

DOKUMENTACIJA

<https://techbabette.github.io/library-site/>

Web Programiranje 2

Ilija Krstić 155/21

Smer: Internet tehnologije

Modul: Web programiranje

Predmet: Web programiranje 2

1 Uvod	4
1.1 Korišćeni programski jezici	4
1.2 Opis funkcionalnosti	4
Generalna funkcionalnost.....	4
Stranica index.html	4
Stranica knjige.html	4
Stranica knjiga.html	5
Stranica favoriti.html	5
Stranica autor.html	5
1.3 Template	5
2. Organizacija.....	6
2.1 Organizaciona šema.....	6
2.2 Mapa sajta	6
2.3 Slike stranica i opis funkcionalnosti	8
2.3.1 Index.html	8
2.3.2 Knjige.html	10
2.3.3 Knjiga.html	13
2.3.4 Favoriti.html.....	15
2.3.5 Univerzalna ponašanja.....	16
3. Kodovi	17
3.1 HTML.....	17
3.1.1 index.html	17
3.1.2 knjige.html	20
3.1.3 knjiga.html	23
3.1.4 Favoriti.html.....	27
3.2 CSS.....	29
3.2.1 style.css	29
3.3 Javascript	42
3.3.1 main.js.....	42
3.3.2 Strani JQuery plugin jquery.zoom.js modifikovan da dodeljuje klase umesto style atributa gde moguće	68
3.4 JSON	74
3.4.1 books.json	74

3.4.2 navbar.json80

3.4.3 footer.json82

1 Uvod

1.1 Korišćeni programski jezici

Jezici korišćeni pri izradi sajta su HTML, CSS, Javascript, JQuery biblioteka Javascripta kao i Bootstrap

1.2 Opis funkcionalnosti

Generalna funkcionalnost

- Dinamičko ispisivanje navigacionog menija i footera.
- Računanje aktivne stranice kao deo navigacionog menija.
- Omogućeno dodavanje knjiga u favorite.
- Omogućeno uklanjanje knjiga iz favorita.
- Omogućen pregled favorita klikom na link iz navigacionog menija.
- Na desktop uređajima JQuery bibliotekom omogućeno automatsko vraćanje na vrh stranice dugmetom koje se pojavljaju nakon scrollovanja.
- Deskripcije knjiga su limitirane na trideset karaktera tako da ne seku reči, dodajući “...” ako je deskripcija skraćena.
- Hvatanje grešaka pri komunikaciji sa serverom (hvatanju JSONa) prikazom modala koji obaveštava korisnika o tome.

Stranica index.html

- Dinamičko ispisivanje četiri najnovije knjige u biblioteci sortirane po godini izdanja.
- Dinamičko računanje broja godina prošlih od otvaranje biblioteke kao i broja naslova u biblioteci (zajedno sa dinamičkim ispisom).
- Dinamičko računanje broja izdatih knjiga na osnovu podataka iz knjiga
- Dinamičko ispisivanje najpopularnijih knjiga na osnovu broja puta koje su pozajmljene.
- Dinamičko kretanje “levo-desno” po najpopularnijim knjiga.

Stranica knjige.html

- Dinamičko ispisivanje prvih četiri knjiga u slučaju da nije prosleđen/sačuvan filter.
- Dinamičko filtriranje knjiga na osnovu kategorije, autora, i godine izdanja.
- Omogućeno sortiranje knjiga na osnovu popularnosti, naziva i godine izdanja.
- Dinamičko ispisivanje dugmića za stranice (straničenje) na osnovu knjiga koje prolaze postavljene filtere.
- Dinamičko ispisivanje knjiga na osnovu trenutne stranice.
- Prikazivanje poruke korisniku ako ne postoje knjige koje odgovaraju filterima.
- Čuvanje podešavanja filtera i sortiranja u local storagu.
- Uklanjanje svih podešenih filtera klikom na prikladan taster.

Stranica knjiga.html

- Dinamičko ispisivanje podataka o knjizi prosleđenoj kroz URL (Klikom na knjigu na bilo kojoj od ranijih stranica).
- U slučaju da knjiga prosleđena kroz URL ne postoji, vraća korisnika na knjige.html.
- Klikom na ime autora knjige ili na godinu izdavanja se vrši pretraga na stranici knjige.html sa prikladnim filterima.
- Dinamičko prikazivanje (modal) forme za rezervaciju klikom na prikladno dugme kroz JQuery.
- Obrada formi elemenata kada se klikne van njih.
- Obrada svih elemenata forme kada se pritisne odgovarajuće dugme i prikaz podataka o uspelosti predaje forme (Tri tekstualna polja, jedna radio selekcija, jedan datum i jedan broj).
- Dinamičko limitiranje zahteva forme, ako korisnik odabere lično preuzimanje polje za adresu postane disabled i ignoriše se.
- Primenom JQuery plugina omogućeno zumiranje prelaskom miša preko slike knjige.
- Omogućen prikaz knjiga koje su u JSON fajlu sačuvane kao srodne ako postoje.

Stranica favoriti.html

- Omogućen prikaz knjiga ranije sačuvanih kao favoriti
- Omogućen prikaz poruke korisnicima u slučaju da korisnik nema sačuvane knjige
- Pri otklanjanju knjige iz favorita na ovoj stranici, njen sadržaj se promeni tako da se uklonjena knjiga više ne prikazuje među favoritima

Stranica autor.html

Stranica o autoru je statički urađena sem delova za učitavanje kategorija i ikona footera navedenih u opisu index.html stranice i sadrži samo osnovne podatke o autoru.

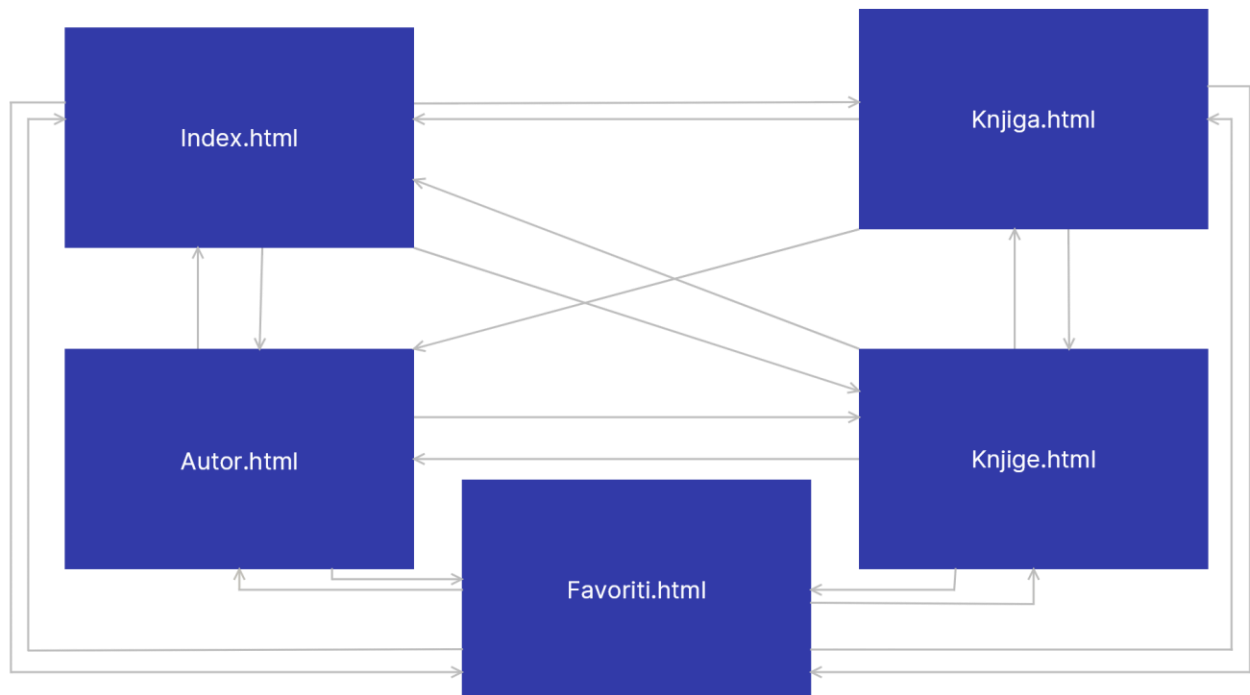
1.3 Template

Za izradu sajta je za osnovu korišćen sajt napravljen za potrebe predmeta Web Programiranje 1

<https://github.com/techbabette/library-site> (Main branch, trenutno je aktivan branch za Web Programiranje 2)

2. Organizacija

2.1 Organizaciona šema



2.2 Mapa sajta

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://techbabette.github.io/library-site/</loc>
    <lastmod>04-03-2023</lastmod>
    <changefreq>daily</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>https://techbabette.github.io/library-site/pages/knjige.html</loc>
    <lastmod>04-03-2023</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.95</priority>
  </url>
  <url>
    <loc>https://techbabette.github.io/library-site/pages/favoriti.html</loc>
    <lastmod>04-03-2023</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.90</priority>
  </url>
  <url>
```

```
<loc>https://techbabette.github.io/library-site/pages/knjiga.html</loc>  
<lastmod>04-03-2023</lastmod>  
<changefreq>monthly</changefreq>  
<priority>0.85</priority>  
</url>  
  <url>  
    <loc>https://techbabette.github.io/library-site/pages/autor.html</loc>  
    <lastmod>04-03-2023</lastmod>  
    <changefreq>yearly</changefreq>  
    <priority>0.80</priority>  
  </url>  
</urlset>
```

2.3 Slike stranica i opis funkcionalnosti

2.3.1 Index.html



- Landing deo stranice, prvo što se vidi kada se uđe na sajt.
- U navigacionom meniju je sav sadržaj dinamički ispisano



- Ispod se nalaze četiri knjige sortirane tako da najnovije budu tu.
- Klikom na knjigu se otvara knjiga.html stranica o kliknutoj knjizi.
- Klikom na prazno srce se srce popuni i knjiga se dodaje u favorite, obrnuto ako je već puno.

<h3>Kratka istorija</h3> <p>Odluka da se, za potrebe grada, osnuju opštinska biblioteka i muzej doneta je 24. oktobra 1930. godine. Na mesto bibliotekara je postavljena dr Marija Ilić Agapova, koja je dve godine kasnije, 30. septembra 1932. godine, postavljena za bibliotekara-upravnika.</p>	<p>75 Članova</p>	<p>93 Godine postojanja</p>
	<p>11 Naslova</p>	<p>68 Izdatih knjiga</p>

- Ispod je kratka istorija biblioteke, vrednosti se dižu korakom od jedan do tačnih u zadatom vremenu (trenutno 2.5 sekunde)
- Godine postojanja se dinamički računa u odnosu na godinu navedenu u istoriji.
- Broj naslova se dinamički računa u odnosu na broj knjiga dostupnih na sajtu.
- Broj izdatih knjiga se dinamički računa u odnosu na to koliko puta su knjige sa sajta pozajmljene (na osnovu čega se računa popularnost), povremeno dodatno inkrementira za jedan radi efekta aktivnosti.

Najpopularnije knjige

Intelektom ispred svih

Potreba da ostanete u formi na...

Popularna psihologija

Henrik Fekševs

Sapijens

Pre stotinu hiljada godina na...

Istorijska dela

Juval Noa Harari

Umeće ratovanja

Sun Tzuova knjiga Umeće...

Istorijska dela

Sun Tzu

Homo Deus

Šta će sada biti prioritetni...

Popularna nauka

Juval Noa Harari

©2023 Gradska biblioteka

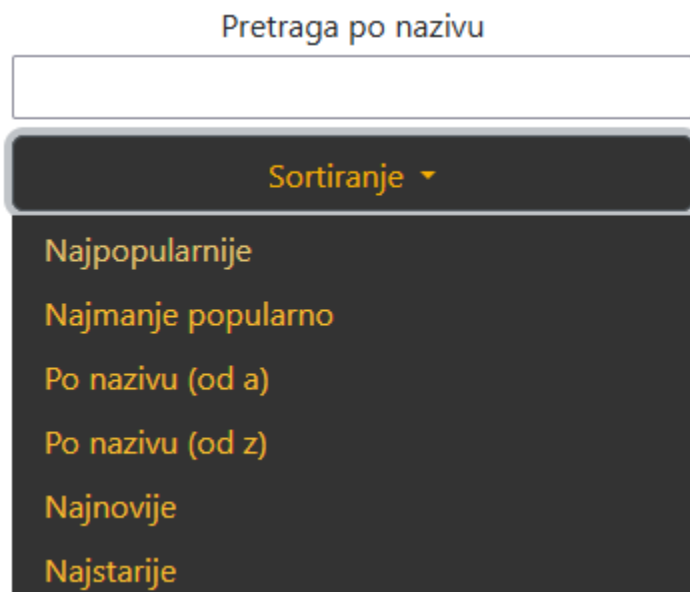
f

- Linkovi u futeru su dinamički generisani, i trenutna godina se stavlja kao godina copyrighta
- Footer linkuje ka sajtovima facebook i twitter kao i ka dokumentaciji i sitemapu.

2.3.2 Knjige.html



- Ulaskom bez filtriranja se prikazuju sve knjige, četiri po četiri.
- Ulaskom sa url filterom se prikazuju samo knjige koje odgovaraju filteru, isto četiri po četiri, ako ih ima, i pregazuje već postavljane filtere ako ih ima
- Klikom na taster numerisane tastere se učitavaju druge “stanice” koje sadrže makar jednu knjigu bez osvežavanja stranice sajta
- Ukoliko ima manje knjiga nego što se prikazuje na dve stranice, paginacija se ne prikazuje jer nije potrebna



- Po defaultu je odabrana opcija za prikaz najpopularnijih knjiga prvo, ovo se računa na osnovu toga koliko puta su knjige pozajmljene
- Tip sortiranja se čuva u local storagu pri promeni i učitava korisniku ako je neki sačuvan
- Promenom tipa sortiranja se ponovo inicijalizuje generisanje knjiga

Kategorije ▾

Popularna nauka (4)

☐

Istorijska dela (3)

☐

Popularna psihologija (2)

☐

Jezici (2)

☐

- Kategorije po kojima se može filtrirati se dinamički učitavaju na osnovu knjiga i njihovih kategorija, isto važi za filtriranje po autorima i godinama izdanja pošto ih sve generiše jedna funkcija
- Kada se odabere kategorija/više kategorija (ponovo, isto važi i za ostale filtere), u ostalim filterima se prikazuju samo opcije koje odgovaraju već postavljenim filterima
- Na primer, ako odaberemo popularnu nauku i istorijska dela, u autorima će se prikazivati samo oni autori koji su pisali knjige u tim kategorijama, i u godinama samo godine kada su one pisane
- Broj pored filtera se računa na osnovu broja knjiga koje prolaze taj filter i ostale filtere podešene u tom trenutku
- Filteri se čuvaju u local storagu i učitavaju korisniku kakve su ih ostavili ako jesu
- Promenom filtera se ponovo inicijalizuje generisanje knjiga

Ukloni filtere

- Taster ukloni filtere čisti sve filtere iz lokalnog storaga, regeneriše filtere i inicijalizuje generisanje knjiga

Pretraga po nazivu

- Unosom teksta u polje za pretragu po nazivu se knjige filtriraju na osnovu toga da li sadrži zadati tekst, bio u formi malog ili velikog slova, ili ne.
- Pretraga po nazivu takođe utiče na ostale filtere koji mogu da se odaberu, jer se prikazuju samo one opcije koje odgovaraju tekstualnoj pretrazi.
- Promenom filtera se ponovo inicijalizuje generisanje knjiga.



- Taster aktivne “stranice” u paginaciji je obojen svetlijom bojom radi lakšeg orijentisanja korisnika

2.3.3 Knjiga.html



- Prikazuju se specifični detalji o knjizi prosleđenoj kroz URL, uključujući opis, autora, godinu izdavanja, broj dostupnih kopija i koliko puta je ukupno pozajmljena.
- Prelaskom miša preko korica se ona zumira odnosno tome gde je miš.
- Klikom na srce se dodaje/uklanja iz favorita
- Klikom na ime autora ili godinu izdavanja se vrši pretraga na knjige.html sa tim filterom.
- Klikom na dugme “Rezerviši” se otvara forma za rezervisanje (Ostatak ekrana se zatamni)

The screenshot shows a reservation form with a dark background. It contains several input fields and buttons. The fields are labeled: 'Ime*' (with 'Luka' entered), 'Prezime*' (with 'Lukić' entered), 'Adresa (ulica i broj)*' (with 'Kralja Aleksandra 13' entered), 'Preuzimanje*' (with radio buttons for 'Dostava' and 'Lično'), 'Rezervisati od*' (with a date input 'dd / mm / yyyy' and a calendar icon), and 'Broj dana rezervacije*' (with a numeric input '20' and a dropdown arrow). At the bottom, there are two buttons: a yellow 'Rezerviši' button and a red 'Odustani' button.

- Na izlasku korisnika iz svakog polja se vrši provera o validnosti unetih podataka u polju.
- Tačno uneti podatci dobijaju zeleno okvirje.
- Pogrešno uneti podatci dobijaju crveno okvirje i ispisuju grešku neposredno ispod polja u kojem se dogodila.
- Unosom dobrih podataka u polje koje je ranije bilo pogrešno uneto se uklanja klasa za grešku i greška se krije, a dodaje se klasa za uspeh i obratno.

- Odabirom opcije “Lično” u polju “Preuzimanje” se polje adresa prazni, gubi svojstvo required, dobija zeleni okvir, i postaje disabled (vraća se u početno stanje ako se ponovo odabere “Dostava”).
- Klikom na dugme “Rezerviši” unutar forme se elementi forme proveravaju i ako nema grešaka ispisuje se poruka o uspehu rezervacije.
- Polje “Rezervisati od” očekuje datum počeci od sutrašnjeg.
- Polje “Broj dana rezervacije” očekuje broj između jedan i četrdeset.
- Rezervisanje je zamišljeno tako da se sprovodi ne vezano za računar/sesiju/veb pretraživač već na osnovu preuzimanja tako da se podatak o rezervisanim knjigama ne čuva na nivou sajta

2.3.4 Favoriti.html

Favoriti



- Stranica za favorite prikazuje knjige koje su ranije sačuvane kao favoriti
- Stranica ima opciju za paginaciju ako je potrebna, to jest ako postoji više sačuvanih knjiga nego što se prikazuje po stranici
- Klikom na ispunjeno srce se knjiga uklanja iz favorita i generisanje knjiga se ponovo pokreće (samim tim, ako je potrebno, i paginacije)

2.3.5 Univerzalna ponašanja

Dogodila se greška pri komunikaciji sa bazom

- Ukoliko na bilo kojoj od stranica dođe do greške pri hvatanju JSON fajlova, korisnicima se ispisuje podatak o tome i vidu modala



- Taster služi za vraćanje na vrh stranice i prikazuje se ako ima potrebe, to jest ako je korisnik skrolovao na dole određenu dužinu

3. Kodovi

3.1 HTML

3.1.1 index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <meta name="author" content="Ilija Krstić"/>
  <meta name="keywords" content="Gradska Biblioteka, Biblioteka, Biblioteka
  Beograd, Knjige, Naslovi, BGB"/>
  <meta name="description" content="Gradska Biblioteka Beograda sa preko sto
  izdatih knjiga"/>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
  beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
  0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfhXaWutUpFmBp4vmVor"
  crossorigin="anonymous"/>
  <link rel="stylesheet" type="text/css" href="css/style.css"/>
  <link rel="icon" type="image/x-icon" href="imgs/favicon.ico"/>
  <title>Gradska Biblioteka Beograda</title>
</head>
<body>
  <header class="container-fluid">
    <div class="row">
      <nav class="navbar navbar-expand-lg navbar-dark mk-background">
        <div class="container-fluid" id="actual-navbar">

        </div>
      </nav>
    </div>
  </header>
  <div id="errorModal" class="mk-modal">
    <div class="mk-modal-content">
      <p id="error-content">Generička greška</p>
    </div>
  </div>
  <!--First page-->
  <div id="landing-page" class="mk-page page container-fluid ">
    <div class="row mk-clear">
      <div class="col-12">
        <div class="mk-introduction d-flex flex-column justify-content-
        center">
          <h2>Dobrodošli u gradsku biblioteku</h2>
```



```

        <h3>Naslova</h3>
    </div>
    <div class="col-12 col-md-6 info-block mk-flex-column-center">
        <h2 id="leNum">0</h2>
        <h3>Izdatih knjiga</h3>
    </div>
</div>
</div>
</div>
</div>
<!--Fourth page-->
<div class="mk-page page container-fluid">
    <div class="row w-100">
        <div class="col-12 container-fluid w-100">
            <div class="row mk-sorted-book-holder mk-clear">
                <div class="col-1 mk-flex-column-center"><a href="#"
class="moveButton" id="moveLeftButton">&lArr;</a></div>
                <div class="container-fluid col-12 col-md-10">
                    <div class="row">
                        <h2 id="mk-book-category">Najpopularnije knjige</h2>
                        <hr/>
                    </div>
                    <div class="row" id="pop">
                    </div>
                </div>
                <div class="col-1 mk-flex-column-center"><a href="#"
class="moveButton" id="moveRightButton">&rArr;</a></div>
            </div>
        </div>
    </div>
</div>
<!--Footer-->
<footer class="container-fluid mk-background">
    <div class="row w-100" >
        <div class="mk-logo col-6 flex" id="footer-title">
        </div>
        <div class="col-6 flex flex-md-row">
            <ul class="row align-items-center list-unstyled p-0">
                <li class="col-md-3 col-6 icon-holder">
                </li>
                <li class="col-md-3 col-6 icon-holder">
                </li>
                <li class="col-md-3 col-6 icon-holder">
                </li>
                <li class="col-md-3 col-6 icon-holder">

```

```

        </li>
    </ul>
</div>
</div>
</footer>
    <a href="#" class="invisible" id="goBackUp">&uArr;</a>
<script src="js/jquery-3.6.1.js"></script>
<script src="js/main.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
pprn3073KE6t16bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
<script src="https://code.iconify.design/2/2.2.1/iconify.min.js"></script>
</body>
</html>

```

3.1.2 knjige.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="author" content="Ilija Krstić"/>
    <meta name="keywords" content="Gradska Biblioteka, Naslovi, Knjige, Sve
knjige"/>
    <meta name="description" content="Knjige dostupne u Gradskoj Biblioteci"/>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfhXaWutUpFmBp4vmVor"
crossorigin="anonymous"/>
    <link rel="stylesheet" type="text/css" href="../css/style.css"/>
    <link rel="icon" type="image/x-icon" href="../imgs/favicon.ico">
    <title>Knjige</title>
</head>
<body>
    <header class="container-fluid">
        <div class="row">
            <nav class="navbar navbar-expand-lg navbar-dark mk-background">
                <div class="container-fluid" id="actual-navbar">

                    </div>
                </nav>
            </div>

```

```

</header>
<div id="errorModal" class="mk-modal">
  <div class="mk-modal-content">
    <p id="error-content">Generička greška</p>
  </div>
</div>
<div class="mk-solo-page page container-fluid">
  <div class="row spec-row d-flex flex-row">
    <div id="filters">
      <div class="d-flex flex-column text-center">
        <label for="textSearch">Pretraga po nazivu</label>
        <input type="text" name="textSearch" class="form-text"
id="textSearch">
      </div>
      <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-expanded="false">
          Sortiranje
        </button>
        <ul class="dropdown-menu" id="sortHolder" aria-
labelledby="dropdownMenuButton1">
        </ul>
      </div>
      <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-expanded="false">
          Kategorije
        </button>
        <ul class="dropdown-menu" id="categoryHolder" aria-
labelledby="dropdownMenuButton1">
        </ul>
      </div>
      <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-expanded="false">
          Autori
        </button>
        <ul class="dropdown-menu" id="authorHolder" aria-
labelledby="dropdownMenuButton2">
        </ul>
      </div>
      <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-expanded="false">
          Godina izdanja

```

```

        </button>
        <ul class="dropdown-menu" id="yearHolder" aria-
labelledby="dropdownMenuButton1">
        </ul>
    </div>
    <a class="btn btn-dark w-100" id="resetFilterButton"
href="#">Ukloni filtere</a>
    </div>
    <div class="mk-holder" id="filter-holder">
        <div class="container-fluid">
            <div class="row">
                <h2 id="mk-book-category">Sve knjige</h2>
                <hr>
            </div>
            <div id="event-div" class="row mk-clear w-100">

                </div>
            </div>
            <p id="loadMore"></p>
        </div>
    </div>
</div>
<footer class="container-fluid mk-background">
    <div class="row w-100" >
        <div class="mk-logo col-6 flex" id="footer-title">
        </div>
        <div class="col-6 flex flex-md-row">
            <ul class="row align-items-center list-unstyled p-0">
                <li class="col-md-3 col-6 icon-holder">
                </li>
                <li class="col-md-3 col-6 icon-holder">
                </li>
                <li class="col-md-3 col-6 icon-holder">
                </li>
                <li class="col-md-3 col-6 icon-holder">
                </li>
            </ul>
        </div>
    </div>
</footer>
    <a href="#" class="invisible" id="goBackUp">&uArr;</a>
<script src="../../js/jquery-3.6.1.js"></script>
<script src="../../js/main.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-

```

```
pprn3073KE6t16bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
<script src="https://code.iconify.design/2/2.2.1/iconify.min.js"></script>
</body>
</html>
```

3.1.3 knjiga.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="author" content="Ilija Krstić"/>
  <meta name="keywords" content="Gradska Biblioteka, Autor, Kopija,
Rezerviši, Rezervisanje, Knjiga"/>
  <meta name="description" content="Detaljan pogled knjige iz Gradske
Biblioteke, sadržeći autora i datum izdavanja"/>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfhXaWutUpFmBp4vmVor"
crossorigin="anonymous"/>
  <link rel="stylesheet" type="text/css" href="../css/style.css"/>
  <link rel="icon" type="image/x-icon" href="../imgs/favicon.ico">
  <title>Detalji o knjizi</title>
</head>
<body>
  <header class="container-fluid">
    <div class="row">
      <nav class="navbar navbar-expand-lg navbar-dark mk-background">
        <div class="container-fluid" id="actual-navbar">

          </div>
        </nav>
      </div>
    </header>
    <div id="errorModal" class="mk-modal">
      <div class="mk-modal-content">
        <p id="error-content">Generička greška</p>
      </div>
    </div>
    <div id="myModal" class="mk-modal">
      <div class="mk-modal-content">
        <form class="row" action="uspeh.php" method="post">
```

```
<div class="mb-3 col-12 col-md-6">
  <label for="firstName" class="form-label">Ime<span class="mk-
red">*</span></label>
  <input type="text" class="form-control" id="firstName"
placeholder="Luka" aria-describedby="emailHelp"/>
  <p id="firstNameError" class="form-text mk-red error-alert
hidden"></p>
</div>
<div class="mb-3 col-12 col-md-6">
  <label for="lastName" class="form-label">Prezime<span class="mk-
red">*</span></label>
  <input type="text" class="form-control" id="lastName"
placeholder="Lukić" aria-describedby="emailHelp"/>
  <p id="lastNameError" class="form-text mk-red error-alert
hidden"></p>
</div>
<div class="mb-3 col-md-7">
  <label for="address" id="addDesc" class="form-label">Adresa (ulica
i broj)<span class="mk-red">*</span></label>
  <input type="text" class="form-control" id="address"
placeholder="Kralja Aleksandra 13"/>
  <p class="form-text mk-red error-alert hidden"></p>
</div>
<div class="mb-3 form-check m-0 col-md-5">
  <p>Preuzimanje<span class="mk-red">*</span></p>
  <div class="mk-delivery row">
    <div class="col-6">
      <input name="delivery" checked="checked" id="delivered"
value="1" type="radio" class="form-check-input"/>
      <label class="form-check-label"
for="delivered">Dostava</label>
    </div>
    <div class="col-6">
      <input name="delivery" id="personal" value="0" type="radio"
class="form-check-input"/>
      <label class="form-check-label" for="personal">Lično</label>
    </div>
    <p id="deliveryError" class="form-text mk-red error-alert
hidden"></p>
  </div>
</div>
<div class="mb-3 col-md-5">
  <label for="reserveStart">Rezervisati od<span class="mk-
red">*</span></label>
  <input type="date" id="reserveStart" name="reserveStart"/>
```



```

        <p class="form-text mk-red error-alert hidden"></p>
    </div>
    <div class="mb-3 col-md-7">
        <label for="length">Broj dana rezervacije<span class="mk-
red">*</span></label>
        <input type="number" id="length" name="quantity" min="1"
placeholder="20" max="40"/>
        <p class="form-text mk-red error-alert"></p>
    </div>
    <div class="col-12 mk-modal-buttons">
        <button class="btn mk-send">Rezerviši</button>
        <button type="submit" class="mk-close btn btn-
danger">Odustani</button>
    </div>
    <p id="successAlert" class="form-text mk-yellow error-alert
hidden"></p>
</form>
</div>
</div>
<div class="mk-solo-page page container-fluid flex-column">
    <div class="row" id="bookPage">
        <div class="col-12 col-lg-3">
            <div class="position-relative">
                
                <a href="#" data-id="0" class="mk-favorite-icon-holder">
                    <span class="iconify mk-favorite-icon" data-icon="mdi:cards-
heart-outline">Heart</span>
                </a>
            </div>
        </div>
        <div class="col-12 col-lg-9" id="bookInfo">
            <div class="w-100 mk-yellow">
                <h2 id="book-title">Ime knjige</h2>
            </div>
            <p id="book-description"></p>
            <div>
                <div class="dynamic-field" id="author-field">
                    <p>Autor:</p>
                    <ul class="list-unstyled">
                        <li>
                            <a id="author-link" href="#"></a>
                        </li>
                    </ul>
                </div>
            </div>

```

```

<div class="dynamic-field" id="category-field">
  <p>Kategorija:</p>
  <ul class="list-unstyled">
    <li>
      <a id="category-link" href="#"></a>
    </li>
  </ul>
</div>
<div class="dynamic-field" id="date-field">
  <p>Godina izdavanja: </p>
  <ul class="list-unstyled">
    <li>
      <a id="date-link" href="#"></a>
    </li>
  </ul>
</div>
<div class="dynamic-field">
  <p id="availability-field">Broj dostupnih kopija: x</p>
</div>
<div class="dynamic-field">
  <p id="loaned-field">Izdato 0 puta</p>
</div>
  <a class="btn btn-light w-100" id="openModal">Rezerviši</a>
</div>
</div>
<div id="relatedBooks">
  <div class="mk-holder">
    <div class="container-fluid">
      <div class="row">
        <h2 id="mk-book-category">Slične knjige</h2>
        <hr>
      </div>
      <div id="event-div" class="row mk-clear w-100 align-items-center justify-content-center">

        </div>
      </div>
    </div>
  </div>
</div>
</div>

</div>
<footer class="container-fluid mk-background">
  <div class="row w-100" >

```

```

<div class="mk-logo col-6 flex" id="footer-title">
</div>
<div class="col-6 flex flex-md-row">
  <ul class="row align-items-center list-unstyled p-0">
    <li class="col-md-3 col-6 icon-holder">
    </li>
    <li class="col-md-3 col-6 icon-holder">
    </li>
    <li class="col-md-3 col-6 icon-holder">
    </li>
    <li class="col-md-3 col-6 icon-holder">
    </li>
  </ul>
</div>
</div>
</footer>
<a href="#" class="invisible" id="goBackUp">&uArr;</a>
<script src="../js/jquery-3.6.1.js"></script>
<script src="../js/jquery.zoom.js"></script>
<script src="../js/main.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
pprn3073KE6tL6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
<script src="https://code.iconify.design/2/2.2.1/iconify.min.js"></script>
</body>
</html>

```

3.1.4 Favoriti.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="keywords" content="Gradska Biblioteka, Rezerviši, Favoriti,
Knjiga"/>
  <meta name="description" content="Prikaz svih knjiga sačuvanih kao
favoriti"/>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfhXaWutUpFmBp4vmVor"
crossorigin="anonymous"/>
  <link rel="stylesheet" type="text/css" href="../css/style.css"/>
  <link rel="icon" type="image/x-icon" href="../imgs/favicon.ico">

```

```

    <title>Favoriti</title>
</head>
<body>
    <header class="container-fluid">
        <div class="row">
            <nav class="navbar navbar-expand-lg navbar-dark mk-background">
                <div class="container-fluid" id="actual-navbar">

                    </div>
                </nav>
            </div>
        </header>
        <div id="errorModal" class="mk-modal">
            <div class="mk-modal-content">
                <p id="error-content">Generička greška</p>
            </div>
        </div>
        <div class="mk-solo-page page container-fluid">
            <div class="row">
                <div class="mk-holder">
                    <div class="container-fluid w-100">
                        <div class="row">
                            <h2 id="mk-book-category">Favoriti</h2>
                            <hr>
                        </div>
                        <div id="event-div" class="row mk-clear w-100">

                            </div>
                            <a id="loadMore" href="#">Učitaj još</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <footer class="container-fluid mk-background">
            <div class="row w-100" >
                <div class="mk-logo col-6 flex" id="footer-title">
                </div>
                <div class="col-6 flex flex-md-row">
                    <ul class="row align-items-center list-unstyled p-0">
                        <li class="col-md-3 col-6 icon-holder">
                        </li>
                        <li class="col-md-3 col-6 icon-holder">
                        </li>
                        <li class="col-md-3 col-6 icon-holder">

```

```

        </li>
        <li class="col-md-3 col-6 icon-holder">
        </li>
    </ul>
</div>
</div>
</footer>
<a href="#" class="invisible" id="goBackUp">&uArr;</a>
<script src="../../js/jquery-3.6.1.js"></script>
<script src="../../js/main.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
pprn3073KE6t16bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
<script src="https://code.iconify.design/2/2.2.1/iconify.min.js"></script>
</body>
</html>

```

3.2 CSS

3.2.1 style.css

```

body{
    overflow-x:hidden;
}
hr{
    color: black;
}
.navbar{
    min-height: 8vh;
}
.mk-light-yellow{
    color: #f7bd2f;
}
.mk-yellow{
    color: #f5af05;
}
.page{
    position: relative;
    display: flex;
    align-items: center;
    justify-content: space-around;
}
.spec-row{
    width: 100vw;
}

```

```
.mk-flex-column-center{
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}
.mk-part-page{
  min-height: 60vh;
}
.mk-page{
  min-height: 92vh;
}
.mk-solo-page{
  min-height: 77vh;
}
.mk-full-page{
  min-height: 100vh;
  padding: 0px;
  padding: 10px 0px 10px 0px;
}
.mk-red{
  color: red;
}
.nav-link, .navbar-brand{
  color: #f7bd2f !important;
}
a.active{
  color: #e7c776 !important;
}
.dropdown-item:active{
  background-color: white !important;
}
.dropdown-menu{
  background-color: #333333 !important;
}
#landing-page{
  padding: 0px;
  background-image: url(../imgs/main.jpg);
  background-size: cover;
  background-position: center;
}
#landing-page .row{
  width: 100%;
}
.mk-introduction{
```

```
width: 100vw;
height: 25vh;
color: #f7bd2f;
background-color: #333333;
text-align: center;
padding: 10px;
}
.mk-holder{
  position: relative;
  min-height: 65vh;
  width: 100vw;
  border-radius: 2%;
  padding-top: 2px;
  text-align: center;
}
.card{
  border: none;
}
.mk-card-limit{
  width: 90%;
}
.info-holder{
  background-color: #333333;
  padding: 20px;
}
.info-block{
  background-color: #333333;
  color: #f5af05;
  border-top: 1px solid #f5af05;
  border-bottom: 1px solid #f5af05;
  text-align: center;
  height: 25vh;
}
#lib-desc{
  background-color: #333333;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  color: #f5af05;
  padding: 40px;
}
#libHist{
  margin-top: 20px;
  text-align: center;
```

```
}
.mk-favorite-icon-holder{
  position: absolute;
  top: 20px;
  right: 10px;
  background-color: white;
  padding: 4px;
  color: red !important;
  border-radius : 50%;
}
.mk-favorite-icon-holder:hover{
  color: red !important;
}
.mk-favorite-icon-holder:hover{
  scale: 1.10;
}
.mk-favorite-icon{
  font-size: 50px;
}
#mk-book-category{
  color: #333333;
  font-weight: bold;
  margin-top: 10px;
}
.card-title{
  font-weight: bold;
}
.moveButton{
  box-shadow: none;
  display: none;
  background-color: #333333;
  color: aliceblue !important;
  text-decoration: none;
  width: 2.5em;
  height: 2.5em;
  line-height: 2.2em;
  border-radius: 20%;
  border: 2px solid #f7bd2f;
}
.moveButton:hover{
  background-color: #222222;
  color: aliceblue !important;
}
.book:hover{
  transform: scale(1.02);
```



```
    transition: 0.2s;
}
#filters{
    margin: 0% auto;
    display: flex;
    flex-direction: column;
    justify-content: center;
    gap: 8px;
    align-items: center;
    width: 100vw;
    min-height: 35vh;
    padding-bottom: 0px;
}
.form-check-label{
    background-color: #333333 !important;
}
#filters .text-center{
    width: 100%;
}
#filters .text-center .form-text{
    width: 100%;
}
#filters .dropdown button{
    background-color: #333333 !important;
    color: #f5af05 !important;
}
#resetFilterButton{
    background-color: #f7bd2f;
    color: #333333 !important;
}
#resetFilterButton:hover{
    background-color: #f5af05;
}
.dropdown-menu{
    border: 0px;
}
.dropdown-menu .dropdown-item{
    background-color: #333333 !important;
    color: #f7bd2f;
}
.dropdown{
    padding-bottom: 0px;
    width: 100% !important;
}
.dropdown .btn{
```

```
    width: 100% !important;
}
.dropdown-menu{
    background-color: white !important;
    width: 100%;
}
.form-check-label{
    background-color: white;
}
.form-text{
    height: 2rem;
}
.book{
    color: white;
    background-color: #333333;
    border: 1px solid #333333;
    transform: scale(1.0);
    transition: 0.2s;
    padding: 0px;
    margin-bottom: 10px;
}
#loadMore{
    display: flex;
    border-radius: 5%;
    margin: 10px auto;
    margin-bottom: 10px;
    color: white !important;
    padding: 5px;
    width: 100%;
    gap: 15px;
    justify-content: center;
}
#loadMore a{
    padding: 20px;
    border-radius: 20%;
    background-color: #222222;
}
#loadMore a:hover{
    color: #f5af05;
}
#bookPage{
    padding: 20px;
    background-color: #333333;
    border: 2px solid white;
    color: white;
```

```
border-radius: 1.5%;
text-align: center;
width: 100%;
margin-top: 2%;
}
.book-prev{
border: 1px solid #f7bd2f;
height: 300px;
}
.book-title{
color: #f5af05;
min-height: 3rem;
}
.book-desc{
min-height: 3rem;
}
.book-body{
display: flex;
flex-direction: column;
justify-content: space-between;
align-items: center;
}
.zoom-span{
border: 2px solid #f7bd2f !important;
display:inline-block;
}
#bookInfo{
margin-top: 10px;
display: flex;
flex-direction: column;
justify-content: space-between;
}
#book-description{
padding: 5%;
}
.success{
border: 2px solid green;
}
.mk-modal {
display: none;
position: fixed;
z-index: 1;
left: 0;
top: 0;
width: 100%;
```

```
height: 100%;
overflow: hidden;
background-color: rgba(0,0,0,0.4);
}
.mk-modal-content {
margin: 8% auto;
color: #f7bd2f !important;
background-color: #333333;
padding: 20px;
border: 1px solid #888;
width: 90%;
border-radius: 1%;
}
.mk-modal-buttons{
display: flex;
justify-content: space-between;
}
#openModal{
background-color: #f7bd2f;
color: #333333 !important;
border: 1px solid #333333;
}
#openModal:hover{
background-color: #f5af05;
}
#openModal:focus{
background-color: #f5af05;
}
form{
color: #f7bd2f;
text-align: center;
}
input{
text-align: center;
}
.mk-send{
background-color: #f7bd2f;
color: #333333;
}
.mk-send:hover, .mk-send:focus {
background-color: #f5af05;
color: #333333;
}
.form-check-input:checked{
background-color: #f5af05;
```

```
}
.mk-delivery{
  flex-direction: row;margin-left: 10px;
}
#deliveryError{
  padding-left: 0px;
}
#reserveStart{
  display: block;
  width: 100%;
  margin: 0% auto;
  margin-top: 8px;
}
.success{
  border: 2px solid green !important;
}
.failure{
  border: 2px solid red;
}
.error-alert{
  margin-bottom: 0px;
}
#length{
  width: 100%;
  margin-top: 8px;
}
#autor-div{
  display: flex;
  flex-direction: row;
  justify-content: center;
  gap: 0px;
  width: 80%;
}
#event-div{
  flex-direction: row;
  justify-content: flex-start;
  text-align: center;
}
.mk-clear{
  --bs-gutter-x: 0rem;
}
.mk-events .container-fluid{
  padding-bottom: 2%;
}
.mk-quote{
```

```
    font-style: italic;
}
.mk-background{
    background-color: #333333;
}
.mk-logo h2{
    font-size: 16px;
    display: inline;
}
a{
    color: #f7bd2f !important;
}
a:link {
    text-decoration: none;
}
a:hover{
    color: #f5af05 !important;
}
.mk-logo a{
text-decoration: none;
color: #f5af05;
}
ul, li{
    display: inline;
}
.dynamic-field{
    margin: 5px 0px;
}
.dynamic-field p{
    display: inline;
}
#goBackUp{
    position: fixed;
    right: 50px;
    bottom: 10vh;
    border-radius: 50%;
    background-color: #333333;
    color: white !important;
    border: 2px solid #f5af05;
    width: 2.5em;
    height: 2.5em;
    font-size: 22px;
    text-align: center;
    line-height: 2.25em;
    display: none;
```

```
}
#goBackUp:hover{
  background-color: black;
}
.invisible{
  visibility: none;
}
.hidden{
  display: none !important;
}
footer .row{
  height: 100%;
  margin: 0%;
  width: 100%;
}
.mk-sorted-book-holder{
  margin-top: 10px;
  text-align: center;
  flex-direction: row;
  align-items: center;
  justify-content: space-around;
}
footer{
  height: 15vh;
}
.flex{
  display: flex;
  align-items: center;
  text-align: center;
}
#error-content{
  text-align: center;
}
@media screen and (min-width: 768px){
  .mk-full-page{
    padding: 0px;
  }
  .mk-collapsible-page{
    min-height: 92vh;
  }
  .mk-solo-page{
    min-height: 84vh;
  }
  footer{
    margin-top: 0vh;
  }
}
```

```
        height: 8vh;
    }
    .mk-holder{
        min-height: 60vh;
        border-radius: 1%;
    }
    .mk-modal-content{
        width: 90%;
    }
    .moveButton{
        display: block;
    }
    .mk-delivery{
        flex-direction: row;
        justify-content: space-between;
        align-content: space-between;
    }
    #delivery{
        display: inline;
    }
    form{
        text-align: left;
    }
    input{
        text-align: left;
    }
    #reserveStart{
        display: inline;
    }
    .mk-modal-buttons{
        display: block;
    }
    .mk-close{
        margin-left: 10px;
    }
    .dropdown-menu{
        padding: 0px;
    }
    #loadMore{
        gap: 40px;
    }
}
@media screen and (min-width: 992px) {
    .navbar{
        height: 8vh;
    }
}
```



```
        padding: 0% 2% 0% 2%;
    }
    #filters{
        width: 19vw;
        margin-top: 40px;
        height: 45vh;
    }
    footer .row{
        padding: 0% 2% 0% 2% !important;
    }
    .mk-introduction{
        width: 100vw;
        height: 18vh;
    }
    .mk-card-holder{
        margin-bottom: 0%;
    }
    .mk-sorted-book-holder{
        margin-top: 0px;
    }
    .mk-modal-content{
        width: 40%;
    }
    .mk-responsive-width{
        width: 75%;
    }
    .mk-holder{
        width: 80vw;
        padding-top: 5px;
    }
    #filter-holder{
        width: 78vw;
    }
    #bookInfo{
        margin-top: 0px;
    }
    #bookPage{
        width: 75%;
    }
    input{
        text-align: left;
    }
    #loadMore{
        width: 30%;
    }
}
```

```

    #goBackUp{
      display: block;
    }
    .mk-book-category{
      margin-top: 0px;
    }
  }
}

```

3.3 Javascript

3.3.1 main.js

```

var mainPage;
var sPath = window.location.pathname;
var sPage = sPath.substring(sPath.lastIndexOf('/') + 1);
var favoritesOnly = false;
const JSONPATH = "assets/"
let books = new Array();
let sortOptions = ["Najpopularnije", "Najmanje popularno", "Po nazivu (od a)", "Po nazivu (od z)", "Najnovije", "Najstarije"];
let currentSort = sanitizeForIdentifier(sortOptions[0]);
let searchBoxes = new Array(
  {"title" : "Kategorije", "prop" : "category", "complex" : true, "holder" : "categoryHolder"},
  {"title" : "Autori", "prop" : "author", "complex" : true, "holder" : "authorHolder"},
  {"title" : "Godine", "prop" : "releaseDate", "complex" : false, "holder" : "yearHolder"});
let currentPage = 1;
let perPage = 4;
const currentYear = new Date().getFullYear();
window.onload = function(){
  mainPage = sPage == "index.html" || sPage.length == 0 ? true : false;
  callback("navbar.json", generateNavBar, ["#actual-navbar", "#footer-title"]);
  callback("footer.json", generateFooter, ["icon-holder"])
  callback("books.json", initializeBooks);
  window.onscroll = function(){
    let upButton = $("#goBackUp");
    if ($(window).scrollTop() > 300){
      upButton.removeClass("invisible");
    }
    else{
      upButton.addClass("invisible");
    }
  }
}

```

```

    }
    let upButton = $("#goBackUp");
    upButton.click(function(event){
        event.preventDefault();
        $(window).scrollTop(0);
    })
    //If currently on index page
    if(sPage === "index.html" || sPage.length === 0){

    }
    if(sPage === "knjiga.html"){
        $('#openModal').click(function(event) {
            event.preventDefault();
            $("#myModal").show("fast")
        });
        $('#mk-close').click(function(event){
            event.preventDefault();
            $("#myModal").hide("fast")
        })
        $(window).click(function(event){
            if (event.target === $('#myModal')[0])
                $("#myModal").hide("fast")
        })
        $(".mk-send").click(function(event){
            event.preventDefault();
            checkForm();
        })
        let deliveryRadios = $("input[name='delivery']");

        deliveryRadios[0].addEventListener("click",function(){addressRequired(true)});

        deliveryRadios[1].addEventListener("click",function(){addressRequired(false)});
        let reName = /^[A-ZŠĆČĐŽ][a-zšđćčž]{2,13}(\s[A-ZŽŠĆČĐ][a-zžšđćč]{2,13}){0,3}$/
        let firstName = document.querySelector("#firstName")
        let lastName = document.querySelector("#lastName")
        let address = document.querySelector("#address")
        let reserveDate = document.querySelector("#reserveStart");
        let length = document.querySelector("#length");
        let radioSelected;
        for(let radio of deliveryRadios){
            if(radio.checked){
                radioSelected = radio.value;
                break;
            }
        }
    }

```

```

    }
  }
  addressBool = radioSelected>0? true:false;
  addressRequired(addressBool);
  firstName.addEventListener("blur", function(){checkFormRegex(firstName,
reName,"Ime sme da sadrži samo slova i ne sme biti prazno");})
  lastName.addEventListener("blur", function(){checkFormRegex(lastName,
reName,"Prezime sme da sadrži samo slova i ne sme biti prazno");})
  address.addEventListener("blur", checkAddress);
  reserveDate.addEventListener('blur', checkDate);
  length.addEventListener('blur', checkLength);
}
}
//Funkcija .zoom plugina se vidi samo u globalnom scopu iz nekog razloga
if(sPage === "knjiga.html"){
  const queryString = window.location.search;
  const urlParams = new URLSearchParams(queryString);
  const urlBook = urlParams.get('knjiga')
  if(urlBook != undefined)
  {
    zoomOnHover("#bookImage", urlBook.toLowerCase());
  }
}
function book(name, category, author, description, copies, releaseDate){
  var name, category, author;
  this.name = name;
  this.category = category;
  this.author = author;
  this.description = description;
  this.copies = copies;
  this.releaseDate = releaseDate;
}

function saveToLocalStorage(value, args){
  localStorage.setItem(args[0], JSON.stringify(value));
}

function getFromLocalStorage(args){
  return JSON.parse(localStorage.getItem(args[0]));
}

function callback(file, handler, args = [], asynchronicity = true){

```

```

let request = createRequest();
request.onreadystatechange = function(){
    if(request.readyState == 4){
        if(request.status >= 200 && request.status < 300){
            if(args == []){
                handler(JSON.parse(this.responseText));
            }
            else{
                handler(JSON.parse(this.responseText), args);
            }
        }
        else if(request.status >= 400 && request.status < 600){
            let errorModal = $("#errorModal");
            let errorModalP = $("#error-content");
            errorModalP.text("Dogodila se greška pri komunikaciji sa bazom");
            errorModal.show("fast");
            errorModal.click(function(){
                errorModal.hide("fast");
            })
        }
    }
}

request.open("GET", `${mainPage? "" : "../"}${JSONPATH+file}`,
asynchronicity)
request.send();
}

```

```

function createRequest(){
    let request = false;
    try{
        request = new XMLHttpRequest();
    }
    catch(windows){
        try{
            request = new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch(otherWindows){
            try{
                request = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch{
                console.log("Ajax nije podrzan");
            }
        }
    }
}

```

```

    }
    return request;
}

function zoomOnHover(zoomElement, zoomImg)
{
    $(zoomElement)
    .wrap('<span class="zoom-span"></span>')
    .css('display', 'block')
    .parent()
    .zoom({
        url: `../imgs/${zoomImg}.jpg`
    })
}

function limitToFullWords(text, length){
    let textSplit = text.split(" ");
    text = "";
    for(let word of textSplit){
        if(word.length + text.length <= length) text += word + ' ';
        else{
            text = text.trim() + '...';
            break;
        }
    }
    return text;
}

function negativeToBCE(year){
    return Math.abs(year) + (year < 0 ? ". P.N.E." : ".")
}

function fillColumns(elementList, columns, numberOfColumns){
    for(let column of columns){
        column.innerHTML = "";
    }
    for(let element in elementList){
        columns[element % numberOfColumns].innerHTML += elementList[element];
    }
}

function fillBooks(elementList, holder){
    holder.innerHTML = "";
    for(let element of elementList){
        $(element).hide();
        holder.append(element)
    }
}

```

```

    }
    for(let element of elementList){
        $(element).slideDown(200);
    }
    let newElementList = document.getElementsByClassName("mk-favorite-icon-
holder")
    for(let element of newElementList){
        if(element.hasAttribute('listener')) continue;
        element.addEventListener("click", function(event){
            event.preventDefault();
            eventAddToFavorite(element);
            if(favoritesOnly){
                loadMore(false);
            }
        });
        element.setAttribute('listener', true);
    }
}

function eventAddToFavorite(element){
    addToFavorite(element);
    displayNumberOfFavorites(['favoriti', '#Favoriti'])
}

function displayNumberOfFavorites(args){
    let element = document.querySelector(args[1]);
    element.innerHTML = `Favoriti: ${getNumberOfFavorites()}`
    function getNumberOfFavorites(){
        let arrayFromLocalStorage = getFromLocalStorage([args[0]]);
        if(arrayFromLocalStorage == null){
            return 0
        }
        return arrayFromLocalStorage.length;
    }
}

function addToFavorite(element){
    let idOfBookFavorited = element.dataset.id;
    //Kod za vizuelni prikaz promene na već prikazanim knjigama
    let allLoadedBookElements = document.getElementsByClassName("mk-favorite-
icon-holder");
    for(let iconHolder of allLoadedBookElements){

```

```

    if(iconHolder.dataset.id == idOfBookFavorited){
        var iconElement = iconHolder.firstElementChild;
        if(iconElement.dataset.icon == "mdi:cards-heart-outline"){
            iconElement.dataset.icon = "mdi:cards-heart";
            element.setAttribute('favorite', true);
            continue;
        }
        iconElement.dataset.icon = "mdi:cards-heart-outline"
        element.setAttribute('favorite', false);
    }
}
//Kod za dodavanje u local storage
addToLocalStorage(["favoriti", idOfBookFavorited, false, true]);
}

```

```

function generateUrl(object, redirect = ""){
    let url = "";
    if(object["primary"]){
        url += mainPage? '' : '../';
    }
    else{
        url += mainPage? `${redirect}` : '';
    }
    url += object["url"];
    return url;
}

```

```

function generateNavBar(objects, args){
    let htmlContent = "";
    let holderElement = document.querySelector(args[0]);
    let active;
    let url;
    let id;
    for(object in objects){
        active = false;
        url = generateUrl(objects[object], 'pages/');
        id = objects[object]["text"].replaceAll(" ", "_");
        if(objects[object]["url"] == sPage) active = true;
        if(object === "0"){
            htmlContent +=
                `<a class="navbar-brand" id=${id} href=${url}><h1
class="h4">${objects[object]["text"]}</h1></a>

```



```

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button><div class="collapse navbar-collapse"
id="navbarSupportedContent"><ul class="navbar-nav ms-auto mb-2 mb-lg-0">`
    }
    else if(object == objects.length - 1){
        htmlContent +=
        `<li class="nav-item">
            <a class="nav-link ${active ? "active" : ""}" id=${id} href="${url}">
                ${objects[object]["text"]}
            </a>
        </li></ul></div>`
    }
    else{
        htmlContent +=
        `<li class="nav-item">
            <a class="nav-link ${active ? "active" : ""}" id=${id} href="${url}">
                ${objects[object]["text"]}
            </a>
        </li>`
    }
}
holderElement.innerHTML = htmlContent;
displayNumberOfFavorites(['favoriti', '#Favoriti']);

generateFooterTitle();

function generateFooterTitle(){
    let footerTitleHolder = document.querySelector(args[1]);
    let firstObject = objects[0];
    let linkWrap = document.createElement("a");
    linkWrap.href = generateUrl(firstObject, 'pages/');
    linkWrap.innerHTML = "&copy;";
    linkWrap.innerHTML += currentYear + " ";
    let header = document.createElement("h2");
    header.textContent = firstObject.text;
    linkWrap.appendChild(header);
    footerTitleHolder.appendChild(linkWrap);
}
}
function generateFooter(objects, args){

```

```

let columns = document.getElementsByClassName(args[0]);
let elementList = new Array();
for(let i = 0; i < objects.length; i++){
  elementList.push(
    `

```

```

    }
  }
  if(search.active) {
    activeSearches.push(search);
    numberOfSearchFactors++;
  }
  if(acceptedCategories.length > 0){
    if(search.complex){
      filteredBooks = filteredBooks.filter(b =>
acceptedCategories.some(c => b[search.prop].id == c))
    }
    else{
      filteredBooks = filteredBooks.filter(b =>
acceptedCategories.some(c => b[search.prop] == c));
    }
    for(let searchY of searchBoxes){
      if(search.active == true){
        if(searchY.title != search.title){
          let properties = getPropertiesFromBooks([searchY.prop,
searchY.complex, filteredBooks]);
          let propertyHolder =
document.getElementById(searchY.holder);
          showCategories([propertyHolder, properties,
searchY.title]);
        }
      }
    }
  }
  saveValueOfFilter([search.title]);
}
if(textToSearch){
  numberOfSearchFactors++;
  filteredBooks = filteredBooks.filter(b =>
b.name.toLowerCase().includes(textToSearch));
}

if(numberOfSearchFactors < 2){
  generateAllFilters([activeSearches, books]);
}
if(numberOfSearchFactors >= 2 || textToSearch){
  generateAllFilters([searchBoxes, filteredBooks]);
}
if(numberOfSearchFactors == 0){
  generateAllFilters([searchBoxes, books]);
}

```

```

    }

    if(filteredBooks.length < 1){
        displayNoBooksFitYourParamaters([columns, "Ni jedna knjiga ne odgovara
pretrazi"]);
        if(paginate){
            fillPageButtons(0, searchable);
        };
        return;
    }
    filteredBooks = sort([filteredBooks, currentSort]);
}
numberOfPages = Math.ceil(filteredBooks.length / perPage);
if(numberOfPages == 1) currentPage = 1;
for(let booksLoaded = (currentPage - 1) * perPage;
booksLoaded<currentPage*perPage;booksLoaded++){
    if(booksLoaded>=filteredBooks.length){returnCode = true; break;}
    if(booksLoaded== filteredBooks.length - 1){returnCode = true;}
    let currentBook = filteredBooks[booksLoaded];
    elementList.push(bookToElement(currentBook));
}
fillBooks(elementList, columns);
if(paginate){
    fillPageButtons(numberOfPages, searchable);
}
return returnCode;
}

function generateAllFilters(args){
    let searchBoxes = args[0];
    let filteredBooks = args[1];
    for(let searchY of searchBoxes){
        let properties = getPropertiesFromBooks([searchY.prop, searchY.complex,
filteredBooks]);
        let propertyHolder = document.getElementById(searchY.holder);
        showCategories([propertyHolder, properties, searchY.title]);
    }
}

function fillPageButtons(numberOfPages, searchable){
    let holder = document.querySelector("#loadMore");
    holder.innerHTML = "";

```

```

    if(numberOfPages < 2){
        return;
    }
    for(let i = 1; i <= numberOfPages; i++){
        let pageButton = document.createElement("a");
        pageButton.dataset.page = i;
        pageButton.innerText = i;
        pageButton.href = "#";
        pageButton.addEventListener("click", function(event){
            event.preventDefault();
            currentPage = this.dataset.page;
            loadMore(searchable);
        })
        if(pageButton.dataset.page == currentPage)
pageButton.classList.add("active");
        holder.appendChild(pageButton);
    }
}

function displayNoBooksFitYourParamaters(args){
    args[0].innerHTML = `

${args[1]}

`;
}

function setLinkValue(selector,filter, href, text)
{
    document.querySelector(selector).setAttribute("href",`knjige.html?${filter}=${href}`);
    document.querySelector(selector).innerText = text;
}
function loadMore(searchable = true){
    let holder = document.querySelector("#event-div")
    generateBooks([books, holder, true, searchable]);
}
let addressBool;
function addressRequired(req){
    let address = document.querySelector("#addDesc");
    let addressInput = document.querySelector("#address")
    addressBool = req;
    if(req){
        address.innerHTML = `Adresa (ulica i broj)<span class="mk-red">*</span>`;
        addressInput.setAttribute("required", "required");
        addressInput.removeAttribute("disabled")
    }
}

```

```

        addressInput.setAttribute("placeholder", "Kralja Aleksandra 13");
        addressInput.classList.remove("success");
    }
    else{
        address.innerHTML = `Adresa (ulica i broj)`;
        removeError(addressInput);
        addressInput.setAttribute("disabled", "disabled");
        addressInput.value = "";
        addressInput.removeAttribute("required");
        addressInput.removeAttribute("placeholder");
    }
}
function checkForm(){
    let success = 0;
    let deliveryRadios = document.getElementsByName("delivery");
    let firstName = document.querySelector("#firstName");
    let lastName = document.querySelector("#lastName");
    let radioSelected;
    let reName = /^[A-ZŠĆČĐŽ][a-zšđćčž]{2,13}(\s[A-ZŽŠĆČĐ][a-zžšđćć]{2,13}){0,3}$/;
    for(let radio of deliveryRadios){
        if(radio.checked){
            radioSelected = radio.value;
            break;
        }
    }
    if(typeof radioSelected !== 'undefined'){
        addressBool = radioSelected>0? true:false;
    }
    else success-=1;
    if(firstName.value == ""){displayError(firstName, "Ime ne sme biti prazno")
    success--};
    if(lastName.value == ""){displayError(lastName, "Prezime ne sme biti prazno")
    success--};
    if(radioSelected == null){
        let errorHolder =
deliveryRadios[0].parentNode.parentNode.lastElementChild;
        errorHolder.innerText = 'Morate odabrati način preuzimanja'
        errorHolder.removeAttribute("hidden");
        success -= 1;
    }
    else{
        let errorHolder =
deliveryRadios[0].parentNode.parentNode.lastElementChild;

```

```

        errorHandler.setAttribute("hidden", "hidden");
    }
    success += checkFormRegex(firstName, reName, "Ime sme da sadrži samo slova i
ne sme biti prazno");
    success += checkFormRegex(lastName, reName, "Prezime sme da sadrži samo slova
i ne sme biti prazno");
    success += checkAddress();
    success += checkDate();
    success += checkLength();
    if(success == 0){
        showSuccess();
    }
    else{
        hideSuccess();
    }
}
function showSuccess(){
    let element = document.querySelector("#successAlert");
    element.innerText = "Uspešno ste rezervisali knjigu";
    show(element);
}
function hideSuccess(){
    let element = document.querySelector("#successAlert");
    hide(element);
}
function checkLength(){
    let reserveLength = document.querySelector("#length");
    if(reserveLength.value == ""){
        displayError(reserveLength, "Morate odabrati trajanje rezervacije [1-
40]");
        return -1;
    }
    if(reserveLength.value < 0 || reserveLength.value > 40){
        displayError(reserveLength, "Broj dana trajanja rezervacije može biti od 1
do 40");
        return -1;
    }
    removeError(reserveLength);
    return 0;
}
function checkAddress(){
    let reAddress = /^[([A-ZŠĐČĆŽ][a-zšććđž\d]+)|([0-9][1-9]*\.\.?))(\s[A-Za-
zŠĐŽĆĆšććđž\d+){0,7}\s(((1-9)[0-9]{0,5}[\/-]?[A-Z])|((1-9)[0-
9]{0,5})|(BB))\.\.?$/
    let address = document.querySelector("#address");

```

```

    if(addressBool){
        if(address.value === ""){
            displayError(address,"Adresa ne sme biti prazna za preuzimanje
dostavom");
            return -1;
        }
        else{
            return checkFormRegex(address, reAddress, "Adresa ne odgovara formatu,
primer: 'Kralja Aleksandra 13'");
        }
    }
    return 0;
}
function checkDate(){
    if(reserveStart.value === ""){
        displayError(reserveStart, "Morate odabrati datum početka rezerve");
        return -1;
    }
    let inputDate = new Date(reserveStart.value);
    let currentDate = new Date()
    if(inputDate < currentDate){
        displayError(reserveStart, "Rezerva ne može početi pre sutrašnjeg
dana")
        return -1;
    }
    else{
        removeError(reserveStart);
        return 0;
    }
}
function displayError(element, message){
    let errorHolder = element.nextElementSibling;
    element.classList.add("failure");
    element.classList.remove("success");
    errorHolder.innerText = message;
    show(errorHolder)
}
function removeError(element){
    let errorHolder = element.nextElementSibling;
    element.classList.remove("failure");
    element.classList.add("success");
    hide(errorHolder);
}
function show(element){
    if(element.classList.contains("hidden")) element.classList.remove("hidden");
}

```



```

}
function hide(element){
  element.classList.add("hidden");
}
function checkFormRegex(element, test, message){
  if(test.test(element.value)){
    removeError(element);
    return 0;
  }
  else{
    displayError(element, message);
    return -1;
  }
}
function countTo(element, from, to, timeToLoad){
  let interval = timeToLoad/Math.abs(to - from);
  let step = to > from ? 1 : -1;
  if(from === to){
    element.textContent = from;
    return;
  }
  let counter = setInterval(function(){
    from += step;
    element.textContent = from;
    if(from == to){
      clearInterval(counter);
    }
  }, interval);
}
function bookToElement(currentBook){
  let bookDescription = currentBook.description;
  let wrapper = document.createElement("div");
  let href = `${mainPage ? "pages/" : ""}knjiga.html?knjiga=${currentBook.name}`;
  let favorites = getFromLocalStorage(['favoriti']);
  let favorite = false;
  if(favorites != null){
    favorite = favorites.includes(currentBook.id);
  }
  let imgPart;
  wrapper.classList.add("flex", "col-12", "col-md-6", "col-lg-3", "justify-content-center")
  wrapper.href = `knjiga.html?knjiga=${currentBook.name}`
  if (currentBook.description.length > 30){
    bookDescription = limitToFullWords(currentBook.description, 30);
  }
}

```

```

    }
    imgPart = `
    <div class="position-relative">
    <a href=${href}></a>
    <a href="#" data-id="${currentBook.id}" class="mk-favorite-icon-holder">
    <span class="iconify mk-favorite-icon" data-icon="${favorite ? "mdi:cards-
heart" : "mdi:cards-heart-outline"}">Heart</span>
    </a>`
    wrapper.innerHTML = `
    <div class="card book mk-card-limit">
    ${imgPart}
    <div class="card-body book-body">
    <a href="${href}"> <h5 class="card-title book-
title">${currentBook.name.replaceAll("_", " ")}</h5> </a>
    <p class="card-text book-desc"><em>${bookDescription}</em></p>
    <p class="card-text mk-light-yellow">
    <a href="${mainPage ? "pages/" :
""}knjige.html?kategorija=${currentBook.category.name}">
${currentBook.category.name.replaceAll("_", " ")} </a>
    </p>
    <p class="card-text">
    <a href="${mainPage ? "pages/" :
""}knjige.html?autor=${currentBook.author.name}">${currentBook.author.name.repl
aceAll("_", " ")}</a>
    </p>
    </div>
    </div>
    `
    return wrapper;
}
let bookId = 0;
function moveBooks(books, holder, direction){
    holder.innerHTML = '';
    let elementList = new Array()
    if(direction === 1){
        bookId++;
        for(let i = 0; i<4; i++){
            elementList.push(bookToElement(books[(i+bookId)%books.length]));
        }
    }
    if(direction === -1){
        bookId--;
    }
}

```

```

        if(bookId < 0)bookId = books.length - 1;
        for(let i = 0; i<4; i++){
            elementList.push(bookToElement(books[(i+bookId)%books.length]));
        }
    }
    fillBooks(elementList, holder);
}
//Used for getting categories, authors and years
function getPropertiesFromBooks(args){
    let uniqueProperties = new Array();
    let saughtProperty = args[0];
    let complex = args[1];
    let booksToSearch = args[2];
    let propertyOfBook;
    if(complex){
        for(let b of booksToSearch){
            if(uniqueProperties.some(cat => cat.id == b[saughtProperty].id)){
                propertyOfBook = uniqueProperties.filter(c => c.id ==
b[saughtProperty].id)[0];
                propertyOfBook.count += 1;
            }
            else{
                b[saughtProperty].count = 1;
                uniqueProperties.push(b[saughtProperty]);
            }
        }
    }
    else{
        for(let b of booksToSearch){
            if(uniqueProperties.some(cat => cat.name == b[saughtProperty])){
                propertyOfBook = uniqueProperties.filter(c => c.name ==
b[saughtProperty])[0];
                propertyOfBook.count += 1;
            }
            else{
                uniqueProperties.push({"name" : b[saughtProperty], "count" : 1});
            }
        }
    }
    uniqueProperties = uniqueProperties.sort((a,b) =>
    {
        return b.count-a.count;
    })
    return uniqueProperties;
}

```

```

function showSortOptions(args){
  let resultHolder = args[0];
  resultHolder.innerHTML = "";
  let active;
  for(let opt of args[1]){
    let li = document.createElement("li");
    let dataName = sanitizeForIdentifier(opt)
    active = dataName == currentSort;
    li.innerHTML = `
      <div class="d-flex flex-row justify-content-between dropdown-item">
        <a id="${dataName}" class="w-100 ${active ? "active" : ""}" data-
sort="${dataName}" href="#">${opt}</a>
      </div>
    `
    resultHolder.appendChild(li);
    document.querySelector(`#${dataName}`).addEventListener("click",
function(){
      currentSort = this.dataset.sort;
      currentPage = 1;
      showSortOptions([sortHolder, sortOptions]);
      saveValueOfSort(["sacuvanSort"]);
      loadMore();
    })
  }
}

```

```

function sanitizeForIdentifier(string){
  let replaceRegex = new RegExp(/[(\)]/, "g");
  string = string.replaceAll(" ", "_").replaceAll(replaceRegex, "");
  return string
}

```

```

function sort(args){
  let booksToSort = args[0];
  let sortType = args[1];
  if(sortType == sanitizeForIdentifier(sortOptions[0])){
    return sortBooksByPopularity([booksToSort, ["desc"]]);
  }
  if(sortType == sanitizeForIdentifier(sortOptions[1])){
    return sortBooksByPopularity([booksToSort, ["asc"]]);
  }
}

```

```

    if(sortType == sanitizeForIdentifier(sortOptions[2])){
        return booksToSort.sort((a,b) =>
            a.name > b.name ? 1 : b.name > a.name ? -1 : 0
        )
    }
    if(sortType == sanitizeForIdentifier(sortOptions[3])){
        return booksToSort.sort((b,a) =>
            a.name > b.name ? 1 : b.name > a.name ? -1 : 0
        )
    }
    if(sortType == sanitizeForIdentifier(sortOptions[4])){
        return booksToSort.sort((a,b) => {
            return b.releaseDate - a.releaseDate;
        })
    }
    if(sortType == sanitizeForIdentifier(sortOptions[5])){
        return booksToSort.sort((a,b) => {
            return a.releaseDate - b.releaseDate;
        })
    }
    return booksToSort;
}

```

```

function showCategories(args){
    let resultHolder = args[0];
    let storeName = args[2];
    let currentlySelected = getFromLocalStorage([storeName]);
    tempHolder = document.createDocumentFragment();
    for(let cat of args[1]){
        let checked;
        let value = cat.id ? cat.id : cat.name;
        if(currentlySelected != null){
            if(currentlySelected.includes(value)) {
                currentPage = 1;
                checked = true
            }
        }
        let li = document.createElement("li")
        li.innerHTML += `
            <div class="d-flex flex-row justify-content-between dropdown-item">
                <label class="form-check-label text-wrap"
for="s${cat.name}">${String(cat.name).replaceAll("_", " ")}
                (${cat.count})</label>

```

```
        <input class="form-check-input float-right" type="checkbox" ${checked?
"checked='checked'" : ""} id="s${cat.name}" name="${args[2]}"
value="${value}"/>
```

```
    </div>
```

```
    tempHolder.appendChild(li);
```

```
}
```

```
resultHolder.innerHTML = "";
```

```
resultHolder.appendChild(tempHolder);
```

```
for(let elem of resultHolder.children){
```

```
    elem = elem.firstElementChild.lastElementChild;
```

```
    elem.addEventListener("change", function(){
```

```
        currentPage = 1;
```

```
        saveValueOfFilter([storeName]);
```

```
        loadMore();
```

```
    });
```

```
}
```

```
}
```

```
function saveValueOfSort(args){
```

```
    let storeName = args[0];
```

```
    localStorage.removeItem(storeName);
```

```
    addToLocalStorage([storeName, currentSort, true, false]);
```

```
}
```

```
function saveValueOfFilter(args){
```

```
    let storeName = args[0];
```

```
    let elems = document.getElementsByName(storeName);
```

```
    localStorage.removeItem(storeName);
```

```
    let exists = false;
```

```
    for(let elem of elems){
```

```
        if(elem.checked){
```

```
            addToLocalStorage([storeName, elem.value, true, true]);
```

```
            exists = true;
```

```
        }
```

```
    }
```

```
    if(!exists){
```

```
        localStorage.removeItem(storeName);
```

```
    }
```

```
}
```

```
function addToLocalStorage(args){
```

```
    let storeName = args[0];
```

```

let value = args[1];
let alreadyFiltered = args[2];
let numeric = args[3];
let arrayFromLocalStorage = getFromLocalStorage([storeName]);
if(numeric) value = parseInt(value);
if(alreadyFiltered){
    addAlreadyFiltered();
}
else{
    addIfNotThereElseRemove();
}
//Removes items if clicked twice
function addIfNotThereElseRemove(){
    if(arrayFromLocalStorage == null){
        let arrayOfFavorites = new Array();
        arrayOfFavorites.push(value);
        saveToLocalStorage(arrayOfFavorites, [storeName]);
        return;
    }
    if(arrayFromLocalStorage.includes(value)){
        arrayFromLocalStorage = arrayFromLocalStorage.filter(x => x !== value);
        if(arrayFromLocalStorage.length == 0){
            localStorage.removeItem(storeName);
            return;
        }
        saveToLocalStorage(arrayFromLocalStorage, [storeName]);
        return;
    }
    arrayFromLocalStorage.push(value);
    saveToLocalStorage(arrayFromLocalStorage, [storeName]);
    return;
}
//Doesn't remove items, doesn't add copies
function addAlreadyFiltered(){
    if(arrayFromLocalStorage == null){
        let arrayOfFavorites = new Array();
        arrayOfFavorites.push(value);
        saveToLocalStorage(arrayOfFavorites, [storeName]);
        return;
    }
    if(arrayFromLocalStorage.includes(value)){
        return;
    }
    arrayFromLocalStorage.push(value);
    saveToLocalStorage(arrayFromLocalStorage, [storeName]);
}

```

```

        return;
    }
}

function initializeBooks(data){
    books = data;
    popularBooks = sortBooksByPopularity([books, "desc"]);
    if(mainPage){
        let timeToLoad = 2500;
        $("#moveLeftButton").click(function(event){
            event.preventDefault();
            moveBooks(popularBooks, popularHolder, -1);
        });
        $("#moveRightButton").click(function(event){
            event.preventDefault();
            moveBooks(popularBooks, popularHolder, 1);
        })
        let popularHolder = document.querySelector("#pop")
        generateBooks([popularBooks, popularHolder, false, false]);
        recentHolder = document.querySelector("#rec")
        let copyOfBooks = [...books];
        sortedByDate = books.sort(function(a ,b){
            if(a.releaseDate > b.releaseDate){
                return -1
            }
            else if (a.releaseDate < b.releaseDate){
                return 1;
            }
            else return 0;
        })
        books = copyOfBooks;
        let numberOfLentBooks = 0;
        books.forEach(e1 => numberOfLentBooks += e1.timeLoaned);
        generateBooks([sortedByDate, recentHolder, false, false]);
        countTo(document.querySelector("#titNum"), 0, books.length, timeToLoad);
        countTo(document.querySelector("#memNum"), 0, 75, timeToLoad);
        countTo(document.querySelector("#yrNum"), 0, currentYear - 1930,
timeToLoad);
        countTo(document.querySelector("#leNum"), 0, numberOfLentBooks,
timeToLoad);
        countTo(document.querySelector("#leNum"), numberOfLentBooks, 1000,
7000000);
    }
}

```



```

if(sPage == "knjige.html"){
    let localSort = getFromLocalStorage(["sacuvanSort"]);
    if(localSort) {
        currentSort = localSort[0];
    };
    const queryString = window.location.search;
    const urlParams = new URLSearchParams(queryString);
    let resetFilterButton = document.querySelector("#resetFilterButton");
    resetFilterButton.addEventListener("click", function () {
        currentPage = 1;
        clearAllFilters();
        loadMore();
    });
    let textSearch = document.querySelector("#textSearch");
    let savedTextSearch = localStorage.getItem("savedText");
    if(savedTextSearch){
        textSearch.text = savedTextSearch;
    }
    let sortHolder = document.querySelector("#sortHolder");
    textSearch.addEventListener("input", function(){
        currentPage = 1;
        if(textSearch.value.length > 0){
            localStorage.setItem("savedText", textSearch.value);
        }
        else{
            localStorage.removeItem("savedText");
        }
        loadMore();
    })
    showSortOptions([sortHolder, sortOptions])
    let urlFilter = false;
    //If the category paramater is set, filter the books by category, else
    show all books
    if(urlParams.has('kategorija')){
        urlSearch(["Kategorije", "kategorija"]);
        return;
    }
    if(urlParams.has('autor')){
        urlSearch(["Autori", "autor"]);
        return;
    }
    if(urlParams.has('godina')){
        urlSearch(["Godine", "godina"]);
        return;
    }
}

```

```

    if(!urlFilter){
        generateAllFilters([searchBoxes, books]);
        loadMore();
    }
    function clearAllFilters(){
        for(let sb of searchBoxes){
            localStorage.removeItem(sb.title);
        }
        localStorage.removeItem("savedText");
        generateAllFilters([searchBoxes, books]);
    }
    function urlSearch(args){
        let propertyToCheck = args[0];
        let urlParamName = args[1];
        urlFilter = true;
        let propValue = urlParams.get(urlParamName);
        clearAllFilters();
        automaticallyCheckValue([propertyToCheck, propValue]);
        loadMore();
    }
}
if(sPage == "knjiga.html"){
    const queryString = window.location.search;
    const urlParams = new URLSearchParams(queryString);
    const urlBook = urlParams.get('knjiga')
    let currentBook = books.filter(book => book.name === urlBook)[0]
    if(!currentBook)window.location.href =
"https://techbabette.github.io/library-site/pages/knjige.html";

document.querySelector("#bookImage").src=`../imgs/${currentBook.name.toLowerCase()}.jpg`
    document.title = currentBook.name.replaceAll("_", " ");
    document.querySelector("#book-title").innerHTML =
currentBook.name.replaceAll("_", " ");
    document.querySelector('#book-description').innerHTML =
currentBook.description;
    var iconHolder = document.querySelector(".mk-favorite-icon-holder");
    iconHolder.dataset.id = currentBook.id;
    iconHolder.addEventListener("click", function(event){
        event.preventDefault();
        eventAddToFavorites(iconHolder);
    });
    var favorite;
    var favorites = getFromLocalStorage(["favoriti"]);
    if(favorites != null){

```

```

        favorite = favorites.includes(currentBook.id);
    }
    document.querySelector(".mk-favorite-icon").dataset.icon = favorite ?
"mdi:cards-heart" : "mdi:cards-heart-outline";
    setLinkValue("#author-link", "autor", currentBook.author.name,
currentBook.author.name.replaceAll("_", " "));
    setLinkValue("#category-link", "kategorija", currentBook.category.name,
currentBook.category.name.replaceAll("_", " "))
    setLinkValue("#date-link", "godina", currentBook.releaseDate,
negativeToBCE(currentBook.releaseDate));
    document.querySelector("#availability-field").innerHTML = "Broj kopija: "
+currentBook.copies;
    let relatedHolder = document.querySelector("#relatedBooks");
    relatedHolder.style.visibility = currentBook.relatedBooks.length > 0;
    let relatedBookHolder = document.querySelector("#event-div");
    document.querySelector("#loaned-field").innerText = `Pozajmljena
${currentBook.timeLoaned} ${currentBook.timeLoaned != 1 ? "puta" : "put"}`
    if(currentBook.relatedBooks.length > 0){
        let bookList = findBooksFromId([currentBook.relatedBooks]);
        generateBooks([bookList, relatedBookHolder, false, false]);
    }
    else{
        relatedHolder.classList.add("hidden");
    }
}
if(sPage == "favoriti.html"){
    favoritesOnly = true;
    loadMore(false);
}
}

function automaticallyCheckValue(args){
    let search = args[0];
    let value = args[1];
    let elements = document.getElementsByName(search)
    for(let ele of elements){
        if(ele.id.toLowerCase() == "s" + value.toLowerCase()){
            ele.checked = true;
            return;
        }
    }
}
}

```

```

function findBooksFromId(args){
  let arrayOfIds = args[0];
  let bookList = new Array();
  bookList = books.filter(b => arrayOfIds.includes(b.id));
  return bookList;
}

function sortBooksByPopularity(args){
  let arrayOfBooks = args[0];
  let direction = args[1];
  let helperVariable = -1;
  if (direction == "asc") helperVariable = 1;
  let booksToReturn = arrayOfBooks.sort((a,b) => {
    return (a.timeLoaned - b.timeLoaned) * helperVariable;
  })
  return booksToReturn;
}

```

3.3.2 Strani JQuery plugin jquery.zoom.js modifikovan da dodeljuje klase umesto style atributa gde moguće

```

/*!
  Zoom 1.7.21
  license: MIT
  http://www.jacklmoore.com/zoom
*/
(function ($) {
  var defaults = {
    url: false,
    callback: false,
    target: false,
    duration: 120,
    on: 'mouseover', // other options: grab, click, toggle
    touch: true, // enables a touch fallback
    onZoomIn: false,
    onZoomOut: false,
    magnify: 1
  };

  // Core Zoom Logic, independent of event listeners.
  $.zoom = function(target, source, img, magnify) {
    var targetHeight,
        targetWidth,
        sourceHeight,

```

```

        sourceWidth,
        xRatio,
        yRatio,
        offset,
        $target = $(target),
        position = $target.css('position'),
        $source = $(source);

        // The parent element needs positioning so that the zoomed element can
        be correctly positioned within.
        target.style.position = /(absolute|fixed)/.test(position) ? position :
        'relative';
        target.style.overflow = 'hidden';
        img.style.width = img.style.height = '';

        $(img)
            .addClass('zoomImg')
            .css({
                position: 'absolute',
                top: 0,
                left: 0,
                opacity: 0,
                width: img.width * magnify,
                height: img.height * magnify,
                border: 'none',
                maxWidth: 'none',
                maxHeight: 'none'
            })
            .appendTo(target);

        return {
            init: function() {
                targetWidth = $target.outerWidth();
                targetHeight = $target.outerHeight();

                if (source === target) {
                    sourceWidth = targetWidth;
                    sourceHeight = targetHeight;
                } else {
                    sourceWidth = $source.outerWidth();
                    sourceHeight = $source.outerHeight();
                }
            }
        };
    }
}

```

```

    }

    xRatio = (img.width - targetWidth) / sourceWidth;
    yRatio = (img.height - targetHeight) / sourceHeight;

    offset = $source.offset();
  },
  move: function (e) {
    var left = (e.pageX - offset.left),
        top = (e.pageY - offset.top);

    top = Math.max(Math.min(top, sourceHeight), 0);
    left = Math.max(Math.min(left, sourceWidth), 0);

    img.style.left = (left * -xRatio) + 'px';
    img.style.top = (top * -yRatio) + 'px';
  }
};

$.fn.zoom = function (options) {
  return this.each(function () {
    var
      settings = $.extend({}, defaults, options || {}),
      //target will display the zoomed image
      target = settings.target && $(settings.target)[0] || this,
      //source will provide zoom location info (thumbnail)
      source = this,
      $source = $(source),
      img = document.createElement('img'),
      $img = $(img),
      mousemove = 'mousemove.zoom',
      clicked = false,
      touched = false;

    // If a url wasn't specified, look for an image element.
    if (!settings.url) {
      var srcElement = source.querySelector('img');
      if (srcElement) {

```

```

        settings.url = srcElement.getAttribute('data-src') ||
srcElement.currentSrc || srcElement.src;
    }
    if (!settings.url) {
        return;
    }
}

$source.one('zoom.destroy', function(position, overflow){
    $source.off(".zoom");
    target.style.position = position;
    target.style.overflow = overflow;
    img.onload = null;
    $img.remove();
}).bind(this, target.style.position, target.style.overflow));

img.onload = function () {
    var zoom = $.zoom(target, source, img, settings.magnify);

    function start(e) {
        zoom.init();
        zoom.move(e);

        // Skip the fade-in for IE8 and lower since it chokes on
fading-in
        // and changing position based on mousemovement at the same
time.

        $img.stop()
            .fadeTo($.support.opacity ? settings.duration : 0, 1,
$.isFunction(settings.onZoomIn) ? settings.onZoomIn.call(img) : false);
    }

    function stop() {
        $img.stop()
            .fadeTo(settings.duration, 0,
$.isFunction(settings.onZoomOut) ? settings.onZoomOut.call(img) : false);
    }

    // Mouse events

```

```

        if (settings.on === 'grab') {
            $source
                .on('mousedown.zoom',
                    function (e) {
                        if (e.which === 1) {
                            $(document).one('mouseup.zoom',
                                function () {
                                    stop();

                                    $(document).off(mousemove,
                                        zoom.move);

                                }
                            );

                            start(e);

                            $(document).on(mousemove, zoom.move);

                            e.preventDefault();
                        }
                    }
                );
        } else if (settings.on === 'click') {
            $source.on('click.zoom',
                function (e) {
                    if (clicked) {
                        // bubble the event up to the document to
                        trigger the unbind.

                        return;
                    } else {
                        clicked = true;
                        start(e);
                        $(document).on(mousemove, zoom.move);
                        $(document).one('click.zoom',
                            function () {
                                stop();
                                clicked = false;
                                $(document).off(mousemove, zoom.move);
                            }
                        );
                        return false;
                    }
                }
            );
        }
    }

```



```

    }
  }
);
} else if (settings.on === 'toggle') {
  $source.on('click.zoom',
    function (e) {
      if (clicked) {
        stop();
      } else {
        start(e);
      }
      clicked = !clicked;
    }
  );
} else if (settings.on === 'mouseover') {
  zoom.init(); // Preemptively call init because IE7 will
  fire the mousemove handler before the hover handler.

  $source
    .on('mouseenter.zoom', start)
    .on('mouseleave.zoom', stop)
    .on(mousemove, zoom.move);
}

// Touch fallback
if (settings.touch) {
  $source
    .on('touchstart.zoom', function (e) {
      e.preventDefault();
      if (touched) {
        touched = false;
        stop();
      } else {
        touched = true;
        start( e.originalEvent.touches[0] ||
e.originalEvent.changedTouches[0] );
      }
    })
    .on('touchmove.zoom', function (e) {
      e.preventDefault();
      zoom.move( e.originalEvent.touches[0] ||
e.originalEvent.changedTouches[0] );
    })

```

```

        .on('touchend.zoom', function (e) {
            e.preventDefault();
            if (touched) {
                touched = false;
                stop();
            }
        });
    }

    if ($.isFunction(settings.callback)) {
        settings.callback.call(img);
    }
};

img.setAttribute('role', 'presentation');
img.alt = '';
img.src = settings.url;
});
};

$.fn.zoom.defaults = defaults;
}(window.jQuery));

```

3..4 JSON

3.4.1 books.json

```

[
  {
    "id" : 1,
    "name":"Manipulacija_i_moć",
    "category":{
      "id" : 1,
      "name": "Popularna_psihologija"
    },
    "author":{
      "id" : 1,
      "name" : "Henrik_Feksevs"
    },
    "description":"Želite da odmah primetite da vas neko laže? Flertujete
na pravi način? Prodajte svoju priču svima? Naravno, reći ćete, ko ne želi da
posедуje ove veštine? Henrik Feksevs je fascinantna ličnost. Neki bi ga mogli

```

opisati kao pomalo luckastog ekscentrika, ali ako pročitate njegovu knjigu, otkrićete da on savršeno dobro zna o čemu priča. A uz to je i duhovit. Verovatno se pitate da li je ova knjiga praktični vodič za iluzioniste? Ili možda priručnik za mađioničare početnike? Ili čak neki ezoterijski tekst? Ne, ova knjiga nije ništa od toga. Ona je namenjena svima koji žele da čitaju misli, bez potrebe da budu čarobnjaci. Položaj tela, intonacija govora, korak, pogled i pokreti otkrivaju naša osećanja. Često će ti neverbalni signali biti u direktnom sukobu sa porukom koju izražavamo rečima. ",

```
    "copies":5,
    "releaseDate":2022,
    "relatedBooks" : [
        4, 5, 6, 3
    ],
    "timeLoaned" : 5
},
{
    "id" : 2,
    "name":"Francuski_jezik",
    "category":{
        "id" : 2,
        "name": "Jezici"
    },
    "author":{
        "id" : 2,
        "name" : "Biljana_Aksentijević"
    },
    "description":"Udžbenik iz francuskog",
    "copies":2,
    "releaseDate":2021,
    "relatedBooks" : [
        10, 11, 6, 5
    ],
    "timeLoaned" : 5
},
{
    "id" : 3,
    "name":"Prosvećeni_svet",
    "category":{
        "id" : 3,
        "name": "Popularna_nauka"
    },
    "author":{
        "id" : 4,
        "name" : "Steven_Pinker"
    },
}
```

"description": "U ovoj elegantnoj proceni stanja čovečanstva na početku trećeg milenijuma, kognitivni psiholog, intelektualac i mislilac Stiven Pinker poziva nas da se odaljimo od zastrašujućih naslova kojima smo zasuti i crnih proročanstava „stručnjaka“, koji igraju na kartu naših psihološki uslovljenih predrasuda. Umesto toga, on predlaže da jednostavno pratimo podatke. Pinker pokazuje da su život, zdravlje, prosperitet, sigurnost, mir, znanje i sreća u porastu, ne samo na Zapadu, već u čitavom svetu. Taj napredak nije rezultat neke kosmičke sile. To je dar prosvetiteljstva, uverenja da razum i nauka mogu da dovedu do procvata čovečanstva.",

```
    "copies": 2,
    "releaseDate": 2019,
    "relatedBooks" : [
        8, 7, 6, 4
    ],
    "timeLoaned" : 6
},
{
    "id" : 4,
    "name": "Homo_Deus",
    "category": {
        "id" : 3,
        "name": "Popularna_nauka"
    },
    "author": {
        "id" : 5,
        "name" : "Juval_Noah_Harari"
    },
    "description": "Šta će sada biti prioritetni problemi čovečanstva umesto gladi, bolesti i rata? Kakvu ćemo sudbinu nameniti ljudskom rodu, kao samoproklamovani bogovi planete Zemlje, koje korake ćemo preduzimati da to ostvarimo? Homo Deus istražuje projekte, snove i noćne more koji će oblikovati dvadeset prvi vek – od prevazilaženja smrti do stvaranja veštačkog života. Pred nama su osnovna pitanja na koja moramo da damo odgovore: u kom pravcu ćemo se dalje kretati? I kako ćemo zaštititi ovaj krhki svet od sopstvenih destruktivnih moći? Pred nama je sledeća faza evolucije: Homo Deus.",
```

```
    "copies": 3,
    "releaseDate": 2022,
    "relatedBooks" : [
        1, 3, 6, 5
    ],
    "timeLoaned" : 8
},
{
    "id" : 5,
    "name": "Sapijens",
```

```
    "category":{
      "id" : 4,
      "name": "Istorijska_dela"
    },
    "author":{
      "id" : 5,
      "name" : "Juval_Noa_Harari"
    },
    "description":"Pre stotinu hiljada godina na Zemlji je živelo najmanje šest ljudskih vrsta. Danas postoji samo jedna - homo sapijens. Kako je naša vrsta uspjela da pobedi u bici za prevlast? Zašto su naši preci lovci-sakupljači udružili snage da bi gradili gradove i osnivali carstva? Kako smo počeli da verujemo u bogove, nacije i ljudska prava, u novac, knjige i zakone; kako smo pali u ropstvo birokratije, radnog vremena i konzumerizma? Kako će naš svet izgledati u budućnosti?",
    "copies":3,
    "releaseDate":2022,
    "relatedBooks" : [
      9, 1, 2, 3
    ],
    "timeLoaned" : 10
  },
  {
    "id" : 6,
    "name":"Umeće_ratovanja",
    "category":{
      "id" : 4,
      "name": "Istorijska_dela"
    },
    "author":{
      "id" : 6,
      "name" : "Sun_Tzu"
    },
    "description":"Sun Tzuova knjiga Umeće ratovanja, je jedno od najznačajnijih klasičnih kineskih dela.Ova knjiga ne sadrži ni jednu zastarelu maksimu ili nejasno uputstvo. Najbolje je pobediti bez borbe, rekao je Sun Tzu. Za njega je rat bio sastavni deo života.Pažljivo pročitajte ovu knjigu, i sve savremene knjige koje govore o upravljanju državom više vam se neće činiti dostojne pažnje.",
    "copies":3,
    "releaseDate":-500,
    "relatedBooks" : [
      4, 5, 7, 10
    ],
    "timeLoaned" : 10
  }
```

```
    },
    {
      "id" : 7,
      "name":"Intelektom_ispred_svih",
      "category":{
        "id" : 1,
        "name": "Popularna_psihologija"
      },
      "author":{
        "id" : 1,
        "name" : "Henrik_Feksevs"
      },
      "description":"Potreba da ostanete u formi na mentalnom nivou, koja se poslednjih decenija uvukla u kolektivnu svest, dobar je pristup i malim sivim ćelijama.Kao kada je reč o fizičkom zdravlju, postoje svojevrсни metodi za unapređenje mentalnog: vežbe i alatke koje možete koristiti da bi vaše misli postale jače, hitrije i prilagodljivije, što će vam pomoći da ostvarite i održite vrhunski učinak u životu, a koji će vas zaštititi od stresa i teškoća, s kojima ćete se neizostavno susretati.",
      "copies":4,
      "releaseDate":2021,
      "relatedBooks" : [
        4, 5, 6, 3
      ],
      "timeLoaned" : 15
    },
    {
      "id" : 8,
      "name":"Brojevi_ne_lažu",
      "category":{
        "id" : 3,
        "name": "Popularna_nauka"
      },
      "author":{
        "id" : 7,
        "name" : "Vaclav_Smit"
      },
      "description":"Kako da bolje shvatite moderni svet. Moj omiljeni autor. Bez imalo dvoumljenja preporučujem ovu knjigu svima koji vole da saznaju nešto novo.“– Bil Gejts",
      "copies":2,
      "releaseDate":2022,
      "relatedBooks" : [
        4, 5, 6, 9
      ],
    },
  ],
}
```

```
    "timeLoaned" : 2
  },
  {
    "id" : 9,
    "name":"Stari_gradovi_srbije",
    "category":{
      "id" : 3,
      "name": "Popularna_nauka"
    },
    "author":{
      "id" : 8,
      "name" : "Dragan_Bosnić"
    },
    "description":",,Nema grada na svetu oko koga su se jagmili toliki narodi, pod čijim su se bedemima vodile tolike bitke, koji je toliko puta menjao vlasnika i pedeset puta uništavan da bi se svih pedeset puta ponovo iz istorijskog groba podigao – kao što je naš Beograd. Pa zar ta činjenica što je taj grad srpski ni najmanje o Srbima ne govori?“ – Borislav Pekić",
    "copies":2,
    "releaseDate":2019,
    "relatedBooks" : [
      3, 5, 6, 1
    ],
    "timeLoaned" : 4
  },
  {
    "id" : 10,
    "name":"Isus_to_nije_rekao",
    "category":{
      "id" : 4,
      "name": "Istorijska_dela"
    },
    "author":{
      "id" : 9,
      "name" : "Bart_D_Erman"
    },
    "description":"Rafinirano, ali i na jedan krajnje izazovan način, Erman iznosi dokaze o tome kako je puno omiljenih biblijskih priča i široko prihvaćenih verovanja koje se odnose na Isusovu božanstvenost, Trojstvo, božansko poreklo i nadahnutost Biblije, nastalo kao plod namernih i slučajnih izmena načinjenih od strane drevnih pisara - a ove izmene su, pak, dramatično uticale na sve naredne verzije Biblije.",
    "copies":1,
    "releaseDate":2019,
    "relatedBooks" : [
```

```

        4, 5, 6, 9
    ],
    "timeLoaned" : 1
},
{
    "id" : 11,
    "name":"Nemački_za_neupućene",
    "category":{
        "id" : 2,
        "name": "Jezici"
    },
    "author":{
        "id" : 10,
        "name" : "Wendy_Foster"
    },
    "description":"Guten Tag! Ukoliko želite da naučite nemački jezik -
    bilo radi posla, putovanja ili zabave - Nemački za neupućene će zadovoljiti sve
    vaše potrebe. Pored poglavlja koja objašnjavaju gramatiku i njenu primenu,
    knjiga sadrži i dijaloge koji će vam omogućiti da koristite i govorite nemački
    jezik kao maternji. Osim toga, ona sadrži i CD koji pruža dodatne mogućnosti za
    vežbanje govornog jezika. ",
    "copies":1,
    "releaseDate":2018,
    "relatedBooks" : [
        4, 5, 2, 10
    ],
    "timeLoaned" : 2
}
]

```

3.4.2 navbar.json

```

[
    {
        "text" : "Gradska biblioteka",
        "url" : "index.html",
        "primary" : true
    },
    {
        "text" : "Pocetna",
        "url" : "index.html",
        "primary" : true
    },
    {
        "text" : "Knjige",
        "url" : "knjige.html",
    }
]

```



```
    "primary" : false
  },
  {
    "text" : "Favoriti",
    "url" : "favoriti.html",
    "primary" : false
  },
  {
    "text" : "0 autoru",
    "url" : "autor.html",
    "primary" : false
  }
]
```

3.4.3 footer.json

```
[
  {
    "url" : "https://www.facebook.com/",
    "name" : "facebook",
    "icon" : "icomoon-free:facebook",
    "primary" : false
  },
  {
    "url" : "https://www.twitter.com/",
    "name" : "twitter",
    "icon" : "la:twitter",
    "primary" : false
  },
  {
    "url" : "documentation.pdf",
    "name" : "fa-light",
    "icon" : "fa-file",
    "primary" : true
  },
  {
    "url" : "sitemap.xml",
    "name" : "sitemap",
    "icon" : "bx:sitemap",
    "primary" : true
  }
]
```