

Aspekti korištenja React biblioteke pri izradi reaktivnih mrežnih aplikacija

Aspects of Using the React Library for Building Reactive Web Applications

¹Florijan Kos, ²Sanja Brekalo

^{1,2}Međimursko veleučilište u Čakovcu, ¹student

e-mail: ¹florijan.kos@student.mev.hr,

²sbrekalo@mev.hr

Sažetak: *U radu se analiziraju prednosti i nedostaci React biblioteke u razvoju web aplikacija, s naglaskom na njenu primjenu u stvaranju dinamičnih, stabilnih i učinkovitih korisničkih sučelja. Među ključnim prednostima React biblioteke ističu se jednostavnost učenja i korištenja, višekratna uporaba komponenti, poboljšanje performansi zahvaljujući virtualnom DOM-u te mogućnost prilagodbe SEO optimizaciji. Također, alat React Developer Tools omogućuje lakše razvijanje, otklanjanje pogrešaka i optimizaciju aplikacija, dok React Native podržava razvoj mobilnih aplikacija koristeći istu logiku kao za web. Ipak, pri korištenju React biblioteke potrebno je ukazati na izazove poput čestih ažuriranja koja zahtijevaju kontinuirano praćenje promjena u samoj biblioteci, ograničenu dokumentaciju i oslanjanje na vanjske alate za osnovne funkcionalnosti poput formi i validacije podataka. Unatoč izazovima, React se potvrđuje kao moćan i fleksibilan alat koji potiče profesionalni rast i omogućuje inovativna rješenja u razvoju modernih web i mobilnih aplikacija.*

Ključne riječi: *React JavaScript, React Developer Tools, React Native, virtualni DOM, SEO optimizacija*

Summary: *The paper analyzes the advantages and disadvantages of React JavaScript in web application development, emphasizing its application in creating dynamic, stable, and efficient user interfaces. Key advantages include ease of learning and use, component reusability, performance improvements through the virtual DOM, and adaptability to SEO*

optimization. Additionally, React Developer Tools enable easier development, debugging, and optimization of applications, while React Native supports mobile app development using the same logic as web applications. However, the paper highlights challenges such as frequent updates requiring continuous tracking of changes, limited documentation, and reliance on external tools for basic functionalities like forms and data validation. Despite these challenges, React proves to be a powerful and flexible tool fostering professional growth and enabling innovative solutions in modern web and mobile application development.

Keywords: *React JavaScript, React Developer Tools, React Native, virtual DOM, SEO optimization*

1. Uvod

Facebook je 2011. godine prvi implementirao React.js, JavaScript biblioteku koju je razvio Jordan Walke, tadašnji softverski inženjer navedene globalne društvene platforme. Walke je bio jedan od pionira u stvaranju ovoga alata koji je ubrzo postao ključan za moderne web projekte.

React je osmišljen kako bi riješio izazove s kojima su se programeri suočavali prilikom razvoja web stranica i raznih softverskih rješenja. Ključni cilj bio je omogućiti izradu korisničkoga sučelja koje je istovremeno funkcionalno, vizualno privlačno te reaktivno. Danas su usluge razvoja temeljenoga na React.js-u izuzetno tražene jer ova biblioteka omogućuje stvaranje složenih korisničkih sučelja čije se komponente mogu dinamički mijenjati, bez potrebe za pisanjem velike količine različitoga koda. Pomoću Reacta moguće je mijenjati podatke unutar aplikacije bez utjecaja na stabilnost sustava ili korisničko iskustvo.

React se često smatra "V" slojem u okviru softverskog obrasca model-pogled-kontroler (engl. MVC: Model – View – Controller). U ovom pristupu, aplikacija se dijeli na tri ključna segmenta: model (M), koji upravlja pohranom i dohvatanjem podataka; pogled (V), koji je zadužen za prikaz korisničkog sučelja; i kontroler (C), koji obrađuje korisničke unose i komunicira s modelom. React preuzima ulogu pogleda, dok se za model i kontroler koriste drugi alati.

Svrha ovoga rada je analizirati prednosti i nedostatke korištenja React.js-a u izradi reaktivnih mrežnih aplikacija, s posebnim naglaskom na njegovu ulogu u optimizaciji korisničkoga iskustva i učinkovitosti razvoja. Razmatraju se ključni aspekti koji pridonose popularnosti Reacta među programerima, ali i izazovi s kojima se suočavaju prilikom njegove primjene. Ova analiza pruža uvid u mogućnosti koje React nudi, kao i potencijalne prepreke koje je potrebno uzeti u obzir pri donošenju odluke o njegovom korištenju u projektima.

2. Prednosti React JavaScript web aplikacija

React aplikacije mogu se koristiti kao web okvir (engl. *framework*) kada se uz njih primijene dodatni paketi, no mogu se i integrirati u postojeće klasične aplikacije. Većina JavaScript okvira koristi se za izradu jednostraničnih aplikacija (engl. *SPA - Single Page Application*), dok React omogućuje postupan prijelaz s tradicionalne arhitekture web aplikacija na arhitekturu jednostraničnih aplikacija, bez potrebe za potpunim prepisivanjem postojećega koda. [1], [2], [3].

2.1. Jednostavnost učenja i upotrebe

React aplikacije prepoznatljive su po jednostavnosti učenja i upotrebe, što ih čini atraktivnim izborom za programere. Bogata baza dokumentacije, raznovrsni tutorijali i brojni obrazovni resursi olakšavaju početak rada s Reactom. Osim toga, React omogućuje intuitivni način razvoja aplikacija, razdvajajući ih na manje, ponovno iskoristive komponente. Ovaj pristup smanjuje složenost projekata i olakšava njegovo održavanje, čineći integraciju novih funkcionalnosti u postojeće sustave bržom i jednostavnijom [1], [4].

2.2. Jednostavnije kreiranje dinamičnih web aplikacija

React omogućuje razvoj dinamičnih web aplikacija s manje koda i većom funkcionalnošću, zahvaljujući korištenju JSX-a. JSX je posebna sintaksa koja omogućuje uključivanje HTML elemenata unutar JavaScript koda, čime se pojednostavljuje izrada komponenti i optimizira struktura sučelja. Ova značajka rezultira čitljivijim i učinkovitijim kodom, olakšavajući održavanje i unaprjeđenje aplikacija [2], [5].

2.3. Višekratne komponente

Jedna od ključnih prednosti Reacta je mogućnost stvaranja višekratno upotrebljivih komponenti. Svaka komponenta ima vlastitu logiku i generira manji dio HTML-a, što omogućuje bolju organizaciju koda i lakšu suradnju u timu. Upotrebom virtualnog DOM-a, React analizira promjene i optimizira ažuriranje, mijenjajući samo one dijelove aplikacije koji su se izmijenili, što značajno poboljšava performanse [1], [6].

2.4. Poboljšanje performansi

React koristi virtualni DOM kako bi unaprijedio performanse aplikacija. Virtualni DOM djeluje kao memorijska reprezentacija stvarnoga DOM-a i omogućuje učinkovito upravljanje ažuriranjima podataka unutar aplikacije. Ovaj pristup smanjuje opterećenje resursa preglednika i ubrzava rad aplikacija, čime se omogućuje glatko korisničko iskustvo, čak i kod kompleksnih aplikacija [1], [7].

2.5. Dobra podrška praktičnih alata

React pruža širok spektar specijaliziranih alata, od kojih je najpoznatiji React Developer Tools. Ovaj alat omogućuje detaljan pregled hijerarhije React komponenti, njihova svojstva i trenutno stanje, olakšavajući identificiranje problema i optimizaciju aplikacija. Time se značajno povećava učinkovitost i kvaliteta razvojnog procesa [1], [8].

2.6. Prilagođeno SEO optimizaciji

React omogućuje renderiranje aplikacija na poslužitelju, gdje se virtualni DOM pretvara u statičke HTML stranice, čime se poboljšava brzina učitavanja stranica i optimizacija za tražilice. Ova prednost osigurava kompatibilnost s web preglednicima i pridonosi boljoj vidljivosti aplikacija na tražilicama [1], [9].

React omogućuje poboljšanje SEO performansi putem Server-Side Rendering (SSR) alata poput Next.js-a. Ovaj okvir proširuje funkcionalnost React-a, omogućujući generiranje statičkih stranica i bolje rezultate na tražilicama. Ipak, kod aplikacija koje koriste samo Client-Side Rendering (CSR), potrebno je dodatno prilagoditi optimizaciju kako bi tražilice ispravno indeksirale sadržaj.

2.7. Jednostavnost testiranja koda

React aplikacije jednostavne su za testiranje zahvaljujući alatima poput Jesta i React Testing Library, koji omogućuju izolirano testiranje komponenti i otkrivanje pogrešaka u ranoj fazi razvoja. Ovo osigurava stabilnost i pouzdanost aplikacija, čak i tijekom ažuriranja [1], [10].

2.8. Razvoj mobilnih aplikacija

React Native omogućuje razvoj mobilnih aplikacija koristeći istu logiku kao i za web aplikacije. Ova platforma omogućuje primjenu istog koda za iOS i Android, smanjujući vrijeme razvoja i troškove, dok optimizirane performanse osiguravaju glatko korisničko iskustvo [2], [11].

3. Nedostaci React JavaScript aplikacija

3.1. Neprestano ažuriranje

Stalni razvoj tehnologije i ažuriranja nose sa sobom prednosti i nedostatke. S jedne strane, stalne inovacije osiguravaju da alati poput React-a ostanu moderni, učinkoviti i prilagođeni potrebama programera. S druge strane, brzo mijenjajuće okruženje može biti izazovno za programere koji se teško prilagođavaju novostima. Kako bi mogli slijediti promjene, programeri su često prisiljeni kontinuirano raditi na unapređenju svojih vještina i usvajanju novih metoda. Iako taj proces pruža priliku za profesionalni rast i razvoj, može biti zahtjevan i vremenski intenzivan [2]. Brze promjene utječu i na timove koji rade na dugotrajnim projektima. Takva okolina može otežati održavanje starijih aplikacija i prilagodbu novim verzijama React-a, osobito kada se koristi u kombinaciji s alatima i knjižnicama koje također prolaze kroz stalne promjene.

3.2. Loša dokumentacija

Zbog stalnih ažuriranja i ubrzanoga tempa razvoja biblioteke, često nema dovoljno vremena za izradu detaljnih i sveobuhvatnih priručnika. Dokumentacija često ne uspijeva držati korak s razvojem novih značajki i ažuriranja. Kako bi se riješio taj problem, zajednica React programera izrađuje vodiče te snima tutorijale za lakše korištenje novih verzija i alata. Iako taj pristup smanjuje potrebu za službenom dokumentacijom, može biti zahtjevan za početnike programere. Ovo posebno otežava integraciju React-a sa starijim

sustavima ili prilagodbu aplikacija u kojima je potrebno povezati više različitih tehnologija. Iako zajednica pruža brojne resurse, mnoge ključne informacije ostaju fragmentirane, što povećava složenost za programere početnike [1], [4].

3.3. Loša tehnološka pokrivenost

React omogućava fleksibilnost, ali zahtijeva da se programeri oslanjaju na vanjske alate za mnoge osnovne funkcionalnosti. To ga čini odličnim za prilagodljive projekte, ali manje pogodnim za projekte koji trebaju cjelovito rješenje. React ne nudi ugrađene module za forme, internacionalizaciju ili validaciju podataka. Programer mora ručno birati alate i konfigurirati ih. Kao primjer možemo navesti dodavanje alata poput Formik-a ili React Hook Form-a za rad s formama. Budući da React pokriva samo korisničko sučelje aplikacija, često je potrebno integrirati dodatne tehnologije kako bi se kreirala potpuno funkcionalna aplikacija [2], [6], [7].

4. Diskusija

React JavaScript postao je jedan od najpopularnijih alata za razvoj web i mobilnih aplikacija, ali njegova primjena ima i dobre i loše strane. Ovom analizom jasno su se istaknule ključne prednosti i mane koje utječu na odluku programera prilikom odabira tehnologije.

Kao prednosti React aplikacija trebalo bi istaknuti da je React poznat po svojoj jednostavnosti za učenje i primjenu, što ga čini privlačnim alatom za programere, bez obzira na razinu iskustva. Dokumentacija, tutorijali i podrška zajednice značajno olakšavaju proces razvoja, dok višekratne komponente omogućuju organiziraniji kod i jednostavnije održavanje [1], [3].

Velika prednost je i virtualni DOM, koji omogućuje brže performanse, osobito kod aplikacija s puno dinamičkih elemenata. Ovo korisnicima pruža glatko iskustvo, a programerima olakšava posao [7]. Osim toga, alati poput React Developer Tools čine razvoj intuitivnijim i pomažu pri rješavanju problema [8].

React nije samo prilagođen za web aplikacije, već i za mobilne, zahvaljujući React Native-u. Ta kompatibilnost omogućuje korištenje istog koda za različite platforme, čime se štede vrijeme i resursi [2], [11]. Prilagođenost SEO optimizaciji također je bitna, posebno za projekte u kojima je vidljivost na tražilicama ključna [9].

Unatoč brojnim prednostima, React ima i svoje izazove. Stalna ažuriranja i promjene mogu biti frustrirajuće za timove koji nemaju dovoljno resursa ili vremena za stalno praćenje novih verzija [2]. Zahtjev za kontinuiranim učenjem može biti iscrpljujući, posebno za manje timove ili programere početnike.

Loša dokumentacija često otežava početak rada s Reactom. Iako zajednica pomaže kroz vodiče i tutorijale, mnogi početnici osjećaju nedostatak detaljnih i jasnih uputa, što može otežati usvajanje novih funkcionalnosti [1], [4].

Još jedan problem je što React pokriva samo korisničko sučelje, dok za osnovne funkcionalnosti poput rada s formama, validacije podataka ili internacionalizacije

programeri moraju posezati za dodatnim alatima. Iako ovo daje fleksibilnost, povećava kompleksnost razvoja [2], [6].

Iako React sve više dominira kao izbor za razvoj modernih aplikacija, suočava se s izazovima zbog konkurencije iz drugih tehnologija poput Vue.js-a, Angulara ili Svelte-a. Ove alternative često nude jednostavnije pristupe za manje projekte ili uključuju ugrađene funkcionalnosti koje React zahtijeva od vanjskih alata. Uz to, brzo zasićenje tržišta tehnologijama za razvoj korisničkoga sučelja može otežati razlikovanje između različitih rješenja. Ovisnost o velikim korporacijama, poput Meta-e koja upravlja razvojem Reacta, također može predstavljati rizik za dugoročnu stabilnost tehnologije.

Proučavanjem prednosti i nedostataka React biblioteke donesena je SWAT analiza programskoga jezika prikazana u Tablici 1.

Tablica 1. SWAT analiza korištenja React biblioteke

SNAGE	SLABOSTI
Jednostavnost učenja i široka zajednica podrške	Oslanjanje na vanjske alate za osnovne funkcionalnosti
Višekratne komponente za skalabilnost i održavanje	Često nedostatna dokumentacija kod novih verzija
Virtualni DOM za bolje performanse	Potreba za stalnim usavršavanjem zbog brzih ažuriranja
Fleksibilnost i integracija s drugim alatima	Složenost kod integracije sa starijim sustavima
Podrška za SSR za bolju SEO optimizaciju	Izazovi u SEO optimizaciji kod CSR aplikacija
Kompatibilnost s mobilnim razvojem putem React Native-a	
PRILIKE	PRIJETNJE
Razvoj alata poput Next.js za bolju SEO optimizaciju	Konkurencija s alternativama poput Vue.js, Angulara ili Svelte-a
Porast potražnje za reaktivnim web i mobilnim aplikacijama	Brze promjene u JavaScript ekosustavu
Širenje tržišta za <i>full-stack</i> razvoj s Reactom	Kompleksnost za manje iskusne timove

Prilagodba industrijskim standardima	Potencijalno zasićenje tržišta tehnologijama za razvoj sučelja
Korištenje React Native-a za mobilne aplikacije	Ovisnost o velikim korporacijama poput Mete

Iz navedenog moguće je zaključiti da je React odličan izbor za projekte koji zahtijevaju visoke performanse, prilagodljivost i skalabilnost. Njegove prednosti dolaze do izražaja kod velikih projekata, dok su mane uglavnom povezane s manjim timovima ili projektima s ograničenim resursima.

Za timove koji mogu pratiti promjene i brzo učiti, React nudi priliku za profesionalni razvoj i modernizaciju aplikacija. Međutim, za projekte koji zahtijevaju cjelovita rješenja ili imaju stroge rokove, može biti korisno razmotriti druge opcije.

5. Zaključak

React.js je jedna od najpopularnijih JavaScript biblioteka za izradu modernih web i mobilnih aplikacija. Njegova modularnost, fleksibilnost i značajke poput višekratnih komponenti, virtualnog DOM-a i prilagodbe SEO optimizaciji čine ga moćnim alatom za razvoj složenih i prilagodljivih aplikacija.

U ovom radu analizirane su ključne prednosti i izazovi ove tehnologije. Prednosti, poput jednostavnosti učenja, učinkovitosti u kreiranju dinamičnih aplikacija te bogate podrške zajednice i alata, čine React idealnim izborom za programere koji rade na projektima koji zahtijevaju skalabilnost i visok stupanj prilagodljivosti. Ipak, rad ukazuje i na nedostatke, kao što su potreba za stalnim usavršavanjem zbog učestalih ažuriranja te oslanjanje na vanjske alate za određene funkcionalnosti.

React ostaje bitan u razvoju modernih aplikacija, no njegovo pozicioniranje na tržištu zahtijeva stalnu prilagodbu konkurenciji i promjenama u ekosustavu. Alati poput Next.js-a otvaraju nove mogućnosti za SEO optimizaciju, dok prisutnost Vue.js-a i Svelte-a potiče inovacije unutar zajednice. Programeri koji žele iskoristiti puni potencijal React-a moraju biti spremni uložiti dodatno vrijeme u prilagodbu i praćenje novih trendova.

Unatoč izazovima, React.js ostaje izvanredna platforma za izradu aplikacija prilagođenih potrebama suvremenoga digitalnoga tržišta. Njegova prilagodljivost, performanse i široka podrška zajednice jamče daljnji rast i relevantnost ove tehnologije. Programeri koji su spremni prihvatiti brze promjene i uložiti vrijeme u usvajanje dodatnih alata, mogu u potpunosti iskoristiti potencijal React-a te razviti aplikacije koje postavljaju nove standarde u industriji.

6. Literatura

- [1] M. Grbić, "Izrada React JavaScript web aplikacija: prednosti i mane," PopWebDesign Blog, 2023. [Online]. Dostupno:

https://www.popwebdesign.net/popart_blog/2023/05/izrada-veb-aplikacija-uz-react-js-prednosti-i-mane/. [Pristupljeno: 03.12.2024.].

- [2] "10 Key Advantages of ReactJS: Why Choose It for Your Next Project?," ProCoders Blog, 2024. [Online]. Dostupno: <https://procoders.tech/blog/advantages-of-using-reactjs/>. [Pristupljeno: 04.12.2024.].
- [3] "Quick Start" React Official Documentation, 2024. [Online]. Dostupno: <https://react.dev/learn>. [Pristupljeno: 05.12.2024.].
- [4] C. Boyd, "React JS for Beginners: Understanding the Why, Mastering the How, and Exploring the What," Medium, 9. travnja 2024. [Online]. Dostupno: <https://medium.com/@castonboyd/react-js-for-beginners-understanding-the-why-mastering-the-how-and-exploring-the-what-2082aad56052>. [Pristupljeno: 05.12.2024.].
- [5] "Writing Markup with JSX," React.dev, 2024. [Online]. Available: <https://react.dev/learn/writing-markup-with-jsx>. [Pristupljeno: 05.12.2024.].
- [6] "React Components and Props," React Official Documentation, 2024. [Online]. Dostupno: <https://legacy.reactjs.org/docs/components-and-props.html>. [Pristupljeno: 03.12.2024.].
- [7] "Inside Fiber: In-depth Overview of the New Reconciliation Algorithm in React," Angular.love, 15. ožujka 2020. [Online]. Dostupno: <https://angular.love/inside-fiber-in-depth-overview-of-the-new-reconciliation-algorithm-in-react>. [Pristupljeno: 04.12.2024.].
- [8] "React Developer Tools," Chrome Web Store, 2024. [Online]. Dostupno: <https://chrome.google.com/webstore>. [Pristupljeno: 08.12.2024.].
- [9] "Server Components," API Reference, 2024. [Online]. Dostupno: <https://react.dev/reference/rsc/server-components>. [Pristupljeno: 01.12.2024.].
- [10] "Testing React Apps," React Testing Library Documentation, 2024. [Online]. Dostupno: <https://testing-library.com/docs/react-testing-library/intro/>. [Pristupljeno: 02.12.2024.].
- [11] "React Native Introduction," React Native Documentation, 2024. [Online]. Dostupno: <https://reactnative.dev/docs/getting-started>. [Pristupljeno: 01.12.2024.].