Izrada web aplikacije pomoću React JavaScript razvojnog okvira

Stamenić, Filip

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:345745

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-27**

Repository / Repozitorij:

University North Digital Repository

Sveučilište
Sjever

Završni rad br. 598/MM/2018

Izrada web aplikacije pomoću React JavaScript razvojnog okvira

Filip Stamenić, 0818/336

Varaždin, rujan 2018. godine

Sveučilište
Sjever

Odjel za Multimediju, oblikovanje i primjenu

Završni rad br. 598/MM/2018

Izrada web aplikacije pomoću React JavaScript razvojnog okvira

Student

Filip Stamenić, 0818/336

Mentor

mr.sc.Vladimir Stanisavljević

Varaždin, rujan 2018. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za multimediju, oblikovanje i primjenu

PRISTUPNIK: Filip Stamenić | MATIČNI BROJ: 0818/336

DATUM: 19.09.2018. | KOLEGIJ: Programski alati 2

NASLOV RADA: Izrada web aplikacije pomoću React JavaScript razvojnog okvira

NASLOV RADA NA ENGL. JEZIKU: Development of a web aplication with Rect JavaScript framework

MENTOR: mr.sc. Vladimir Stanisavljević | ZVANJE: viši predavač

ČLANOVI POVJERENSTVA:
1. doc.dr.sc. Dejan Valdec - predsjednik
2. dr.sc. Andrija Bernik, pred. - član
3. mr.sc. Vladimir Stanisavljević, v.pred. - mentor
4. dr.sc. Robert Logožar, v. pred. - zamjenski član
5.

Zadatak završnog rada

BROJ: 598/MM/2018

OPIS:

Programski jezik JavaScript osnova je rada svih dinamičkih web stranica. Dugi niz godina standardiziran je kroz ECMA script specifikaciju. Razvojni okviri olakšavaju rad s JavaScriptom dok druga programska rješenja modificiraju sam jezik kako bi se kod čim lakše pisao. ReactJS već neko vrijeme spada u takve razvojne okvire odnosno programska rješenja.

U ovom radu potrebno je:
* opisati osnove ReactJS-a razvojnog okvira
* opisati postupak instalacije i osnovnog korištenja iz odabranog razvojnog okruženja te postavljanje na javni posàužitelj
* detaljno obraditi mogućnosti i sintaksu ReactaJS-a te na primjerima pokazati njihovo korištenje,
* osmisliti i oblikovati demo web stranicu na kojoj će biti prikazane glavne mogućnosti, načini korištenja ReactJS-a, a u radu objasnit primjere koda,
* ukratko usporediti ReactJS sa sličnim programskim rješenjima

Detaljno opisati sve korištene tehnologije i korake u radu potrebne da bi se ostvarilo traženo te opisati stečena iskustva i postignute rezultate.

ZADATAK URUČEN: 21.09.2018. | POTPIS MENTORA:

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER

Predgovor

Ovim putem bih se želio zahvaliti svojoj obitelji i prijateljima na podršci tijekom tri godine studiranja te svom mentoru mr.sc.Vladimiru Stanisavljeviću za pomoć i savjetima kod izrade ovog rada.

Sažetak

React JavaScript biblioteka je vrlo popularna u današnje vrijeme i važan je dio za front-end development. Cilj ovog rada je pokazati kako se React razvija iz godine u godinu te kako je to jedna od najkvalitetnijih JavaScript biblioteka za izradu aplikacija kada se uspoređuje sa sličnim JavaScript bibliotekama ili razvojim okvirima. U ovom radu ću objasniti što je React, što nam omogućuje, od kakvih se komponenti sastoji te što je potrebno znati kako bi izrada aplikacije bila moguća. Za praktični dio sam odlučio realizirati jednu od brojnih ideja koje se mogu realizirati koristeći se React JavaScript bibliotekom, u ovom slučaju kalkulator pomoću kojeg se mogu izračunati jednostavne računske operacije.

Ključne riječi: programiranje, kodiranje, IDE, komponente, virtualni DOM, ReactJS

Summary

React JavaScript library is very popular nowadays and it is significant for front-end development. The purpose of this thesis is to show how React develops year after year and to show that this is high quality JavaScript library for application creation in comparison with similar JavaScript libraries or frameworks. In this thesis I am going to explain what React is, what can we do with it, of what components it consists of and what is important to understand to make application creation possible. For practical part I have decided to realize one of many ideas that can be realized with React JavaScript library, in this case this is calculator that can calculate basic operations of arithmetic.

Keywords: programming, coding, IDE, components, virtual DOM, ReactJS

Popis korištenih kratica

CSS Cascading Style Sheets
Opisni jezik namijenjen opisivanju izgleda web stranice.

DOM Document Object Mode
Apstrakcija HTML strukture stranice.

ES6 ECMAScript 6
Specifikacija skriptnog jezika.

HTML HyperText Markup Language
Sintaksa za obilježavanje hipertekştualnih dokumenata.

IDE Integrated development environment
Softver za programere koji se bave softver developmentom.

JS JavaScript
Programski jezik namijenjen za izradu dinamičnih i interaktivnih elemenata.

JSX JavaScript XML
Ekstenzija sintakse JavaScript-e te se koristi kod React-a za opisivanje korisničkog sučelja.

W3C World Wide Web Consortium
Organizacija za razvoj i standardizaciju web tehnologija.

XML Extensible Markup Language
Definira skupinu pravila koja se koriste za enkodiranje dokumenata u format koji je čitljiv od strane računala i korisnika.

Sadržaj

1. Uvod

U današnje vrijeme bismo mogli reći da tehnologija raste eksponencijalno. Potrebno je sve više znati o trenutnim trendovima i alatima koji se koriste za razvoj raznih web stranica ili web aplikacija koje su funkcionalne i kvalitetne kako bi se ostvarilo zadovoljavajuće korisničko iskustvo. Smatram kako je važno konstantno pratiti koji se razvojni okvir (eng. framework) ili JavaScript biblioteka najviše koristi za određenu svrhu kako bismo mogli napraviti aplikaciju koja će se svidjeti korisnicima.

Sva je veća potražnja za onima koji imaju volju i znanje posvetiti se programiranju ili dizajniranju neke web stranice ili web aplikacije pa nije neobično kako se mnogi posvete tom području jer su voljni učiti jedni od drugih svaki dan ili čekati nove nadogradnje nekog alata kako bi mogli raditi na nekom novom zadatku koristeći razne nove funkcije. Iz tih razloga smatram da je to jednostavno posao budućnosti.

Tijekom studiranja sam shvatio koliko je važno naučiti jezike HTML, CSS i JavaScript te od kakve su važnosti. Posebno mi je zanimljiv programski jezik JavaScript i njegove razne mogućnosti u izradi raznih dinamičnih ili interaktivnih elemenata. U današnje vrijeme se sve više i izrađuju takve stranice za razliku od statičnih stranica koje su se još koristile za vrijeme web 1.0.

Jedna od JavaScript biblioteka za koju smatram da je kvalitetna i zanimljiva za naučiti je React JavaScript biblioteka. To je jedna od najpopularnijih JavaScript biblioteka te nastaje kao biblioteka za izradu korisničkih sučelja. Smatram da je učenje ove biblioteke dobra za programere i za one koji žele učiti raditi s raznim razvojnim okvir-ima i bibliotekama.

React je odlična JavaScript biblioteka za izradu raznih aplikacija na jednoj stranici kao što su kalendari u kojima se mogu upisivati razne osobne bilješke ili kalkulatori za izračunavanje jednostavnih računskih operacija.

React je koristan za učenje jer nam omogućuje izradu raznih aplikacija u front-end području, ali osim što je React kvalitetna biblioteka, učenje te biblioteke nam pomaže kako bismo bolje shvatili JavaScript programski jezik, ali i opisne jezike HTML i CSS.

U ovom radu sam se odlučio posvetiti komponentama koje se nalaze u React JavaScript biblioteki te alatima koji nam omogućavaju izradu React aplikacija kao što su alati JetBrains Webstorm i CodePen za pisanje programskih kodova te pomoću njih napraviti funkcionalni kalkulator za izračunavanje jednostavnih računskih operacija.

Korištenjem ovih alata je izrada svih aplikacija pojednostavljena te se bez većih problema može napisati programski kod našeg željenog zadatka.

Smatram da je ovakav pristup izrade aplikacije na jednostavniji način dobro došao s obzirom na razvoj tehnologije u današnje vrijeme gdje je potrebno veliko znanje za pisanje programskog koda. React ovdje dosta olakšava posao. Iako se React-ova struktura sastoji od mnogo manjih komponenti, učenje React JavaScript biblioteke nije zahtjevno ako se već znamo koristiti HTML-om, CSS-om ili JavaScript-om te smatram će React svakoj osobi pomoći u svijetu programiranja.

2. Kodiranje i programiranje

Programiranje je pisanje uputa računalu što i kako učiniti, a izvodi se u nekom od programskih jezika. Programiranje je umjetnost i umijeće u stvaranju programa za računala. Stvaranje programa sadrži u sebi pojedine elemente dizajna, umjetnosti, znanosti, matematike kao i inžinjeringa. Osoba koja stvara program zove se programer.

Programi ili upute za računalo pišu se u programskom jeziku upotrebom određene sintakse i pravila koja vrijede za svaki programski jezik (ili tip), koji se potom prevodi u strojni jezik koje je osobito za određeno računalo te je ovisno o njegovoj arhitekturi. Prevođenje s višeg programskog jezika na strojni provodi se putem programa prevodioca (kompajler) ili se naredbe u višem jeziku izravno prevode preko takozvanog p_koda u strojni jezik.

Primjeri programskih jezika:

Assembler, BASIC, Pascal C/C++/C#, Java, Logo, Fortran...

Primjer prevoditeljskog (interpretiranog) programskog jezika bio bi Basic (izvršava se unutar posebnog programa tzv. prevoditelja (eng. interpreter) koji stoji između stroja računala i programa). Postoje niži i viši programski jezici kao što je assembler ili strojni kod koji je primjer nižeg programskog jezika (obično se radi o izravnom pozivanju "naredbi centralne jedinice" (CPU) ili instrukcija kao npr. INC (uvećanje), MOV (kopiranje spremnika (registara)) itd.).

Primjer višeg programskog jezika je C koji se mora prvo prevesti, a potom prilagoditi da radi na određenom operacijskom sustavu (eng. compile & link). Postoje drugi oblici klasifikacija i tipovi programskih jezika od čega je vjerojatno bitno spomenuti objektno-orijentirane jezike (C++, Java, …) koji su danas najraširenij u primjeni koristeći razne mehanizme kakvi nisu bili u uporabi u npr. C-u i Pascal-u.

Najbitniji pojmovi u tom slučaju su objekt, enkapsulacija, nasljeđivanje itd. Pri programiranju se također koriste razne metodologije razvoja software-a (programske podrške) pri kojima je dobro početi od vodopadnog modela, spiralnog, prototypinga i sličnih kao jednostavnih modela razvoja.[1]

Pri programiranju se također koriste razne metodologije razvoja software-a (programske podrške) pri kojima je dobro početi od vodopadnog modela, spiralnog, prototypinga i sličnih kao jednostavnih modela razvoja. Standardi za razvoj i osiguranje kvalitete (kakvoće) software-a također postoje od kojih je bitno spomenuti ISO standarde. Veće tvrtke koje se bave razvoje software-a obično koriste svoje metodologije razvijane godinama ili prilagođavaju postojeće

kako bi pratile njihovu filozofiju razvoja, a sve u svrhu osiguranja kvalitete i uniformnosti razvoja.[2]

Svaki programski jezik ima svoja određena pravila i način na koji funkcionira, no bez obzira na to postoje mnoge sličnosti zbog kojih je moguće jednostavno naučiti više programskih jezika ako imamo potrebno znanje u bar jednom programskom jeziku.

Kodiranje je jedan od procesa u izradi web stranica. Kao i dizajn ima više metoda rada, ovisno o sadržaju stranice i sklonosti webmastera. Dizajn stranice napravi se u jednom od grafičkih programa (najbolji za rastersku grafiku je Adobe Photoshop, a za vektorsku grafiku Adobe Illustrator). Kada je dizajn gotov pristupa se prijelomu u HTML oblik. Izrađuju se HTML i CSS datoteke koje se sa ostalim komponentama web stranice smještaju na server.

HTML (Hyper Text Markup Language) je jezik koji opisuje sadržaj web stranice. Hipertekst je tekst koji sadrže veze ili linkove ka drugim dokumentima. HTML naredbe pišu se u vidu tzv. TAG-ova. Tag je ustvari naredba koja govori pregledniku što i kako da uradi sa sadržajem na web stranici. Kada posjetimo neku web stranicu događa se slijedeće: preglednik na našem računalu ili mobitelu šalje zahtjev serveru. Server na kojemu je smještena web stranica generira HTML datoteku (koju je prije webmaster izradio) i vraća našem pregledniku.

Naš preglednik (Google Chrome, Internet Explorer, Mozilla Firefox, Safari i dr.) razumije HTML kod i na osnovu njega prikazuje web stranicu na zaslonu računala ili mobitela. Ako je web stranica u CMS sustavu, HTML datoteka sadrži i programske skripte (PHP ili ASP) koje se povezuju sa bazom podataka. HTML datoteka sadrži i JavaScript kod. On se izvršava kada naš preglednik preuzme od servera HTML datoteku. HTML je abeceda web dizajna i svaki bi ga webmaster trebao znati. Sada je standard verzija HTML 5 koja u odnosu na verziju HTML 4 ima mnogo više mogućnosti. HTML datoteku sadrži svaka web stranica. Izvorni HTML kod svake web stranice se može vidjeti ako kliknemo desnim klikom miša i odaberemo "Pogledaj izvor stranice".

CSS (Cascading Style Sheets) definira izgled HTML elemenata. Dok HTML opisuje kako prikazati sadržaj, CSS stilovi definiraju kako će taj sadržaj na web stranici izgledati. CSS kod u najjednostavnijem obliku određuje koja će biti boja teksta, boja pozadine, koji će se font prikazati itd.

Što je web stranica zahtjevnija i veća potrebno je više vremena za njeno kodiranje. Web stranice se danas prikazuju na zaslonima raznih uređaja (monitori, laptopi, tableti, mobiteli). Svaki od uređaja koristi različite rezolucije ekrana. Kod responzivnog dizajna postoji više CSS datoteka.

CSS kodiranje se vrši na način da se obuhvate svi mogući uređaji sa svim mogućim rezolucijama. Naš preglednik automatski izabere i prilagodi CSS datoteku koja je prikladna za

naš uređaj. Tako će biti drugačije odabran i prikazan CSS kod za pristup internetu sa mobitela od onog ako pristupate tabletom u horizontalnom položaju. Ili ako pristupate internetu sa stolnog računala i monitorom od 27 inča od pristupa sa tabletom u okomitom položaju. CSS je vrlo široko područje i za njegovu uspješnu primjenu potrebna je napredna razina znanja. Sadašnja verzija CSS-a je CSS3.

Tehnologije HTML 5, CSS3 i JavaScript su ključne za uspješno kodiranje modernih web stranica. Pored njih dobro je poznavati i ostale web tehnologije.[3]

```
<!DOCTYPE html>
<html>
<head>

<meta charset="UTF-8"/>
<title>Filip Stamenić</title>
<script src=
"https://unpkg.com/react@16/umd/react.development.js">
</script>
<script src=
"https://unpkg.com/react-dom@16/umd/react-dom.development.js">
</script>
<script src=
"https://unpkg.com/babel-standalone@6.15.0/babel.min.js"> </script>
</head>

<body>
<div id="root"></div>
<script type="text/babel">

    ReactDOM.render{
    <h1>Hello, world!<h1>,
    document.getElementById('root')

};
</script>
</body>
</html>
```

Programski kod 2.1. – primjer zadatka u React-u

U ovom jednostavnom primjeru je napisan programski kod koji će u web pregledniku ispisati jednostavan tekst „Hello, world!“ bez dodatnog uljepšavanja teksta. Iako je ovo jednostavan kod, potrebno je voditi brigu o urednosti pisanja kako bi se mogle izbjeći razne pogreške kod pisanja ili kako se neki dio ne bi slučajno izostavio. To je razlog zašto se kod pisanja složenijih programskih kodova koristi tabulator, ostavljaju komentari ili prazni redovi zbog urednosti i omogućuje se drugima i samom programeru lakše čitanje cijelog programskog koda.

Također je poželjno zbog bolje produktivnosti i kvalitete programskog koda koristiti code editor koji podržava bojanje i struktuiranje koda, hijerarhijsko grupiranje i autocomplete funkciju.[4]

Uz HTML i CSS, treći jezik kojim je važno znati rukovati je Javascript. JavaScript je skriptni programski jezik, koji se izvršava u web pregledniku na strani korisnika. Izvorno ga je razvila tvrtka Netscape.

JavaScript s AJAX (Asynchronous JavaScript and XML) tehnikom omogućuje web stranicama komunikaciju sa serverskim programom, što čini web aplikaciju više interaktivnom i lakšom za korištenje.[5]

JavaScript nije pojednostavljena inačica programskog jezika Java. Povezuje ih jedino slična sintaksa i to što se koriste za izvršavanje određenih radnji unutar preglednika. Izvorno se JavaScript trebao zvati LiveScript, ali da bi se potakla uporaba novog skriptnog jezika, nazvan je slično jeziku Java.

Danas je podrška za JavaScript izvrsna u svim preglednicima kao što su popularni web preglednici Google Chrome ili Mozilla Firefox, tako da se autori preglednika više ne natječu u podršci nego u brzini izvođenja određenih algoritama.

JavaScript se može pisati u bilo kojem uređivaču teksta (editor) koji podržava standard ASCII. Blok za pisanje (Notepad) prisutan je u svim inačicama operacijskog sustava Windows pa se nameće kao pogodan i za programiranje u JavaScriptu.

Da bi se programiranje ubrzalo i učinilo učinkovitijim, poželjno je rabiti uređivač teksta koji imamo mogućnost otvaranja više dokumenata u jednom prozoru, razlikuje dijelove programskog koda i prikazuje ih u različitim bojama, podrška za UTF-8 što je korisno za pisanje slova iz hrvatske abecede, automatsko poravnavanje (tidy) i provjera stila kodiranja u skladu s preporučenim postupcima (linter).[6]

2.1. JetBrains WebStorm 2018.1.5

JetBrains je tvrtka koja se bavi razvojem softvera čijim se alatima koriste uglavnom programeri, ali i dizajneri. Jedan od alata kojeg sam ja odlučio koristiti u svrhu izrade zadatka je JetBrains WebStorm 2018.1.5.

JetBrains Webstorm je profesionalni IDE (Integrated Development Environment) za web programere. WebStorm nam pomaže da jednostavno kreiramo web stranice uređivanjem HTML-a, CSS-a i JavaScript, jednostavno kretanje kroz datoteke i korištenje automatskog dovršetka za sve što je u našem kodu.

WebStorm nas također obavještava o problemima kodova u trenutku pisanja i jednostavno rukuje kompliciranim kombinacijama jezika, npr. HTML markup ili SQL unutar JavaScript. WebStorm prati sve izmjene na izvornim datotekama, štiteći ih od slučajnih gubitaka ili izmjena, čak i ako ih izrađuju drugi programi. U bilo kojem trenutku možemo pregledati povijest bilo određene datoteke ili direktorija i vratiti se na bilo koju od prethodnih verzija pomoću WebStorma.

WebStorm je baziran na HTML/XHTML i XML kodu te je sposoban za rad i dovršavanje raznih funkcija kao što su: stilovi, nazivi oznaka, zatvaranje oznaka, atributi. U CSS-u je također prisutan takav pristup kod HTML ID-a, ključnih riječi, vrijednosti i svojstva.

WebStorm također provjerava greške u kodu te ih može automatski ispraviti u slučaju pojave. Pogreške kod CSS selektora, greške kod CSS svojstva i mnoge druge greške se mogu brzo popraviti kako bi se moglo nastaviti pisati kod.[7]

Program sam odabrao jer mnogo ljudi na internetu smatra da je kvalitetan pa sam ga odlučio isprobati. Ovaj program mi omogućuje rad s React JavaScript bibliotekom te je program jednostavan za korištenje i pomaže mi kod pisanja na način da ga automatski dovršava. Uz to koristi više boja što mi pomaže kod čitanja koda, a i cijeli kod izgleda uredno, lakše za čitanje te uz komentare mogu jednostavno pronaći dio koda koji tražim.

Prije programa WebStorm sam radio s programima Notepad++ i NetBeans pa mi nije bilo potrebno puno vremena kako bih mogao normalno raditi u WebStorm-u. Programi su slični i imaju istu zadaću pa je stoga i način na koji se koriste sličan.

2.1.1. Prednosti i nedostaci programa WebStorm 2018.1.5

Poznato je da svaki program ima određene prednosti i nedostatke. Nakon što sam koristio mnogo programa za različite svrhe sam shvatio da se svaki program barem malo razlikuje od drugog. Na temelju tih razlika bih odlučio koji program će mi na najlakši i najkvalitetniji mogući način pomoći da ostvarim svoj cilj. Broj programa koji se mogu koristiti za određenu svrhu je ogroman te je svaki program jedinstven na neki način. Svaka tvrtka želi pružiti program koji će izvršavati svoju zadaću na jednostavan, ali i kvalitetan način te je krajnji cilj stvoriti što bolje korisničko iskustvo Na taj način se želi privući broj veliki broj korisnika čime će rasti popularnost tog programa, zajedno sa zaradom.

Prednosti:

Korištenjem ovog programa sam shvatio da je za mene po mnogim stvarima bolji i kvalitetniji za korištenje u usporedbi sa sličnim programima.

Mogućnost automatskog nadopunjavanja mi je mnogo olakšalo kod pisanja programskih kodova i zbog te pomoći sam imao manje greške nego što bih to inače imao. Pisanje programskih kodova je lakše, pregledan je i jednostavan za čitanje. Debugger se nalazi unutar IDE, daje podršku za brojne razvojne okvire kao što su Angular, Node, Require, ali i React. Postoji mogućnost za izradu raznih vlastitih prečica kako bi se moglo lakše i brže raditi. Koristeći se tim prečicama sam mogao brže raditi na pisanju svog programskog koda.

Korisničko sučelje se može prilagoditi onako kako korisnik želi što je i meni osobno pomoglo kod pisanja programskog koda. Tražilica u programu je također vrlo korisna. Korištenjem „Shift+Shift“ se mogu pronaći sve datoteke, funkcije, postavke i ostali sadržaj koji je vezan uz program i naš rad.

Kada sam želio urediti stil tako da, primjerice, promijenim boju nekog elemenata, u tom slučaju nisam morao ništa pisati nego sam u novom prozoru mogao vidjeti sve boje vizualno i odabrati koju želim bez potrebe upisivanja HTML koda za određenu boju.

Nedostaci:

Postoji još mnogo prednosti kod ovog programa, iako smatram da su to prilično važne prednosti. Korištenjem ovog programa nisam naišao na mnogo nedostataka. Za pokretanje ovog programa je potrebno malo više vremena, ali kada se cijeli sistem pokrene to više nije problem. [8] Osim vremena potrebnog za pokretanja programa, za mene je najveći problem predstavljala činjenica da ovaj program nije besplatan.

No tvrtka JetBrains nudi svim studentima ovaj program besplatno na godinu dana ako se registriramo koristeći našu e-mail adresu koju dobijemo početkom studiranja pa ta činjenica što program nije besplatan nije predstavljao problem za mene. Osim tih sitnih problema nisam naišao na poteškoće prilikom korištenja ovog programa što je po mom mišljenju najvažnije i razlog više zašto sam se odlučio koristiti ovaj program.

2.1.2. CodePen code editor

CodePen je web mjesto za pisanje HTML, CSS i JavaScript kodova. To je open-source online uređivač teksta (kodova) koji sam također koristio uz JetBrains Webstorm zbog jednostavnije izrade zadatka, ali i za testiranje raznih drugih zadataka.

Kada napišem programski kod koji radi i s kojim sam zadovoljan, taj kod se nakon toga može preuzeti HTML, CSS, JS (Babel) datoteke) i jednostavno pokrenuti u WebStorm-u za daljnju obradu bez javljanja neke greške.

CodePen je sličan WebStorm-u po načinu rada i po izgledu te je zbog toga korisnicima lakše pisati kodove za svoje aplikacije, a osim toga ima bogat izbornik sa raznim korisnim postavkama koji pomažu korisnicima kod kodiranja.

3. Što je ReactJS

ReactJS ili jednostavno React je JavaScript biblioteka koja spada u view biblioteku te nam omogućuje izradu korisničkog sučelja. React je razvijen od strane Facebook-a i koristi se za izradu kvalitetnih aplikacija.

Ova JavaScript biblioteka je nastala u 2013. godine u ožujku te se konstatno ažurirala. Vrlo je popularna u današnje vrijeme te se često koristi kao kombinacija s drugim bibliotekama i razvojnim okvirima za izradu raznih aplikacija.

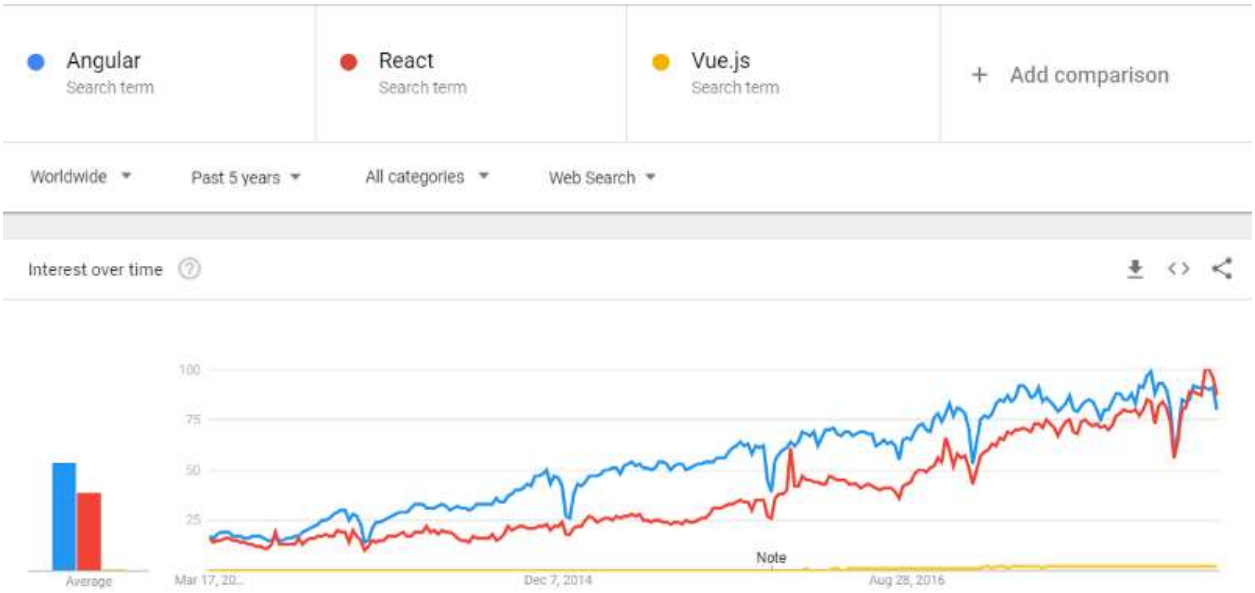
Mnogo ljudi smatra da React nije razvojni okvir, već JavaScript biblioteka. Na njihovoj službenoj web stranici piše da je React JavaScript biblioteka za izradu korisničkih sučelja[9].

No, unatoč tome postoji još uvijek skupina ljudi koji smatraju da je React razvojni okvir, iako i ja smatram da je React zapravo biblioteka. Postoje neke razlike između razvojni okvir i JavaScript biblioteka, neke koje su ključne i koje mogu odgovoriti na pitanje je li React razvojni okvir ili JavaScript biblioteka.

Razvojni okvir definira okvir u kojem se razvija aplikacija, dok JavaScript biblioteka definira klase/metode koje rade specifične operacije. Ono što ih ponajviše razlikuje zove se Inversion of Control (IoC). Kada pozivamo metodu iz biblioteke, mi imamo kontrolu - za razliku od razvojnog okvira, gdje on poziva naš programski kod.[10] IoC je stoga ključna razlika između razvojnog okvira i JavaScript biblioteka.

Iako je React biblioteka, često se uspoređuje s raznim razvojnim okvirima po pitanju efikasnosti, sigurnosti, cijeni i raznim drugim parametrima. Konkretno se AngularJS i ReactJS često spominju i uspoređuju s naglaskom na njihove prednosti i mane.

Trenutna verzija React-a koja se koristi je verzija 16.4.2. Ova JavaScript biblioteka se konstantno nadograđuje i popravljaju se razne greške što omogućuje bolji rad na željenim zadacima.



Slika 3.1. – usporedba popularnosti biblioteke React sa svojim suparnicima

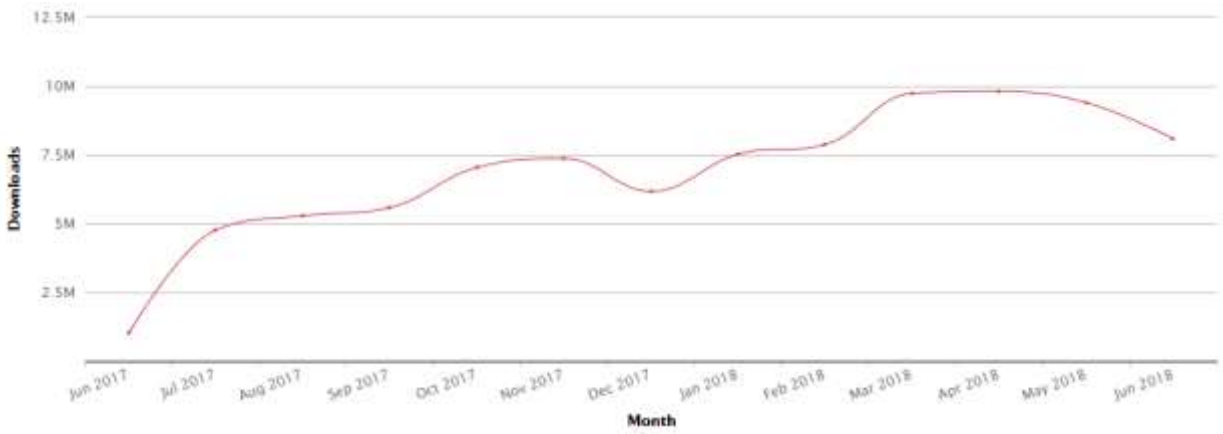
React nije bio popularan kada je nastao kao što je bio u to vrijeme razvojni okvir AngularJS. Kroz par godina se razlika u interesu između biblioteke i razvojnog okvira smanjila i React je postao sve popularniji i postao je ozbiljna konkurencija raznim drugim JavaScript bibliotekama i razvojnim okvirima. U zadnjih par godina, AngularJS i React su postali vrlo popularni te se najviše koriste za front-end development.[11]

2018. godine i dalje raste popularnost React-a do te mjere da je daleko ispred razvojnog okvira AngularJS.

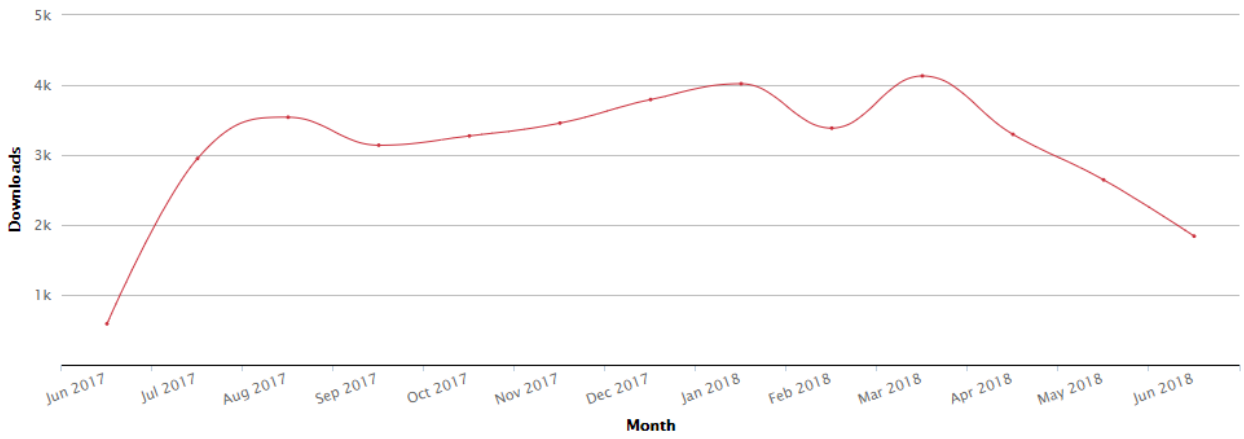
Korištenjem grafova na web stranici https://npm-stat.com/ usporedio sam količinu skidanja u posljednjih godinu dana kako bi se mogla usporediti popularnost JavaScript biblioteke React i razvojnog okvira AngularJS.

Prema podacima dobivenim iz grafova vidi se da je React daleko ispred AngularJS, a popularnost mu ne prestaje rasti. Radi usporedbe, u posljednjih 12 mjeseci je React biblioteka pregledana i preuzeta 2 238 više puta nego razvojni okvir AngularJS.

Podaci na grafovima ispod su dobiveni 2018. godine u lipnju te se može vidjeti broj skidanja u svakom mjesecu.



Slika 3.2. – skidanje React biblioteke izražen u milijunima



Slika 3.3. – Angular izražen u tisućicama

React je bila najpopularnija JavaScript biblioteka 2017. godine. Kako bismo se mogli koristiti tom bibliotekom, potrebno je znati HTML opisni jezik što je i jedan od razloga zašto mu popularnost raste.

Potrebno je naglasiti da sadrži mogućnost responzivnosti što je važno s obzirom koji su trendovi prisutni u današnje vrijeme, a tu se prvenstveno misli na razvoj tehnologije u području mobilnih uređaja.

Danas su brojna web mjesta i aplikacije responzivna što znači da se mogu prilagođavati širini korisnikova ekrana što nam u konačnici olakšava rad i cijela web stranica ili aplikacija će biti preglednija.

U praktičnom djelu u ovom radu se može vidjeti kako se širina rada prilagođava, odnosno rad će promijeniti izgled kada odlučimo promijeniti širinu prozora našeg web preglednika kojeg koristimo.

3.1. Područja primjene React-a

React JavaScript biblioteka je jedna od najviše korištenih biblioteka u današnje vrijeme za izradu korisničkih sučelja. Mnogo web mjesta koriste React za izradu svojih web stranica kao što su BBC, Facebook, Imgur, Instagram, Netflix, PayPal, Reddit, Uber, WhatsApp i brojna druga web mjesta.

Osim što Facebook koristi ReactJS, Facebook je razvio tu JavaScript biblioteku i sadrži preko 20000 React komponenti na svojoj stranici.[12] Svako web mjesto koje koristi React sadrži mnogo komponenti te se često uz React koriste razni drugi razvojni okviri ili JavaScript biblioteke kao što je React Native kod Facebook-a za reklame.

Instagram koristi React zbog responzivnosti, Reddit zbog bržeg učitavanja linkova ili nekog drugog sadržaja i boljeg korisničkog iskustva. Svako web mjesto koje je uvelo React funkcionira bolje prema mišljenjima većine korisnika.

React se može koristiti i kod izrade jednostavnih aplikacija kao što su kalkulatori, kalendari, planeri, a uz upotrebu i drugih biblioteka se može i izraditi složenija aplikacija poput music playera ili video playera.

Brzo renderiranje s virtualnim DOM-om je samo jedan od razloga zašto mnogo korisnika preferira koristiti React JavaScript biblioteku za svoje potrebe kako bi njihova aplikacija radila brzo i efikasno.

React doprinosi boljoj fleksibilnosti i jednostavnosti web stranicama za razliku od brojnih drugih razvojnih okvira ili biblioteka.

3.2. ReactJS komponente

Komponente su poput JavaScript funkcija. One prihvaćaju proizvoljne ulaze pod nazivom „props“ i vraćaju React elemente koji određuju što će se pojaviti na ekranu. React nam omogućuje kreiranje komponenti na dva načina: kao ES6 klasa ili pomoću React.createClass.

Komponente koje su definirane kao ES6 pružaju više mogućnosti. Potrebno je definirati render() u React.Component podklasi, a ako bi se definirala React EcmaScript klasa koristi se sintaksa u slijedećem obliku:

```
class Component extends React.Component {
    render ()
    {
        return (
            <p>Dobar dan!</p>
        );
    }
}
```

Programski kod 3.1. – ispis poruke u React-u

React nas ipak ne prisiljava da koristimo samo ES6 način za kreiranje komponenti. Moguće je i koristiti create-react-class metodu. Tijekom korištenja React.createClass metode moguće se prebaciti na ES6 metodu raznim alatima koji nam automatski prebacuju jednu metodu u drugu. Sintaksa je malo različita u usporedbi s prvim primjerom, no u React-u nema dobrog ili lošeg načina deklariranja komponenti, stoga se mogu koristiti ona metoda koja je korisniku jednostavnija ili praktična. No unatoč tome, Facebook je nekoliko puta izrazio želju da se React.createClass u potpunosti ukloni kako bi se samo ES6 metoda mogla koristiti.

```
const Component = React.createClass({
    render ()
    {
        return (
            <p>Dobar dan!</p>
        );
    }
});
```

Programski kod 3.2. – ispis poruke u React-u

Kao prototipni jezik, JavaScript nije imao način na koji bi se mogle kreirati klase neko vrijeme. To je bio razlog zašto je React izmislio svoj sistem po kojemu bi se klase mogle kreirati, a to je bio React.createClass sistem.

Pojavom ES6, dolazi do promjena te nam React omogućuje implementaciju komponente pomoću class sintakse. Rezultat je isti, ali se koristi drugačiji stil, odnosno drugačiji način pisanja programskog koda.[13]

3.2.1. JSX

JSX je ekstenzija sintakse JavaScript-e te se koristi kod React-a za opisivanje korisničkog sučelja. JSX nam omogućuje kreirati React elemente koji se renderiraju u DOM-u., no kod korištenja React-a nije nužno koristiti JSX, već samo JavaScript.[14]

No, postoje situacije kada se preporučuje koristiti JSX, a to su situacije u kojima sintaksa nije više efikasna i nije više čitljiva. Ipak, korištenje se u nekim situacijama se preporučuje jer je korisno kao vizualna pomoć kada se radi na korisničkim sučeljima pisanjem u JavaScript kodu. Korištenje JSX-a nam donosi:

- brzinu jer koristi optimizaciju kod kompajliranja u JavaScript
- pomoć kod uočavanja grešaka tijekom kompajliranja
- jednostavnije i brže pisanje predloška uz znanje HTML-a [15]

Tijekom korištenja JSX-a, naš programski kod izgleda slično kao i kada se koristi samo HTML uz neke razlike. Možemo koristiti vlastite kreirane atribute uz korištenje već postojećih HTML elemenata i atributa. Kada želimo kreirati vlastite atribute, koristimo prefiks „data-“ kao u slijedećem primjeru gdje je korišten naziv „data-atribut“:

```
import React from "react";
import ReactDOM from "react-dom";
import "./styles.css";

class App extends React.Component {
    render() {
        return (
            <div>

                <h1>Naslov</h1>
                <h2>Tekst</h2>
                <p data-atribut="somevalue">Dobar dan.</p>

            </div>
        );
    }
}

const rootElement = document.getElementbyId("root");
ReactDOM.render(<App />, rootElement);
```

Programski kod 3.3. – korištenje JSX-a

Korištenjem JSX možemo koristiti razne izraze kao što je zbrajanje ili kod korištenja uvjetnih izraza. Ne mogu se koristiti „if else“ izrazi kao što se mogu koristiti na primjer u programskom jeziku C++, ali se može stvoriti uvjet koji nam govori je li neka radnja istina (true) ili laž (false).

Osim toga je također poželjno koristiti camelCase sintaksu ako želimo dodati stil našem tekstu. Na taj način možemo mijenjati boju teksta ili font, a osim stila možemo dodati uvjet koji će nam u određenim situacijama dati rješenje koje može biti „true“ ili „false“.

No stil se također može jednostavno promijeniti u posebnom CSS dokumentu. Oba dva načina funkcioniraju te ovisi o korisnicima koji način žele koristiti.

```
import React from "react";
import ReactDOM from "react-dom";
import "./styles.css";

class App extends React.Component {
    render() {
        var a = 0;
        var b = 1;
        var stil = {
            color: "#FFA500"
        }
        return (

            <div>
            <h1 style = {stil}>{a == 10 ? 'True' : 'False'}</h1>
            <h2>{a == 0 ? 'True' : 'False'}</h2>
            <h3>{b == 0 ? 'True' : 'False'}</h3>
            </div>

        );
    }
}

const rootElement = document.getElementbyId("root");
ReactDOM.render(<App />, rootElement);
```

Programski kod 3.4. – korištenje true i false izraza

3.2.2. Babel.js

Babel je besplatni open-source JavaScript kompajler koji se sastoji od raznih plugins-a. U današnje vrijeme se vrlo često koristi kod JavaScript programskog jezika. Ovaj alat se koristi kod konvertiranja ECMAScript 2015+ koda tako da će taj JavaScript kod biti čitljiv u svim web preglednicima. [16]

Uz to valja i naglasiti da neki web preglednici imaju probleme sa određenim stavkama kod ES6 standarda pa se često Babel koristi u svrhu pretvorbe ES6 standarda u ES5 standard zbog bolje funkcionalnosti.

Babel može konvertirati JSX sintaksu te se često koristi kod izrade React aplikacija, no kao i JSX, nije potrebno koristiti Babel kod izrade svake React aplikacije.

3.3. State i props objekti

Props (skraćeno od „properties“ što u prijevodu znači svojstva) i state ili stanje su JavaScript objekti. Props je zapravo način na koji komponente komuniciraju međusobno. U Reactu props se koristi kako bi se podaci prebacili od „parent“ komponente do „child“ komponente (odnos roditelj/dijete).

Vrijednost im je konstantno ista, što znači da se neće promijeniti. Protok podataka gdje komponente primaju podatak od strane „parent“ znači da je u React-u protok jednosmjeran. No, to je samo u slučaju za props. Prije je postojao način mijenjanja props pomoću funkcije setPropr() i replaceProps(), ali se to ukinulo prije brojnih ažuriranja.

Ako imamo podatke koji neće ostati nepromjenjivi, tada koristimo state objekt i funkciju setState() gdje će se nakon ažuriranja komponenta renderirati. State sadrži „privatne“ informacije o komponenti.

Na prvi pogled programski kod izgleda kompliciranije kada se koristi state objekt, no važno je samo naglasiti nekoliko važnih točaka po čemu se state i props objekti razlikuju.

State objekt je kreiran u komponenti, moguće ga je promijeniti funkcijom setState() nakon čega se automatski renderira komponenta ili se može koristiti funkcija prevState() koja se koristi za povratak na prijašnje stanje.

Ukratko, state objekti djeluju slično kao i props objekti, ali je state objekt kreiran u komponenti i nalazi se u jednoj komponenti te može se promijeniti kada mi to želimo, a props objekti su vrsta objekata koji se često i nazivaju read-only objekti.

Primjer pisanja programskog koda gdje se koriste state i props objekti kod trake za pretraživanje (eng. search bar). [17]

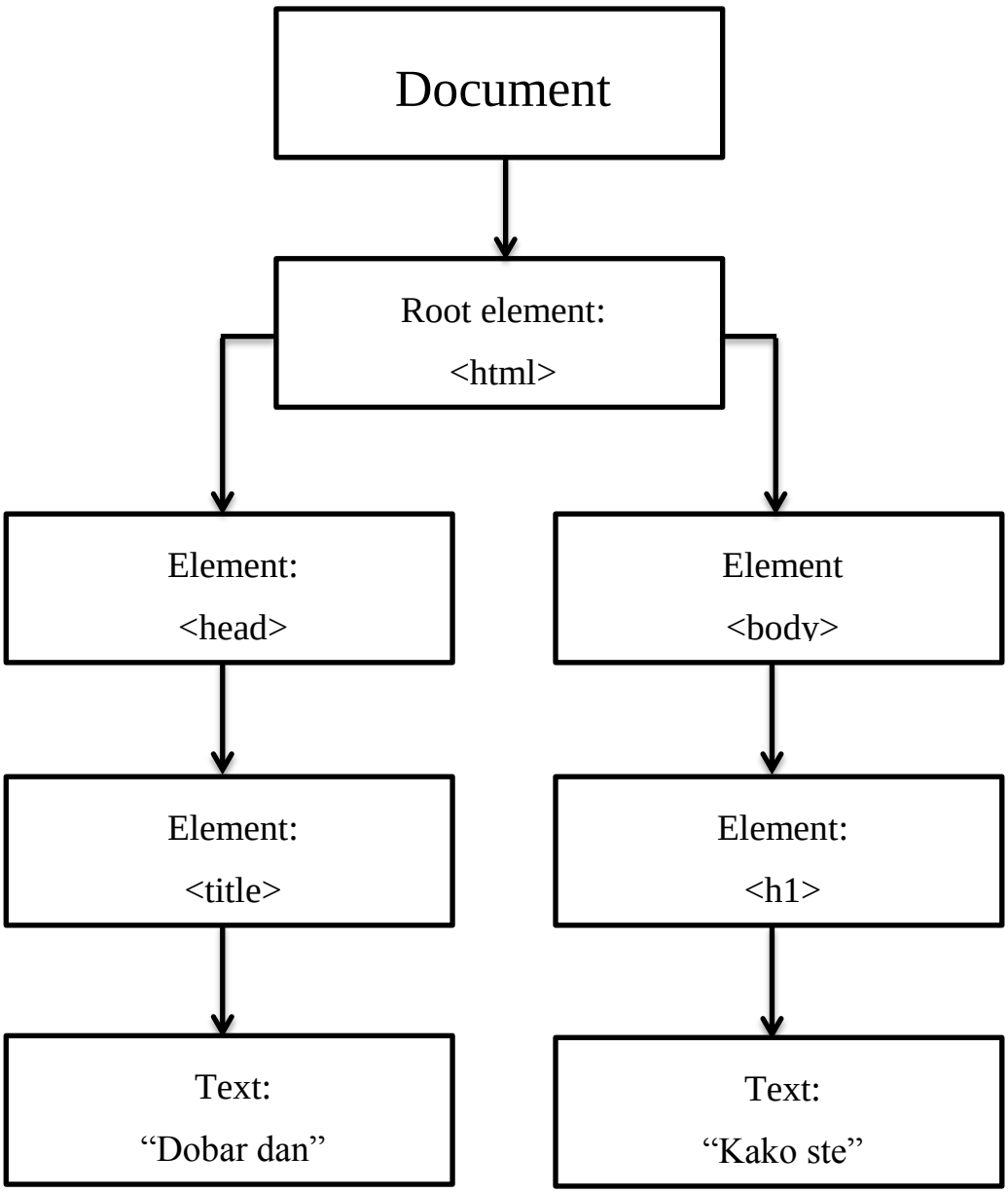
```
Class SearchBar extends Component {
    constructor(props) {

        super(props);
        this.state = { term: '' };
    }
    render() {
        return (
            <div className="search-bar">
                <input

value={this.state.term}
onChange={event => this.onInputChange(event.target.value)} />
            </div>

        );
    }
    onInputChange(term) {
        this.setState({term});
        this.props.onSearchTermChange(term);
    }
}
```

Programski kod 3.5. – korištenje state i props objekata

4. Virtualni DOM

DOM (eng. Document Object Model) je apstrakcija HTML strukture stranice nekog web mjesta i to je W3C standard. Uzima HTML elemente i pakira ih u jedan objekt u hijerarhijskom obliku te zadržava odnos roditelj/djeca (eng. parent/child relationship) kod tih HTML elemenata. Za manipuliranjem DOM-a se koristi JavaScript, no treba biti oprezan jer se zbog manipuliranja smanjuje efikasnost s obzirom da je DOM bio predviđen za statične web stranice, a ne dinamične.

HTML DOM struktura objekata može izgledati ovako:



Slika 4.1 – prikaz HTMl DOM strukture

Problem se javlja kod renderiranja. Kada imamo veći broj stavki i jedna od tih stavki zahtjeva ažuriranje, odnosno update. Kada se jedna stavka ažurira, cijela lista stavki se renderira što nije efikasno zbog mogućeg velikog kapaciteta pa je iz tog razloga React počeo koristi virtualni DOM.[18]

Virtualni DOM je zapravo kopija DOM-a. Kada želimo izvršiti promjenu nad nekom stavkom, stvara se novi virtualni DOM. Pomoću metode setState() možemo napraviti novi virtualni DOM, a sam proces je brz i ne utječe na performanse sadržaja. React upravlja s dva virtualna DOM-a koja su u dva različita stanja – jedno je novo stanje, a drugo je prijašnje stanje, odnosno stanje prije ažuriranja. Na taj način se pravi DOM može ažurirati u minimalnom broju koraka koristeći se diff algoritmom koji se koristi za takvu situaciju gdje se uspoređuju dva virtualna DOM-a.[19]

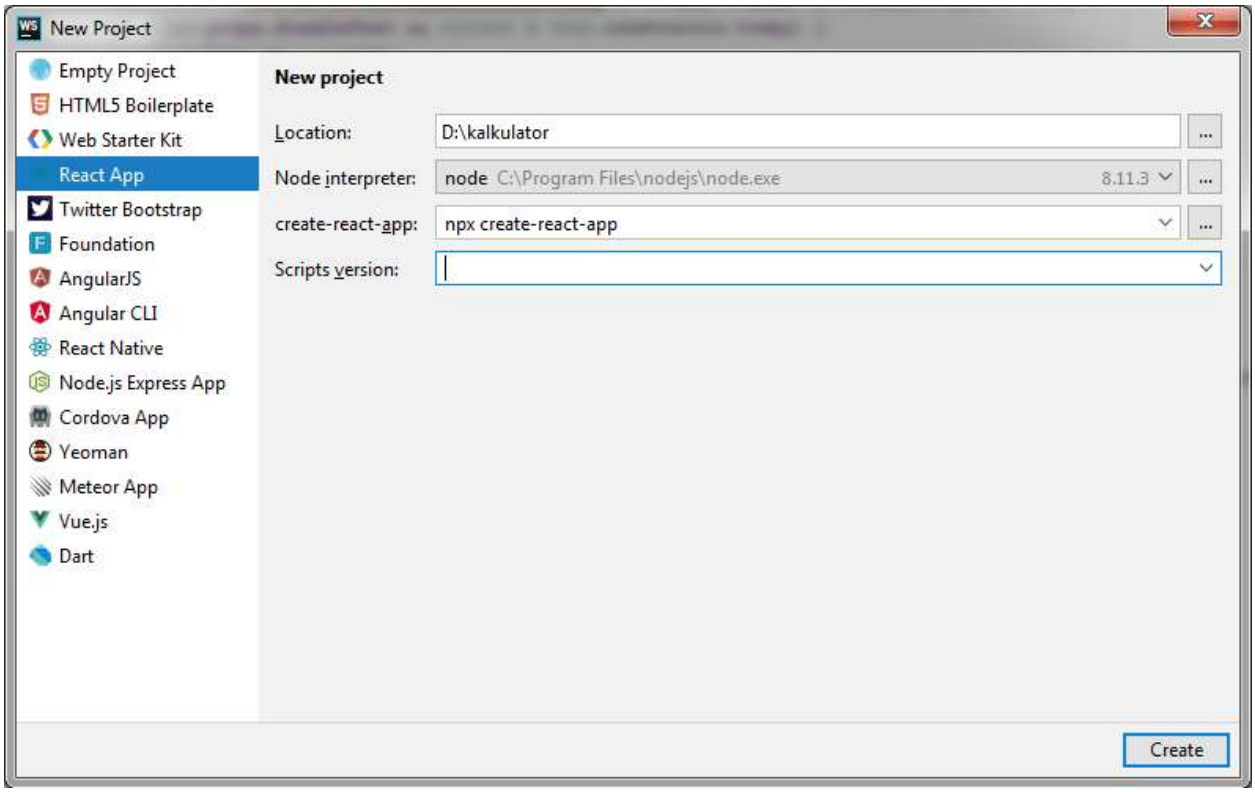
React koristi virtualni DOM i to je vrlo efikasan način za ažuriranje potrebnih podataka. Renderiranje podataka se izvršava u onom trenutku kada opažamo da su podaci „dirty“ te se za renderiranje koristi virtualni DOM kako bismo dobili završni pravi DOM.

5. Primjer korištenja ReactJS-a u Jetbrains WebStorm

5.1. Opis rada

Za praktični dio sam odlučio napraviti kalkulator za računanje jednostavnih računskih operacija kao što su plus (+), minus (-), puta (*) i dijeljeno (/), a uz te računske operacije su prisutne i tipke koje mogu obrisati naš posljednji unos ili cijelu operaciju te znak jednakosti. Kalkulator će promijeniti svoj izgled ako promijenimo veličinu prozora na zaslonu pa stoga ima određenu responzivnost. Za rad sam koristio uglavnom IDE JetBrains Webstorm uz manju pomoć CodePen na početku. Ovo nije jedini način za izradu kalkulatora, iako je način izrade kalkulatora sličan bez obzira na koji način se unaša željeni programski kod. Cijeli programski kod može se vidjeti na web stranici:

https://codepen.io/Stamen9000/pen/MqWzRM



Slika 5.1. – IDE JetBrains WebStorm

5.2. Izgled i funkcionalnost kalkulatora

Svoj programski kod sam započeo pisati kreiranjem komponenti kako bi kalkulator mogao normalno funkcionirati. Koristio sam metodu React.createClass(). Ovdje sam napisao najvažnije dijelove programskog koda svojeg zadatka.

```
const unos = React.createClass({
  displayName: "unos",
  render: function render() {
    return React.createElement(
      "div",
      { className: "unos" },
      React.createElement(
        "p",
        { className: "prije" },
        this.props.prijeString
      ),
      React.createElement(
        "p",
        { className: "iznos" },
        this.props.iznosString
      )
    );
  }
});
const tipka = React.createClass({
  displayName: "tipka",
  render: function render() {
    return React.createElement(
      "div",
      { className: "tipka", onClick: this.props.onClick },
      this.props.sadrzaj
    );
  }
});
```

Programski kod 5.1. – pisanje unosa

Kako bi kalkulator mogao normalno funkcionirati, napravljen je na način da se sastoji od više dijelova koji zajedno funkcioniraju i omogućuju unos podataka i konačan ispis.

Prvi dio je dio koji se odnosi na unos podataka, odnosno brojeva. Svaki kalkulator sadrži ovaj dio te će se tipkom lijevog miša unijeti željena vrijednost koja se kasnije može izbrisati ili se može nastaviti s pisanjem kako bi se mogla izračunati neka računska operacija. Programski kod za ovaj dio kalkulatora može izgledati ovako:

```
const broj = React.createClass({
  displayName: "broj",
  render: function render() {
    return React.createElement(
      "div",
      { className: "broj" },
      React.createElement(tipka, { sadrzaj: "7",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "8",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "9",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "4",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "5",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "6",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "1",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "2",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "3",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "0",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: ".",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "*0.1",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "*10",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "*100"
,onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj:
"*1000",onClick:this.props.unos })
    );
  }
});
```

Programski kod 5.2. – definiranje brojeva na kalkulatoru

Ovdje se mogu dodavati razne vrijednosti ako želimo tisućice, neki decimalni broj ili decimalna točka. Također se može promijeniti redoslijed unosa ako želimo da nam određena tipka bude na mjestu na kojem mi želimo da ta tipka bude. Kada pritisnemo na neku od ovih vrijednosti one će se pojaviti u gornjem dijelu kalkulatora.

Uz ovaj dio koji služi za unos brojeva, potreban je dio na kojemu se nalaze standardni aritmetički operatori (/, *, -, +) te dio u kojemu se nalazi znak jednakosti (=) i dvije tipke pod nazivom „DEL“ i „CLR“ koje se koriste za brisanje jedne unesene vrijednosti ili brisanja svega što smo unijeli.

```
const rijesi = React.createClass({
    displayName: "rijesi",
    render: function render() {
        return React.createElement(
            "div",
            { className: "rijesi" },
            React.createElement(tipka, { sadrzaj: "DEL", onClick:
this.props.del }),
            React.createElement(tipka, { sadrzaj: "CLR", onClick:
this.props.clear }),
            React.createElement(tipka, { sadrzaj: "=",   onClick:
this.props.iznos }),
        );
    }
});

const Operator = React.createClass({
  displayName: "Operator",
  render: function render() {
    return React.createElement(
      "div",
      { className: "operator" },
      React.createElement(tipka, { sadrzaj: "/",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "*",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "-",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "+",
onClick:this.props.unos }),
      React.createElement(tipka, { sadrzaj: "MOP",
onClick:this.props.iznos })
    );
  }
});
```

Programski kod 5.3. – definiranje funkcionalnih tipki

Ovdje sam napisao programski kod koji rješava dio za brisanje jednog znaka, brisanje cijelog procesa ili za znak jednakosti, odnosno iznos.

```
clear: function clear() {
  this.setState({
    iznosString: "",
    prijeString: "" });
},
del: function del() {
  const string = this.state.iznosString.slice(0, -1);
  this.setState({ iznosString: string });
},
iznos: function iznos() {
  const result = eval(this.state.iznosString),
      prije = this.state.iznosString;
  this.setState({ iznosString: result, prijeString: prije });
},
```

Programski kod 5.4. – pisanje koda za znak jednakosti i brisanje vrijednosti

Kada god se unese neka vrijednost, ta vrijednost se pojavi u gornjem dijelu kalkulatora. Kada se pokazivač miša nalazi na bilo kojoj tipki, ta tipka se zacrveni (hover).

Na kraju dokumenta se nalazi programski kod ReactDOM.render.

```
  render: function render() {
    return React.createElement(
      "div",
      { className: "kalkulator" },
      React.createElement(unos, { prijeString: this.state.prijeString,
iznosString: this.state.iznosString }),
      React.createElement(klik, { unos:this.unos,iznos:this.iznos,
del: this.del, clear: this.clear })
    );
  }
});
ReactDOM.render(React.createElement(kalkulator, null),
document.querySelector('body'));
```

Programski kod 5.5. – dio koda za renderiranje

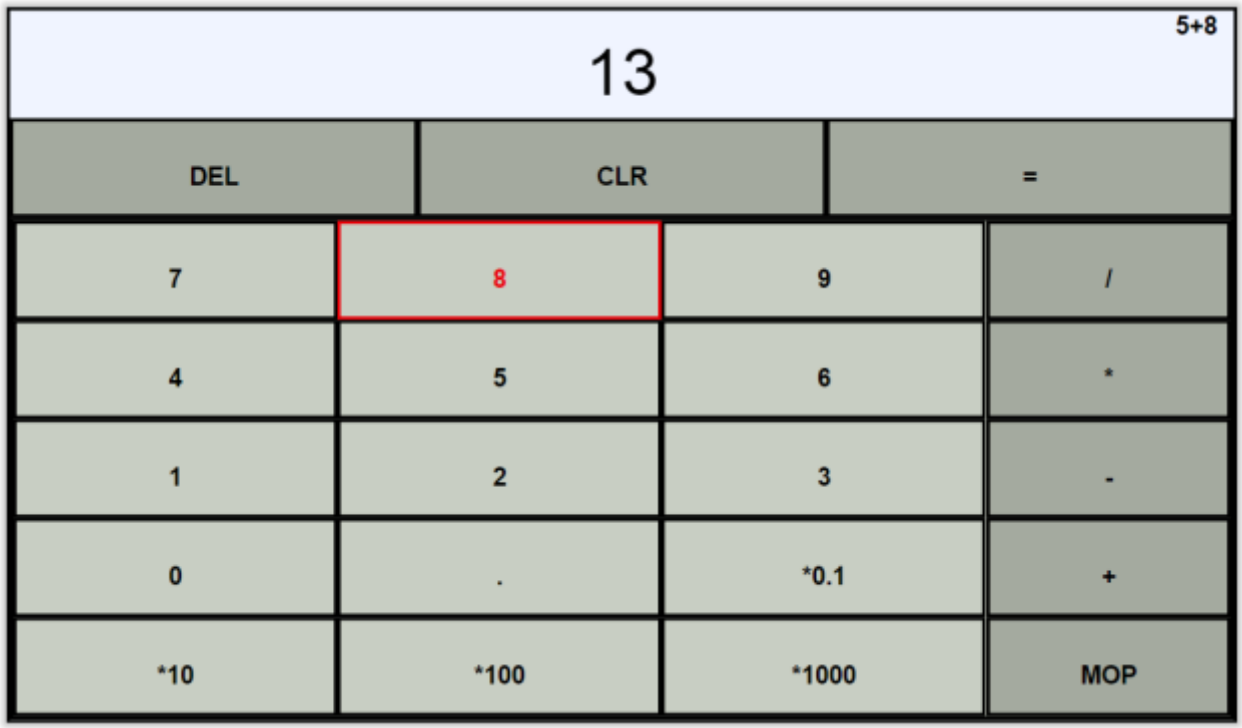
```css
/*CSS stil kalkulatora*/
* {
  position: relative;
  box-sizing: border-box;
  font-family:"arial";
  text-align: center;
  font-weight: 600;
  font-size: 30px;
}
/*stil tijela digitrona (položaj)*/
body, html {
  height: 950px;
  margin: 0;
  overflow: hidden;
}
/*stil digitrona*/
.kalkulator {
  width:70%;
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  background-color: #f0f4ff;
  box-shadow: 0 0 10px 5px rgba(0, 0, 0, 0.2);
  border-top: solid 4px;
  border-bottom: solid 4px;
  border-left: solid 4px;
  border-right: solid 4px;
}
/*stil digitrona gornji dio*/
.kalkulator .unos .prije {
  margin: 0;
  padding-right: 20px;
  height: 10px;
  line-height: 40px;
  text-align: right;
}
/*stil digitrona gornji dio za unos*/
.kalkulator .unos .iznos {
  width: 80%;
  height: 120px;
  line-height: 130px;
  padding: 10px;
  padding-top: 0;
  margin: 0 auto;
  font-size: 80px;
  overflow: hidden;
  font-weight: 500;
  font-family: Arial;
  user-select: auto;
}
/*stil za C DEL =*/
.kalkulator .rijesi {
  display: flex;
  width: 100%;
  background-color: #a4aa9f;
}
```

```css
/*stil za C DEL =*/
.kalkulator .rijesi .tipka {
  width: 100%;
  border-top: solid 4px;
  border-bottom: solid 4px;
  border-left: solid 4px;
  border-right: solid 4px;
}
/*stil za donji dio digitrona*/
.kalkulator .box {
  width: 100%;
  display: flex;
  flex-wrap: wrap;
  background-color: #c8cec3;
  border-top: solid 4px;
  border-bottom: solid 4px;
  border-left: solid 4px;
  border-right: solid 4px;
}
/*stil za donji dio digitrona*/
.kalkulator .box .broj {
  width: 80%;
  display: flex;
  flex-wrap: wrap;
}
/*stil za donji dio digitrona*/
.kalkulator .box .tipka {
  width:33.3%;
  border-top: solid 4px;
  border-bottom: solid 4px;
  border-left: solid 4px;
  border-right: solid 4px;
  font-size: 30px;
}
/*stil za donji dio digitrona*/
.kalkulator .box .operator {
  width: 10%;
  display: flex;
  flex-direction: column;
}
/*stil + - / **/
.kalkulator .box .operator .tipka {
  width: 200%;
  background-color: #a4aa9f;
  border-top: solid 4px;
  border-bottom: solid 4px;
  border-left: solid 4px;
}
/*stil + - / **/
.tipka {
  height: 120px;
  cursor: pointer;
  line-height: 4.5em;
}
/*stil kada je pokazivač miša na tipki*/
.tipka:hover {
  color: #eb0008;
```

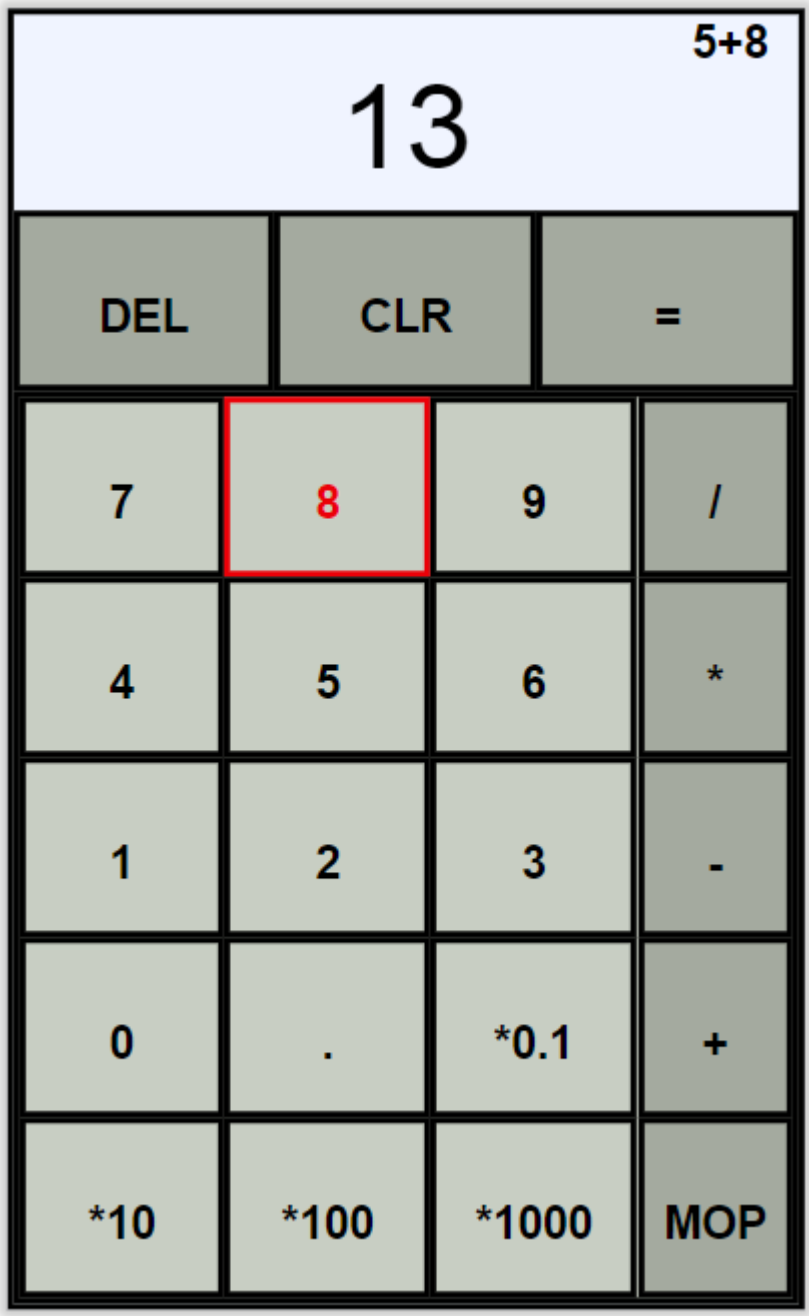
```
}
/*stil za promjenu veličine (responzivnost)*/
@media screen and (max-width: 480px) {
  body, html {
    height: 91vh;
  }
  .kalkulator {
    width: 90%;
  }
  .tipka {
    height: 3.5em;
    line-height: 3.5em;
  }
  .kalkulator .unos .iznos {
    height: 1.7em;
    line-height: 1.7em;
  }
}
```

Programski kod 5.6. – kod za izgled kalkulatora

Izgled kalkulatora će se mijenjati na drugačijoj širini ekrana što se može vidjeti na dvije fotografije ispod teksta. Taj pojam se još naziva responzivni dizajn.



Slika 5.2. – konačan izgled kalkulatora 1



Slika 5.3. – konačan izgled kalkulatora 2

6. Zaključak

Koji je najbolji razvojni okvir ili JavaScript biblioteka? Na to pitanje ne postoji odgovor. Prije svega moramo znati koji je naš zadatak, odnosno koji cilj želimo postići. Mi sami biramo što želimo koristiti prema našim osobnim mišljenjima nakon što isprobamo nekoliko opcija. Naravno, svaka JavaScript biblioteka je različita i možda ćemo ponekad morati koristiti onu koju ne preferiramo jer nam omogućuje izradu našeg zadatka, no ja ne mislim da je to loša stvar. Mogućnost izbora JavaScript biblioteke nam daje mnogo više opcija i poznavanje većeg broja razvojnih okvira ili JavaScript biblioteka nam omogućuje bolje aplikacije što je potrebno kako bi se ostvarilo bolje korisničko iskustvo.

Do pisanja ovog rada sam uglavnom koristio HTML i CSS opisne jezike za kreiranje web stranica te sam stoga imao dovoljno predznanje kako bih se mogao koristiti React bibliotekom u kojoj se može dovoljno brzo naučiti osnove rada.

Rad s React JavaScript bibliotekom mi je pomogao da bolje shvatim proces izrade aplikacije i planiram ga koristiti u slobodno vrijeme za daljnje učenje. Osim kalkulatora kojeg sam napravio, u slobodno vrijeme sam pisao programske kodove za nekoliko drugih aplikacija i mogu reći da nam React zaista omogućuje izradu mnogo različitih i funkcionalnih aplikacija.

Sviđa mi se jednostavnost pisanja koda, ali i raspoređenost i funkcionalnost svih dokumenata tijekom moje izrade zadatka u programu JetBrains Webstorm. Programski kod u dokumentima je pregledan i svatko ga može pročitati.

Uz dovoljno znanje je moguće koristiti React sa drugim bibliotekama ili razvojnim okvirima za izradu kompleksnijih aplikacija te imam želju u budućnosti više učiti o JavaScript bibliotekama kako bih mogao izraditi nove zanimljivije aplikacije.

U Varaždinu, _____ _____

Potpis studenta

Sveučilište
Sjever

SVEUČILIŠTE SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, FILIP STAMENIĆ *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom IZRADA WEB APLIKACIJE POMOĆU REACT JAVASCRIPT KNJIŽNOG OKVIRA *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)
Filip Stamenić
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, FILIP STAMENIĆ *(ime i prezime)* neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom IZRADA WEB APLIKACIJE POMOĆU REACT JAVASCRIPT KNJIŽNOG OKVIRA *(upisati naslov)* čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)
Filip Stamenić
(vlastoručni potpis)

Literatura

[1] http://www.kako.hr/c/kako-i-sto-je-programiranje-programiranje-opcenito,dostupno 15.06.2018.

[2] https://hr.wikipedia.org/wiki/Ra%C4%8Dunalno_programiranje, dostupno 15.06.2018.

[3] https://web-usluge.com/blog/58-kodiranje/7-kodiranje-web-stranice,dostupno 16.06.2018.

[4] M. Tomiša, M. Čačić, Grafički i programski alati u web dizajnu, dostupno na: http://app.eva-sms.com/claroline/claroline/backends/download.php?url=LzAzX1dEX0dyYWZpYy1raV9pX3Byb2dyYW1za2lfYWxhdGkucGRm&cidReset=true&cidReq=VZWD1718

[5] https://hr.wikipedia.org/wiki/JavaScript, dostupno 16.06.2018.

[6] https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf, dostupno 17.06.2018.

[7] https://www.componentsource.com/product/webstorm, dostupno 17.06.2018.

[8] https://www.quora.com/What-are-the-pros-and-cons-of-WebStorm-JetBrains-IDE-for-JavaScript, dostupno 18.06.2018.

[9] https://reactjs.org/, dostupno 18.06.2018.

[10] http://www.mono.hr/2016/12/12/javascript-frameworks-code-camp/,dostupno 20.06.2018.

[11] https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d, dostupno 21.06.2018.+

[12] https://reactjs.org/blog/2016/04/07/react-v15.html, dostupno 12.07.2018.

[13] https://www.fullstackreact.com/articles/react-create-class-vs-es6-class-components/, dostupno 22.06.2018.

[14] https://reactjs.org/docs/introducing-jsx.html, dostupno 24.06.2018.

[15] https://www.tutorialspoint.com/reactjs/reactjs_jsx.htm, dostupno 24.06.2018.

[16] https://babeljs.io/docs/en, dostupno 02.07.2018.

[17] https://hackernoon.com/understanding-state-and-props-in-react-94bc09232b9c, dostupno 25.06.2018.

[18] https://medium.com/@hidace/understanding-reacts-virtual-dom-vs-the-real-dom-68ae29039951, dostupno 27.06.2018.

[19] https://hackernoon.com/virtual-dom-in-reactjs-43a3fdb1d130, dostupno 29.06.2018.

Popis slika

Popis kodova

.

.