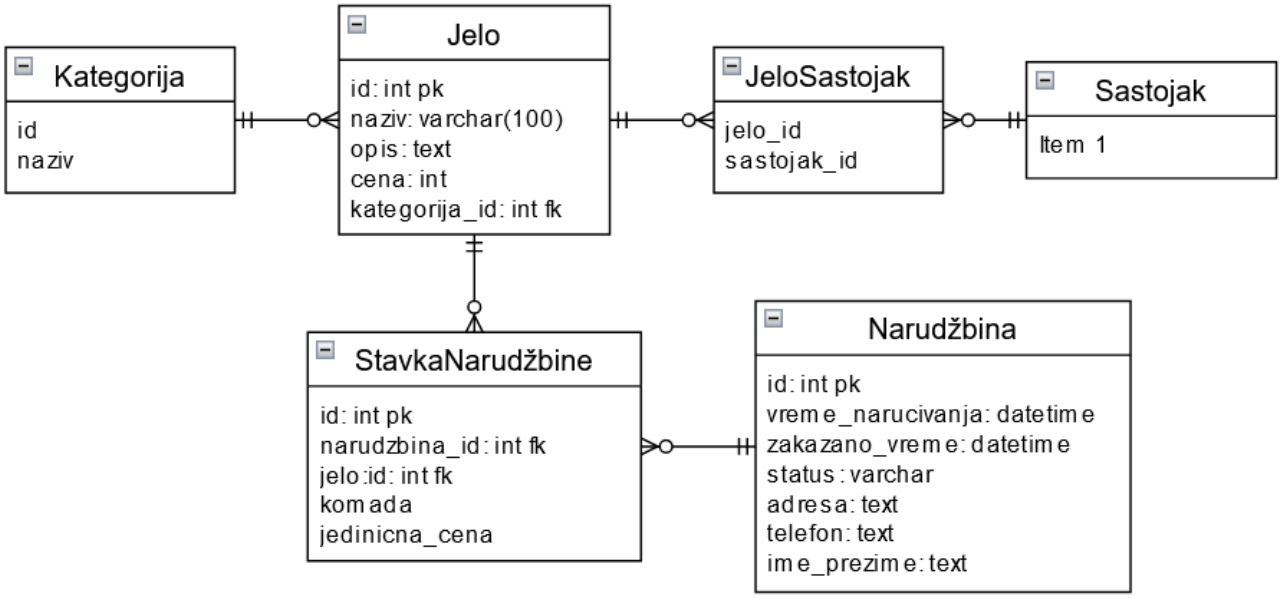
Vežba 1

Rok za završetak vežbe je 22.10.2023. Vežbu treba postaviti na github repozitorijum i poslati link asistentu na mail. Instrukcije za slanje završene vežbe biće poslate blagovremeno preko mail liste. Ako student vežbu radi na času asistent može bodovati vežbu kao deo ispita.

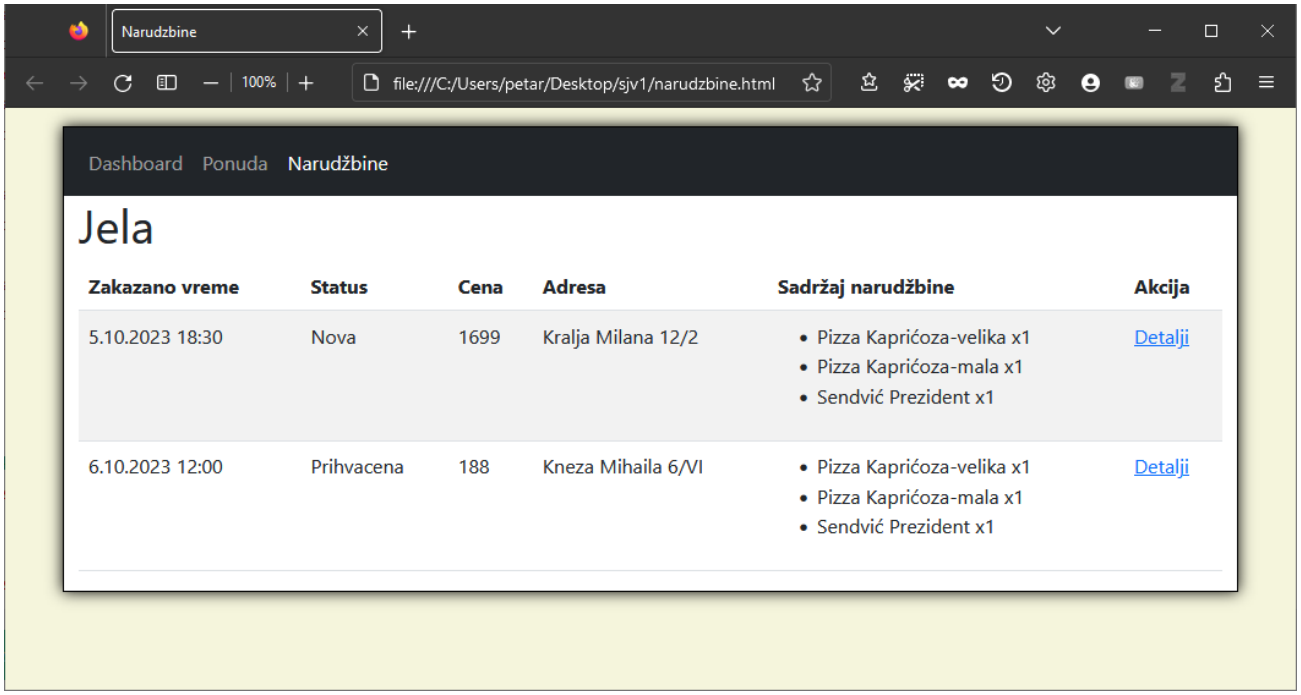
U ovoj vežbi razvićemo front-end interfejs adminske aplikacije. Koristićemo bootstrap za osnovu a dodaćemo i potrebnog CSS koda. Primenom JavaScript-a dodaćemo dinamičnost stranicama.

Napravićemo admin panel za sistem za naručivanje hrane - stranice za spisak, unos i izmenu jela i obradu narudžbina. Model podataka koji se koristi u vežbama je na slici.

Naredni koraci opisuju proces izrade. Prilagodićete nazive fajlova, naslove, nazive polja itd. vašem projektu. Npr. ako je projekat knjižara, umesto jela.html napravićete knjige.html, narudžbina će sadržati knjige, umesto sastojaka biće autori itd.



Na slici ispod je izgled stranice sa spiskom narudžbina, koju ćemo napraviti u ovoj vežbi.



Postavljanje kostura aplikacije - html, css, bootstrap

1. Napravite folder *vezba1*. Otvorite ga u Visual Studio Code ili editoru po izboru.
2. Napravite fajl *index.html*.
3. Na https://www.w3schools.com/bootstrap5/bootstrap_get_started.php možete naći primere osnovnih bootstrap stranica. Iskoristite **Container Example**.

Na https://startbootstrap.com/theme/sb-admin-2 možete naći gotov bootstrap admin template koji je takođe dobra opcija i koji vam daje gotove stranice koje možete da kopirate i prilagodite po želji.

4. Promenite sadržaj title taga u *Vežba 1*
5. U head tag dodajte <link rel="stylesheet" href="main.css"> pred kraj head taga.
6. Napravite fajl *main.css*
7. Umesto div taga koji ima klasu container stavite main tag, radi bolje semantike stranice
    a. <main class="container"> … </main>
8. Promenite sadržaj h1 naslova u *Dobrodošli na dashboard.* Pasuse možete da obrišete
    a. Ovde ćemo kasnije dodati spisak najnovijih narudžbina i neki chart.
9. Iznad naslova dodajte div sa klasom *row*, u koji ćemo smestiti navigaciju.
    a.
10. Na w3schools u sidebaru levo pronađite **navbar**
    a. Dodajte u index.html neku varijantu **Colored Navbar** unutar div taga sa klasom *row*.
11. U navbar dodajte linkove
    a. Dashboard, vodi na index.html
    b. Ponuda, vodi na jela.html
    c. Narudbžine, vodi na narudzbine.html
12. U main.css dodajte pravilo koje će promeniti boju pozadine stranice, boju pozadine main kontejnera i dodati border i senku oko main kontejnera. Dodaćemo i gornju marginu da bi se kontejner vizuelno odvojio od ivice browsera.

```
body{ background-color: #fefeec; }
main.container{
    background-color:white;
    box-shadow: 0px 0px 12px black;
    border:1px solid black;
    margin-top:15px;
}
```

   a. Prilagodite po želji. Improvizujte sa background-image i tiled background.

13. Napravite fajlove
    a. narudzbine.html
    b. narudzbina.html
    c. jela.html
    d. jelo.html
    e. novo-jelo.html
14. Kopirajte html kod iz index.html u sve ove fajlove.
15. Promenite title i h1 naslov u skladu sa fajlom.
16. U navbar primenite klasu active na odgovarajući link.

Stranice sa spiskovima - jela i narudžbine

Stranice jela.html i narudzbine.html

1. Otvorite stranicu jela.html
2. Na w3schools u sidebaru pronađite **Buttons**
3. Ispod naslova dodajte link *Novo jelo* koji vodi ka *novo-jelo.html*. Dodajte klasu primary i klasu float-end kako bi se prikazalo na desnoj strani ekrana.
    a. Novo jelo
4. Na w3schools u sidebaru pronađite **Tables**. Na Try it yourself možete videti primer koda.
5. Dodajte tabelu i primenite bootstrap klase po želji.
6. U thead dodajte red sa tri ćelije/kolone Kategorija, Jelo, Cena, Akcija
7. U tbody dodajte nekoliko redova:
    a. Pizza, Kaprićoza-velika, 1000
    b. Pizza, Kaprićoza-mala, 600
    c. Sendvič, Prezident, 99
8. U poslednju kolonu svakog reda stavite button *Promena cene* i link *Izmeni*
    a. Button treba da ima klasu primary i atribut onclick koji je prazan
        i. <button class="btn btn-primary" onclick="">Promena cene</button>
    b. Link treba da vodi na jelo.html?id=1 (trenutno nije povezano sa bazom, stavljamo ovo kao primer za kasnije)
        i. Izmeni

9. Slično primenite i na narudzbine.html. Tabela treba da ima kolone: Zakazano vreme, Status, Cena, Adresa, Sadržaj narudžbine, Akcija
    a. U koloni akcija biće samo link *Detalji* koji vodi na narudbina.html?id=1
10. Dodajte nekoliko redova. Primer:

Zakazano vreme	Status	Cena	Adresa	Sadržaj narudžbine	Akcija
5.10.2023 18:30	Nova	1699	Kralja Milana 12/2	Pizza Kapričoza-velika x1 Pizza Kapričoza-mala x1 Sendvić Prezident x1	**Detalji**
6.10.2023 12:00	Prihvaćena	188	Knez Mihailova 6/6	Sendvič Prezident x2	**Detalji**

Forme i validacija

Stranica novo-jelo.html

Sada ćemo napraviti stranicu za dodavanje novog jela. Na ovoj stranici nalaze se polja za naziv jela, kategoriju, opis, cenu.

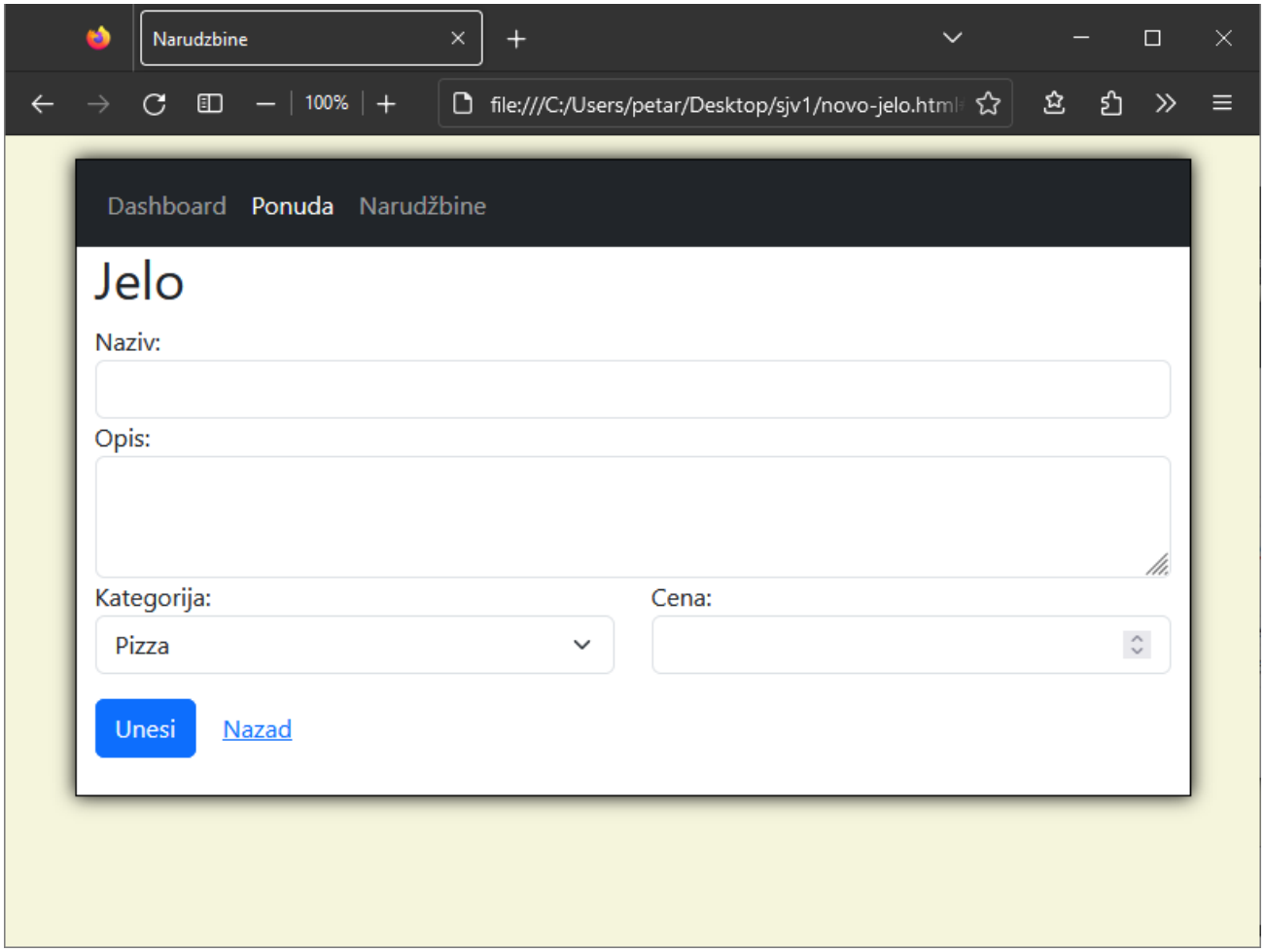
1. Otvorite fajl novo-jelo.html
2. Na w3schools pronađite **Forms** (https://www.w3schools.com/bootstrap5/bootstrap_forms.php)
3. Ispod naslova dodajte form tag čiji je method=post i action=http://postman-echo.com/post
    a. U vežbama 2 i 3 dodaćemo pravi action koji zaista radi nešto, za sad će nam postman-echo omogućiti da vidimo šta forma šalje
4. Dodajte input text za polje Naziv, i odgovarajući label. Ovo polje je obavezno, pa dodajte i atribut *required*.
```
<div>
      <label for="naziv">Naziv:</label>
      <input type="text" class="form-control" id="naziv"
                  name="naziv" required>
</div>
```
5. Dodajte textarea za Opis
```
<textarea id="opis" name="opis"
class="form-control"></textarea>
```
6. Dodajte select za kategoriju, sa opcijama Pizza, Sendvič čije su pridružene vrednosti 1 i 2
```
<select class="form-select" name="kategorija"
id="kategorija">
      <option value="1">Pizza</option>
      <option value="2">Sendvič</option>
</select>
```
7. Dodajte input number za cenu, odredite minimum 0, step 1, želimo samo cele brojeve.
```
<input type="number" min="0" step="1" required id="cena"
name="cena" class="form-control">
```
8. Dodajte button sa klasom btn-primary i tekstom Unesi
```
<button class="btn btn-primary">Unesi</button>
```

9. Dodajte link sa klasom btn-link i tekstom Nazad, koji vodi na jela.html
```
<a href="jela.html" class="btn btn-link">Nazad</a>
```

10. Opciono
    a. Button i link možete da smestite u div sa klasama mt-3 i mb-5 kako bi se dodao prostor iznad i ispod
    b. Kategoriju i cenu možete da smestite u div sa klasom row, i da divovima dodate klasu col-6

```
<div class="row">
    <div class="col-6">
        kategorija…
    </div>
    <div class="col-6">
        Cena…
    </div>
</div>
```

Dobićete stranicu kao na slici



11. Pred kraj stranice, pre </body> dodajte <script> tag, u koji će se umetnuti kod funkcije za validaciju forme.
12. Definišite funkciju

```
function validacija(){    }
```

13. Inicijalizujte promenljivu validno=true. Pokušaćemo da osporimo validnost forme.
```
var validno = true;
```
14. Dodajte uslov koji dohvata input sa id-em naziv i proverava da li je dužina stringa veća od 3. Ako nije, obojte border u crveno i stavite validno=false. Ako jeste obojte border u zeleno.
```
if( document.getElementById("naziv").value.length < 3 ){
    validno=false;
    document.getElementById("naziv").style.borderColor =
'red';
}
else {
    document.getElementById("naziv").style.borderColor =
'green';
}
```
15. Dodajte return
    return validno;
16. U form tag dodajte onsubmit="return validacija()" koji će pridružiti događaj *submit* formi (izaziva ga button) takav da poziva funkciju validacija() i prosleđuje njen rezultat. Ako se dobije *false* submit događaj će biti suspendovan.
17. Probajte da li ovo radi. Šta se dešava kada se napiše ime jela *a* a zatim pokuša izmena? Bilo bi dobro da se crveni border skloni kada se vrednost polja menja.
18. Otvorite main.css i definišite klasu error i success
    .error{ border-color:red; }
    .success{ border-color:green; }
19. Promenite funkciju validacija() tako da se umesto promene style atributa vrši pridruživanje css klase
    document.getElementById("naziv").classList.add("error");
    document.getElementById("naziv").classList.remove("success");
    i obrnuto u else delu
20. U input polje naziv dodajte događaj onkeypress koji će da ukloni klase error i success. Atribut događaja je zapravo telo metoda tog taga, u kome postoji referenca *this*
    <input …. name="naziv" onkeypress=" this.classList.remove('success');
    this.classList.remove('error') ">
    a. Vodite računa da vrednost atributa ne može da bude multiline.
    b. Vodite računa o navodnicima.
21. Probajte…

Stranica narudzbina.html

Napravićemo i stranicu za prikaz jedne narudžbine. Stavićemo statičke podatke kao primer, a kasnije ćemo povezati ovo sa backend-om i upisivati stvarne podatke iz baze. Jedino što želimo da menjamo u ovoj formi je status narudžbine.

1. Otvorite fajl narudzbina.html
2. Promenite h1 naslov u *Detalji narudžbine*
3. Dodajte formu čiji je method post, action je http://postman-echo.com/post
4. Dodajte id atribut formi, id="forma"
5. Dodajte <dl> tag u kome ćete ispisati detalje narudžbine.
    a. Upotrebite <ul> listu za sadržaj narudžbine

```
<dl class="row">
  <dt class="col-sm-3">Zakazano vreme</dt>
  <dd class="col-sm-9">5.10.2023 16:15</dd>

  <dt class="col-sm-3">Sadržaj narudžbine</dt>
  <dd class="col-sm-9">
    <ul id="sadrzaj">
      <li>Pizza Kaprićoza-velika 1x</li>
      <li>Pizza Kaprićoza-mala 1x</li>
    </ul>
  </dd>
</dl>
```

itd. popunite ostalim poljima

6. Dodajte *Status* <select> sa opcijama: novo, prihvaćeno, odbijeno, u pripremi, u dostavi, završeno, takođe u <dl>
7. U <select> dodajte onchange u kome ćemo da iniciramo slanje forme kada se promeni status.
    a. onchange=" document.getElementById('forma').submit() "

8. Probajte…

Dinamični elementi stranice

Stranica jelo.html

Sada ćemo u jelo dodati mogućnost izbora sastojaka. Sastojci će se u klijentskoj aplikaciji prikazivati uz opis jela. Želimo da omogućimo izbor sastojka iz liste, odnosno unos novog sastojka ukoliko ne postoji na listi. Potrebno je da možemo i da uklonimo sastojak iz liste. Kada se forma submituje sastojke šaljemo u JSON formatu, kao niz parova (id,naziv). Da bismo ovo postigli potrebno je da dinamički kreiramo i brišemo čvorove u DOM-u. Ovo ćemo uraditi na stranici za izmenu jela.

1. Otvorite fajl jelo.html
2. Kopirajte u njega sadržaj fajla novo-jelo.html, dogradićemo na ovoj osnovi.
3. Pošto je ovo stranica za izmenu postojećeg jela u bazi
    a. button *Unesi* treba da bude *Sačuvaj*
    b. U nastavku treba dodati i link *Obriši*, koji trenutno ne vodi nigde, pa u href stavite #
        ```
        <a href="#" class="btn btn-danger">Obriši</a>
        ```

4. Pre </head> dodajte script tag koji će uključiti fajl jelo.js
    a. <script src="jelo.js"></script>
5. Napravite fajl jelo.js

Prvo ćemo uraditi refaktorizaciju postojećeg koda tako da izdvojimo javascript iz html-a u potpunosti.

6. U jelo.js dohvatite objekat prozora *window* i pridužite mu event listener događaja *load*

```
window.addEventListener("load", function(){
        //sadrzaj funkcije koja ce se pozvati kada browser proglasi stranicu ucitanom
        //tj DOM tree potpuno formiranim
});
```

7. U jelo.html form tagu dodajte id="forma" i obrišite onsubmit atribut
8. U jelo.js u funkciju dodajte naredbu koja dohvata formu po id-u i pridružuje joj event listener za događaj click, i kopirajte sadržaj funkcije validacija() unutra, pošto je prethodna namena onsubmit atributa bila da prosledi povratnu vrednosti funkcije validacija()

```
document.getElementById("forma").addEventListener("submit", function(){
        //kod funkcije validacija() ide ovde…
});
```

U ovom kontekstu return false ne radi suspendovanje dogadjaja, pa ocekujemo da iako return validate vrati false forma bude submitovana. Umesto toga treba da nad objektom dogadjaja pozovemo preventDefault(). U deklaraciju funkcije dodajte parametar event, a na mestu return false stavite event.preventDefault();

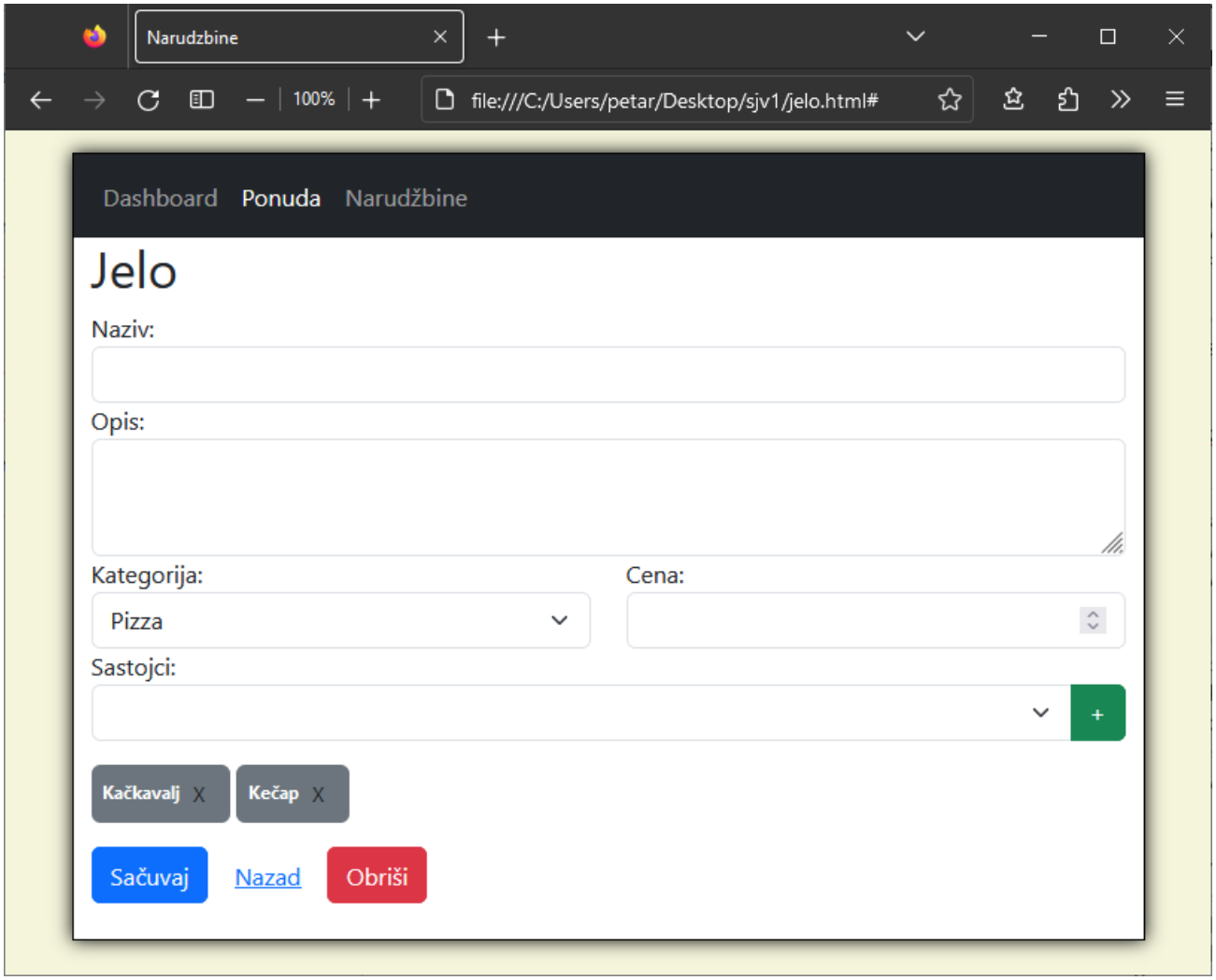
```
…addEventListener("submit", function(event){ ….
```

9. Zatim dohvatite input naziv i pridružite event listener za keypress. U telo funkcije stavite kod iz atributa onkeypress, a u jelo.html obrišite taj atribut.

```
document.getElementById("naziv").addEventListener("keypress", function(){
        this.classList.remove('success');
        this.classList.remove('error');
});
```

   Primetite da se this ponaša očekivano, ova funkcija se ugrađuje kao metod objekta koji predstavlja input polje naziv u DOM-u

Sada ćemo dodati sastojake. Sastojak će se birati iz liste sastojaka. Na + dodajemo novi sastojak u spisak sastojaka jela, i disable-ujemo ga u listi za izbor (da onemogućimo dodavanje jednog sastojka više puta).
Na X brišemo sastojak iz spiska sastojaka jela. Na slici je konačni izgled ove komponente.



10. Na w3schools pogledajte **Input groups** https://www.w3schools.com/bootstrap5/bootstrap_form_input_group.php i **Badges**
11. Napravićemo input group koji ima select i button desno

```
<div>
        <label for="naziv">Sastojci:</label>
        <div class="input-group mb-3">
          <select class="form-select" id="spisak-sastojaka">
            <option></option>
            <option value="1">Šunka</option>
            <option value="2">Kačkavalj</option>
            <option value="3">Kečap</option>
            <option value="4">Parizer</option>
            <option value="5">Majonez</option>
          </select>
          <button class="btn btn-success" type="button"
id="dodaj-sastojak">+</button>
        </div>
```

a. Prvi option ostaje prazan jer će nam to koristiti kao default vrednost

12. Zatim ćemo dodati div sa id-em *sastojci* u kome će se nalaziti izabrani sastojci. Napravićemo i primer izabranih sastojaka, kako bismo proverili da li se to prikazuje kako treba. Iskoristićemo bootstrap badge u kome je naziv sastojka i button X za uklanjanje tog sastojka iz izabranih.

```
            <div id="sastojci">
               <span class="badge bg-secondary" data-id="1">
                  Kečap
                  <button type="button" class="btn btn-default btn-sm">X</button>
               </span>
               <span class="badge bg-secondary" data-id="1">
                  Šunka
                  <button type="button" class="btn btn-default btn-sm">X</button>
               </span>
            </div>
         </div>
```

    a. data-id je data atribut u kome ćemo čuvati id sastojka

Pošto imamo završen prikaz ove komponente, dodaćemo potreban javascript kod.

13. U jelo.js u funkciju listenera događaja load dodaćemo kod koji pridružuje event listener događaja click na button #dodaj-sastojak. Kada se klikne na + treba da se proveri da li je u listi izabran neki sastojak i da se pozove funkcija dodajSastojak(id), odnosno da se prikaže odgovarajuća poruka

```
document.getElementById("dodaj-sastojak").addEventListener("click", function(){
   var id = document.getElementById("spisak-sastojaka").value;
   if(!id){
      alert("Izaberi sastojak");
      return;
   }
   dodajSastojak( id );
});
```

14. Zatim treba da definišemo funkciju dodajSastojak(). Dodajte je nakon event listenera za load događaj

```
function dodajSastojak(id){   }
```

15. U funkciji treba da dohvatimo option unutar select#spisak-sastojaka koji ima dati id i da ga disable-ujemo, a select da postavimo na prvu, praznu, opciju.
    a. Koristimo querySelector i pišemo relativno složenu putanju, koja kaže: option tag koji ima atribut value čija je vrednost *id*, i koji je direktni potomak taga koji ima id *spisak-sastojaka*
    b. Obratite pažnju na backtick navodnike u prvom redu. JS radi preprocessing ovakvih stringova i ${variable} se koristi kao sintaksa za umetanje vrednosti promenljive

```
document.querySelector(`#spisak-sastojaka > option[value='${id}']`).disabled = true;
document.getElementById("spisak-sastojaka").selectedIndex = 0;
```

16. Treba nam naziv sastojka - tekst optiona koji ima dati id
    var naziv = document.querySelector(`#spisak-sastojaka >
    option[value='${id}']`).innerHTML;

17. Zatim treba da kreiramo DOM elemente i povežemo ih u stablo, tako da dobijemo strukturu koja odgovara tagu koji prikazuje izabrani sastojak.

```
<span class="badge bg-secondary" data-id="1">
  Kečap
  <button type="button" class="btn btn-default btn-sm">X</button>
</span>
```

    a. Pravimo span
    ```
    var span = document.createElement("span");
    span.classList.add("badge");
    span.classList.add("bg-secondary");
    span.dataset.id = id;
    span.innerHTML = naziv;
    ```
    b. Pravimo button
    ```
    var button = document.createElement("button");
    button.type="button";
    button.classList.add("btn");
    button.classList.add("btn-default");
    button.classList.add("btn-sm");
    button.innerHTML = "X";
    ```
    c. Pridružujemo button u span
    ```
    span.appendChild(button);
    ```
    d. Pridružujemo span u #sastojci
    ```
    document.getElementById("sastojci").appendChild(span);
    ```

Interesantna anomalija - kada pokrenete primetićete da su spanovi spojeni, a kada smo pisali html bili su razdvojeni. To je zbog znaka razmaka između njih. Tačnije, u našem primeru između dva span-a postoji newline i nekoliko space ili tab karaktera. Sve to html parser tumači kao jedan razmak, i taj razmak se zaista renderuje (širine je 4-5px). Otud razmak između dva span-a, koji nam zapravo odgovara i želimo da postoji. Možemo to da rešimo kroz css, dodavanjem desne margine, a možemo i da zaista umetnemo razmak nakon span taga. Uradićemo ovo drugo.

    e. Nakon span-a u #sastojci dodajemo jedan tekstualni element koji u sebi sadrži samo znak razmak
    ```
    document.getElementById("sastojci").appendChild(document.createTextNode(" "));
    ```

Potrebno je da omogućimo da se klikom na X uklanja sastojak iz izabrani i enable-uje u spisku

18. U funkciji dodajSastojak() dohvatićemo *button* i pridružićemo mu event listener za click
    ```
    button.addEventListener("click", function(){    });
    ```

19. Kada se klikne želimo da dohvatimo id, koji je sačuvan u data-id atributu span taga, i da obrišemo taj span
    a. Dohvatamo id
        var id = this.parentNode.dataset.id;

    b. Span je parent element button-a. Event je pridružen button-u, pa je this referenca na kliknuti button, a this.parentNode referenca ka parentu, tj. span tagu. Brisanje taga vrši se dereferenciranjem iz njegovog parenta - imamo button->span->parent koji briše button->span
        this.parentNode.parentNode.removeChild(this.parentNode);

    c. Enable-ujemo option u spisku sastojaka
        document.querySelector(`#spisak-sastojaka > option[value='${id}']`).disabled = false;

Razlog zašto smo baš ovakav deo funkcionalnosti izvukli u funkciju dodajSastojak() je taj što ćemo ovu funkciju koristiti i za postavljanje default vrednosti kasnije kada se povežemo sa serverom. Ovako možemo da na osnovu niza id-eva iteriranjem i pozivanjem funkcije postavimo početno stanje forme prilično jednostavno.

Sada ćemo rešiti slanje liste sastojaka prilikom submitovanja forme. Potrebno je da se prilikom submitovanja dohvate svi sastojci i pripreme kao niz u JSON formatu, i spakuju u input polje koje treba da dodamo u formu. To polje ne treba da se vidi, pa ćemo koristiti type="hidden"

20. U formu dodajemo <input type="hidden" name="sastojci" id="sastojci-input">
21. U submit handler dodajemo kod koji postavlja vrednost ovog inputa na osnovu sastojaka koji postoje u #sastojci
    a. Dohvatamo sve span-ove u #sastojci
        var spanovi = document.querySelectorAll("#sastojci > span.badge");
    b. Pravimo novi niz u koji ćemo da smestimo sve id-eve sastojaka
        var niz = [];
    c. Prolazimo kroz span-ove i smeštamo id-eve iz data-id atributa u niz
        for(let i=0; i<spanovi.length; i++){
          niz.push(spanovi[i].dataset.id);
        }
    d. Pretvaramo niz u JSON string i upisujemo to kao novu vrednost hidden input polja

Samostalni rad - uključen u vežbu

Napravite CRUD stranice za kategorije i sastojke. I kategorije i sastojci imaju samo jedan atribut - naziv. Potrebno je napraviti fajlove

- kategorije.html
- kategorija.html
- nova-kategorija.html
- sastojci.html
- sastojak.html

- novi-sastojak.html

Time je postavljen osnov za front-end uredničkog dela aplikacije za naručivanje hrane. U narednim vežbama ćemo napraviti serverski deo koji isporučuje ove stranice i omogućava pristup podacima u bazi. Dopunićemo funkcionalnost ovih stranica AJAX pozivima ka serveru kako bismo povukli podatke i postavili potrebne default vrednosti.