

ANALIZA ZNAČAJKI I PERFORMANSI PROGRESIVNIH WEB APLIKACIJA

ANALYSIS OF PROGRESSIVE WEB APPLICATION FEATURES AND PERFORMANCE

Ognjen Staničić, Sanja Kraljević

Tehničko veleučilište u Zagrebu, Vrbik 8, Zagreb, Hrvatska

SAŽETAK

Progressivne web aplikacije su trend u svijetu web programiranja jer pružaju moćni skup funkcionalnosti spajajući najbolje strane web i mobilnih aplikacija. Kod planiranja izrade, česta je dilema da li se odlučiti za web ili mobilnu aplikaciju, a najčešće je potrebno razviti obje verzije aplikacije kako bi došli do što većeg broja korisnika. Progressivne web aplikacije koriste moderne mogućnosti web preglednika kako bi postigle korisničko iskustvo vrlo slično mobilnim aplikacijama a u okviru klasičnih web aplikacija, time kombinirajući najbolje od oba svijeta. U ovome radu su analizirane funkcionalnosti progresivnih web aplikacija, te je obrazloženo zašto pružaju odlično korisničko iskustvo i zašto su sve češća pojava na tržištu. Osim prednosti su proučene i mane, te su na primjeru gotove progresivne web aplikacije testirane performanse i značajke takve aplikacije.

***Ključne riječi:** progresivna web aplikacija, mobilna aplikacija, service worker, manifest*

ABSTRACT

Progressive web applications are a buzzword in the world of web development because they provide a powerful set of features combining the advantages of web and mobile applications. When starting to develop an application, there is often a dilemma whether to choose a web application or a native mobile application and frequently one must develop both versions of the application to get maximum user reach. Progressive web applications are web applications that use the capabilities of modern web browsers to achieve a user experience which is very mobile application

like within the classic web application, combining best of both worlds. In this paper features of progressive web applications are analyzed, why they provide great user experience and why are they on the rise. Pros and cons are also examined and on a finished example progressive web app its performance and features are tested.

***Keywords:** progressive web application, mobile application, service worker, manifest*

1. UVOD

1. INTRODUCTION

Ornare aenean euismod elementum nisi. Faucibus interdum posuere lorem ipsum. Nulla posuere sollicitudin aliquam ultrices sagittis orci a scelerisque purus. Amet facilisis magna etiam tempor orci eu lobortis elementum nibh. Vel pharetra vel turpis nunc.

Enim neque volutpat ac tincidunt vitae semper quis lectus. Morbi tristique senectus et netus. Purus semper eget duis at tellus at urna condimentum. Sit amet facilisis magna etiam tempor orci eu. Placerat vestibulum lectus mauris ultrices eros in. Non diam phasellus vestibulum lorem. Tellus elementum sagittis vitae et leo. Quis auctor elit sed vulputate mi. Facilisis mauris sit amet massa vitae tortor. Bibendum at varius vel pharetra vel turpis nunc. Nec ultrices dui sapien eget mi proin sed.

Progressivna web aplikacija (PWA) je noviji termin koji označava skup određenih funkcionalnosti koje možemo dodati modernim web aplikacijama kako bi ih poboljšali. Moderni web preglednici omogućavaju web aplikacijama velik broj novih mogućnosti koje prije nisu

bile moguće, a pomoću kojih web aplikacije mogu sve više sličiti nativnim mobilnim aplikacijama, kako u izgledu tako i u ponašanju. Epitet progresivno se odnosi na progresivno dodavanje funkcionalnosti kako bi web aplikacija izgledala kao nativna mobilna aplikacija, te kako bi korisnik imao korisničko iskustvo kao da koristi nativnu mobilnu aplikaciju. Naravno ako korisnik koristi stariji web preglednik, web aplikacija će i dalje funkcionirati sa osnovnim skupom funkcionalnosti kao i do sada. Tri glavne karakteristike progresivnih web aplikacija su pouzdanost (momentalno učitavanje, rad čak u izvanmrežnom načinu rada), brzina (brzi odgovori na korisničku interakciju s glatkim animacijama i prijelazima) i privlačnost (osjećaj nativne aplikacije s unaprijedom korisničkim iskustvom). [1] Ovo dovodi do sve većeg porasta progresivnih web aplikacija na tržištu [2], te će se u nastavku analizirati funkcionalnosti koje čine progresivne web aplikacije tako atraktivnima.

2. USPOREDBA PWA I NATIVNIH MOBILNIH APLIKACIJA

2. COMPARISON OF PWA AND NATIVE MOBILE APPS

Nativne mobilne aplikacije su trenutno dominantne na tržištu, a odlikuju ih moćne funkcionalnosti poput pristupa kameri uređaja, instalacije na početni ekran, push notifikacija, izvanmrežnog (engl. offline) načina rada i slično. S druge strane sam proces razvoja nativnih mobilnih aplikacija zbog različitih operacijskih sustava je kompleksniji nego razvoj web aplikacija. Također klasične web aplikacije su znatno pristupačnije. Nije ih potrebno instalirati, te ih je moguće pronaći kroz Google pretragu ili im pristupiti pomoću poveznice. Istraživanje je pokazalo da prosječan korisnik u prosjeku instalira nula novih aplikacija mjesečno kroz App store ili Google Play store. [3] Dodatno, novije verzije se učitavaju automatski za razliku od mobilnih aplikacija za koje je novije verzije potrebno ručno instalirati.

Progresivne web aplikacije uvode sve ove funkcionalnosti i još neke druge u klasične web aplikacije, što zapravo kombinira sve prednosti nativnih mobilnih aplikacija s prednostima

klasičnih web aplikacija, te njihov postepeni ulazak u široku uporabu će zasigurno dovesti do velikog porasta korištenja ovakvih web aplikacija uz potiskivanje nativnih mobilnih aplikacija.

Progresivnim web aplikacijama možemo jednostavno pristupiti pomoću URL-a, mogu se pretraživati i slati linkom kao klasične web aplikacije ili web stranice, a moguće ih je instalirati i to na puno jednostavniji način nego nativne mobilne aplikacije.

Bitno je napomenuti da progresivnost web aplikacije nije binarna, nego je spektar, odnosno aplikacija je u određenom postotku progresivna u ovisnosti koje sve značajke implementira.

3. PWA FUNKCIONALNOSTI

3. PWA FEATURES

Imajući to na umu, postoje tri nužna uvjeta koje web aplikacija mora zadovoljiti kako bi bila progresivna.

1. Sadrži manifest datoteku, pomoću koje se uređuju postavke aplikacije. Ovo također služi za omogućavanje vrlo jednostavnog instaliranja aplikacije na početni ekran jednim klikom.

2. Koristi mehanizam servisnog radnika (engl. service worker) web preglednika, kako bi omogućila neprekidno korištenje aplikacije što uključuje aplikacijsku ljusku (engl. application shell), izvanmrežni način rada, predmemoriju (engl. cache) za brže učitavanje i automatsku sinkronizaciju pri ponovnom spajanju na mrežu, te slanje push notifikacija.

3. Koristi isključivo HTTPS protokol koji je preduvjet za rad service workera, te kako bi omogućila razinu sigurnosti potrebnu za korištenje samostojeće verzije aplikacije nakon instalacije. [4]

Krvotok progresivnih aplikacija su već spomenuti service workeri. Oni su novija funkcionalnost modernih preglednika koji omogućuju JavaScript procesima da se izvršavaju u pozadini, odvojeni od HTML stranice u posebnoj dretvi, dok aplikacija nije aktivna. Služe primarno za reagiranje na različite događaje koji su došli od aplikacije, korisnika ili poslužitelja.

U nastavku je opisan skup funkcionalnosti koje web aplikacije čine progresivnima.

1. Application manifest

Application manifest je JSON datoteka koja omogućuje progresivnoj web aplikaciji da ju korisnik može dodati odnosno instalirati na početni ekran radi kasnijeg lakšeg pristupa. U ovoj datoteci se uređuju postavke o imenu, ikonama, boji teme, načinu prikaza i početnom URL-u aplikacije. Ovime se postiže dostupnost i osjećaj nativne aplikacije, sa puno bržom i jednostavnijom instalacijom.

2. Caching/offline podrška

Ovo se postiže tako da se pomoću service workera određeni dijelovi aplikacije pohrane u cache web preglednika, za koje želimo da budu dostupni kada korisnik nije spojen na mrežu. Moguće je čak i u cache pohraniti odgovore osnovnih AJAX GET upita s osnovnim sadržajem aplikacije. Ovo pruža puno bolje korisničko iskustvo. Također pomoću navedenog se postiže i brže učitavanje aplikacije jer statične dijelove možemo učitati iz cache-a i ako smo spojeni na mrežu. Takav osnovni dio aplikacije koji se brzo učita se još zove i aplikacijska ljuska.

3. Background sync

Pozadinska sinkronizacija (engl. background sync) je mehanizam slanja podataka web poslužitelju kada smo u izvanmrežnom načinu rada, primjerice kod privremenog prekida konekcije na internet. Sinkronizacija funkcionira tako da service worker ako nema pristupa mreži prilikom slanja podataka na web poslužitelj, pohrani podatke i stvori zadatak slanja na poslužitelj koji se aktivira kada mreža opet postane dostupna. Kombinacija ovoga i offline cachea čini privremeni gubitak mreže jedva osjetan korisniku što unapređuje korisničko iskustvo.

4. Web push

Web push je mehanizam pomoću kojega se korisniku šalje notifikacija sa web poslužitelja pomoću service workera čak i ako aplikacija nije otvorena. Ovo je iskustvo na koje su korisnici mobilnih aplikacija naviknuli i vrlo je korisno što je ovo moguće postići i u web aplikacijama.

5. Responzivni dizajn

Prava progresivna web aplikacija koja ima osjećaj mobilne web aplikacije mora implementirati sve principe responzivnog web dizajna kako bi se aplikacija ispravno prikazivala na svim uređajima.

6. Geolocation API

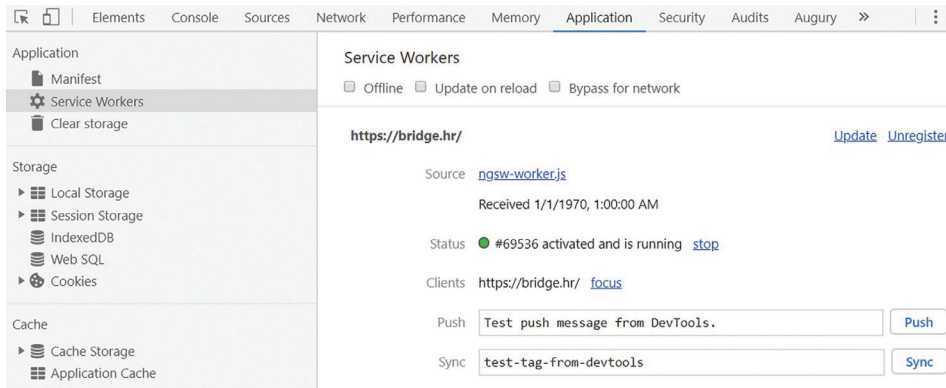
Geolocation API je mogućnost web preglednika koje nam pruža pristup lokaciji korisnika pomoću navigator objekta web preglednika, naravno ako korisnik isto dopusti. Ova funkcionalnost je često korištena u nativnim mobilnim aplikacijama a dostupna je i u progresivnim web aplikacijama.

7. Media API

Pomoću Media API-ja omogućen je pristup audio i video funkcionalnostima mobilnog uređaja. Navedeno se također postiže pomoću navigator objekta.

Korištenjem svih ovih mogućnosti postiže se izrazito bolje korisničko iskustvo kombinirajući najbolje stvari nativnih mobilnih i web aplikacija. Ako korisnik koristi web preglednik koji još ne podržava neke od ovih funkcionalnosti svejedno će sve ostale funkcionalnosti aplikacije raditi. No service workeri su trenutno dostupni u svim modernim web preglednicima [5], pa će sve funkcionalnosti aplikacije biti dostupne samo ako korisnik koristi najnovije inačice web preglednika. Važno je da ako korisnik koristi stariju inačicu koja ovo ne podržava, on i dalje može koristiti aplikaciju kao klasičnu web aplikaciju.

Imajući ovo sve na umu može se očekivati sve veći broj web aplikacija koji implementiraju neke ili sve ove funkcionalnosti u bliskoj budućnosti. Većina tvrtki koja je izdala PWA verziju svoje aplikacije je primijetila vrlo veliku korištenost te verzije i bolji conversion rate. Conversion rate se odnosi na postotak korisnika koji ostvari željeni cilj tvrtke pri korištenju njihove aplikacije. Najveća indijska tvrtka za e-poslovanje Flipkart je uvođenjem progresivne web aplikacije primijetila 70% poboljšani conversion rate te trostruku porast u provedenom vremenu korisnika na njihovoj stranici. Twitter je zabilježilo povećanje tweetova od 75% nakon izdavanja PWA verzije, te 65% porasta u broju posjećenih stranica po korisniku, uz smanjeni prijenos podataka i brže učitavanje. [2]



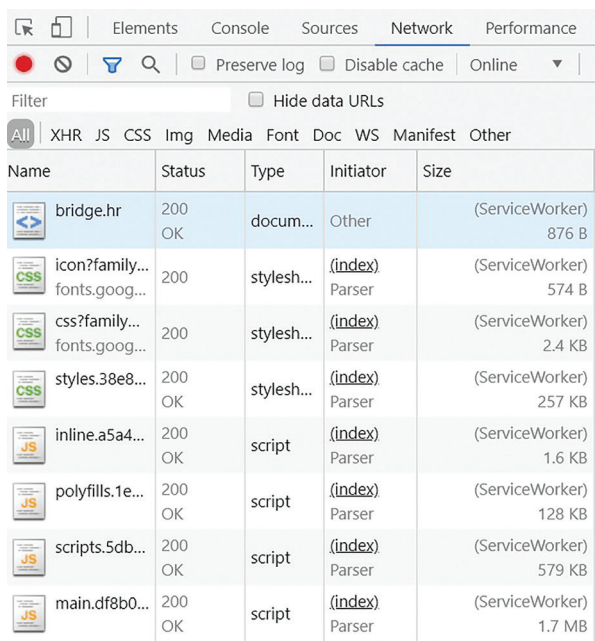
Slika 1 Service worker radi u pozadini

Figure 1 Service worker in the background

Jedan od razloga zašto PWA nije još u potpunosti preuzela vodstvo nad tržištem je nešto slabija podržanost svih funkcionalnosti na iOS uređajima, no očekuje se da će se i to popraviti ubrzo, te da će progresivna web aplikacija postati primarna opcija prilikom odabira vrste aplikacije. [6]

4. PWA POMOĆU ANGULARA 4. ENABLING PWA USING ANGULAR

Service workere, koji omogućavaju sve ove funkcionalnosti, je moguće upotrijebiti korištenjem čistog JavaScripta bez ikakvih radnih okvira. Service worker je dostupan pomoću serviceWorker svojstva navigator objekta u web preglednicima koji ga podržavaju. No ovaj način zahtjeva dosta koda i ručnog konfiguriranja.



Slika 2 Učitavanje iz cachea pomoću Service workera

Figure 2 Loading the application from cache via service worker

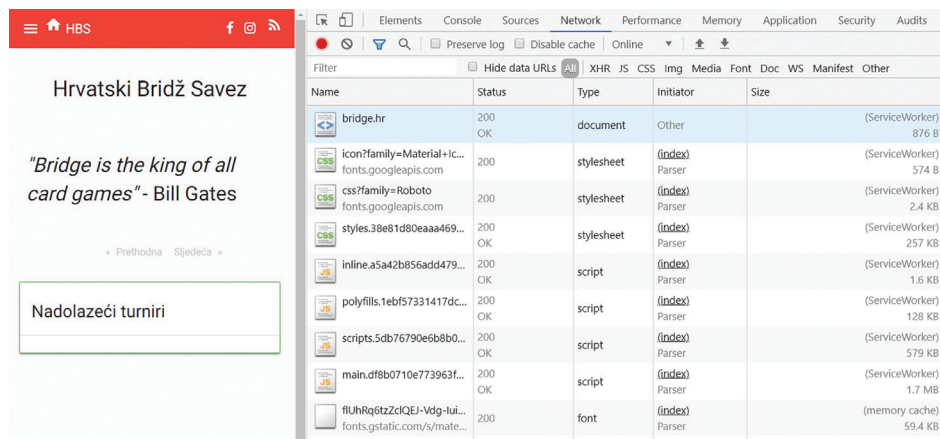
Postoje vanjske biblioteke koje olakšavaju rad sa service workerima kao i vlastita rješenja svih popularnih modernih JavaScript radnih okvira poput Angulara ili Reacta. Workbox je vanjska JavaScript biblioteka koja se može uključiti u projekt, time se mogu koristiti service workeri na mnogo jednostavniji način. Workbox pruža skup najboljih praksi i štedi nas pisanja boilerplate koda koji svaki programer mora inače pisati kada koristi service workere. [7]

U ovom radu će se analizirati progresivna web aplikacija izrađena pomoću radnog okvira Angular. U Angularu se service workeri omogućavaju pomoću posebnog ServiceWorkerModule modula koji je dio @angular/service-worker paketa kojeg je prethodno potrebno instalirati unutar Angular projekta. Instalacija ovoga paketa automatski postavlja pohranu osnovnih resursa aplikacije u cache.

Moguće je naknadno dodati pohranu i ostalih željenih resursa u cache, te dodatno konfigurirati service workera pomoću ngsw-config.json datoteke. Potom je potrebno napraviti manifest.json datoteku i uvesti ju u index.html aplikacije. Push notifikacije se omogućuju pomoću SwPush objekta koji je isto dostupan unutar ovog paketa.

5. TESTIRANJE PWA FUNKCIONALNOSTI NA PRIMJERU “HBS” APLIKACIJE 5. TESTING PWA FEATURES USING “HBS” APP

Glavne funkcionalnosti progresivne web aplikacije pokazane su na primjeru aplikacije Hrvatskog bridž saveza (HBS) koja je dostupna na URL-u https://bridge.hr.



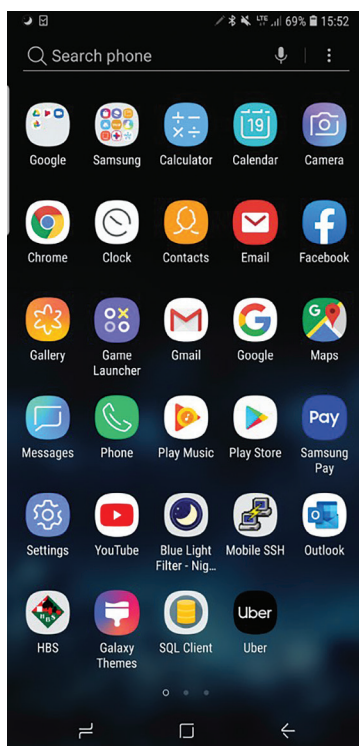
Slika 3 Izgled stranice ako ju učitamo bez pristupa mreži

Figure 3 The application in offline mode

HBS aplikacija koristi Angular radni okvir (engl. framework) i pomoću njega je instaliran service worker i dodatne PWA funkcionalnosti.

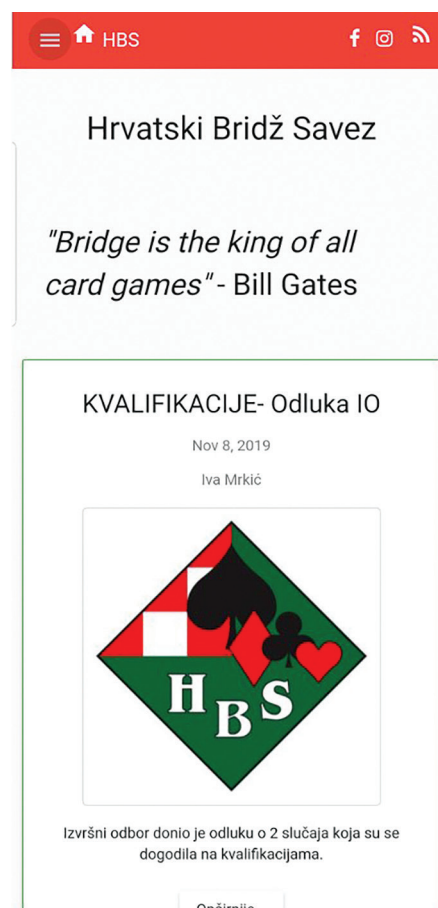
Service worker nakon instalacije i početne pohrane u cache prilikom prvog pokretanja dalje služi za brže učitavanje statičnih resursa stranice, kao i njihov prikaz u izvanmrežnom načinu rada. HBS aplikacija ima manifest.json datoteku pomoću koje, kada korisnik prvi puta pristupi aplikaciji u web pregledniku pomoću URL-a https://bridge.hr, mu se nudi opcija da ju doda na početni ekran, ukoliko to web preglednik podržava.

Nakon što se aplikacija otvori, izgleda kao nativna mobilna aplikacija, nema web preglednika niti alatne trake nego samo izbornik unutar aplikacije.



Slika 4 Instalirana aplikacija na početnom ekranu mobilnog uređaja

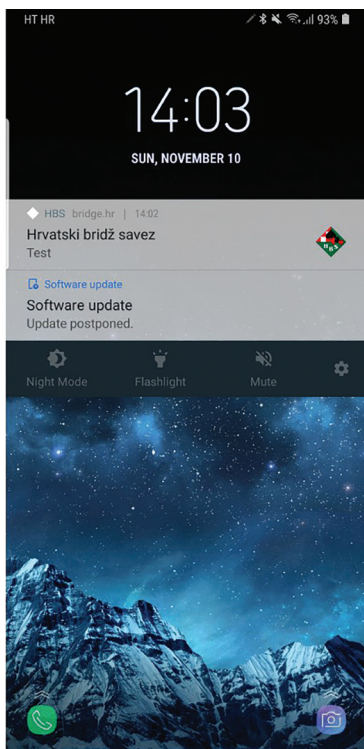
Figure 4 installed application on the home screen of the mobile device



Slika 5 Izgled otvorene aplikacije

Figure 5 The application after loading

U aplikaciji je također dodana opcija da se pošalje korisniku push notifikacija kada se objavi nova vijest. Za ovo se prvo mora pretplatiti pomoću gumba na vrhu izbornika sa prethodne slike. Pomoću tog gumba se i odjavi ako više ne želi primati notifikacije.



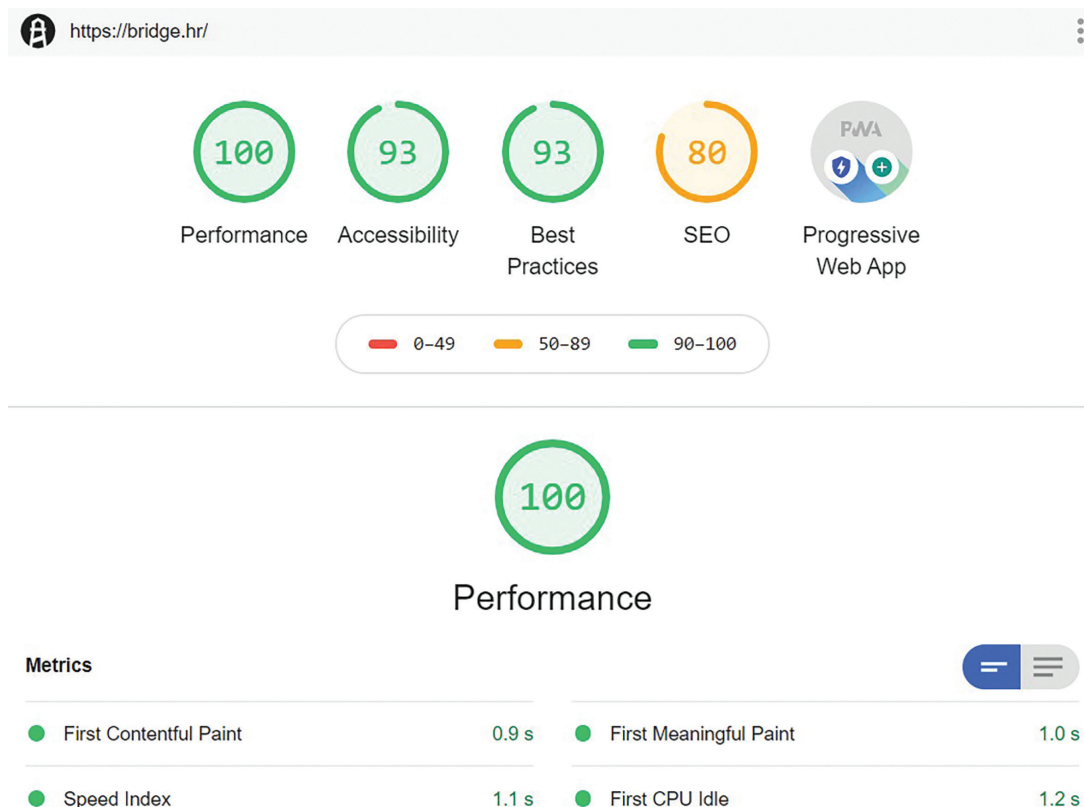
Slika 6 Notifikacija nakon nove vijesti
 Figure 6 Notification on the screen after a new news item

6. TESTIRANJE PERFORMANSI “HBS” APLIKACIJE

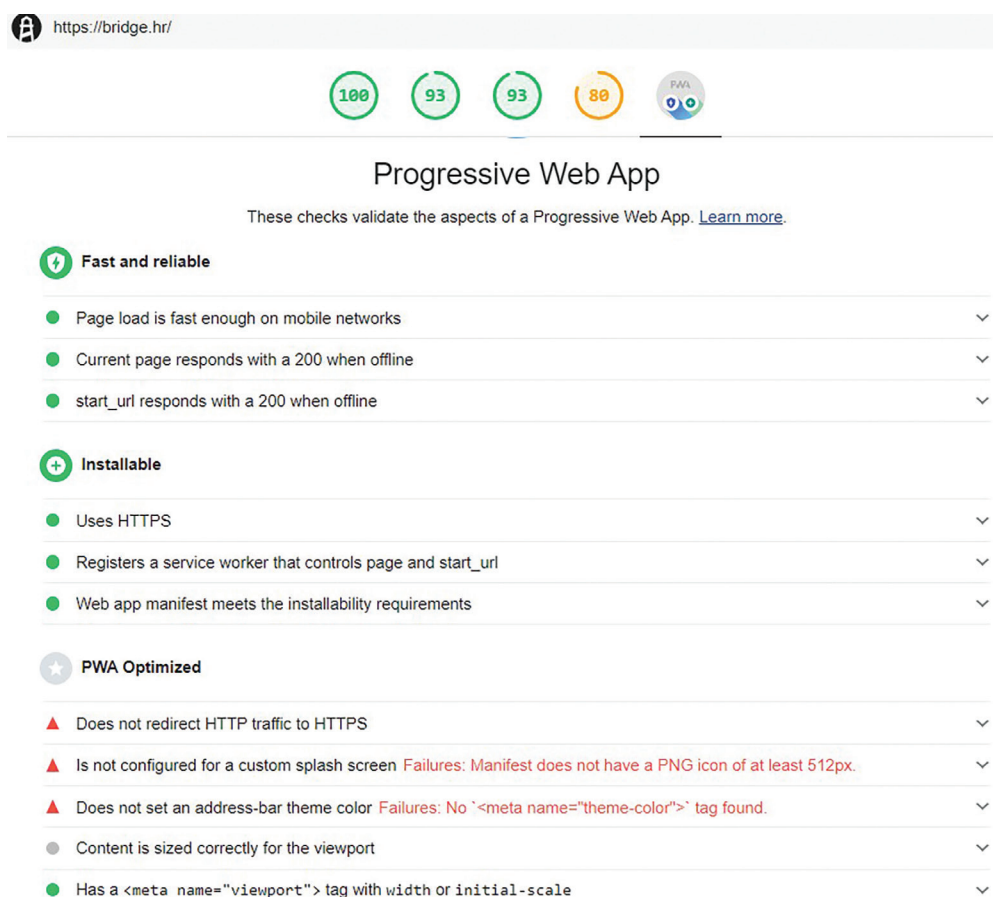
6. TESTING PERFORMANCE OF THE “HBS” APP

U developer tools dijelu Google Chrome web preglednika postoji aplikacija Lighthouse koja služi za analizu performansi web aplikacija. Uz analizu performansi, pristupačnosti, poštivanja najboljih praksi i optimizaciju za pretraživače analizira se i implementiranost funkcionalnosti progresivnih web aplikacija. U nastavku su navedeni rezultati za HBS aplikaciju.

Aplikacija ima dobre rezultate za promatrane parametre, brzo se učitava, pristupačna je i koristi većinu najboljih preporučenih praksi. Brzina učitavanja je vrlo važna, istraživanje pokazuje da prosječni korisnik odustane od čekanja ako učitavanje potraje dulje od tri sekunde, a svaki drugi korisnik očekuje da se web stranica učita ispod dvije sekunde. [8] HBS web aplikacija se učitava nešto iznad jedne sekunde što je i više nego zadovoljavajući rezultat.



Slika 7 Rezultati analize performansi HBS aplikacije
 Figure 7 Analysis results of the HBS application performances



Slika 8 Analiza PWA aspekata HBS aplikacije

Figure 8 Analysis of the PWA aspects of the HBS app

Malo su lošiji rezultati optimizacije za pretraživače što je očekivano kod JavaScript single page web aplikacija, zbog toga što crawleri pretraživača imaju problema sa sadržajem koji se izrađuje tek prilikom učitavanja na klijentskoj strani, obzirom da im prilikom inicijalnog učitavanja stranice nije u potpunosti dostupan sav sadržaj. Postoje tehnike kako se to može poboljšati u ovakvim aplikacijama i to je jedno moguće poboljšanje ove aplikacije.

Posebni odjeljak u analizi aplikacije se odnosi na PWA karakteristike i funkcionalnosti. U rezultatima možemo vidjeti da aplikacija zadovoljava većinu potrebnih uvjeta. Moguća poboljšanja su dodati automatsko preusmjeravanje HTTP prometa na HTTPS kako korisnik ne bi niti mogao pristupiti aplikaciji preko HTTP protokola, te još ostala manja vizualna poboljšanja. Ako korisnik pristupi aplikaciji preko zastarjelog web preglednika ili preko HTTP protokola, service worker neće raditi, te će uz nemogućnost instalacije aplikacije performanse biti lošije.

7. ZAKLJUČAK

7. CONCLUSION

Web aplikacije i mobilne aplikacije su svuda prisutne, te su do sada tvrtke nerijetko razvijale obje verzije kako bi privukle što više korisnika. Pojava progresivnih web aplikacija velikom brzinom mijenja stanje na tržištu, te sve više tvrtki se okreće takvom programskom rješenju koje kombinira najbolje stvari mobilnih i web aplikacija, te nema razloga da se ovakav trend ne nastavi. Razvoj progresivnih web aplikacija dodatno objedinjuje razvoj više verzija nativnih mobilnih aplikacija u odnosu na operacijski sustav na kojem će se instalirati, primjerice Android ili iOS. Ovo je sve moguće zahvaljujući mehanizmu service workera koji je implementiran u novijim inačicama svih modernih web preglednika. Uz bolje performanse koristeći cache omogućena je i instalacija aplikacije te push notifikacije koje aplikaciju čine vrlo sličnom nativnoj mobilnoj aplikaciji uz puno lakšu dostupnost i instalaciju.

Razvojnim inženjerima koji imaju znanje o web tehnologijama, programiranje progresivnih web aplikacija biti će jednostavno i nastaviti će se upravo u smjeru već naučenih programskih tehnika.

8. REFERENCE

8. REFERENCES

- [1.] Google developer, Progressive web apps, <https://developers.google.com/web/progressive-web-apps> (15. 11. 2019.)
- [2.] Kimberly de Silva, "10 Reasons Why PWAs Are On The Rise", <https://www.businessapps.com/blog/10-reasons-pwas-will-replace-native-mobile-apps-infographic/> (15. 11. 2019.)
- [3.] Austin Knight, "The Data Behind Why Apps Fail", <https://austinknight.com/writing/the-data-behind-why-apps-fail> (15. 11. 2019.)
- [4.] Chris Love, "Progressive Web Applications Features Make the Web More App-Like", <https://love2dev.com/blog/progressive-web-app-features/> (15. 11. 2019.)
- [5.] Jake Archibald, "Is service worker ready", <https://jakearchibald.github.io/isserviceworkerready/> (15. 11. 2019.)
- [6.] Lucas DiPietrantonio, "The Advantages of Progressive Web Apps (PWA) Versus Native Apps", <https://hackernoon.com/the-advantages-of-progressive-web-apps-pwa-versus-native-apps-5o2aq308m> (15. 11. 2019.)
- [7.] Workbox docs, <https://developers.google.com/web/tools/workbox> (15. 11. 2019.)
- [8.] Armando Roggio, "Are PWAs the Future of Mobile Commerce?", <https://www.practicalecommerce.com/pwas-future-mobile-commerce/> (15. 11. 2019.)

AUTORI · AUTHORS

• Ognjen Staničić

Diplomirao je 2010. kao diplomirani inženjer elektrotehnike na Fakultetu Elektrotehnike i Računarstva u Zagrebu, 2011. izabran u suradničko zvanje asistenta, 2015. u zvanje predavača na Tehničkom Veleučilištu u Zagrebu. Nositelj je kolegija Interaktivno programiranje na webu i Napredno JavaScript programiranje na studijima informatike i računarstva na TVZ-u. Primarni interesi rada i obrazovanja je razvoj modernih web aplikacija primarno pomoću programskog jezika JavaScript i njegovih radnih okvira (Angular, Node.js).

Korespondencija · Correspondence

ostanicic@tvz.hr



• Sanja Kraljević

Sanja Kraljević je viši predavač u području Tehničkih znanosti, polje Elektrotehnika. Zaposlena je na Tehničkom veleučilištu u Zagrebu od 2009.-te godine. Rođena je u Zagrebu gdje

2008.-te godine završava diplomski studij na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu te stječe titulu diplomiranog inženjera elektrotehnike. Trenutno je na poslijediplomskom doktorskom studiju na Fakultetu elektrotehnike i računarstva, u polju Računarstvo i računarske znanosti. Područja njenog znanstveno-istraživačkog i stručnog interesa su web tehnologije, sigurnost informacijskih sustava, sigurnost weba aplikacija i baze podataka.

Korespondencija · Correspondence

sanja@tvz.hr