

Marija Jovanović

# **BANKOVNA JAVASCRIPT APLIKACIJA**

17.02.2024.

## SADRŽAJ

1. OPIS REŠAVANOG ZADATKA
2. OPIS KORIŠĆENIH TEHNOLOGIJA I ALATA
3. OPIS REŠENJA
4. PREDLOZI ZA DALJA USAVRŠAVANJA
5. LITERATURA

Napomena: Aplikacija je u razvoju tako da postoji mogućnost da dokumentacija i aplikacija nisu u svakom trenutku usklađene.

## 1. OPIS REŠAVANOG ZADATKA

U ovom dokumentu će biti prezentovan način izrade bankovne aplikacije koristeći javascript programski jezik.

### 1.1. Konkretni zadaci:

- Učitavanje korisnika iz baze podataka (json fajl sa podacima o korisnicima).
- Omogućiti korisniku da se uloguje u aplikaciju koristeći korisničke podatke.
- Prikaz trenutnog stanja na računu, datuma i vremena prijave na sistem.
- Prikaz svih transakcija korisnika. Takođe je potrebno za svaku transakciju prikazati da li se radi o prilivu ili odlivu novca sa računa kao datum kada je izvršena.
- Korisnik može da vrši uplatu sredstava tj. transfer novca sa svog računa na račune drugih korisnika.
- Omogućiti korisniku da izbrise svoj nalog ukoliko želi, pri čemu prethodno mora uneti svoje korisničke podatke u formu za brisanje.
- Prilikom prijave na sistem, korisniku se na ekranu prikazuje štoperica koja se svaki put obnavlja kada korisnik izvrši neku akciju. Ako korisnik ostane prijavljen na sistem i ne izvrši nijednu akciju, a vreme istekne, korisnik će automatski biti izlogovan iz aplikacije.
- Omogućiti korisniku da se odjavi iz sistema ukoliko želi.

### 1.2. Dodatni zadaci:

- Korisnici mogu učitati sliku iz lokalnog skladišta na svom računaru i prikazati na profilu u aplikaciji.
- Implementirati opciju sortiranja transakcija od najveće ka najmanje.
- Omogućiti korisniku da vidi svoje kartice u svim valutama kao i podatke na istim (valuta, količina novca, broj kartice).
- Omogućiti korisnicima uvid u trenutne vrednosti valuta koristeći API za praćenje aktuelnih kursnih listi.
- Implementirati menjačnicu u aplikaciji.
- Implementirati opciju ograničavanja potrošnje za svaku karticu.
- Implementirati opciju prelaska na tamni režim rada.

## 2. OPIS KORIŠĆENIH TEHNOLOGIJA I ALATA

U ovom poglavlju će biti dat sažet opis korišćenih tehnologija i alata na projektu.

### 2.1. Visual Studio Code [1]

Visual Studio Code je alat za uređivanje koda koji podržava mnoge jezike među kojima su JavaScript, C#, TypeScript, Node.js itd. Ima ugrađenu podršku za Git i dostupan je na Windows, macOS i Linux operativne sisteme.

Ovaj alat je izabran za korišćenje u ovom projektu jer je besplatan, brzo se pokreće i pruža bogat set funkcionalnosti.

### 2.2. Programski jezik JavaScript [2]

JavaScript je objektno-orjentisani, više-paradigmeni programski jezik koji se koristi za interakciju sa korisnicima na veb stranicama.

Izabran je zbog lakoće korišćenja i mogućnosti stvaranja interaktivnih korisničkih iskustava.

### 2.3. HTML i CSS [3]

HTML je jezik koji razumeju kompjuteri, dizajniran da opiše veb stranice. Koristi se za kreiranje sadržaja na veb stranicama, pružajući im strukturu.

CSS je jezik za stilizaciju, koristi se da opiše izgled HTML-a.

### 3. OPIS REŠENJA

#### 3.1. Opis fajlova i određenih funkcija korišćenih u projektu

U ovom poglavlju će biti opisana struktura projekta (fajlovi od kojih se projekat sastoji) kao i određene funkcije koje je bitno pomenuti zbog lakšeg razumevanja rada aplikacije.

##### 3.1.1. Fajl – user.json

Ovaj fajl predstavlja bazu podataka. Podaci se odnose na informacije o korisniku-vlasniku računa i njegovim transakcijama i zapisani su u json formatu. Primer jednog json objekta iz ovog fajla:

```
{
  "owner": "Aleksa Adamovic",
  "movements": [200, 450, -400, 3000, -650, -130, 70, 1300],
  "accNum": 0,
  "pin": 1111,
  "movementsDates": [
    "2019-11-18",
    "2019-12-23",
    "2020-01-28",
    "2020-04-01",
    "2020-05-08",
    "2020-07-26",
    "2020-07-28",
    "2020-08-01"
  ],
  "cardNum1": "0000 0000 0000 0000",
  "cardNum2": "0000 0000 0000 1111",
  "movementsRSD": [20, 30, 10000]
}
```

Slika 1-user.json

##### 3.1.2. Fajl – login.html

Ova aplikacija se sastoji od samo jedne veb stranice -login.html čiji se interfejs dinamički menja u zavisnosti od korisničkih interakcija.

Prvi UI čini login forma. Nakon uspešne prijave, login forma bude sakrivena, a korisniku se prikazuje sledeći UI - početna strana aplikacije. Naredni UI koji će se biti prikazan zavisi od daljih korisničkih interakcija i izbora opcija iz ponuđenog menija.

##### 3.1.3. Fajl – style.css

Ovaj fajl se odnosi na dizajn aplikacije, definisanje boja, rasporeda elemenata i ostalih stilskih karakteristika kako bi se postigao željeni vizuelni efekat aplikacije.

##### 3.1.4. Fajl – login.js

U ovom fajlu se poziva asinhrona funkcija za učitavanje podataka iz baze (iz json fajla).

```
async function readData(path)
{
  const response = await fetch(path);
  const data = await response.json();
  return data;
}
```

Slika 2-async function

- Ova funkcija koristi *fetch()* metodu kako bi se poslao zahtev ka određenoj putanji (u ovom projektu je to users.json) za asinhrono čitanje podataka.

- Obzirom da je funkcija označena sa *async*, znači da u sebi može sadržati *await*. U prvoj liniji koda u telu funkcije, *await* omogućava čekanje da se zahtev završi kako bi se dobio odgovor u obliku *response* objekta. Dok u narednoj liniji koda, *await* omogućava čekanje na završetak operacije *json()*.
- Metoda *json()* omogućava parsiranje podataka iz json formata u javascript objekat.

Asinhronne funkcije su pogodne jer se izvršavaju tako da ne blokiraju glavnu nit izvršavanja (main) čime je omogućeno drugim operacijama da se izvršavaju dok se čeka izvršenje asinhronih operacija.

### 3.1.5. Fajl – dashboard.js

U ovom fajlu su implementirane funkcije koje omogućavaju razne funkcionalnosti UI-a koji se korisniku prikazuje na početnoj strani, nakon logovanja u aplikaciju.

3.1.5.1. Funkcija *uni()* - Jedna od funkcija koju je bitno posebno pomenuti. Ona se poziva u svakom narednom opisanom fajlu. Prilikom poziva ove funkcije veb stanica se „deli” na tri diva nejednake veličine. Ova funkcija omogućava i prikaz štoperice kao i datuma i vremena logovanja u aplikaciju.

3.1.5.2. Funkcija *showHomePage()* – vrši poziv *uni()* funkcije, a nakon toga i poziv funkcija prikazanih na slici:

```
const uni = function() ...
}

const showHomePage = function()
{
    var bodyElement = document.body;
    var allElements = bodyElement.children;
    for (var i = allElements.length - 1; i >= 0; i--)
    {
        var currentElement = allElements[i];
        if (currentElement.tagName !== 'SCRIPT' && currentElement.tagName !== 'MAIN')
        {
            currentElement.parentNode.removeChild(currentElement);
        }
    }

    uni();
    //leftDiv
    showHomePageLeftDiv();
    //middleDiv
    showHomePageMiddleDiv();
    //rightDiv
    showHomePageRightDiv();
}
```

Slika 3- *showHomePage()* funkcija

3.1.5.3. Funkcija *showHomePageLeftDiv()* – je „zadužena” za to šta će biti prikazano u jednom od tri div-a na koje je stranica podeljena. Konkretno, omogućava prikaz menija korisniku u okviru kog on bira razne opcije kako bi se kretao i vršio interakcije kroz aplikaciju. Ova funkcija se takođe poziva u svakom narednom opisanom fajlu kako bi meni stalno bio prikazan korisniku radi lakšeg snalaženja u aplikaciji.

### 3.1.6. Fajl – converter.json

U ovom fajlu su implementirane funkcije koje omogućavaju prikaz i funkcionalnosti UI-a koji se korisniku prikazuje kada klikne na opciju „Converter” iz menija.

### 3.1.7. Fajl – cards.json

U ovom fajlu su implementirane funkcije koje omogućavaju prikaz i funkcionalnosti UI-a koji se korisniku prikazuje kada klikne na opciju „Cards” iz menija.

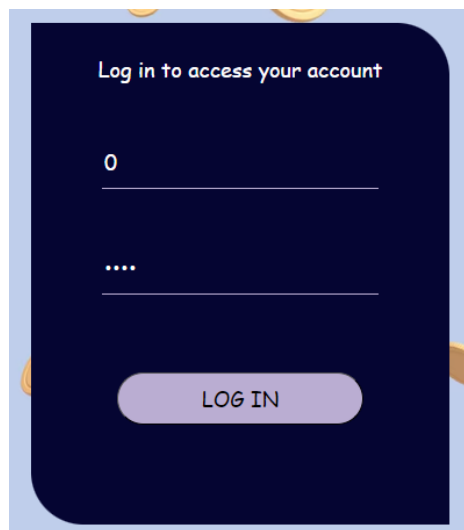
### 3.1.8. Fajl – settings.json

U ovom fajlu su implementirane funkcije koje omogućavaju prikaz i funkcionalnosti UI-a koji se korisniku prikazuje kada klikne na opciju „Settings” iz menija.

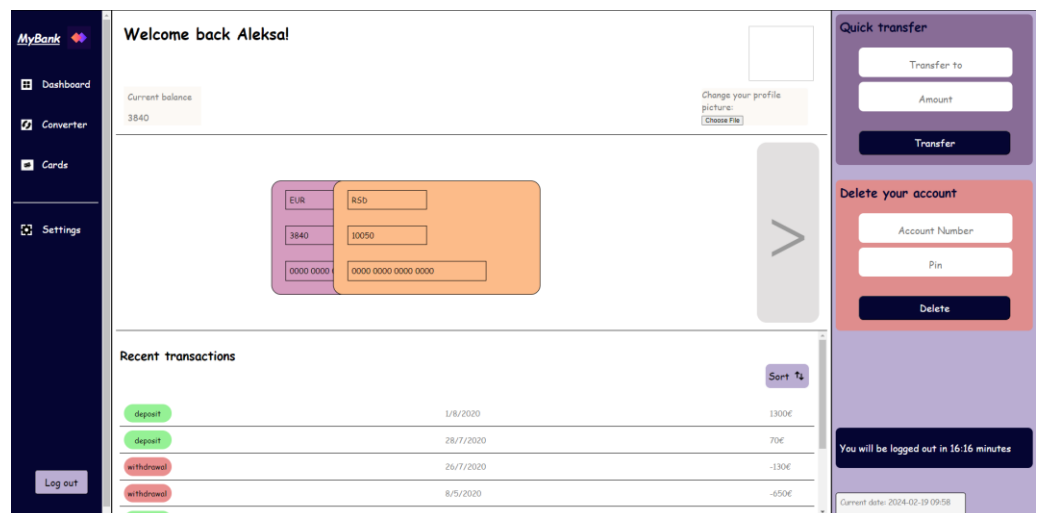
## 3.2. Primeri upotrebe aplikacije

U ovom delu dokumenta biće prikazane funkcionalnosti pri korišćenju aplikacije.

1. Popunjavanjem forme i klikom na dugme „LOG IN” korisnik se prijavljuje na sistem.

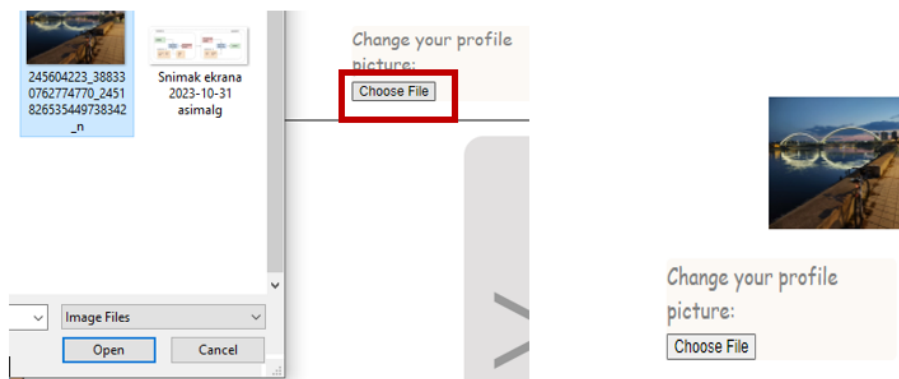


Slika 4 - Login forma



Slika 5-Početna strana aplikacije

2. Klikom na dugme „Choose File” korisnik bira profilnu sliku.



Slika 6 - Izbor slike iz lokalnog skladišta

3. Klikom na dugme sa znakom „>“, korisnik bira koja će mu kartica biti prikazana prva.



Slika 7 - Izbor kartice

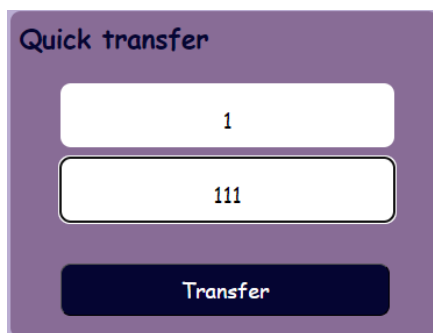
4. Klikom na dugme „Sort“, vrši se sortiranje transakcija – po datumu ili od najveće ka najmanje.

Sort ↑↓		Sort ↑↓
3000€	1/8/2020	1300€
1300€	28/7/2020	70€
450€	26/7/2020	-130€
200€	8/5/2020	-650€

Slika 8 - Simulacija levog klika na dugme Sort

5. Popunjavanjem forme i klikom na dugme „Transfer“ vrši se prenos sredstava sa računa tekućeg korisnika na račun drugog korisnika.



A purple rectangular form titled "Quick transfer". It contains two white input fields. The first field has the number "1" and the second field has the number "111". Below the fields is a dark blue button with the text "Transfer" in white.

Slika 9 - Transfer novca

withdrawal

19/2/2024

-111€

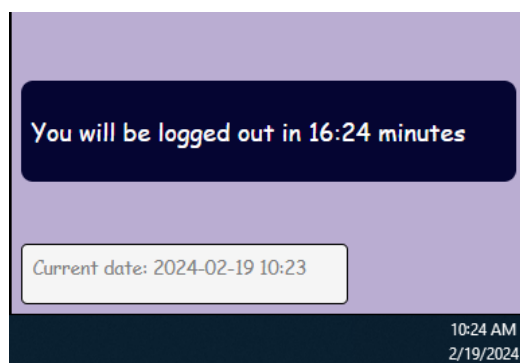
Slika 10 - Izvršena transakcija

- Unosom ispravnih podataka u formu za brisanje, vrši se brisanje korisničkog računa („Fake delete” – sakrivanje UI-a).

A red rectangular form titled "Delete your account". It contains two white input fields. The first field has the number "0" and the second field has the number "1111". Below the fields is a dark blue button with the text "Delete" in white.

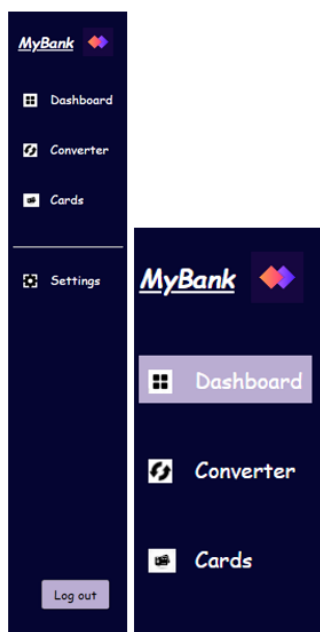
Slika 11 - Simulacija brisanja naloga

- Prilikom prijave na sistem, korisniku se prikazuje datum i vreme prijave kao i štoperica koja pokazuje još koliko vremena će korisnik ostati prijavljen.



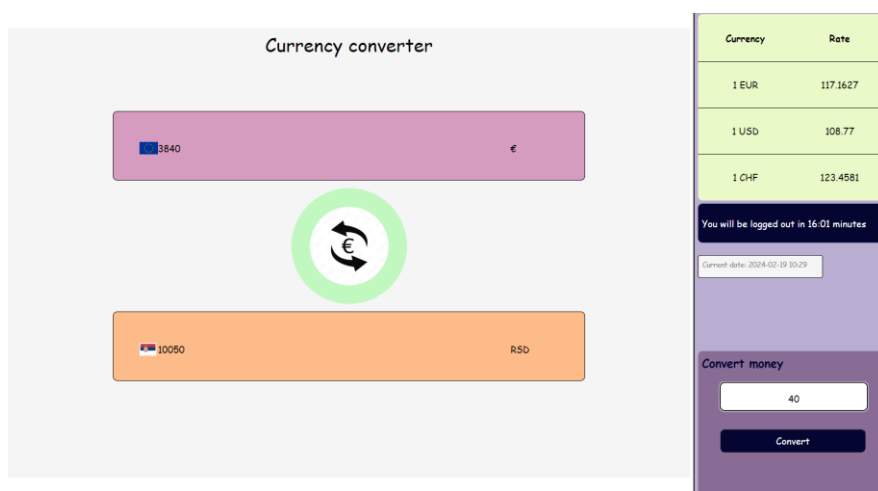
Slika 12 - Štoperica, datum i vreme logovanja

8. Sa leve strane aplikacije se nalazi meni pomoću kojeg se korisnik kreće kroz aplikaciju.

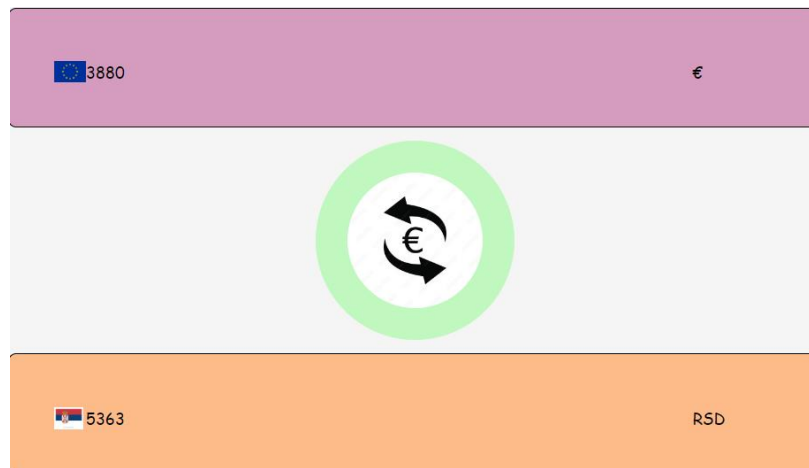


Slika 13 – Meni

9. Na „Convrtter” korisnik može sredstva u valuti RSD konvertovati u EUR. Takođe ima uvid u trenutne vrednosti valuta.

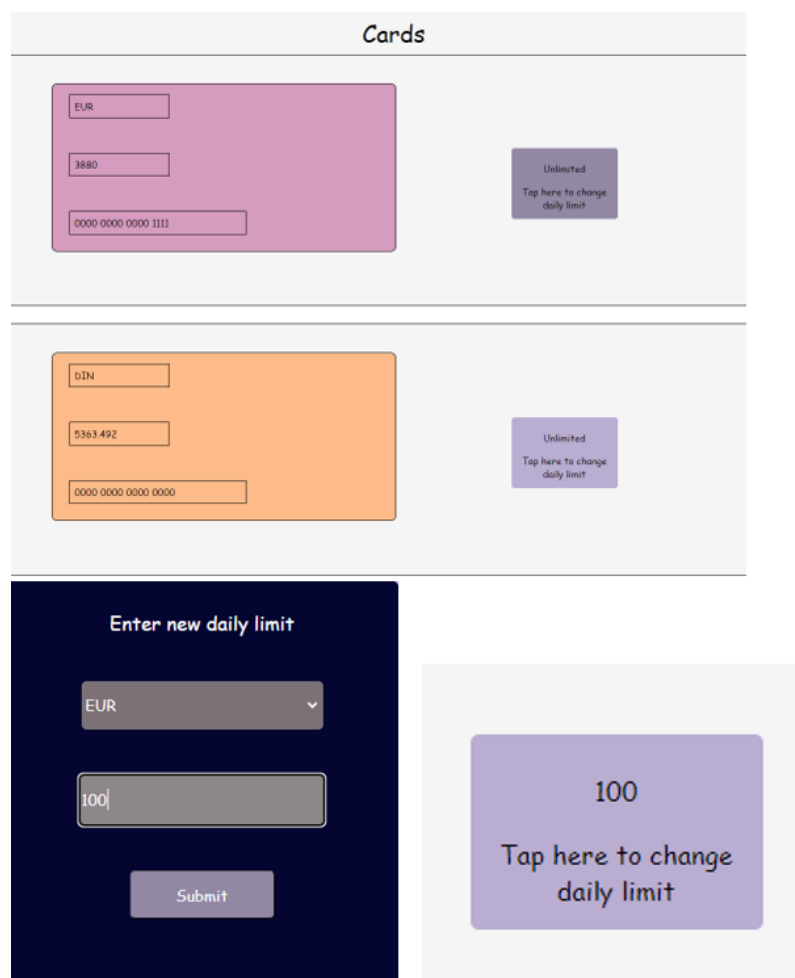


Slika 14 - Menjačnica



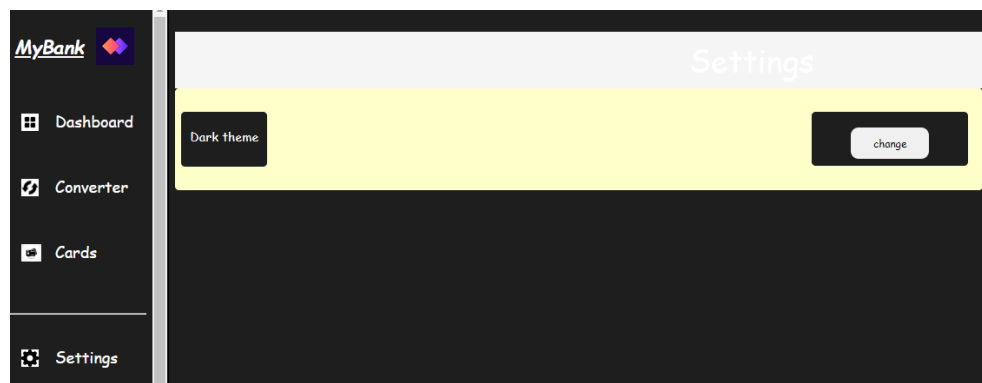
Slika 15 - Izvršen convert novca

10. Klikom na ponuđenu opciju sa slike, sa desne strane se korisniku otvara mogućnost da ograniči količinu novca koju može potošiti prilikom jednog transfer-a.



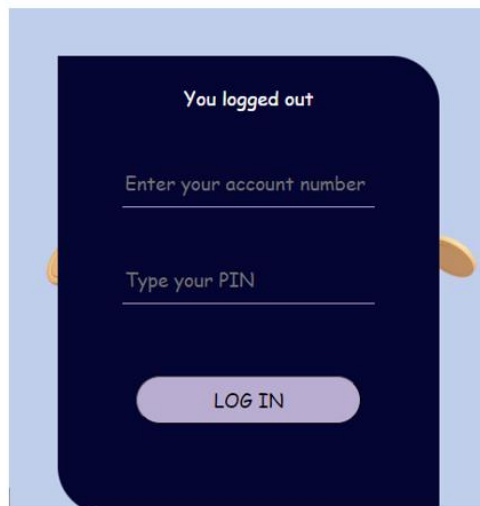
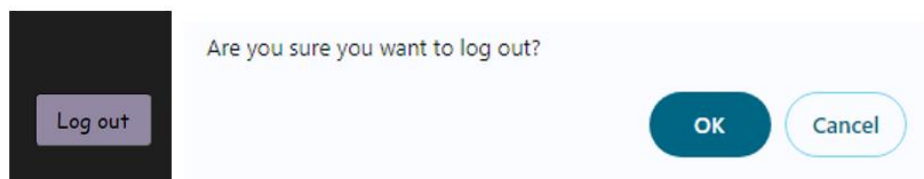
Slika 16 - Ograničivanje potrošnje sa kartica

11. Klikom na dugme „change” , aplikacija dobiva tamnu temu.



Slika 17 - Tamni režim rada

12. Klikom na dugme „Log out” korisnik se odjavljuje iz sistema.



Slika 18 - Simulacija klika na Log out

#### 4. PREDLOZI ZA DALJA USAVRŠAVANJA

Ova aplikacija se može unaprediti usvajanjem sledećih predloga:

- Prelazak sa JSON fajla kao bazu podataka na relacionu bazu podataka čime bi se omogućilo i poboljšalo upravljanje sa velikim količinama podataka. Podaci bi bili lakše organizovani i međusobno povezani.
- Implementacija responsivnog dizajna kako bi se obezbedila prilagodljivost različitim veličinama ekrana i uređajima.
- Dodatna funkcionalnost poput otvaranja novih računa i izbor naziva istog (npr. „Račun za štednju“, „Putovanje“, „ Kuća“, „Auto“ itd.), mogućnost promene pina...
- Ažuriranje podataka u bazi nakon što korisnik izvrši određene aktivnosti (poput transfera novca, konvertovanja i slično).
- Mogućnost registrovanja novih korisnika.
- Obaveštenje na mail korisnika ukoliko dođe do neke transakcije.
- Prikaz detaljnijeg izveštaja o transakcijama (informacije o primaocu, izvršitelju transakcije, vremenu izvršenja...).
- Mogućnost uzimanja kredita.
- Poseban prikaz transakcija za svaki račun (karticu).
- Vizuelne izmene kao što su: prikaz logo-a banke na karticama, da štoperica kao i prikaz datuma i vremena logovanja bude uvek prikazan na istoj poziciji na ekranu prilikom promene UI-a.

## 5. LITERATURA

[1] [Visual Studio Code](#)

[2] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

[3] <https://www.futurelearn.com/info/blog/what-are-html-css-basics-of-coding>