah ReactJS aplikacija pri odjemalcu porabi 17,36 % manj spomina kot Vue aplikacija, hkrati pa se naloži do interaktivnosti 1,12 % hitreje. V primeru, da bi projekt rasel v smeri večje funkcionalnosti in interaktivnosti, pa bi CPE bil pomembnejši, saj je njegova naloga izvajanje in procesiranje kode, navodil in odjemalčevega vnosa (angl. Input) ter upravljanje dinamičnih aspektov aplikacije, vključno z animacijami. V tem primeru je veliko težje določiti, katero ogrodje je bolj primerno, saj je v mojih meritvah pri večji količini podatkov Vue aplikacija porabila le 2,18 % manj pomnilnika. Osredotočil sem se na meritve odjemalca, saj je aplikacija, ki porablja malo sredstev na strani gostitelja, po drugi strani pa veliko sredstev na strani odjemalca nezaželeno, ker lahko poveča čas nalaganja aplikacije, kar lahko odvrne uporabnike (podpoglavje 6.2) ali v skrajnem primeru privede do tega, da operacijski sistem zazna, da aplikacija porablja preveč pomnilnika ali procesorske moči in jo sam prekine, kar lahko povzroči izgubo podatkov uporabnika.

6.5 *Big Data*

Med izvajanjem meritev sem ugotovil, da se aplikaciji obnašata drugače, kadar vsebujeta manjšo ali večjo količino podatkov. Iz poglavij 6.1, 6.3 in 6.4 je razvidno, da se je aplikacija, ustvarjena s knjižnico ReactJS, pri veliki količini podatkov izkazala za bolj učinkovito kot aplikacija, ki sem jo ustvaril z ogrodjem Vue. Glede na ugotovitve iz naštetih poglavij lahko predvidevam, da bi v primeru večje dinamičnosti, funkcionalnosti in dodane interaktivnosti mojim aplikacijam rezultat bil ravno nasprotni, žal pa bi raziskovanje tega, tako kot dodajanje še večje količine podatkov in ugotavljanje obremenjenosti sistemov in hitrosti nalaganja na še več nivojih količine podatkov bilo preobširno in je nekaj, kar bi lahko nekdo drug v svojem delu razširil naprej.

7 SKLEP

Za izdelavo diplomskega dela sem se v prvi fazi naučil delati z ogrodjem Vue in s knjižnico ReactJS. Ta proces mi je dal vpogled v to, katero orodje se je lažje naučiti uporabljati. Postopek izdelave aplikacij je opisan v poglavju 3. S primerjavo učnih virov, ki so za obe orodji na voljo (spletni tečaji, uradna dokumentacija, knjige in elektronske knjige), primerjavo njunih skupnosti na družbenem omrežju Facebook in na spletnih platformah Stack Overflow in Reddit in primerjavo njunih kompleksnosti (sintaksa, virtualni objektni model dokumenta in struktura komponent), sem ugotovil, da se lahko razvijalec hitreje nauči uporabljati ogrodje Vue kot knjižnico ReactJS, kar je potrdilo mojo prvo hipotezo (H1).

Naslednje vprašanje, ki ga je bilo potrebno razrešiti, je bilo, katero orodje porablja več sredstev oziroma bolj obremeni sistem. Odgovor na to vprašanje je kompleksnejši, ker sem meril dva različna parametra na strani gostitelja in nato na strani odjemalca, oboje med uporabo osnovne količine podatkov in nato med uporabo desetkratnika osnovne količine podatkov v aplikacijah. Na strani gostitelja aplikacija, ki je bila izdelana z Vue, povprečno bolj obremeni pomnilnik, pri povečani količini podatkov pa ga aplikacija, izdelana z ReactJS, obremeni bolj. Na strani odjemalca aplikacija, izdelana z Vue bolj obremeni pomnilnik pri obeh variantah količine podatkov. Drugi parameter, s katerim sem meril porabo sredstev, je centralna procesna enota. Na strani gostitelja pri osnovni količini podatkov ReactJS aplikacija bolj obremeni CPE, pri desetkratniku količine podatkov pa Vue aplikacija bolj obremeni CPE. Na strani odjemalca pri osnovni količini podatkov Vue aplikacija bolj obremeni CPE, pri desetkratniku količine podatkov pa ReactJS aplikacija. Na podlagi analize meritev v podpoglavju 6.3 pa sem vendar ugotovil, da skupno aplikacija, ki je bila izdelana z ogrodjem Vue, potrebuje več sredstev, kar je ovrglo mojo drugo hipotezo (H2).

Naslednja zadeva, ki sem jo želel raziskati, je bil tržni položaj obeh orodij. Iz ugotovitev v podpoglavju 5.4 je razvidno, da je ReactJS pogostejše uporabljen kot Vue, hkrati ima večjo skupnost, je bolj priljubljen med razvijalci spletnih aplikacij in nudi višjo stopnjo zaposljivosti tako v Sloveniji kot v tujini. Iz teh razlogov ima ReactJS močnejši tržni položaj, kar potrjuje mojo tretjo hipotezo (H3).

Zadnja dilema, ki sem jo želel razrešiti, je bila, katero orodje je bolj učinkovito glede dela z *Big Data*. V podpoglavjih 6.4 in 6.5 sem na podlagi ugotovitev meritev iz poglavja 6.3 pojasnil, da je za delo z *Big Data* učinkovitejša aplikacija, izdelana s knjižnico ReactJS, kar potrjuje še mojo zadnjo hipotezo (H4). Moja raziskava pa vendar le delno odgovori na zadnje vprašanje, saj predpostavljamo, da narava aplikacije ostane enaka, torej precej osnovna in statična. Vprašanje, ki ostaja in ima morebitni drugačen odgovor, je torej, katera aplikacije je učinkovitejša za aktivno delo z *Big Data* (procesiranje, računanje, primerjanje, animacije in izrisi glede na uporabnikov vnos, dinamičnimi deli, razne funkcionalnosti in tako naprej). Takšna raziskava bi bila za eno diplomsko delo preobširna, lahko pa bi na tej točki nekdo drug v svojem delu nadaljeval s to raziskavo.

V svojem diplomskem delu ugotavljam, da je prednost ogrodja Vue to, da se ga razvijalci lahko hitreje naučijo uporabljati kot knjižnico ReactJS. Na vseh ostalih področjih, ki sem jih raziskoval, pa se je knjižnica ReactJS izkazala za boljšo ali učinkovitejšo. Zaradi manjše porabe sredstev, boljšega tržnega položaja, višje zaposljivosti in učinkovitejšega dela z *Big Data* lahko z zanesljivostjo trdim, da je uporabna vrednost knjižnice ReactJS večja kot uporabna vrednost ogrodja Vue. Rezultati in ugotovitve iz te raziskave so mi pomagali sprejeti končno odločitev, katero od tehnologij bom izbral na svoji poklicni poti. Upam, da bodo te ugotovitve pomagale tudi kolegom razvijalcem in lastnikom podjetij pri njihovih odločitvah med tem, katero tehnologijo izbrati za svoje projekte.

8 VIRI, LITERATURA

Balusamy, B., Abirami, R. N., Kadry, S., & Gandom, A. H. (2021). *Big Data: Concepts, Technology, and Architecture*. Hoboken, New Jersey: Wiley.

Built With: React Usage Statistics. (26. 10. 2023). Pridobljeno s <https://trends.builtwith.com/javascript/React>

Built With: Vue Usage Statistics. (26. 10. 2023). Pridobljeno s <https://trends.builtwith.com/javascript/Vue>

Cobiss: React. (16. 10. 2023). Pridobljeno s <https://plus.cobiss.net/cobiss/si/sl/bib/search?q=react&db=cobib&mat=books&ucf=00400&ucf=68100>

Cobiss: Vue. (16. 10. 2023). Pridobljeno s <https://plus.cobiss.net/cobiss/si/sl/bib/search?q=vue&db=cobib&mat=books&ucf=00400&ucf=68100>

Coursera: react. (16. 10. 2023). Pridobljeno s <https://www.coursera.org/search?query=react>

Coursera: vue. (16. 10. 2023). Pridobljeno s <https://www.coursera.org/search?query=vue>

Facebook: react. (17. 10. 2023). Pridobljeno s https://www.facebook.com/search/groups?q=react&locale=sl_SI

Facebook: vue. (17. 10. 2023). Pridobljeno s https://www.facebook.com/search/groups?q=vue&locale=sl_SI

Gartner Glossary: Big Data. (16. 10. 2023). Pridobljeno s <https://www.gartner.com/en/information-technology/glossary/big-data>

GitHub. (2023). Pridobljeno s <https://github.com/>

Glassdoor: All »react« Results. (26. 10. 2023). Pridobljeno s <https://www.glassdoor.com/Search/results.htm?keyword=react>

Glassdoor: All »vue« Results. (26. 10. 2023). Pridobljeno s <https://www.glassdoor.com/Search/results.htm?keyword=vue>

Google Chrome devTools. (2023). Pridobljeno s <https://developer.chrome.com/docs/devtools/overview>

Indeed: *react.* (26. 10. 2023). Pridobljeno s <https://www.indeed.com/jobs?q=react&l=&vjk=c65709be1139dbc5>

Indeed: *vue.* (26. 10. 2023). Pridobljeno s <https://www.indeed.com/jobs?q=vue&l=&from=searchOnHP&vjk=3323afe11c625f5a>

Iskalci zaposlitve, dobrodošli. (26. 10. 2023). Pridobljeno s <https://www.ess.gov.si/iskalci-zaposlitve>

Legacy React. (2023). Pridobljeno s <https://legacy.reactjs.org/docs/faq-internals.html>

LinkedIn: *react, European Union, Jobs.* (26. 10. 2023). Pridobljeno s https://www.linkedin.com/jobs/search/?currentJobId=3811781519&geoId=91000000&keywords=%20react&location=European%20Union&origin=JOB_SEARCH_PAGE_LOCATION_AUTOCOMPLETE&refresh=true

LinkedIn: *react, Slovenia, Jobs.* (26. 10. 2023). Pridobljeno s https://www.linkedin.com/jobs/search/?currentJobId=3791293215&geoId=106137034&keywords=%20react&location=Slovenia&origin=JOB_SEARCH_PAGE_LOCATION_AUTOCOMPLETE&refresh=true

LinkedIn: *react, United States, Jobs.* (26. 10. 2023). Pridobljeno s https://www.linkedin.com/jobs/search/?currentJobId=3749545378&geoId=103644278&keywords=%20react&location=United%20States&origin=JOB_SEARCH_PAGE_SEARCH_BUTTON&refresh=true

LinkedIn: *vue js, European Union, Jobs.* (26. 10. 2023). Pridobljeno s https://www.linkedin.com/jobs/search/?currentJobId=3592990541&geoId=91000000&keywords=vue%20js&location=European%20Union&origin=JOB_SEARCH_PAGE_LOCATION_AUTOCOMPLETE&refresh=true

LinkedIn: *vue js, Slovenia, Jobs.* (26. 10. 2023). Pridobljeno s https://www.linkedin.com/jobs/search/?currentJobId=3791293215&geoId=106137034&keywords=vue%20js&location=Slovenia&origin=JOB_SEARCH_PAGE_LOCATION_AUTOCOMPLETE&refresh=true

LinkedIn: *vue js, United States, Jobs.* (26. 10. 2023). Pridobljeno s <https://www.linkedin.com/jobs/search/?currentJobId=3730005728&geoId=103644278>

&keywords=vue%20js&location=United%20States&origin=JOB_SEARCH_PAGE_SEARCH_BUTTON&refresh=true

Marz, N., & Warren, J. (2015). *Big Data: Principles and Best Practices of Scalable Real Time Data Systems*. Shelter Island, New York: Manning Publications.

React: Quick Start. (12. 9. 2023). Pridobljeno s <https://react.dev/learn>

React: Quick Start. (12. 9. 2023). Pridobljeno s <https://react.dev/learn>

React: Thinking in React. (12. 9. 2023). Pridobljeno s <https://react.dev/learn/thinking-in-react>

Reddit: react. (17. 10. 2023). Pridobljeno iz <https://www.reddit.com/search/?q=react&type=sr>

Reddit: vue. (17. 10. 2023). Pridobljeno iz <https://www.reddit.com/search/?q=vue&type=sr>

Stack Overflow: 2023 Developer Survey. (26. 10. 2023). Pridobljeno s <https://survey.stackoverflow.co/2023/#most-popular-technologies-webframe-prof>

Stack Overflow: react. (17. 10. 2023). Pridobljeno s <https://stackoverflow.com/search?q=react>

Stack Overflow: vue. (17. 10. 2023). Pridobljeno s <https://stackoverflow.com/search?q=vue>

TatvaSoft: Top 10 Companies that use React. (26. 10. 2023). Pridobljeno s <https://www.tatvasoft.com/outsourcing/2022/02/top-10-successful-companies-using-react-js.html>

TatvaSoft: Top 10 Global Companies using Vue js. (26. 10. 2023). Pridobljeno s <https://www.tatvasoft.com/outsourcing/2021/10/top-companies-using-vue-js.html>

TechTarget. (28. 10. 2023). Pridobljeno s <https://www.techtarget.com/searchdatamanagement/definition/big-data>

The Rick and Morty API. (16. 10. 2023). Pridobljeno s <https://rickandmortyapi.com/>

Udemy: Results for react. (16. 10. 2023). Pridobljeno s <https://www.udemy.com/courses/search/?src=ukw&q=react>

Udemy: Results for vue. (16. 10. 2023). Pridobljeno s <https://www.udemy.com/courses/search/?src=ukw&q=vue>

Understanding Big Data Processing: 2024's Ultimate Guide. (2023). Pridobljeno s <https://hevodata.com/learn/big-data-processing>

Visual Studio Code. (12. 9. 2023). Pridobljeno s <https://code.visualstudio.com>

Vue.js. (2023). Pridobljeno s <https://vuejs.org/guide/extras/rendering-mechanism>

Vue.js: Components Basics. (12. 9. 2023). Pridobljeno s
<https://vuejs.org/guide/essentials/component-basics.html>

Vue.js: Introduction. (12. 9. 2023). Pridobljeno s <https://vuejs.org/guide/introduction.html>

Vue.js: Introduction. (12. 9. 2023). Pridobljeno s <https://vuejs.org/guide/introduction.html>

Windows Performance Monitor. (2023). Pridobljeno s
<https://techcommunity.microsoft.com/t5/ask-the-performance-team/windows-performance-monitor-overview/ba-p/375481>