

RAZVOJ PROGRESIVNIH VEB APLIKACIJA NA PRIMERU IONIC I POLYMER OKRUŽENJA**DEVELOPMENT OF PROGRESSIVE WEB APPLICATIONS USING IONIC AND POLYMER FRAMEWORK**Jelena Kostić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Ovaj rad predstavlja poređenje razvoja progresivnih veb aplikacija pomoću Ionic i Polymer okruženja

Ključne reči: *Progresivne veb aplikacije, Ionic, Polymer, Javascript, HTML, WebRTC, Firebase*

Abstract – *This paper deals with comparison of progressive web applications development using Ionic and Polymer frameworks.*

Keywords: *Progressive web applications, Ionic, Polymer, Javascript, HTML, WebRTC, Firebase*

1. UVOD

Zadatak rada predstavlja poređenje razvoja progresivnih veb aplikacija putem namenskih okruženja. Kao tema rada, odabrana su dva popularna okruženja za razvoj progresivnih veb aplikacija, Ionic [1] i Polymer [2]. Za demonstraciju rada su implementirane dve aplikacije iz domena chat aplikacija, napisane koristeći navedena okruženja.

Ionic i Polymer okruženja su odabrana kao okruženja popularna u oblasti razvoja progresivnih veb aplikacija, ali ne i klasičnih veb aplikacija. Prilikom implementacije, za implementaciju *back-end* dela aplikacije je korišćena *Firebase* [3] platforma za mobilne aplikacije, kao i *WebRTC* [4] tehnologija za pristup multimedijalnim periferijama uređaja.

2. TEHNOLOGIJE PROGRESIVNIH VEB APLIKACIJA

Progresivne veb aplikacije predstavljaju posebnu vrstu klasičnih veb aplikacija koje korisniku pružaju isto iskustvo u korišćenju aplikacije nezavisno od uređaja na kome je koristi. Pravljen su na temelju istih tehnologija na kojima se prave i klasične veb aplikacije, ali se od istih razlikuju u tome što korisniku pružaju osećaj i performanse korišćenja aplikacije pravljen namenski za korisnikov uređaj.

Da bi se stvorio navedeni utisak, kod dizajniranja i implementacije progresivnih veb aplikacija akcenat se stavlja na dva koncepta – *responsive* dizajn, koji omogućava prilagođavanje sadržaja stranice uređaju na kome se ona gleda, i *native* funkcionalnosti, poput brzine, funkcionisanja bez zavisnosti mreže i keširanja resursa.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

Prema definiciji postavljenoj od strane Aleksa Rasela [5], osnovni koncepti progresivnih veb aplikacija su: *responsive* dizajn, nezavisnost od povezanosti na Internet, osećaj korišćenja *native* aplikacije, ažurnost, sigurnost, mogućnost pronalazjenja, mogućnost interakcije sa korisnikom nezavisno od njegove interakcije sa aplikacijom, mogućnost instalacije i mogućnost deljenja.

Minimalni uslovi koje progresivna aplikacija treba da ispuni da bi se mogla tako i nazvati jesu da: bude dostupna putem TLS protokola, bude dostupna *offline* i sadrži *web manifest*.

2.1. Service worker i web manifest

Service workeri predstavljaju posebnu vrstu *web workera*, Javascript koda koji se izvršava u pozadini aplikacije bez blokiranja UI niti. Glavna uloga *service workera* je da presreću HTTP zahteve koji dolaze iz aplikacije za koju je registrovan. Presretanjem HTTP zahteva se omogućava keširanje resursa i podrška za *offline* funkcionisanje aplikacije.

Manifest veb aplikacije predstavlja datoteku koja sadrži podatke o aplikaciji neophodne prilikom preuzimanja i instalacije. Manifest veb aplikacije je napisan u vidu JSON datoteke i sadrži opšte informacije o aplikaciji, informacije o instalaciji, navigaciji i izgledu aplikacije, kao i putanje do resursa korišćenih u aplikaciji. Manifest se dodaje u aplikaciju putem *link* HTML elementa u kome se navodi putanja do JSON datoteke.

Manifest predstavlja jedan od kriterijuma za instalaciju progresivne veb aplikacije na uređaj. Pošto svaki pretraživač različito definiše kriterijume za instalaciju aplikacije, sadržaj manifesta koji je dovoljan za instalaciju na neki uređaj se razlikuje između njih.

2.2. Ionic okruženje

Ionic predstavlja open source okruženje za razvoj aplikacija za veb i mobilne platforme. Može se koristiti sa nekim od popularnih UI okruženja, poput Angular-a, React-a ili Vue.js-a, ili se može koristiti sa Javascriptom, bez posrednog okruženja. Ionic aplikacije se mogu postaviti na Internet kao progresivne veb aplikacije, ili se mogu objaviti na Google Play Store-u ili App Store-u kao hibridne aplikacije.

Ionic sadrži bogatu biblioteku UI komponenti koje se mogu koristiti i u mobilnim i u veb aplikacijama. Navedene komponente su *responsive* i dizajnirane poštujući principe *Material Design-a*.

Ionic Native predstavlja set Ionic biblioteki koje se mogu iskoristiti da bi se korisniku omogućile funkcije dostupne na mobilnim uređajima. Navedene biblioteke se koriste zajedno sa okruženjima za razvoj hibridnih mobilnih aplikacija. Ionic okruženje takođe sadrži i set CLI instrukcija za jednostavniji razvoj aplikacija, pokrivajući ceo životni ciklus istih.

2.2. Polymer okruženje

Polymer predstavlja open source biblioteku razvijenu od strane Google-a. Prednost Polymera jeste to što, sem što pruža gotov skup komponenti koje se mogu koristiti, takođe pruža mogućnost za definisanje i publikovanje sopstvenih komponenti na javni repozitorijum komponenti.

Akcent na Polymer komponenti jeste na jednostavnosti i brzini iscrtaavanja komponenti, čime se povećava efikasnost aplikacije. Najnovija vezija biblioteke, nazvana *LitElement*, uvodi inovaciju u odbacivanju dvosmernog vezivanja za model (*two-way binding*) i koristi samo jednosmerno vezivanje za model (*one-way binding*).

Polymer ne sadrži skup biblioteka za omogućavanje funkcionalnosti vezanih za mobilne aplikacije. Umesto toga, za omogućavanje navedenih funkcija se mogu koristiti druge Javascript biblioteke koje pružaju te mogućnosti. Polymer, poput Ionic-a, takođe sadrži set CLI instrukcija za jednostavniji razvoj aplikacija.

2.3. WebRTC

WebRTC (Web Real-time Communication) projekat predstavlja tehnologiju koja veb aplikacijama omogućava razmenu zvuka i videa koristeći *real-time* komunikacione protokole. Tehnologija je standardizovana od strane W3C-a, dostupna u vidu Javascript API-ja i podržana od većine modernih pretraživača.

Osnovu WebRTC-a predstavljaju protokoli za pristup multimedijalnim uređajima i *peer-to-peer* protokoli. Multimedijalni uređaji koriste standardizovani Javascript Media Device API i tako omogućavaju pristup multimedijalnim periferijama uređaja. *Peer-to-peer* tehnologije omogućavaju razmenu informacija između uređaja, čineći time implementaciju protokola poput VoIP mogućom.

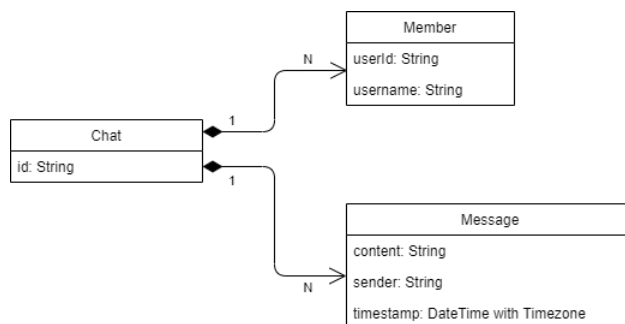
2.4. Firebase

Firebase predstavlja Google platformu za kreiranje mobilnih aplikacija. Platforma pruža niz usluga koje pokrivaju kreiranje, unapređivanje i razvoj mobilnih aplikacija. Kao neke od Firebase usluga, platform pruža mogućnost integracije analitike aplikacije, sistema za autorizaciju, sistema za čuvanje datoteka, NoSQL baze, hostinga aplikacije na cloud serveru itd. Tehnologija se razvija od strane Google-a, i moguća je integracija iste unutar mobilnih i veb platformi.

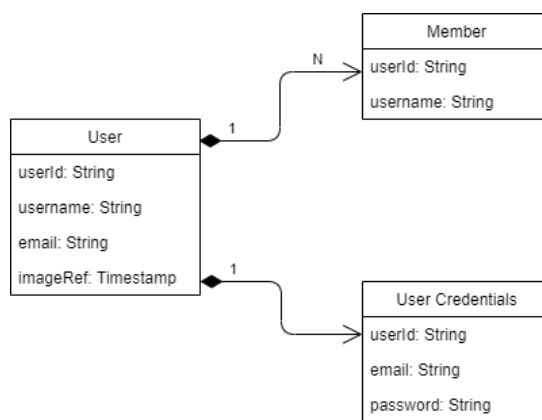
2.5. Specifikacija

Za domen aplikacije je odabrana *chat* aplikacija, pomoću koje korisnici mogu da šalju poruke drugim korisnicima, dodaju ih u kontakte i kreiraju svoje profilne fotografije. Aplikacija je kreirana u dva okruženja koji su tema rada. Aplikacije su koncipirane da izgledom podsećaju jedna na drugu, radi jednostavnijeg poređenja.

Na slikama 1 i 2 je dat model baze podataka. Firebase platforma nudi NoSQL bazu kao rešenje za perzistenciju podataka, tako da je baza sačinjena od dve kolekcije podataka – kolekcija konverzacija i kolekcija korisnika.



Slika 1. Model dokumenta iz kolekcije konverzacija



Slika 2. Model dokumenta iz kolekcije korisnika

Pristizanje poruka se dešava u realnom vremenu, bez ručnog osvežavanja interfejsa. Rad aplikacije je moguć i bez konekcije na Internet pomoću mehanizama keširanja resursa.

Promena profilne fotografije se obavlja koristeći prednju kameru uređaja. Pritiskom na odgovarajuće dugme se otvara interfejs kamere pomoću kog je moguće napraviti fotografiju. Kada korisnik odluči da je zadovoljan napravljenom fotografijom, kamera vraća nazad rezultujuću sliku koja se čuva u bazi i postavlja kao nova slika.

2.6. Implementacija

Prilikom implementacije aplikacija, naglasak je bio na mogućnosti rada aplikacije na glavnim mobilnim platformama- Android i iOS. Pre započinjanja rada su definisani grafički interfejs i glavni elementi interakcije, radi bolje preciznosti prilikom implementacije interfejsa. Oba okruženja sadrže komponente kreirane pomoću *WebComponents* standarda. Komponente su dostupne na javnim repozitorijumima putem *Node Package Manager*-a.

Kao jedan od glavnih izazova prilikom izrade rada, izdvojene su dve glavne funkcionalnosti aplikacije. Prva i osnovna funkcionalnost *chat* aplikacije jeste slanje i primanje poruka u realnom vremenu. Prilikom implementacije funkcionalnosti je korišćen *Firebase Firestore API*, koji pruža mogućnost podešavanja *observera* na određeni

set podataka i definisanja funkcionalnosti koja će se izvršiti svaki put kada se posmatrani podaci promene. Navedena osobina je iskorišćena za osvežavanje podataka prilikom primanja nove poruke. Time je uklonjena potreba za ručnom implementacijom *WebSocket* protokola koji bi omogućavao da server kontaktira korisničku aplikaciju prilikom primanja poruke.

Naredni izazov je predstavljala implementacija funkcionalnosti kamere koja korisniku pruža mogućnost da promeni svoju profilnu sliku. Prilikom implementacije funkcionalnosti je bilo potrebno kreirati poseban interfejs koji će sadržati funkcionalnosti kamere.

Za pristup kameri je iskorišćen navedeni WebRTC protokol, koji omogućava da se, koristeći Media Device API, pristupi kameri, prikažu podaci koji kontinualno dolaze sa kamere i da se ti podaci na kraju preuzmu kao profilna slika korisnika. Na listingu 1 je data implementacija inicijalizacije kamere, dok je na listingu 2 data implementacija preuzimanja fotografije i oslobađanja resursa.

```
openCamera() {
  const constraints = {
    video: true
  };

  navigator.mediaDevices.getUserMedia(constraints)
    .then(stream => {
      this.mediaStream = stream;
      var video = document.querySelector('video');
      video.setAttribute('style', 'border: 1px solid
black; position: fixed; left: 0px; top: 0px');
      document.getElementById('camera-capture')
        .setAttribute('style', 'position: fixed; left:0px;
top: 0px');

      if ("srcObject" in video) {
        video.srcObject = this.mediaStream;
      } else {
        video.src = window.URL
          .createObjectURL(this.mediaStream);
      }
      video.onloadedmetadata = function(e) {
        video.play();
      };
    })
};
```

Listing 1. Inicijalizacija kamere

```
const takePhoto = (() => {
  const canvas = document.getElementById('canvas');
  const context = canvas.getContext('2d');

  const vid = document.querySelector('video');
  canvas.width = vid.videoWidth;
  canvas.height = vid.videoHeight;
  context.drawImage(vid, 0, 0, vid.videoWidth,
vid.videoHeight);

  const imgData = canvas.toDataURL('image/png');
  const imageName = new Date().getTime();
  this.userService.updateImage(this.userId, imageName,
imgData).then(() => {
    this.userService.updateProfile(this.userId,
imageName).then(() => {});
    this.mediaStream.getTracks()[0].stop();
  });

  const video = document.querySelector('video');
  video.setAttribute('style', 'border: 1px solid
black; position: fixed; left: 0px; top: 0px; display:
none');
```

```
const button = document.getElementById('camera-
capture');
button.setAttribute('style', 'position: fixed;
right:0px; top: 0px; display: none');
}).bind(this);
```

```
document.getElementById('camera-capture')
  .addEventListener("click", takePhoto);
```

Listing 2. Prikaz preuzimanja profilne fotografije sa kamere, čuvanja fotografije i oslobađanja resursa

Nakon završetka, obe aplikacije su postavljene na javne domene putem TLS protokola. Funkcionalnost instalacije progresivne aplikacije je proverena na Samsung A50 fizičkom uređaju, dok je za testiranje iOS uređaja korišćen BrowserStack [6] online emulator.

3. ZAKLJUČAK

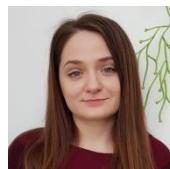
U današnje vreme je primetan porast u broju progresivnih veb aplikacija i preduzeća koja se prebacuju na ovakav vid aplikacija. Nakon prelaska, preduzeća navode primetan porast u broju korisnika, njihovom zadovoljstvu i samim time porastom profita koji aplikacija donosi. Danas se progresivne veb aplikacije smatraju budućnošću *e-commerce* modela poslovanja.

Ionic i Polymer okruženja se i danas aktivno razvijaju, i samim time, postoji mogućnost u unapređivanju aplikacije koja je korišćena za demonstraciju rada. Ionic okruženje prednjači u broju korisnika i načinu dokumentacije sistema, dok je prednost Polymera u njegovoj prilagodljivosti i mogućnosti podešavanja.

4. LITERATURA

- [1] Ionic okruženje, <https://ionicframework.com>
- [2] Polymer okruženje, <https://www.polymer-project.org/>
- [3] Firebase sistem, <https://firebase.google.com/>
- [4] WebRTC protokol, <https://webrtc.org/>
- [5] Progressive apps: Escaping Tabs Without Losing Our Soul, <https://medium.com/@slightlylate/progressive-apps-escaping-tabs-without-losing-our-soul-3b93a8561955>
- [6] Browser Stack, <https://www.browserstack.com/>

Kratka biografija:



Jelena Kostić rođena je u Rumi 1995. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primenjene računarske nauke i informatika odbranila je 2018.god. kontakt: kosticka.jelena@gmail.com