Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

ReactJS

Domaći zadatak 3

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Žarko | Colić | 253/17 |
| Link ka Github-u | <https://github.com/torrent98/Domaci3React> | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc125226267)

[2 Implementacija 3](#_Toc125226268)

[3 Korisničko uputstvo 11](#_Toc125226269)

# Korisnički zahtev

Za realizaciju aplikacije koristili smo React, sa elementima i predznanjem iz HTML, CSS, JavaScripta. Prethodno smo instalirali Visual Studio Code i Node.js koji su nam bili potrebni za ovaj rad. U Command Prompt-u smo preko opcije cd prešli u prethodno napravljen folder u kome smo želeli da inicijalizujemo projekat, a nakon toga smo ga inicijalizovali preko opcije npx create-react-app fast\_, gde se u folderu fast\_ inicijalizovala početna struktura aplikacije.

Pokrećemo server preko opcije npm start i na portu 3000(ili 3001) nam se otvara početni ekran naše aplikacije, na kom se nalazi samo react logo koji se pokreće.

Aplikacija Fast & Furious Services je aplikacija preko koje je moguće vršenje zakazivanja termina za određene popravke na brzim automobilima. Popravke smo kategorisali u par kategorija I oni su opcija za koju možemo da zakažemo termin u servisu. Postoji NavBar meni preko kojeg korisnik vrši navigaciju sa glavnog ekrana, na ekran mogućih servisa I na zakazane termine.

Postoji I Footer deo gde korisnik ima opciju da unese svoju e-mail adresu kako bi dobijao obaveštenja o bilo kojim promenama koje se dešavaju u servisu. Takođe tu se nalaze ikonice koje vode ka stranicama servisa na društvenim mrežama.

# Implementacija

Na početku smo, na način koji je gore naveden, inicijalizovali početnu strukturu projekta, koja se sastoji iz node-modules foldera (koji kroz rad nismo direktno koristili i menjali), public foldera (u kom su po default-u smeštena 2 react logoa različite veličine-koje takođe nismo korisitli ali ih nismo ni obrisali, kao i index.html stranica koju smo na početku malo izmenili), src folder(u kom su smeštene sve komponente koje smo koristili kroz rad), kao i dva .json fajla koje isto direktno nismo dirali, već ukoliko bi došlo do neke instalacije, došlo bi i do promene neke linije koda ili dodavanje nove linije u ta dva .json fajla.

Struktura se sastoji iz src foldera u kojem se nalaze dva foldera: assets gde su skaldištene sve slike potrebne za aplikaciju i components gde se nalaze funkcionalnosti i ostale stranice za aplikaciju, kao i stilizovanje u vidu css fajlova.

U assets folder smo ubacili sliku koja nam sde nalazila na pozadini main container-a na početnoj stranici.



Nakon toga smo napravili MainSection.js i MainSection.css fajlove. U .js fajlu smo importovali potrebne fajlove, a u samoj funkciji preko return-a vratili main-container, u kome se nalazi početna slika na pozadini, preko koje su naslov h1 i tekst. Na dnu smo taj fajl eksportovali kako bismo ga ubacili u App.js fajl. U .css fajlu smo stilizovali .main-container, kao i njegove h1 i p atribute. Podešena je pozadinska slika, veličina kontejnera, kao i opcije za display. Za h1 i p podešeni su boja, font-family, veličina fonta, margine.

.main-container {

    background: linear-gradient(transparent, black),

      url("../assets/img-home.jpg") center center/cover no-repeat;

    height: 90vh;

    width: 100%;

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

    box-shadow: inset 0 0 0 1000px rgba(0, 0, 0, 0.2);

    object-fit: contain;

  }

  .main-container > h1 {

    color: #fff;

    font-size: 100px;

    margin-top: -40px;

  }

  .main-container > p {

    margin-top: 8px;

    color: #fff;

    font-size: 32px;

    font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",

      "Lucida Sans", Arial, sans-serif;

  }

  .main-text {

    height: 75vh;

    width: 100%;

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

    box-shadow: inset 0 0 0 1000px rgba(0, 0, 0);

    object-fit: contain;

  }

  .main-text > h1 {

    color: rgb(255, 255, 255);

    font-size: 70px;

    margin-top: -20px;

  }

  .main-text > h3 {

    color: rgb(255, 255, 255);

    font-size: 38px;

    margin-top: 20px;

    margin-bottom: 30px;

    font-style: normal;

    font-variant: normal;

    font-weight: 400;

    line-height: 20px;

  }

  .main-text > p {

    margin: 20px 100px;

    color: rgb(255, 255, 255);

    font-size: 22px;

    font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",

      "Lucida Sans", Arial, sans-serif;

    text-align: justify;

    font-style: normal;

    font-variant: normal;

    font-weight: 400;

  }

  .main-clients {

    height: 110vh;

    margin-bottom: -70px;

  }

  .main-btns {

    margin-top: 32px;

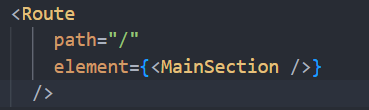
  }

  .main-btns .btn {

    margin: 6px;

  }

Nakon toga prelazimo u App.js fajl, gde importujemo MainSection iz foldera. Pravimo i komponentu MainSection-a.



U App.css fajlu smo podesili osnovna css podešavanja koja se odnose na celu stranicu, kao što su box-sizing, padding, margine i font-family.

\* {

  box-sizing: border-box;

  margin: 0;

  padding: 0;

  font-family: "PT Sans", sans-serif;

}

Nakon toga smo napravili Navbar komponentu. Ona sadrži F&F Services Link ka trenutnoj stranici. Takođe, sadrži ul tag sa dva list-item-a koji su Linkovi ka stranicama Početna i Lokacije koje ćemo kasnije napraviti. U Navbar.css fajlu smo stilizovali bar, uglavnom korišćenjem koda sa bootstrap-a.

import { Link } from 'react-router-dom';

import './Navbar.css';

function Navbar({num}) {

  return (

    <>

      <nav className='navbar'>

        <div className='navbar-container'>

          <Link to='/' className='navbar-logo'>

            F&Fs Services

            <i class="fas fast-furious"></i>

          </Link>

          <ul className='nav-menu'>

            <li className='nav-item'>

              <Link to='/' className='nav-links'>

                Home

              </Link>

            </li>

            <li className='nav-item'>

              <Link

                to='/services'

                className='nav-links'

              >

                Services

              </Link>

            </li>

            <li className='nav-item'>

              <Link

                to='/appointments'

                className='nav-links'

              >

                Appointments

              </Link>

            </li>

            <li className='nav-item'>

              <div>

              <p

                className='nav-links1'

              >

                {num}

              </p>

              </div>

            </li>

          </ul>

        </div>

      </nav>

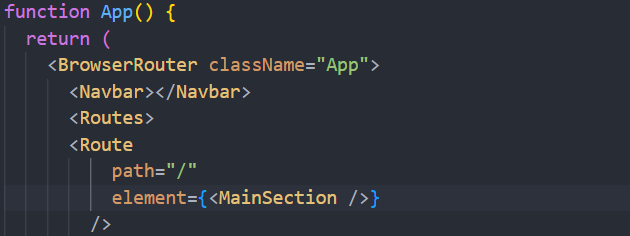
    </>

  );

}

export default Navbar;

Posle smo izmenili App.js fajl. Dodali smo BrowserRouter tag (koji smo prethodno importovali, a pre toga instalirali react-router-dom u konzoli, što je i unelo određene promene u dva .json fajla), u okviru kog smo napravili Routes tag i jedan Route tag u koji smo smestili prethodno kreiranu MainSection komponentu. Iznad Routes taga smo dodali naš Navbar, koji smo prethodno eksportovali iz Navbar.js fajla i importovali u App.js fajl.



Sada podešavamo MainSection i dodajemo kartice na kojima će biti komentari i slike osoba. U assets folder smo dodali 3 slike koje će biti slike korisnika koji su ostavili komentare. Pravimo fajl CardItem.js koji će biti jedna kartica koja će sadržati sliku, ime korisnika i komentar koji je ostavio. Kao parametar šaljemo obj koji je objekat od koga ćemo kasnije uzeti sliku preko .src, ime preko .name i komentar preko .text. Exportujemo taj CardItem, i dodajemo ga u novi fajl Cards.js za koji imamo i .css fajl.

Importujemo i 3 slike koje smo ubacili i imenujemo ih kako bismo ih lakše koristili. u funkciji Cards pravimo niz clients gde svaki od 3 korisnika ima ime, komentar i sliku tj. name,text i src atribute. u return polju vraćamo h1 title i kao untitled listu vraćamo 3 CardItem elementa i redom uzimamo korisnike iz prethodnog niza tako što pristupamo njihovom rednom broju u nizu. Exportujemo Cards fajl, koji je uredjen u css fajlu. U tom fajlu su standardna podešavanja,a ono što je novo je ::after kategorija, koja predstavlja navodnike koji stoje preko slike, kao ikonica za komentar. Prelaskom mišem preko slika zumiraju se, a kada sklonimo miš vrate se na početno stanje. Podešena je i responzivnost.

Uveli smo jednu promenu, iz Cards fajla niz klijenata smo prebacili u MainSection, a u Cards funkciji smo dodali nove parametre, gde data1 i data2 uzimamo preko map funkcije. Tu vidimo i deo realizacije koda preko JavaScript-a.



Kreiramo Home i Locations js fajlove, koji nam predstavljaju naše 2 stranice. Na početku one imaju samo prazan div. Exportujemo ih a zatim importujemo i App.js fajl i dodamo kao 2 nove rute. (Kasnije smo obrisali Home stranicu jer sam sve što je trebalo na njoj da stoji stavila u App.js).

Naredna komponenta koju kreiramo je Button. I za nju imamo i css i js fajl. Tu smo napravili dva niza: stilova i veličina kako bismo ih lakše koristili. Exportujemo dugme kao link koji vodi na neki link koji budemo tražili zavisno od toga gde koristimo dugme. Ime klase se formira kao kombinacija teksta ’btn’ i stila i veličine koja je izabrana, a imamo i funkcije za onClick i sam tip odnosno type tog dugmeta. U css fajlu podešavamo sve veličine za stil i veličine dugmeta, kao i animaciju hover prilikom prelaska mišem preko dugmeta.

import React from 'react';

import { Button } from './Button';

import { useState } from 'react';

function CardItem({name, src, star, checked, type, makeAnAppointment, star1, star2, id, text}) {

  return (

    <>

      <li className='cards\_\_item'>

        <div className='cards\_\_item\_\_link'>

          <figure className='cards\_\_item\_\_pic-wrap' data-category={type=='clients' ? '“' : name}>

            <img

              className='cards\_\_item\_\_img'

              src={src}

            />

          </figure>

          <div className='cards\_\_item\_\_info'>

            <p className='cards\_\_item\_\_text'>{text}</p>

            {type == 'clients' ? <h5 className='cards\_\_item\_\_title'>{name}</h5> : <></> }

            {type == 'clients' ? <></> :

              <div className='star-container'>

                <ul>

                  <li>

                    {checked == false ? <i class="far fa-star" onClick={()=>{star1(id)}}></i> :

                     <i class="fas fa-star" onClick={()=>star2(id)}></i>}

                  </li>

                  <li><p className='star-number'>{star}</p></li>

                  <li>

                    <Button

                      className='btns'

                      buttonStyle='btn--primary'

                      buttonSize='btn--medium'

                      link={'/services'}

                      id = {id}

                      onClick={makeAnAppointment}

                    >

                      MAKE AN APPOINTMENT

                    </Button>

                  </li>

                </ul>

              </div>

             }

          </div>

        </div>

      </li>

    </>

  );

}

//commit greska

export default CardItem;

Sada pravimo Footer fajl. U njega smeštamo paragraf za prijavu na newsletter i paragraf za ostavljanje komentara, kao i dva dugmeta Prijavi se i Pošalji komentar koji su Button komponente. Prvi paragraf je input tipa i tu je placeholder Email jer se upravo on unosi, dok je drugo polje za unos textarea tip kako bi mogao da se proširi ukoliko korisnik želi da unese neki duži komentar. Ovde smo koristili useState hook prilikom definisanja funkcija onClick i onChange. Za obe funkcije onClick izlazi odgovarajući alert, zavisno od toga koje dugme je pritisnuto. Tu vidimo i deo koda realizovan preko JavaScript-a.

import React from 'react';

import './Footer.css';

import { Button } from './Button';

import { Link } from 'react-router-dom';

import { useState } from 'react';

import Swal from 'sweetalert2';

function Footer() {

  const [buttonClick, setButtonClick] = useState('');

  const onClick = () => {

    if(buttonClick != '') {

      setButtonClick('');

      setTimeout(() => {

        //alert('Thank you for subscribing!');

        Swal.fire({

          title: 'Great!',

          text: 'Thank you for subscribing!',

          icon: 'success',

          confirmButtonText: 'OK',

          confirmButtonColor: 'indianred',

        });

      }, 10);

    }

  }

  const onChange = (e) => {

    setButtonClick(e.target.value);

  }

  return (

    <div className='footer-container'>

      <section className='footer-subscription'>

        <p className='footer-subscription-heading'>

          Join the Fast & Furious community and stay up to date!

        </p>

        <p className='footer-subscription-text'>

          You can unsubscribe at any time.

        </p>

        <div className='input-areas'>

          <form>

            <input

              className='footer-input'

              name='email'

              type='email'

              placeholder='Your Email'

              value={buttonClick}

              onChange={onChange}

            />

            <Button buttonStyle='btn--outline' onClick={onClick} link={'#'} >Subscribe</Button>

          </form>

        </div>

      </section>

      <section class='social-media'>

        <div class='social-media-wrap'>

          <div class='footer-logo'>

            <Link to='/' className='social-logo'>

              F&F Services

              <i class="fas fast-furious"></i>

            </Link>

          </div>

          <small class='website-rights'>F&F Services © 2023</small>

          <div class='social-icons'>

            <a

              class='social-icon-link facebook'

              href='https://www.facebook.com/fastandfuriousuk/'

              target='\_blank'

              aria-label='Facebook'

            >

              <i class='fab fa-facebook-f' />

            </a>

            <a

              class='social-icon-link instagram'

              href='https://www.instagram.com/thefastsaga/?hl=en'

              target='\_blank'

              aria-label='Instagram'

            >

              <i class='fab fa-instagram' />

            </a>

            <a

              class='social-icon-link youtube'

              href='https://www.youtube.com/user/fastandfuriousmovie?app=desktop'

              target='\_blank'

              aria-label='Youtube'

            >

              <i class='fab fa-youtube' />

            </a>

            <a

              class='social-icon-link twitter'

              href='https://twitter.com/TheFastSaga?ref\_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor'

              target='\_blank'

              aria-label='Twitter'

            >

              <i class='fab fa-twitter' />

            </a>

            <a

              class='social-icon-link linkedin'

              href='https://www.linkedin.com/company/fast-and-furious-inc/'

              target='\_blank'

              aria-label='LinkedIn'

            >

              <i class='fab fa-linkedin' />

            </a>

          </div>

        </div>

      </section>

    </div>

  );

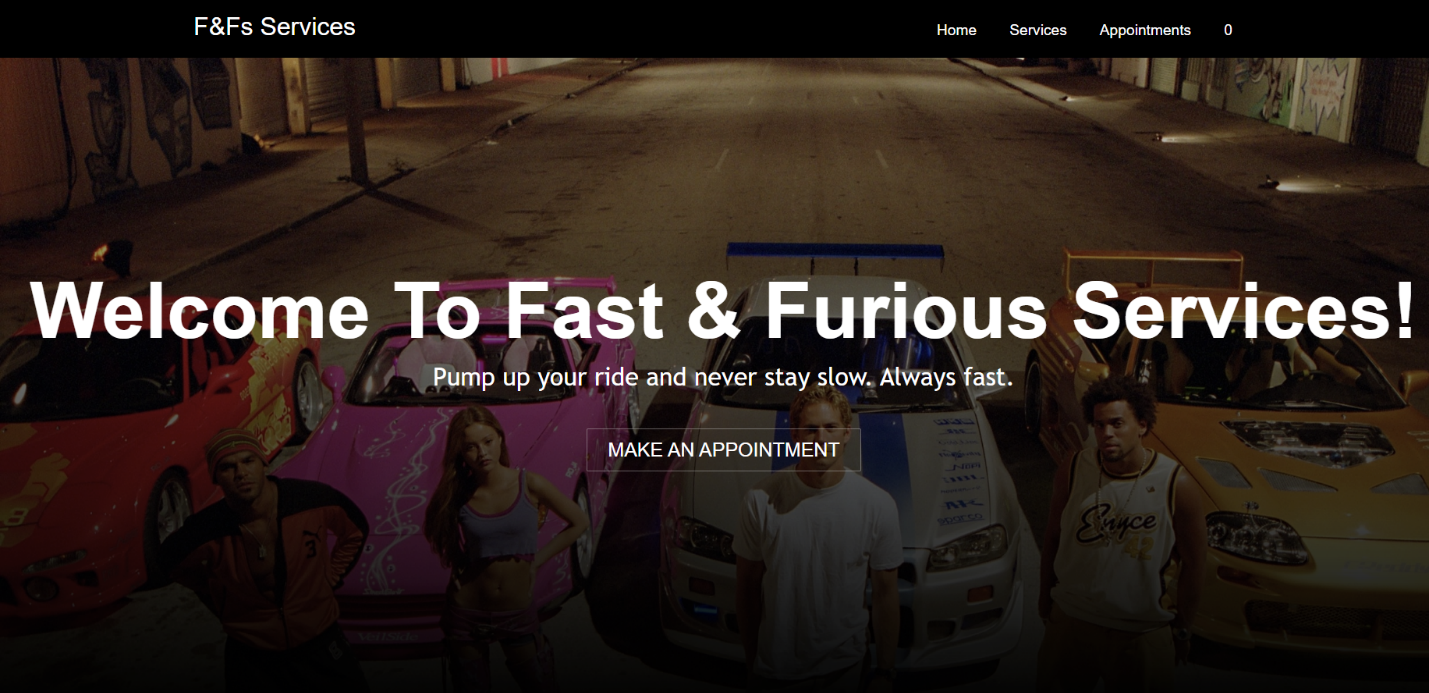
}

export default Footer;

# Korisničko uputstvo

Opis i screenshot-ovi aplikacije

Početna stranica izgleda ovako:



Na vrhu vidimo navbar, koji služi kao putanja od jedne stranice do druge. To je jedna od dve reusable komponente koja se koristi na obe stranice tj više puta kroz sajt.

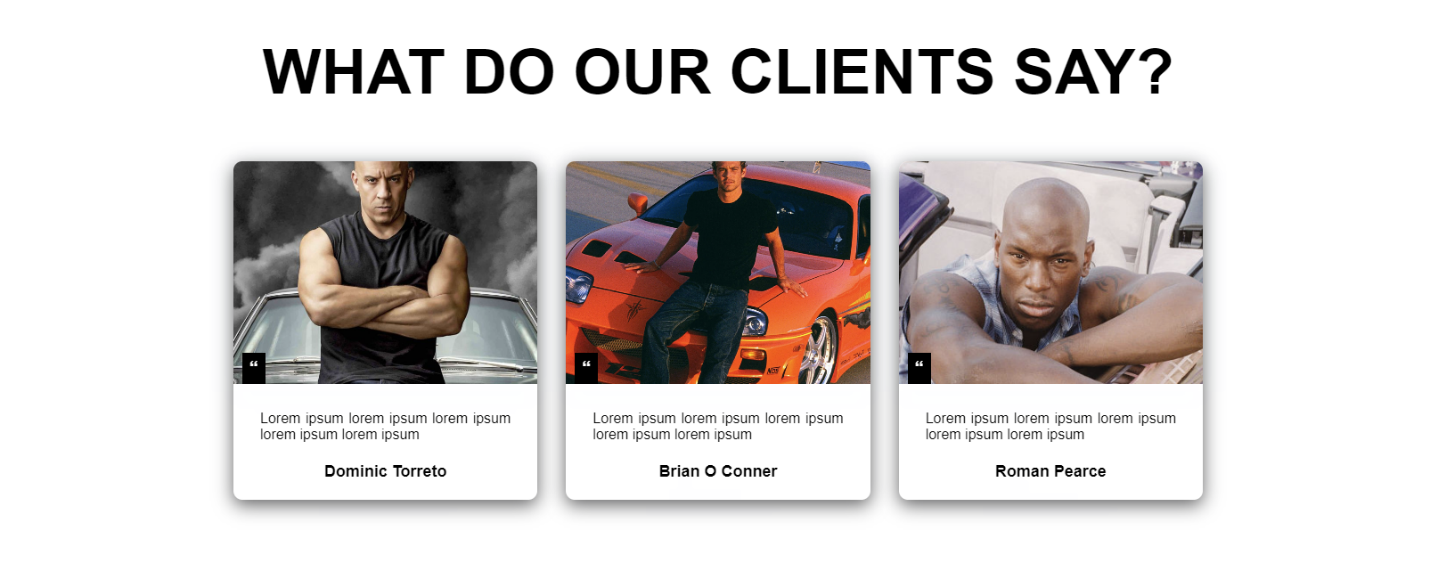
Ispod vidimo MainSection komponentu koja sadrži pozadinsku sliku, h1 naslov i paragraf.

Kada skrolujemo naniže, vidimo ostatak stranice:

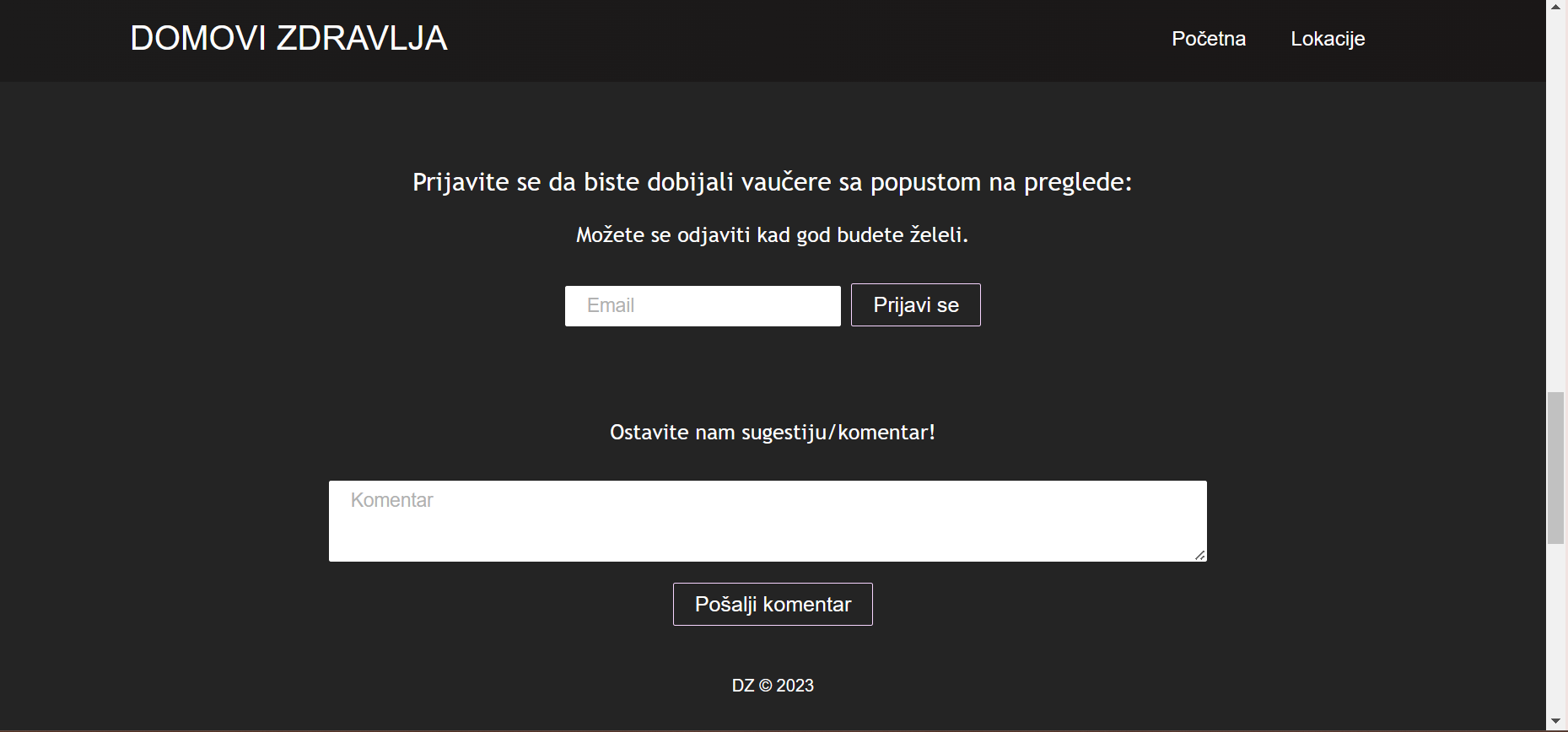


Tu takođe vidimo deo komponente MainSection. Vidimo h1 i h3 naslove, kao i paragrafe ispod njih. Sve je uređeno u MainSection.css fajlu.

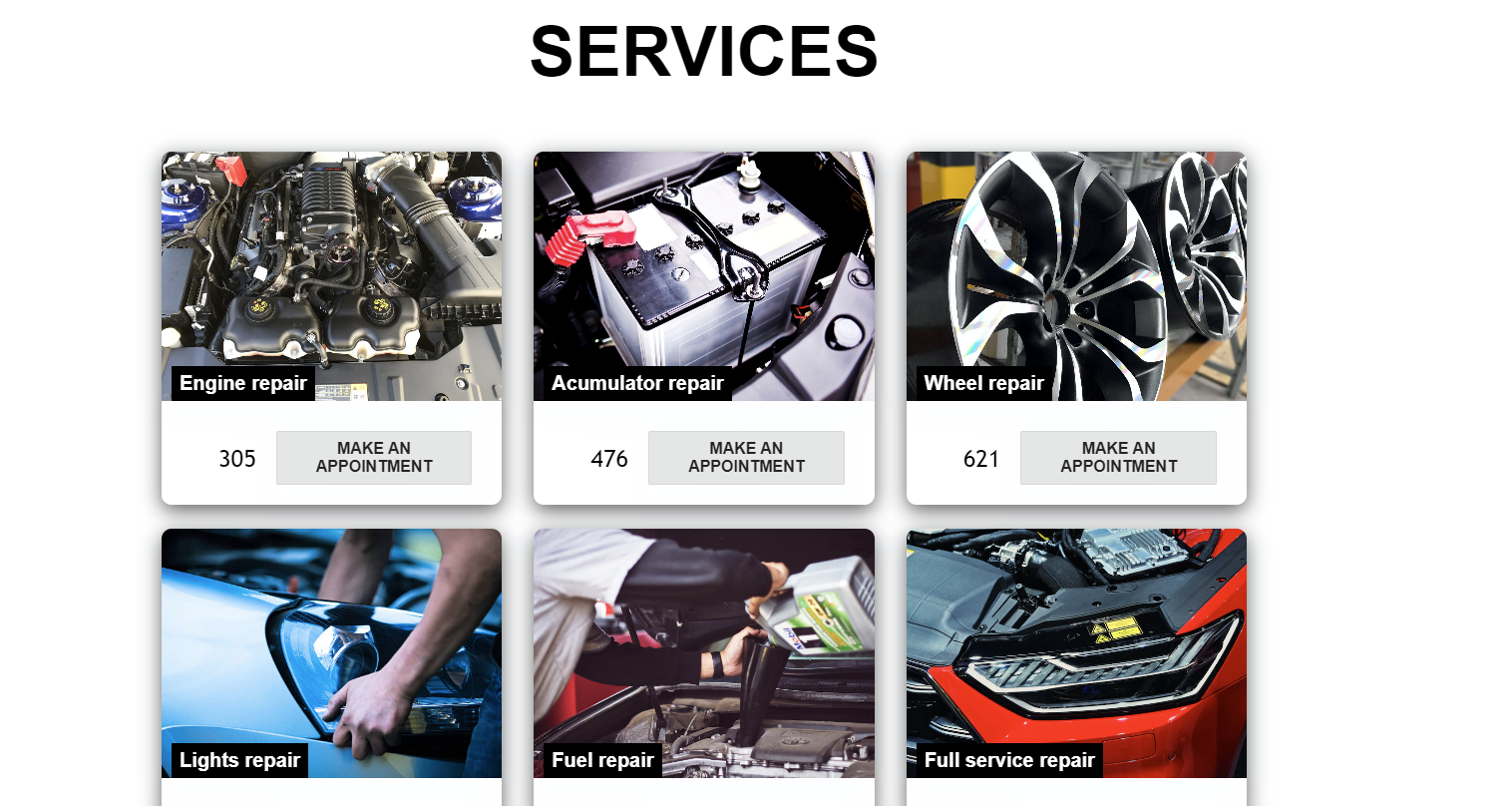
Ispod ovoga vidimo i komentare klijenata:



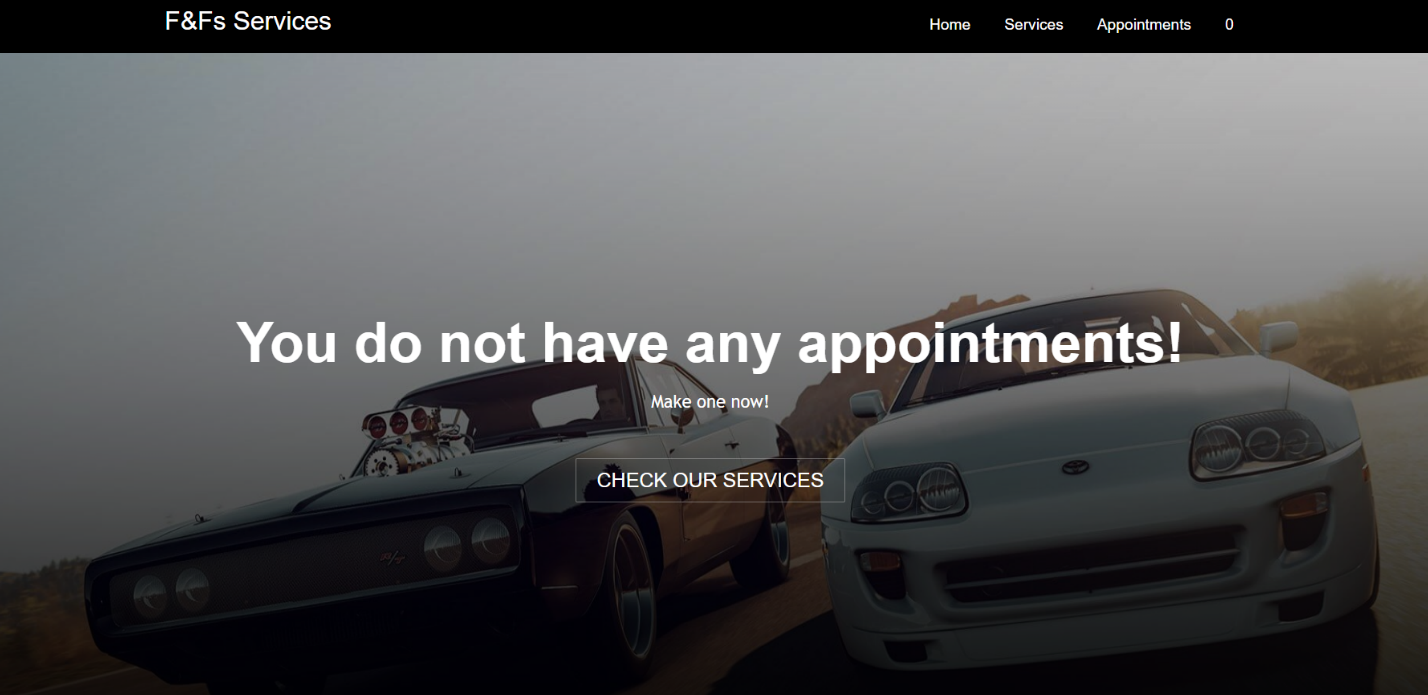
Oni su takođe deo MainSection fajla, implementiran preko Cards tag-a, koji sadrži u sebi CardItem. Prelaskom mišem preko slike, zumira se, a kada se miš skloni sa slike, slika se vraća na svoje mesto.

Na dnu stranice nalazi se footer sa dva polja za unos i dugmićima:

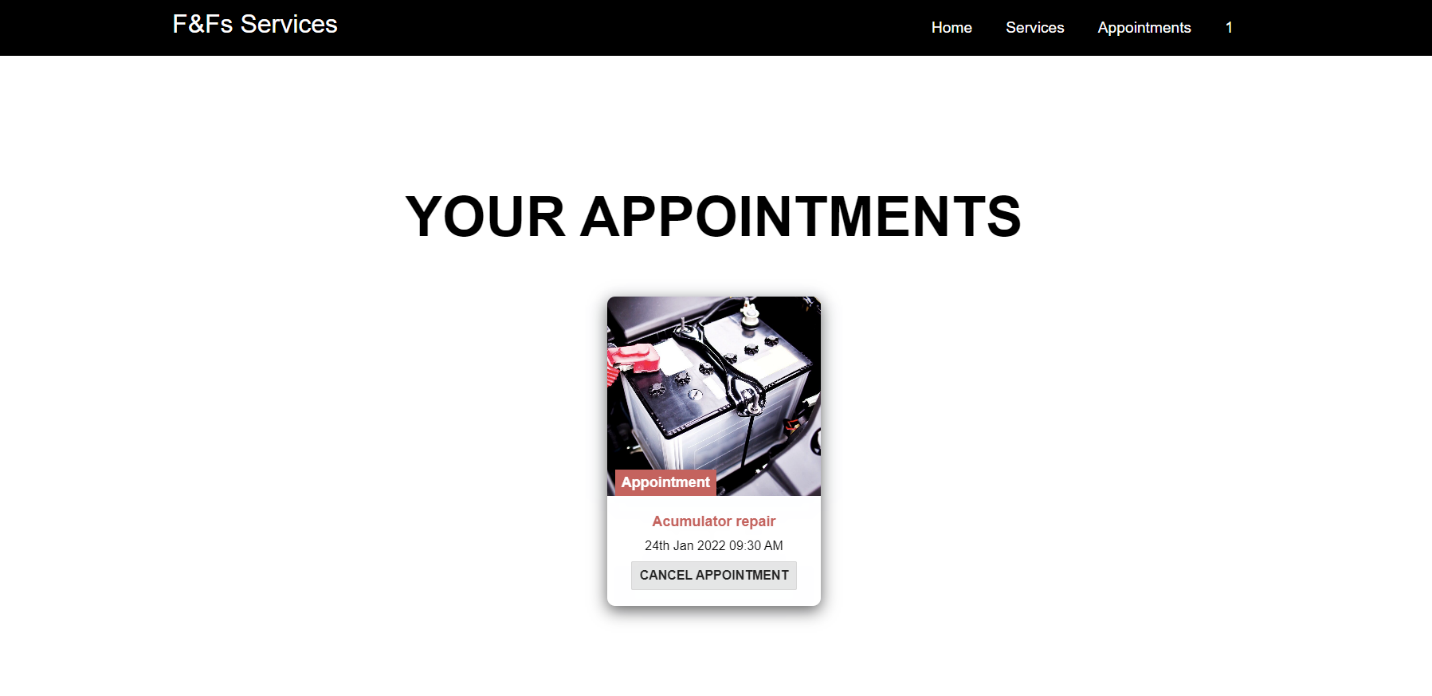
Prelazimo na stranicu sa servisima gde možemo videti moguće usluge tj servise koje možemo kao korisnik dobiti u Fast and Furious Services. Svi Servisi su raspoređeni kao kartice i sadrže određene podatke kao na slici (broj lajkova, dugme za dodavanje termina određenog servisa i naziv servisa kao i sliku).



Poslendja stranica je stranica za termine. Na njoj se nalaze svi termini koje smo zakazali. U slučaju da nema nijednog, prikazuje nam se tekst sa određenom porukom i dugme za prelazak na stranicu sa servisima.



U slučaju da smo zakazali određeni termin za neki servis, on će biti prikazan na stranici sa određenim informacijama, i dugmetom CANCEL u slučaju da želimo da otkažemo zakazan termin.



Na dnu ponovo vidimo Footer koji smo videli i na prethodnoj stranici. 