

Dokumentacija

Naš sajt predstavlja restoran Holycow! Steakhouse. Posetioци mogu da pogledaju ponudu jela, kao i da se informišu o lokacijama na kojima se nalaze objekti, rezervišu sto, ostave utisak i saznaju više o samom preduzeću.

Veb sajt se sastoji od 4 stranice kojima se može pristupiti klikom na jedno od sledećih dugmića u navigacionom baru glavnog menija:

1. Početna stranica
2. Jelovnik
3. Kontakt i rezervacije
4. O nama

„Početna stranica“ predstavlja kratak rezime našeg poslovnog koncepta, a sadrži i sliku unutrašnjosti našeg restorana, kao i pojednostavljeni jelovnik sa cenama. Kod pojednostavljenog jelovnika nalazi se i link ka celokupnom jelovniku, tj. stranici „Jelovnik“.

Na stranici „Jelovnik“ može se videti ponuda našeg restorana Holycow!. Klikom na saznaj više pojaviće se ime jela. Ovo je omogućeno u javascript fajlu koji je povezan sa html funkcijom toggle. Prva četiri elementa su raspoređena korišćenjem Flexbox-a. Svi elementi su responzivni zahvaljujući css fajlu media.css koji je povezan sa njim.

Na stranici „Kontakt i rezervacije“ nalaze se dve forme. Prva omogućava korisniku da unese svoje podatke i preference i rezerviše svoj sto u jednom od naših restorana. Druga forma služi za

feedback od posetilaca; popunjavanjem forme im je omogućeno da ostave svoje utiske i predloge. Elementi ove stranice raspoređeni su uz pomoć inline grid layout-a. Funkcionalnosti ovih formi obezbeđena je uz pomoć JavaScript-a i JQuery-ja, a nešto više o tome nalazi se u odeljku „JavaScript“.

Na stranici „O nama“ korisnik može da sazna nešto više o našim podacima i zanimljivostima u vezi sa restoranom. Takođe na ovoj stranici nalazi se odbrojavanje do otvaranja našeg novog objekta. Ovo je napravljeno uz pomoć flipdown-a (sata koji odbrojava sekunde, stilizovanog tako da se kartice sa brojevima smenjuju dok otkucava).

Na svim stranicama se nalazi meni i futer.

Meni je napravavljen kao obična lista (ul), a elementi liste su u stvari stranice kojima možemo da pristupimo preko menija (li sa klasom nav-item). Takođe, u meniju postoji i jedan element liste koji ima klasu “toggle”. Ovaj element treba da bude vidljiv samo kada se ekran smanji na manje od 650px(prikaz za mobilne telefone i tablete). Klikom na taj element (hamburger meni) se prikazuje ostatak menija.

Meni je stilizovan korišćenjem FlexBox-a. Prvo je napravljn meni za mobilne telefone. Kod takvog menija je potrebno da svi elementi budu pozicionirani jedan ispod drugog. Zbog toga je za svaki element sa klasom “nav-item” podešeno width:100%. Pošto ne želimo da nam stavke menija budu prikazane sve vreme na mobilnom telefonu, ovi elementi će imati svojstvo: “display:none”. To znači da će u mobilnom prikazu biti prikazan samo logo i hamburger meni(element sa klasom “toggle”).

Kada korisnik klikne na hamburger meni, otvara mu se pregled svih stavki menija, prikazane jedna ispod druge. Ovo je omogućeno uz pomoć jQuery-ja:

```
$(function() {  
    $(".toggle").on("click", function() {  
        if ($(".nav-item").hasClass("active")) {  
            $(".nav-item").removeClass("active");  
        } else {  
            $(".nav-item").addClass("active");  
        }  
    });  
});
```

U trenutku kada korisnik klikne na hamburger meni (to je element sa klasom “toggle”), za sve elemente liste(elementi sa klasom nav-item) koja se nalazi u meniju se proverava da li imaju klasu “active”. Ukoliko imaju tu klasu to znači da su one trenutno vidljive na ekranu. To je omogućeno u CSS-u sledećim kodom:

```
.nav-item.active{  
    display:block;  
}
```

U tom slučaju je uz pomoć jQuery-ja samo obrišemo tu klasu, pa ti elementi više neće ni biti vidljivi.

Ukoliko ovi elementi nisu imali klasu “active”, potrebno je da im dodelimo tu klasu kako bismo mogli da ih vidimo na ekranu.

Uz pomoć Media Query-ja je omogućena responzivnost menija. Takođe na taj način je obezbeđeno da se hamburger meni (sa klasom “toggle”) ne vidi na ekranu većem od 650px.

Futer je napravljen korišćenjem GridLayout-a. U futeru se nalaze 3 kolone, čija je širina određena automatski (grid-template-columns: auto auto auto;), a razmak između kolona je 10px (grid-gap: 10px;).

U prvoj koloni futera se nalaze podaci o kontaktu (broj telefona I email). Druga kolona je rezervisana za linkove ka društvenim mrežama. To je omogućeno uz pomoć font awesome ikona. U trećoj koloni se nalaze linkovi ka LinkedIn nalogima članova tima koji su učestvovali u izradi domaćeg zadatka.

Responzivnost futera je obezbeđena uz pomoć Media Query-ja. Ako je prozor manji od 800px, sve tri kolone futera će se poredati jedna ispod druge, u jednu kolonu.

A sada malo o dizajnu našeg sajta:

CSS:

Pozadina svih stranica je stilizovana linerarnim gradijentom:

```
background-image: linear-gradient(#bb383c, lightgray);
```

Sa media query-jem, prilikom smanjenja ekrana na širinu manju od 500px, slike i divovi u kojima se nalaze ispunili bi 90% ekrana a font teksta bio bi smanjen (heading 1 na 20px, ostatak teksta na 15px). Leve margine bile bi izbrisane. Flex-direction prva četiri elementa bio bi kolona. Na stranici Kontakt, formama je dato svojstvo da ispod 1200px širine ekrana leva margina bude postavljena na 0, a na 450px font će se smanjiti na 15px, kao i text-area za utiske, na 200px. Korišćen je grid layout, kao i flexbox ponegde.

JavaScript:

Funkcionalnost formi stranice „Kontakt i rezervacije“ obezbeđena je sledećim funkcijama.

```
function funkcija1() {
    let imePrezime;
    let email;
    let utisak;
    //pokupi imeprezime, mejl i utisak iz forme preko id-ija
    imePrezime = document.getElementById("imeiprezime").value;
    email=document.getElementById("email").value;
    utisak = document.getElementById("textBox").value;
    //ako je imePrezime prazan string ili ako nema znak razmaka
    //Ako je prazan string znači da nista nije uneto
    //ako nema znak razmaka znaci da fali ili ime ili prezime
    if (imePrezime === "" || !imePrezime.includes(" ")) {
        alert("Pogrešan unos! Unesi ime i prezime");
        return false;
    }
    if(email=== "" || !email.includes('@')){
        alert("Pogrešan unos! Unesi ispravan email!");
        return false;
    }
    //utisak ne sme da budeprazan string ili da bude duzi od npr 25
    0 karaktera (ovo mozemo i da obrisemo)
    if(utisak=== "" || utisak.length()>250){
        alert("Pogrešan unos! Unesi neki utisak kraći od 250 karakter
a!");
        return false;
    }
}
```

```

function funkcija2() {
    let imePrezime;
    var brojOsoba;
    let datum; //ovo ce biti string datum rezervacije
    let datumRezervacije;
    let danas = new Date(); //danasnji datum
    //pokupi sve podatke iz forme preko id-ija
    imePrezime = document.getElementById("imeiprezime2").value;
    brojOsoba = document.getElementById("broj").value;
    datum = document.getElementById("datum").value; //vraca datum
    u obliku 2020-11-30
    //uzecemo taj datum i podelicemo ga u niz stringova uz pomoc fun
    ckije split
    let elementiDatuma=datum.split("-
"); //ovo ce viti niz stringova u ovom obliku "2020","11","30"
    //ova promenljiva ce biti tipa Date i u nju cemo smestiti podat
    ke iz niza od gore
    //ali pre toga moramo da ih pretvorimo i int jer su oni string
    ("2020"..)
    //datum se setuje redosledom godina, mesec, dan
    //meseci se broje od nule, pa se oduzima 1
    datumRezervacije = new Date(parseInt(elementiDatuma[0]),parseIn
t(elementiDatuma[1]-1),parseInt(elementiDatuma[2]));

    let brojSekundiDoDanaRezervacije = datumRezervacije.getTime();
    let preostaloVreme =brojSekundiDoDanaRezervacije-
danas.getTime();
    //alert(preostaloVreme);
    if (imePrezime === "" || !imePrezime.includes(" ")) {
        alert("Pogrešan unos! Unesi puno ime i prezime");
        return false;
    }
    if (isNaN(brojOsoba) || brojOsoba < 1 || brojOsoba>50) {
        alert("Pogrešan broj osoba!")
        return false;
    }
}

```

```
if(preostaloVreme<0){  
    alert("Odaberite neki datum iz budućnosti!")  
    return false;  
}  
}
```