



E-Restoran

Front-End dokumentacija

Šta je ReactJS

- ReactJS je biblioteka za izradu korisničkog interfece-a (sučelja)
- To je skup funkcija pisanih u JavaScript programskom jeziku olakšava izradu JavaScript aplikacija
- Najpopularniji Front-End Framework za izradu Web aplikacija
- Koristi se za rukovanjem view layer-a (sloja za prikazivanje) za web i mobilne aplikacije
- Prvi put je implementiran na Facebook newsfeed 2011. godine, a na Instagramu godinu dana poslije.

Šta je ReactJS



React omogucava programerima da izrade velike web aplikacije koje mogu mijenjati podatke bez ponovnog učitavanja stranice.

On je skalabilan (lako se može modifikovati, pri modifikaciji može se podrazumijevati i na modifikaciju koda ali i samog tima ljudi koji radi na jednom projektu (dodavanje ili uklanjanje), brz, jednostavan

Koristi se u kombinaciji sa JavaScript bibliotekama ili okvirima

React upravlja samo prikazom aplikacije. Sam React ne upravlja podacima, povezivanjem podataka ili bilo čim drugim (to rade dodatne biblioteke). Postoje kompletna okruženja, kao što je Angular, koja upravljaju podacima, povezivanjima i korisničkim interfejsom.

Virtualni DOM



DOM (objektni model dokumenta) je struktura stabla HTML elemenata na stranici. DOM API-ji, kao što je `getElementById()`, omogućavaju izbor i modifikaciju elemenata DOM stabla. Međutim, premeštanje i ažuriranje DOM-a nisu mnogo brze operacije, jer za veće aplikacije ovo stablo postaje jako veliko i razgranato. Ovdje na scenu stupa React jer on omogućava da se definije izgled stranice, tačnije dozvoljava da se deklariše samo izgled komponente, a logiku upravljanja istom ostavlja nekom drugom. A da bi riješio problem performansi React koristi nešto što se zove virtualni DOM. On predstavlja apstrakciju HTML DOM-a, kopija lokalne memorije DOM-a. Unutar iste React izvršava sva izračunavanja koja su potrebna za prikaz komponente korisničkog interface-a. Ovaj sistem omogućava brzinu kod aplikacija visokih performansi.

React komponente

- Komponenta je element web stranice. Srž svake React aplikacije su komponente. To su samostalne cjeline koda koje sadrže markup, logiku, stil... U našem slučaju jedna komponenta predstavlja jednu stranicu (jedan dio stranice).
- Komponente možemo posmatrati kao funkcije koje imaju svoje ulazne parametre, a kao izlaz vraćaju React elemet koji opisuje kako će stranica izgledati.
- Iako se ulazni parametri mijenjaju, React ponovo rendera (ispisuje) tu komponentu, stoga bez obzira na ulaz, izlaz je uvijek isti.
- Komponente treba izrađivati tako da opisuju funkciju za samo jedan dio.



React komponente

- Komponente se mogu praviti na dva načina: pomoću funkcija ili pomoću klasa.
- Obična funkcija postaje komponenta ako vraća JSX
- Komponenta može biti i klasa, a razlikuje se od ostalih komponenti jer nasljeđuje iz React.Component klase
- Nazivi se pišu velikim početnim slovom, da bi se razlikovali od ugrađenih HTML tagova, jer se pozivaju na identičan način.



React komponente

- Render je najvažniji i jedina nužna metoda jedne React komponente. Iz metode render vraćamo opis onoga što želimo prikazati. Kao što je već napomenuto, metoda automatski obnavlja prikaz prilikom promjena.
- JSX (JavaScript Syntay Extension) predstavlja ekstenziju sintakse JavaScript-a koji je napravio Facebook kako bi pisanje markup komponenti bio omogućen u poznatoj HTML sintaksi. U pozadini, ova sintaksa se prevodi u validan Javascript. Korištenje JSX-a se može i izbaciti i koristiti čisti JavaScript, sa malim modifikacijama prilikom kreiranja React elemenata.



React komponente

- Stanje (state) komponente je objekat koji sadrži varjable. Ono postoji samo unutar klase i vidljivo je samo unutar iste.
- Stanja nam omogućavaju izmjenu i interakciju na stranci
- Komponenta posjeduje ugrađene funkcije za rad sa stanjima i na taj način stvara veliki spektar mogućnosti. Pored pomenutih funkcija, naravno, mi dodajemo pomoćne funkcije kojima upravljamo podacima i samom funkcijom komponente.



React komponente

- React je uveo i uveo faze „životnog ciklusa“ komponente i na taj način obezbiđuje metode za upravljanje svakom fazom. Neke od najčešće korištenih metoda u ovoj aplikaciji su:
- render – ova metoda se inicijalno izvršava nakon konstruktora komponente
- componentDidMount – odmah po završetku izvršavanja metode render, poziva se metoda componentDidMount. U ovoj fazi možemo pristupiti DOM-u i obavljaju sve interakcije. Unutar ove metode popunjavaju se podaci iz stata
- componentDidUpdate – također se poziva nakon funkcije render, tj nakon ažuriranja. Ovo je dobro mjesto za rad sa DOM-om nakon što je ažuriran ili za rad sa mrežnim zahtjevima (poređenje prethodnog i trenutnog props-a)
- componentWillUnmount – ova metoda nam služi za uklanjanje komponente iz DOM-a, ona skuplja svo smeće



Axios

- Axios je alat koji služi za spajanje Front-end projekta npr ReactJs, Angular, itd. sa REST API-em. Upravo on nam omogućava lak način komunikacije putem HTTP zahtjeva.
- Axios je promise bazira i stoga se može iskoristiti za asinhronizaciju i čekanja za još čitljiviji asinhroni kod.
Možemo presresti i otkazivati zahtjeve, a postoji i ugrađen zaštita na strani klijenta od krivotvorenja zahtjeva.
- Najbolja strana Axios-a je činjenica da je jednostavan za korištenje API-a.



Komponente u našoj aplikaciji

- Chat
- Contact
- Homepage
- Image
- Menu
- Order
- PublicChat
- Recipe
- Register x
- Welcome x

Komponente u našoj aplikaciji su klasne komponente.



Index.js

File index.js je početni file aplikacije koji rendera najveću komponentu aplikacije, a to je sama aplikacija, komponenta App



A screenshot of Visual Studio Code showing the code editor with the file "index.js" open. The code is a React application component named "App". It imports React, ReactDOM, and other components from the "src" directory. It uses the "React.StrictMode" and "App" components to render the application. It also includes a service worker registration comment.

```
File Edit Selection View Go Run Terminal Help index.js - restoran2 - Visual Studio Code
OPEN EDITORS JS app.js JS index.js JS welcome.js JS contact.js JS homepage.js JS menu.js # welcome.css
RESTORAN2
  > contact
  > menu
  > about.jpg
  > about1.jpg
  > food-bg.png
  > Menu
  > Order
  > PublicChat
  > Recipe
  > Register
  > Welcome
    # welcome.css
    JS welcome.js
# app.css
JS app.js
JS app.test.js
JS history.js
# index.css
JS index.js
# icon.svn
TIMELINE index.js
Loading timeline for index.js...
Activate Windows
Go to settings to activate Windows.
Ln 19 Col 1 Spaces: 2 UTF-8 LF JavaScript
```

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './app';
5 import * as serviceWorker from './serviceWorker';
6
7
8 ReactDOM.render(
9   <React.StrictMode>
10   |   <App />
11   </React.StrictMode>,
12   document.getElementById('root')
13 );
14
15 // If you want your app to work offline and load faster, you can change
16 // unregister() to register() below. Note this comes with some pitfalls.
17 // Learn more about service workers: https://bit.ly/CRA-PWA
18 serviceWorker.unregister();
19 |
```

Komponenta App



Najveća komponenta aplikacije je App komponenta koja se može posmatrati kao „roditeljska“ komponenta aplikacije. Ona objedinjuje sve komponente i definiše izgled kompletne stranice. Unutar ove komponente definisan je i Router – biblioteka koja omogućava definisanja ruta za svaku komponentu. Zanimljiva situacija je činjenica da su neke rute zaštićene i nisu dostupne posjetiteljima, samo logovanim (registrovanim) korisnicima, zbog toga imamo dva <Router> elementa unutar <Switch> elemeta, a pozivaju se u zavisnosti od smog korisnika.

Komponenta App jedina funkcijaska komponenta u aplikaciji.

U sklopu ove komponente kreiran je i nav header.

Terminal Help app.js register.js recipe.js publicChat.js order.js welcome.js contact.js chat.js homepage.js

restoran2 > src > JS app.js > App

```

1 import React, { Component } from "react";
2 import {
3   BrowserRouter as Router,
4   Switch,
5   Route,
6   Link
7 } from "react-router-dom";
8 import { Navbar, Nav, NavDropdown, Button } from 'react-bootstrap';
9
10 import Welcome from './components/Welcome/welcome';
11 import Register from './components/Register/register';
12 import History from './history';
13 import Homepage from './components/Homepage/homepage';
14 import Contact from './components/Contact/contact';
15 import Menu from './components/Menu/menu';
16 import './app.css';
17 import Recipe from './components/Recipe/recipe';
18 import Chat from './components/Chat/chat';
19 import PublicChat from './components/PublicChat/publicChat';
20
21 function App() {
22   return (
23     <Router history={History}>
24       <div>
25         <nav>
26           <header className="container">
27             <Navbar expand="lg" className="nzb" variant="dark" sticky="top">
28               <Navbar.Brand className="navlink" href="#home">E-restoran</Navbar.Brand>
29               <Navbar.Toggle aria-controls="basic-navbar-nav" />
30               <Navbar.Collapse id="basic-navbar-nav" className="nvbcollapse">
31                 <Nav className="navnav ml-auto">
32                   <Nav.Link as={Link} className="navlink" to="/">Početna</Nav.Link>
33                   <Nav.Link className="navlink" href="#about">O nama</Nav.Link>
34                   <Nav.Link as={Link} className="navlink" to="/menu">Menù</Nav.Link>
35                   <NavDropdown className="navlink" title={<span className="navlink">Usluge</span>} id="basic-nav-dropdown">
36                     <NavDropdown.Item href="#action/3.1">Rezervacija</NavDropdown.Item>
37                     <NavDropdown.Item as={Link} to="/homepage">Homepage</NavDropdown.Item>
38                     <NavDropdown.Divider />
39                     <NavDropdown.Item as={Link} to="/recipe">Recepti</NavDropdown.Item>

```

restoran2 > src > JS app.js > App

```

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

```

/* A <Switch> looks through its children <Route>s and renders the first one that matches the current URL. */

```

<Switch>
  <localStorage.getItem("token") == null ? (<div>
    <Route exact path="/">
      <Welcome />
    </Route>
  ) : (<div>
    <Route path="/users/register">
      <Register />
    </Route>
    <Route path="/contact">
      <Contact />
    </Route>
    <Route path="/recipe">
      <Recipe />
    </Route>
    <Route path="/menu">
      <Menu />
    </Route>
    <Route path="/pubChat">
      <PublicChat />
    </Route>
  ) : (<div>
    <Route exact path="/">
      <Welcome />
    </Route>
  )
</div>
<Route path="/users/register">
  <Register />
</Route>
<Route path="/contact">
  <Contact />
</Route>
<Route path="/recipe">
  <Recipe />
</Route>
<Route path="/menu">
  <Menu />
</Route>
<Route path="/pubChat">
  <PublicChat />
</Route>
</div>
<Route exact path="/">
  <Welcome />
</Route>
</div>
<Route path="/users/register">
  <Register />
</Route>
<Route path="/contact">
  <Contact />
</Route>

```

Komponenta Welcome



Ovo je vrlo jedostavna komponenta koja najviše služi za definisanaj izgleda početne stranice (stranica koja se prva pojavljuje kada se stranica učita).

Unutar komponente je iskorištena je metoda `componentDidMount()` unutar koje se računa trenutno vrijeme (sat) i korisniku spisuje poruku pozdrava u zavisnosti od doba dana u kojem je posjetio našu stranicu.

```
export default class Welcome extends React.Component {  
  state = {  
    h: 0,  
    min: 0,  
    sec: 0,  
  
    mess: ""  
  }  
  
  componentDidMount() {  
    setInterval(() => {  
      let time = new Date();  
      this.setState({  
        h: time.getHours(),  
        min: time.getMinutes(),  
        sec: time.getSeconds()  
      })  
  
      if (time.getHours() <= 11) {  
        this.setState({  
          mess: "Good morning!"  
        })  
      } else if (time.getHours() > 11 && time.getHours() <= 18)  
        this.setState({ mess: "Good day!"})  
      else this.setState({ mess: "Good evening!"})  
    }, 1000)  
  }  
  
  render() {  
    return [  
      <div className="body">  
        {  
          <h1>{this.state.mess}</h1>  
          <h3>Welcome to our page.</h3>  
          <h1>{this.state.h}:{this.state.min}: {this.state.sec}</h1>  
        }  
      </div>  
    ]  
  }  
}
```



Dio koda komponente
Welcome

Komponenta Register



Kao što samo ime komponente nam aludira, ova komponenta nam služi za kreiranje novih korisnika i prijavu postojećih, kako bi korisnicima omogućili veće pogodnosti i interakcije na stranici. Sve dozvole i zabrane su definisane na Back-End-u, a ovdje se samo ispoljavaju i postavljaju u funkcionalnost.

Kao state spašavamo informacije koje korisnik unosi u formu. Pomoću funkcija register() i login() se vrši autentifikacija i autorizacija postojećeg korisnika ili unos novog korisnika u bazu.

Metodu componentDidMount() smo iskoristili za prebacivanje korisnika na početnu stranicu aplikcije pomoću Axios-a.

Komponenta Register



Na ovoj stranici se nalazi i privatni chat (chat koji je namijenjen samo uposlenicima jednog restoran), to je onda potrebno unutar ove komponente omogućiti i ovaj zahtjev. Zbog toga pomoću metode `handleInput =(event)` kupi se username korisnika i postavljamo su isti kao nick za chat.

Unutar `render()` metode na uobičajan način ispisana je forma koja zadovoljava potrebe unosa svih neophodnih podataka.

```
1 import React, { Component } from 'react';
2 import { useHistory, Redirect } from 'react-router-dom';
3 import Axios from 'axios';
4 import Chat from './Chat/chat';
5
6
7 export default class Register extends Component {
8     state = {
9         firstName:"",
10        lastName:"",
11        username:"",
12        usernameLogin: "",
13        email:"",
14        tel:"",
15        password:"",
16        passwordLogin: "",
17        DOB:"",
18        hireDate: "",
19        restaurantUnit: 0,
20        allRestaurants: [],
21
22        redirect: false
23    }
24
25    handleInput =(event) => {
26        this.setState({
27            [event.target.name]: event.target.value
28        })
29    }
30
31    register =(e) => {
32        e.preventDefault()
33        console.log(this.state)
34        Axios.post("https://localhost:44342/api/register", {
35            firstName: this.state.firstName,
36            lastName: this.state.lastName,
37            email: this.state.email,
38            userName: this.state.username,
39            DOB: this.state.DOB,
40            hireDate: this.state.hireDate
41        })
42        .then(resp => {})
43    }
44
45
46    login =(e) => {
47        e.preventDefault()
48        Axios.post("https://localhost:44342/api/login", {
49            userName: this.state.usernameLogin,
50            password: this.state.passwordLogin
51        })
52        .then((res)=> {
53            console.log(res.data);
54            localStorage.setItem("token", res.data.title);
55            localStorage.setItem("nick", this.state.usernameLogin);
56
57            this.setState({ redirect: true });
58
59            //history.push("/privateChat");
60
61        })
62        .catch(e => { console.log(e); })
63    }
64
65
66
67
68 }
```

```
69     componentDidMount =() => {
70         let id = 1
71         Axios.get("https://localhost:44342/restaurantUnit/restaurant/1")
72             .then((res) => {
73                 console.log(res.data);
74                 this.setState({ allRestaurants: res.data})
75             })
76             .then(resp => {
77                 console.log("GET restaurant units:  ");
78                 console.log(resp.data);
79             })
80             .catch((e) => {
81                 console.log(e)
82             })
83     }
84     ru = (e) => {
85         console.log(e.target.value)
86         this.setState({restaurantUnit : parseInt(e.target.value)})
87     }
88     render() {
```

```
93      <div>
94          <h3>Ukoliko još nemate svoj račun, napravite ga što prije da bi dobili pogodnosti kakve drugi nemaju. A k
95          | se i iskoristit svoje pogodnosti.
96      </h3>
97      <form id="register" onSubmit={this.register}>
98          <label>First Name: </label>
99          <input type="text" name="firstName" onChange={this.handleInput} value={this.state.firstName}/>
100         <br />
101         <label>Last Name: </label>
102         <input type="text" name="lastName" onChange={this.handleInput} value={this.state.lastName}/>
103         <br />
104         <label>Username: </label>
105         <input type="text" name="username" onChange={this.handleInput} value={this.state.username}/>
106         <br />
107         <label>Password: </label>
108         <input type="password" name="password" onChange={this.handleInput} value={this.state.password}/>
109         <br />
110         <label>E-mail:</label>
111         <input type="text" name="email" onChange={this.handleInput} value={this.state.email}/>
112         <br />
113         <label>Tel:</label>
114         <input type="text" name="tel" onChange={this.handleInput} value={this.state.tel}/>
115         <br />
116         <label>Date of Birth:</label>
117         <input type="date" name="DOB" onChange={this.handleInput} value={this.state.DOB}/>
118         <br />
119         <label>Hire Date:</label>
120         <input type="date" name="hireDate" onChange={this.handleInput} value={this.state.hireDate}/>
121         <br />
122         <label>RestaurantUnitId: </label>
123         <select onChange={this.ru}>
124             {this.state.allRestaurants.map(p =>{
125                 |     return <option value = {p.unitId} key = {p.unitId}> {p.unitId} </option>
126             })}
127         </select>
128         <br />
129         <button>SIGN UP</button>
130     </form>
131
```

Komponenta Contact

An aerial photograph showing several wooden tables and chairs arranged outdoors, likely in a restaurant or cafe setting. Several people are seated at the tables, eating and socializing. The scene is set against a dark, possibly paved ground.

Komponenta Contact nam služi za ispisivanje informacija na Contact stranici. Dinamički iz baze se izlistavaju sve poslovnice jednog restorana unutar lijevom divu, dok u desnom divu se prikazuje klasična forma za unos poruke (ukoliko korisnik želi kontaktirati određeni restoran, poslati im poruku). Na dnu render metode postavljen je <Iframe> tag koji omogućava prikaz google mape. Da bi se omogućio ovakav ispis poslovnica potrebna su 3 Axios.get poziva različitih ruta. Unutar ove komponente koristi se Promise. To je drugi način pisanja callback-a, koji osigurava da će se izvršiti ono što mu se proslijedi. Promise je objekat koji upravlja asinhronim komponentama koje imaju neke bitne garancije čije upravljanje callback-om bi bilo teško ili pak nemoguće.

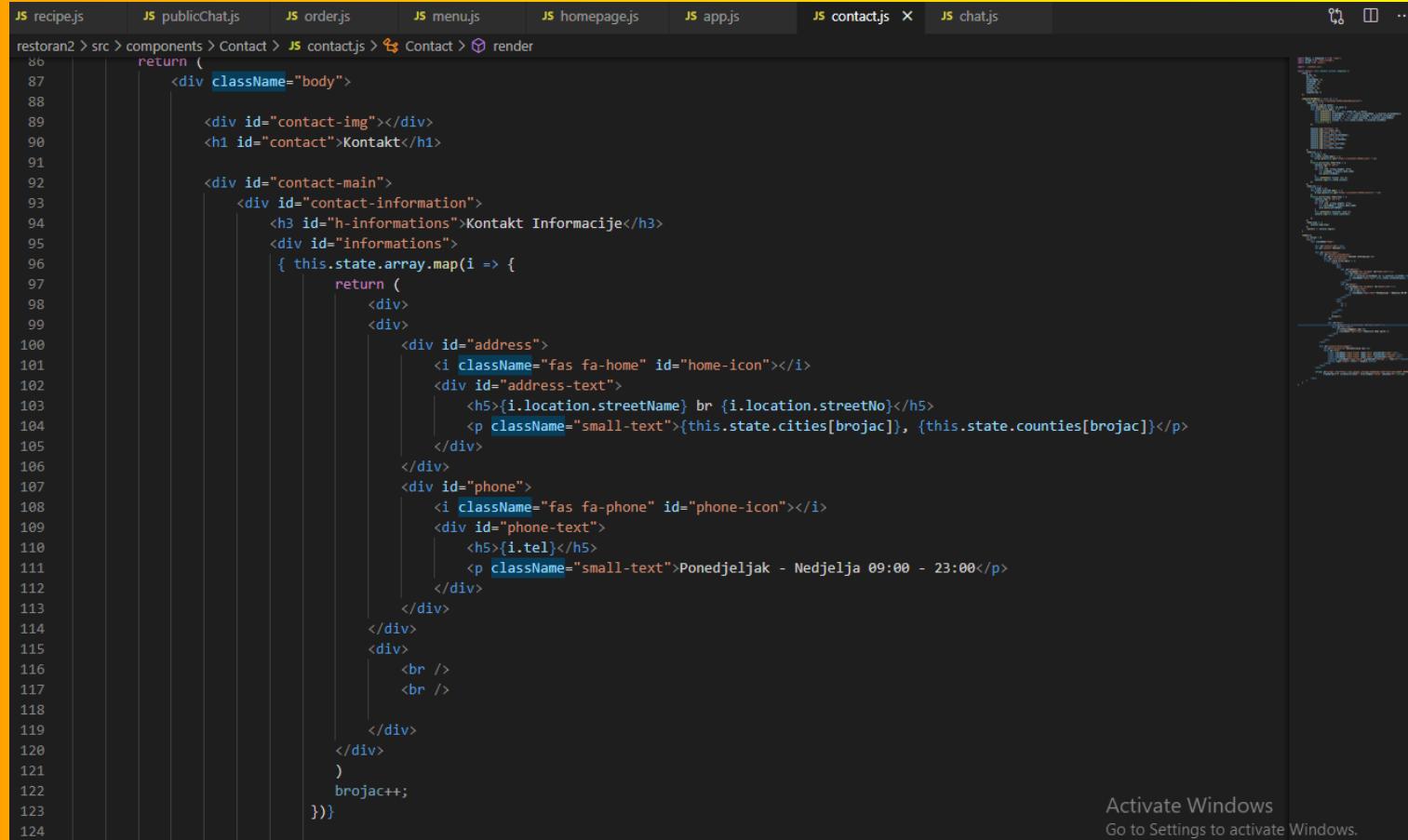
restoran2 > src > components > Contact > JS contact.js > Contact > render

```
20 componentDidMount = async () => {
21     Axios.get("https://localhost:44342/aboutRestaurant")
22     .then((res) => {
23         console.log(res.data);
24         this.setState({ array: res.data })
25         this.state.array.map(i => {
26             this.setState({ tel: [...this.state.tel, i.tel] })
27             this.setState({ streetsName: [...this.state.streetsName, i.location.streetName] })
28             this.setState({ streetsNo: [...this.state.streetsNo, i.location.streetNo] })
29             this.setState({ countryId: [...this.state.countryId, i.location.countryFK] })
30             this.setState({ cityId: [...this.state.cityId, i.location.cityFK] })
31             //console.log(i)
32         })
33
34         console.log("Telefoni: ");
35         console.log(this.state.tel);
36         console.log("Imena ulica: ");
37         console.log(this.state.streetsName);
38         console.log("Brojevi ulica: ");
39         console.log(this.state.streetsNo);
40         console.log("Drzave: ");
41         console.log(this.state.countryId);
42         console.log("Gradovi: ");
43         console.log(this.state.cityId);
44     })
45     .then(res => {
46         let array = [];
47         this.state.cityId.map(i => {
48             array.push(Axios.get("https://localhost:44342/city/" + i))
49         })
50         Promise.all(array).then(resp => {
51             console.log("Tu sam");
52             let niz= [];
53             for (let i=0; i<resp.length; i++){
54                 let cityName = resp[i].data.name;
55                 niz.push(cityName);
56             }
57             this.setState({ cities: niz });
58             console.log(this.state.cities);
59         })
60     })
61 }
```

Primjer korištenja promise-a u
Contact formi

Activate Windows
Go to Settings to activate Windows.

```
<Iframe id="g-map" src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2877.099000014355!2d18.388458550830684!3d43.8537766796" frameborder="0" allowfullscreen="" aria-hidden="false" tabindex="0"></Iframe>
```



```
JS recipe.js   JS publicChat.js   JS order.js   JS menu.js   JS homepage.js   JS app.js   JS contact.js X   JS chat.js
restoran2 > src > components > Contact > JS contact.js > Contact > render
86     return (
87         <div className="body">
88
89             <div id="contact-img"></div>
90             <h1 id="contact">Kontakt</h1>
91
92             <div id="contact-main">
93                 <div id="contact-information">
94                     <h3 id="h-informations">Kontakt Informacije</h3>
95                     <div id="informations">
96                         { this.state.array.map(i => {
97                             return (
98                                 <div>
99                                     <div>
100                                         <div id="address">
101                                             <i className="fas fa-home" id="home-icon"></i>
102                                             <div id="address-text">
103                                                 <h5>{i.location.streetName} br {i.location.streetNo}</h5>
104                                                 <p className="small-text">{this.state.cities[brojac]}, {this.state.counties[brojac]}</p>
105                                             </div>
106                                         <div id="phone">
107                                             <i className="fas fa-phone" id="phone-icon"></i>
108                                             <div id="phone-text">
109                                                 <h5>{i.tel}</h5>
110                                                 <p className="small-text">Ponedjeljak - Nedjelja 09:00 - 23:00</p>
111                                             </div>
112                                         </div>
113                                         <div>
114                                             <br />
115                                         <br />
116                                         <br />
117                                         <br />
118                                         <br />
119                                         <br />
120                                         </div>
121                                         )
122                                         brojac++;
123                                     })}
124                                 </div>
125                             )
126                         )}}>
```

Komponenta Homepage

An aerial photograph showing several wooden picnic tables arranged on a paved area. There are approximately six people seated at the tables, some eating and some looking at their phones. The scene is set outdoors, likely in a park or public square.

Ovo je jedna od komponenti koje mogu vidjeti samo logovani korisnici. Ona podržava ispis poslovnica, update podataka istih ili pak njihovo brisanje. Sve se obavlja preko Axios get, post ili update poziva. Za unos i update koristimo dvije forme, a brisanje se vrši preko dugmeta.

```
125     c2 = (e) => {
126         console.log(e.target.value)
127         this.setState({cityFK: parseInt(e.target.value)})
128     }
129
130     addUnit =(e) => {
131         e.preventDefault();
132         console.log("STATE CIJELI: ");
133         console.log(this.state);
134         Axios.post("https://localhost:44342/location", {
135             StreetName: this.state.streetName,
136             StreetNo: this.state.streetNo,
137             CountryFK: this.state.countyFK,
138             CityFK: this.state.cityFK
139         })
140         .then(resp => {
141             console.log(resp.data.id);
142             Axios.post("https://localhost:44342/restaurantUnits", {
143                 Tel: this.state.tel,
144                 EmployeeNo: this.state.employeeNo,
145                 OpeningDate: this.state.openingDate,
146                 ClosingDate: this.state.closingDate,
147                 Capacity: this.state.capacity,
148                 RestaurantId: this.state.restaurantId,
149                 LocationId: resp.data.id
150             })
151         })
152         .then(() => [
153             alert("DODALI STE NOVU POSLOVNICU U BAZU PODATAKA!")
154         ])
155         .catch(e => console.log(e))
156     }
157
158     azuriraj =(unitId) => {
159         Axios.get("https://localhost:44342/restaurantUnits" + unitId)
160     }
161
162     obrisi =(unitId) => {
```

JS recipe.js

JS publicChat.js

JS order.js

JS menu.js

JS homepage.js X

JS app.js

JS chat.js

File Edit View Insert Tools Help

```
restoran2 > src > components > Homepage > JS homepage.js > Homepage > addUnit
198     <label>Datum otvaranje: </label>
199     <input type="date" name="openingDate" onChange={this.handleInput} value={this.state.openingDate}/><br/>
200     <button>DODAJ</button>
201   </form>
202 </div>
203 <hr />
204 <div id="updateAndDelete">
205   <div id="upadte-unit">
206     <form id="addUnit" onSubmit={this.updateUnite}>
207       {/* <input id="updateUnit" type="number" /> */ }
208       <label>Naziv ulice: </label>
209       <input type="text" name="streetName" onChange={this.handleInput} value={this.state.streetName}/><br/>
210       <label>Broj ulice: </label>
211       <input type="text" name="streetNo" onChange={this.handleInput} value={this.state.streetNo}/><br/>
212       <label>Država: </label>
213       <select onChange={this.c1}>
214         {this.state.dbCountry.map(p =>{
215           return <option value = {p.countryId} key = {p.countryId}> {p.name} </option>
216         })}
217       </select> <br />
218       <label>Grad: </label>
219       <select onChange={this.c2}>
220         {this.state.dbCity.map(p =>{
221           return <option value = {p.cityId} key = {p.cityId}> {p.name} </option>
222         })}
223       </select> <br />
224       <label>Broj telefona poslovnice: </label>
225       <input type="text" name="tel" onChange={this.handleInput} value={this.state.tel}/><br/>
226       <label>Kapacitet (maksimalan broj mogućih radnih mesta) u ovoj poslovniči: </label>
227       <input type="number" name="capacity" onChange={this.handleInput} value={this.state.capacity}/><br/>
228       <label>Datum otvaranje: </label>
229       <input type="date" name="openingDate" onChange={this.handleInput} value={this.state.openingDate}/><br/>
230       <button>DODAJ</button>
231     </form>
232   </div>
233   <div>
234     <h2>A ukoliko želite izbrisati iz baze neku poslovnici kliknite na dugme " DELETE ".</h2>
235     <div id="contact-information">
236       <h3 id="h-informations">Kontakt Informacije</h3>
237       <div id="informations">
```

Activate Windows

Go to Settings to activate Windows.

Komponenta Recipe



Ova komponenta omogućava formu za unos novog recepta. Pošto sam tekst recepta može biti jako dug, omogućeno je slanje pdf file za test recepta. Ovo smo postigli tako što je na Back-End-u implementirana ruta za upload file (tekstualnog file-a). Sada na Frontu samo pozivamo tu rutu i na server šaljemo string koji predstavlja path file-a.

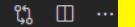
JS recipe.js

JS publicChat.js

JS order.js

JS menu.js

JS chat.js

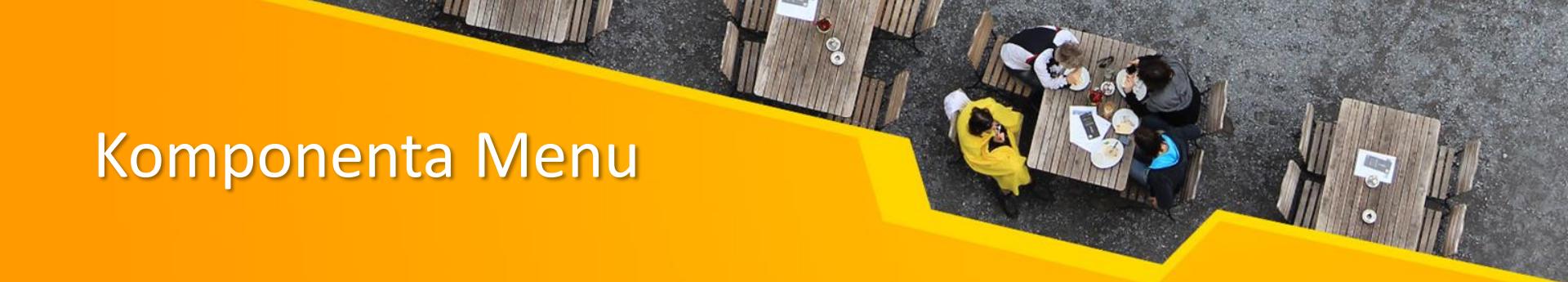


```
restoran2 > src > components > Recipe > JS recipe.js > Recipe > handleInput
  12     this.setState({
  13       [event.target.name]: event.target.value
  14     })
  15   }
  16
  17   postRecipe =(e) => {
  18     e.preventDefault()
  19     console.log(this.state)
  20     Axios.post("https://localhost:44342/recipes", {
  21       name: this.state.name,
  22       ingredients: this.state.ingredients,
  23       preparationTime: parseInt(this.state.preparationTime)
  24     })
  25     .then((resp)=> {
  26       console.log(resp);
  27       let name1 = resp.name;
  28       Axios.post("https://localhost:44342/upload", {
  29         Name: name1,
  30         ObjFile: this.state.recipeText
  31       }, {
  32         headers: {
  33           "Content-Type": 'multipart/form-data'
  34         }
  35       })
  36       .then(resp => {
  37         alert("Dodano");
  38       })
  39       alert("Dodano2");
  40     })
  41     .catch((e)=> console.log(e))
  42   }
  43
  44
  45   render(){
  46     return(
  47       <div>
  48         <div id="left">
  49           </div>
  50           <div id="right">
```

Activate Windows
Go to Settings to activate Windows.

WS.

Komponenta Menu



Menu komponenta je jedna od najbitnijih. Prvenstveno se izlistavaju jela iz baze. Pored svakog jela ispisuje se dugme za narudžbu tog jela. Ovaj dio vide svi korisnici. Ali naručiti jelo mogu samo logovani korisnici. Korisnik bira jela koja naručuje, a kada je siguran da je završio sa narudžbom korisnik treba da klikne na dugme Finish. Potom se otvara modal koji nas pita da li želimo poslati narudžbu, te poruka o uspješnosti iste.

```
restoran2 > src > components > Menu > JS menu.js > Menu
  126     ordersName: a,
  127     ordersCol: b
  128   }, () => {
  129     console.log(this.state.ordersName);
  130     console.log(this.state.ordersCol);
  131   })
  132 }
  133 }
  134
  135 render(){
  136   return(
  137     <div>
  138       <div>
  139         <Button color="danger" onClick={this.toggle}>Zavši narudžbu</Button>
  140         <Modal isOpen={this.state.modal} toggle={this.toggle}>
  141           <ModalHeader toggle={this.toggle}>Modal title</ModalHeader>
  142           <ModalBody id="body">
  143             <ul>
  144               {this.state.ordersName.map((o, i) => {
  145                 return ( <li key={i}>{o}: {this.state.ordersCol[i]}</li> )
  146               })}
  147             </ul>
  148           </ModalBody>
  149           <ModalFooter id="footer">
  150             <Button id="narudzba" onClick={this.mess}>OK</Button>{' '}
  151             <Button id="odustani" color="secondary" onClick={this.toggle}>Prekini narudžbu</Button>
  152           </ModalFooter>
  153         </Modal>
  154       </div>
  155       <div id="meal">
  156         <div>
  157           <h1>Trenutna cijena računa iznosi: {this.state.totalPrice}</h1>
  158           <h3>Recepti:</h3>
  159           { this.state.recipe.map((y, i) => {
  160
  161             return <div key={i}>
  162               <p name="officialPrice" ><b>Redovna cijene:</b> {y.price}</p>
  163               <p name="bonusPrice" >Nova cijena: { y.specialOffers[0] ? y.price - y.specialOffers[0].discount : y.price } / 100</p>
  164               <p name="offerStart" >Ponuda vrijedi od: {y.offerStart}</p>
```

Activate Windows
Go to Settings to activate Windows.

```
JS publicChat.js          JS order.js      JS menu.js  X  JS chat.js
restoran2 > src > components > Menu > JS menu.js >  Menu

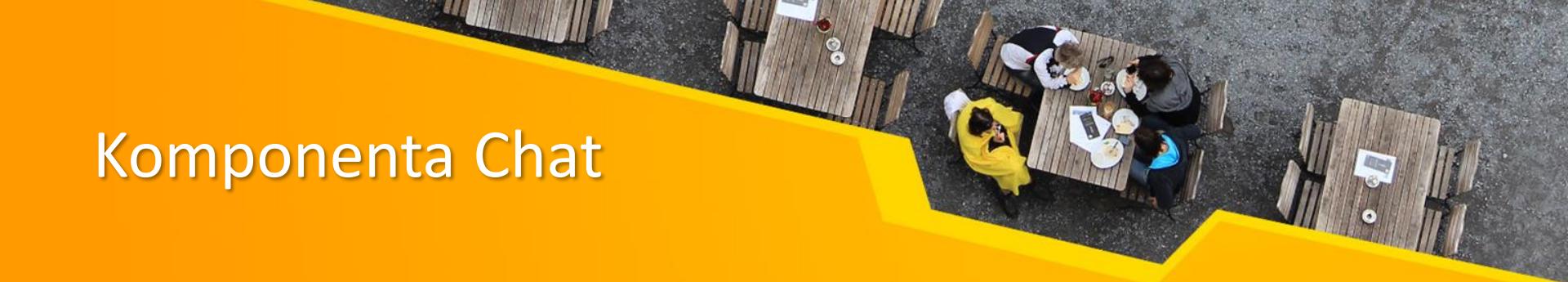
44     this.finish();
45   })
46 }
47
48 componentDidMount =() => {
49   let id = 0;
50   Axios.get("https://localhost:44342/meals/recipes", {
51     headers: {
52       "Authorization": `Bearer ${localStorage.getItem("token")}`
53     }
54   })
55   .then((res) => {
56     console.log("Tu sam!");
57     console.log(res);
58
59     this.setState({
60       recipe: res.data
61     });
62     //id = res.data.specialOffer;
63     console.log(this.state);
64
65   })
66   /*.then(res => {
67     Axios.post("https://localhost:44342/orders", {
68
69     })
70   })*/
71   .catch((e) => {
72     console.log(e);
73     alert("Niste logovani!");
74   })
75 }
76
77 order =(sp, p, n) => {
78   console.log("order", sp, p);
79   if (sp != null) {
80     this.setState({ totalPrice: this.state.totalPrice + p - sp * p / 100});
81   }
82   } else {
83     this.setState({ totalPrice: this.state.totalPrice + p });
84   }
85 }
```

JS publicChat.js JS order.js JS menu.js X JS chat.js

restoran2 > src > components > Menu > JS menu.js > Menu > componentDidMount

```
68
69     })
70   })*/
71   .catch((e) => {
72     console.log(e);
73     alert("Niste logovani!");
74   })
75 }
76
77 order = (sp, p, n) => {
78   console.log("order", sp, p);
79   if (sp != null) {
80     this.setState({ totalPrice: this.state.totalPrice + p - sp * p / 100});
81   } else {
82     this.setState({ totalPrice: this.state.totalPrice + p });
83   }
84   this.setState({ orders: [...this.state.orders, n]});
85   alert("Jedan proizvod naručen!");
86 }
87
88 mess = () => {
89   ReactDOM.render(<p>Narudžba poslana!</p>, document.getElementById('body'));
90   if (this.state.visibility == false){
91     document.getElementById("narudzba").style.visibility = "hidden";
92     document.getElementById("odustani").style.visibility = "hidden";
93   }
94 }
95
96
97 finish = () => {
98   /*if (localStorage.getItem("token") == null) {
99     alert ("niste logovani!");
100    return;
101  }*/
102  this.setState({ visibility: false });
103  this.setState({ isFinis: !this.state.isFinis });
104  console.log("Funkcija Finish");
105  var arr = this.state.orders;
106  var a = [], b = [], prev;
```

Komponenta Chat



Komponenta Chat nam omogućava stranicu za chat tj omogućava zaposlenicima da šalju poruke međusobno.

U componentDidMount() logovanim userima se dodijeli njihov username za nick.

Sam chat je odrađen preko forme za unos poruke i prikazivaja iste na ploču sa porukama

Komponenta PublicChat



Pored komponente Chat, aplikacija posjeduje i komponentu PublicChat. Ova komponenta je gotovo identična komponenti Chat uz malu modifikaciju. Naime, ovom chatu mogu pristupiti svi posjetitelji stranice, stoga oni sam trebaju unositi svoj nickname. Na identičan način se preuzimaju poruke od korisnika i iste ispisuju na ekran zajedno sa korisničkim nickname-om

Izgled PublicChat komponente (dio bez render motode) je prikazan na sljedećem slide-u.

licChat.js X JS chat.js

an2 > src > components > PublicChat > JS publicChat.js > PubChat

```
constructor(props) {
    super(props);

    this.state = {
        nick: '',
        message: '',
        messages: [],
        hubConnection: new HubConnectionBuilder().withUrl("https://localhost:44342/chat").build()
    };
}

componentDidMount() {
    let self = this;

    this.setState({ nick: prompt("Unesite svoj nick: ")});

    this.state.hubConnection.start().then(() => console.log("Connected!"));
    this.state.hubConnection.on("ReceiveMessage", (user, message) => {

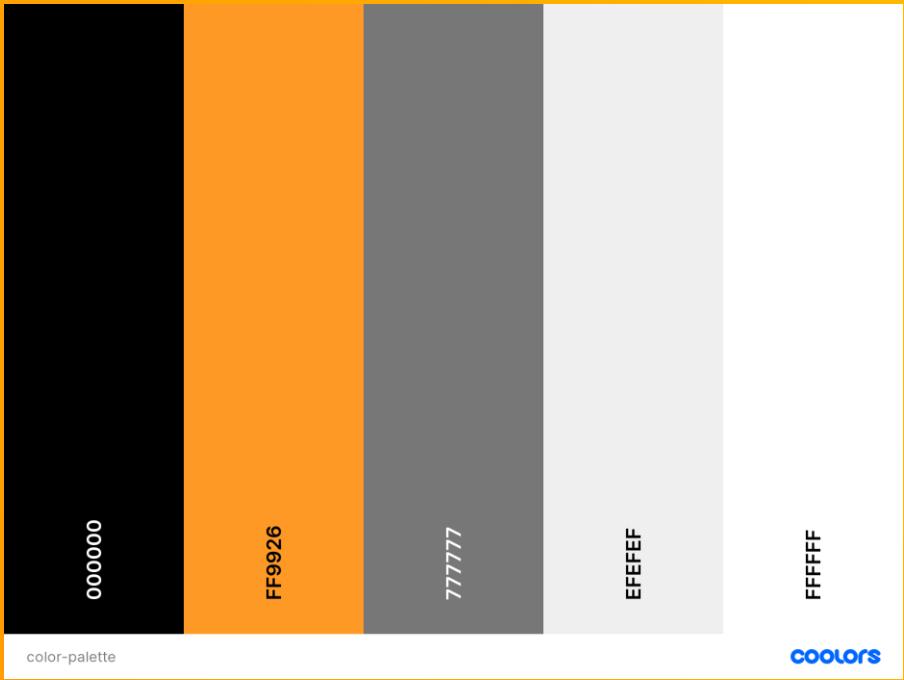
        self.setState({
            messages: [...self.state.messages, `${user}: ${message}`]
        });
    })
}

sendMessage =(e) => {
    e.preventDefault();
    this.state.hubConnection.invoke("SendMessage", this.state.nick, this.state.message);
    this.setState({ message: "" });
}
```

ate Windows
Settings to activate Windows.

Dizajn

Paleta boja korištena pri izradi stranice:



Navigation bar

- Horizontalni meni
- Kreiran uz pomoć biblioteke ‘react-bootstrap’

```
import {Navbar, Nav, NavDropdown, Button} from 'react-bootstrap';
```

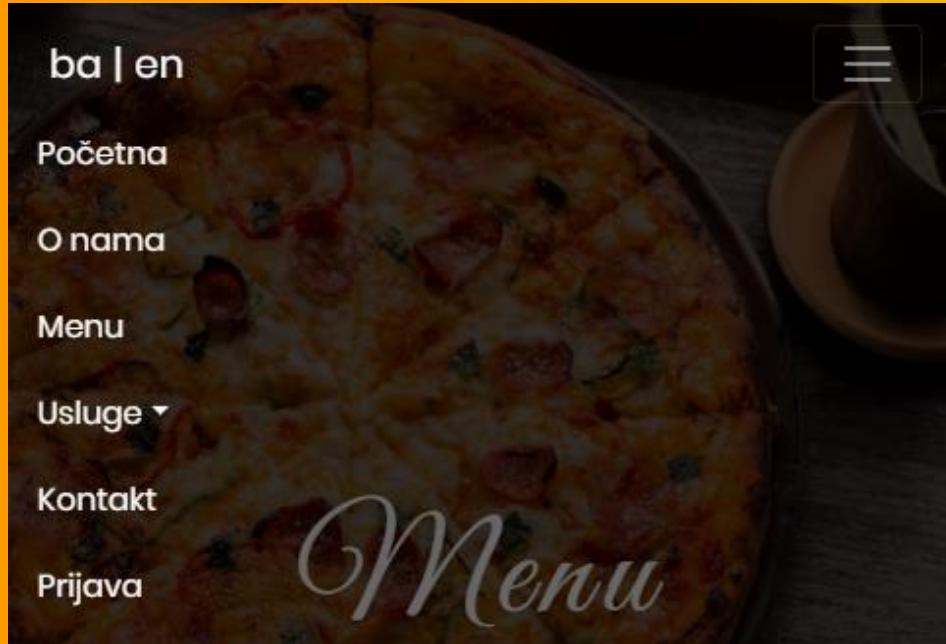
- Sadrži linkove za druge stranice aplikacije
- Sadrži opciju za odabir jezika (bosanski / engleski)
- Dodatna podešavanja izgleda obavljena pomoću css-a
- Fiksirana pozicija na vrhu prozora u čitavoj aplikaciji
- Za mobile uređaje pretvara se u hamburger meni

ba | en

Navigation bar



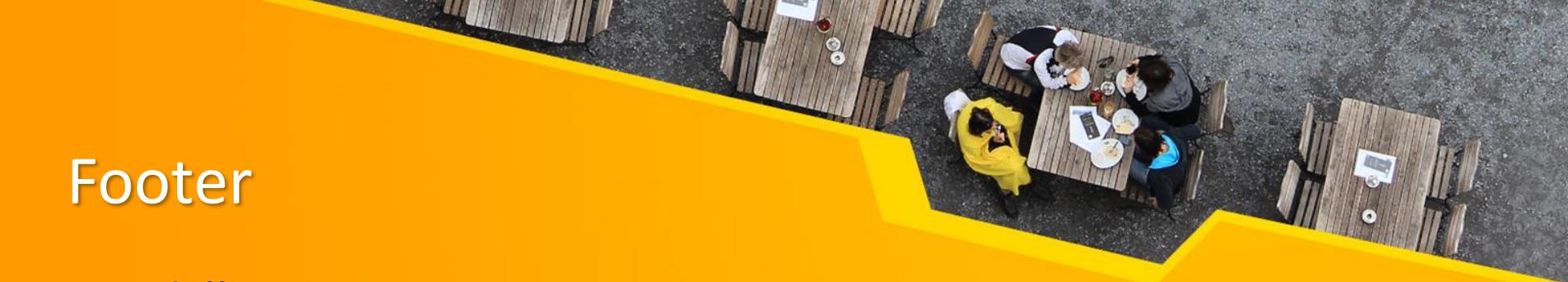
Navigation bar



Footer

Sadrži:

- Kontakt informacije
- Korisni linkovi
- Radno vrijeme
- Linkovi za društvene mreže



Footer



KONTAKT

Zmaja od Bosne 33
 +387 33 10 10 10
 erestoran@gmail.com

LINKOVI

[Rezervacija](#)
[Menu](#)
[Dostava](#)
[Kontaktirajte nas](#)

RADNO VRIJEME

Ponedjeljak - Subota 9:00 - 23:00
Nedjelja 10:00 - 22:00

PRATITE NAS



Welcome Page

- Naslovna stranica aplikacije
- Sadrži kratki prikaz svega što web aplikacija nudi



Welcome Page

ba en

Home About Us Menu Services Contact Sign In

Welcome

HEALTHY & TASTY

Chat See Our Menu

The main visual is a dark, overhead shot of a breakfast spread. It includes a white bowl of cereal or granola, two rectangular bars of granola, a cup of coffee with a basil garnish, a glass of orange juice, and two pieces of toast topped with yogurt and fruit. Sunflower seeds are scattered around the food.

Welcome Page



ba | en

Specijaliteti Naš menu

Početna O nama Menu Usluge Kontakt Prijava



Pohovana piletina 12KM

Piletina, zelena salata, pomfrit

[Naruči](#)



Pohovana piletina 12KM

Piletina, zelena salata, pomfrit

[Naruči](#)



Pohovana piletina 12KM

Piletina, zelena salata, pomfrit

[Naruči](#)



Welcome Page

ba | en

Početna O nama Menü Usluge ▾ Kontakt Prijava

Najbolji Recepti



Tortilja s tartufima



Tortilja s tartufima



Tortilja s tartufima



About Page

ba | en

Početna O nama Menu Usluge Kontakt Prijava

O nama

Restoran

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original

About Page

ba | en

Početna O nama Menu Usluge ▾ Kontakt Prijava

Kuhinja

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections I.10.32 and I.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham. It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.

+387 33 10 10 10

[Rezervišite mjesto](#)



Menu

ba | en

Početna O nama Menu Usluge ▾ Kontakt Prijava

Menu

Doručak



Završi narudžbu

Menu



ba | en

Početna O nama Menu Usluge ▾ Kontakt Prijava

Ručak



Pohovana piletina

12KM

Piletina, zelena salata, pomfrit

Naruči



Pohovana piletina

12KM

Piletina, zelena salata, pomfrit

Naruči



Pohovana piletina

12KM

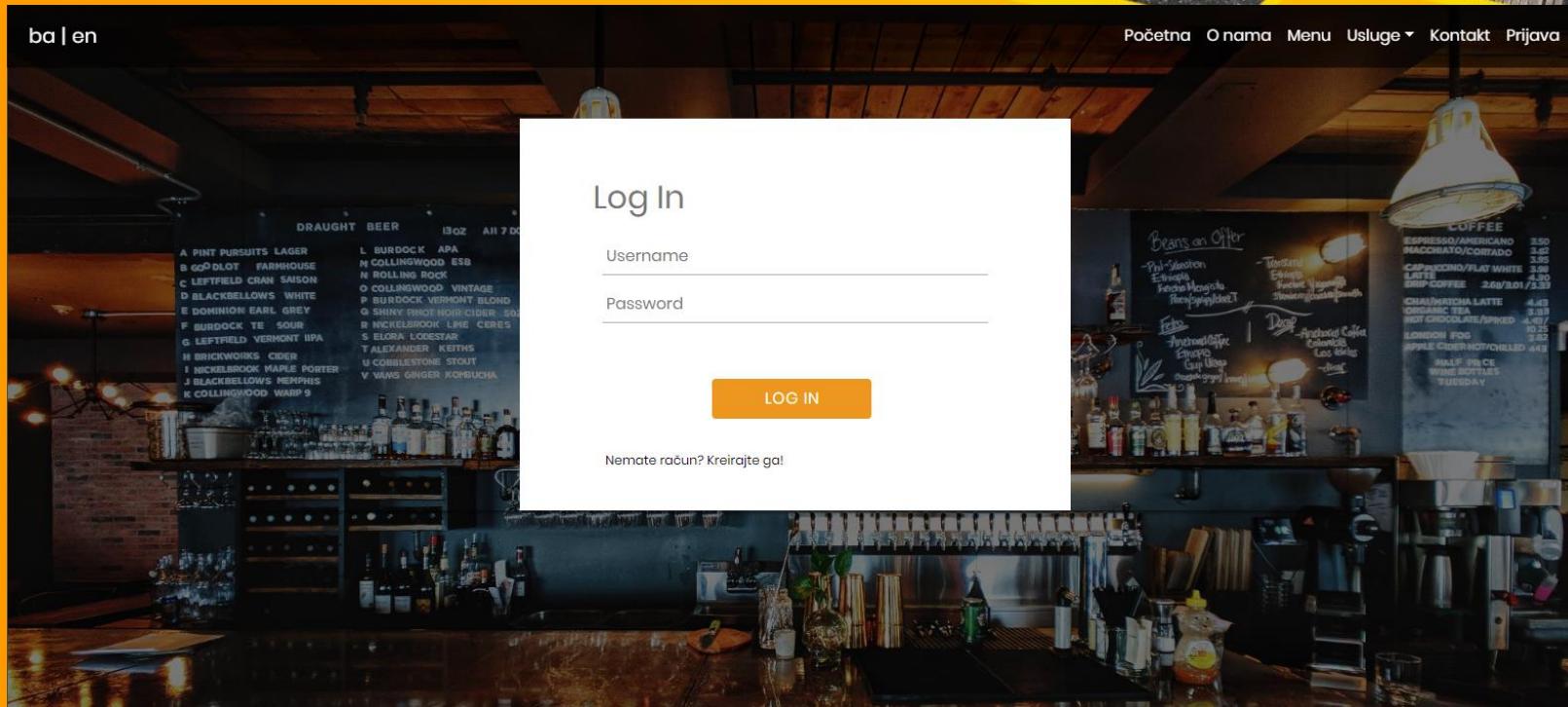
Piletina, zelena salata, pomfrit

Naruči



Završi narudžbu

Sign In Page



Sign Up



Sign Up

Ime

Prezime

Username

Password

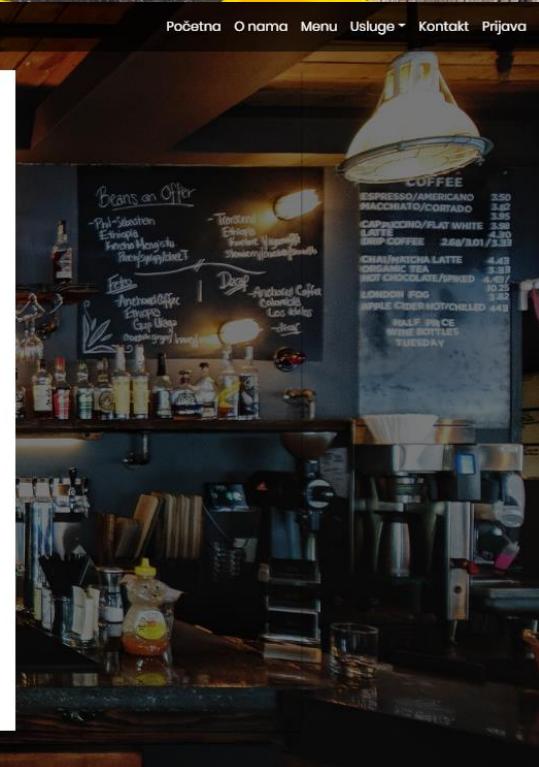
Email

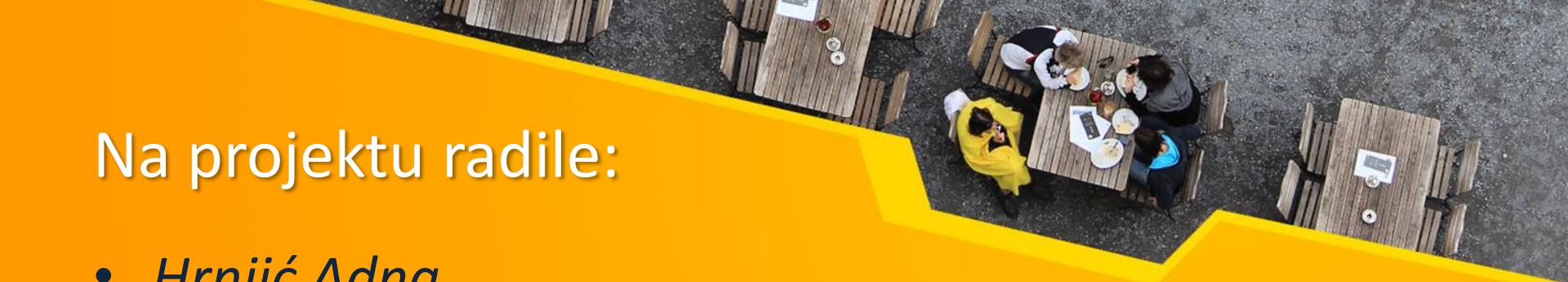
Telefon

Datum rođenja:
 mm/dd/yyyy

Datum zapošlenja:
 mm/dd/yyyy

Restaurant Unit:





Na projektu radile:

- *Hrnjić Adna*
- *Dizdarić Šejla*
- *Alibašić Almedina*

Mentor:

Mr. Skopljaković Selma