Web aplikacija za nadzor rada benzinske postaje

Spajić, Bruno

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:396243

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

Web aplikacija za nadzor rada benzinske postaje

Završni rad

Bruno Spajić

Osijek, 2021.

SADRŽAJ

1. UVOD……1

1.1 Zadatak završnog rada…….....1

2. PREGLED PODRUČJA……...2

3. KORIŠTENE TEHNOLOGIJE…….....4

3.1 HTML……....4

3.2 CSS……5

3.3 JavaScript…….....6

3.4 Node.js…….. 7

3.5 MongoDB…….. 8

4. KORIŠTENJE APLIKACIJE…… 10

4.1 Zahtjevi administratora…….....11

4.2 Zahtjevi djelatnika……..... 16

5. IMPLEMENTACIJA APLIKACIJE…..……..... 20

5.1 Web aplikacija……...20

5.2 Struktura baze podataka……..…26

6. ZAKLJUČAK……28

LITERATURA……29

SAŽETAK……30

ABSTRACT…….....31

ŽIVOTOPIS…….....32

PRILOZI……..33

1. UVOD

Zbog sve veće potražnje tržišta i primjene tehnologije u poslovanju, poslodavci su "osuđeni" na kupovinu ili izradu softvera u svojoj domeni posla. Modernizacija svakog segmenta poslovanja omogućuje što veću produktivnost, preciznost i lakše poslovanje. Upravo radi veće produktivnosti i proširenja posla ulažu se velika financijska sredstva da bih se određeni softveri izradili u svrhu njihovog poslovanja te podigli posao i zaradu na veću razinu. Postoje mnogobrojni primjeri primjene izrade softvera u svrhu poslovanja kao što su web trgovine, softveri u postrojenjima, softveri u trgovinama, bankarski sustavi te mnogi drugi.

U svaku domenu poslovanja gdje se tehnologija primjenjuje, posao i zarada se povećavaju dok se fizički rad i energija uložena u posao značajno smanjuje. Jedna od takvih domena gdje je ključnu ulogu u olakšanju posla omogućio softver je vođenje evidencije rada benzinske postaje. U ne tako davnoj prošlosti benzinske crpke su predstavljale mjesto gdje kupac dolazi kupiti samo naftne derivate iz agregata i to bih bila cjelokupna usluga koju kupac traži. Danas su benzinske crpke skoro pa jedine trgovine koje rade svih 24 sata na dan (čak i blagdanima) i upravo zbog toga razloga je posao otišao u različite domene potražnje kupaca. Pregled područja je opisan u drugom poglavlju, korištene tehnologije prilikom izrade web aplikacije biti će opisane u trećem poglavlju. Način korištenja web aplikacije je opisan u četvrtom poglavlju dok je izrada aplikacije pojašnjena u petom poglavlju.

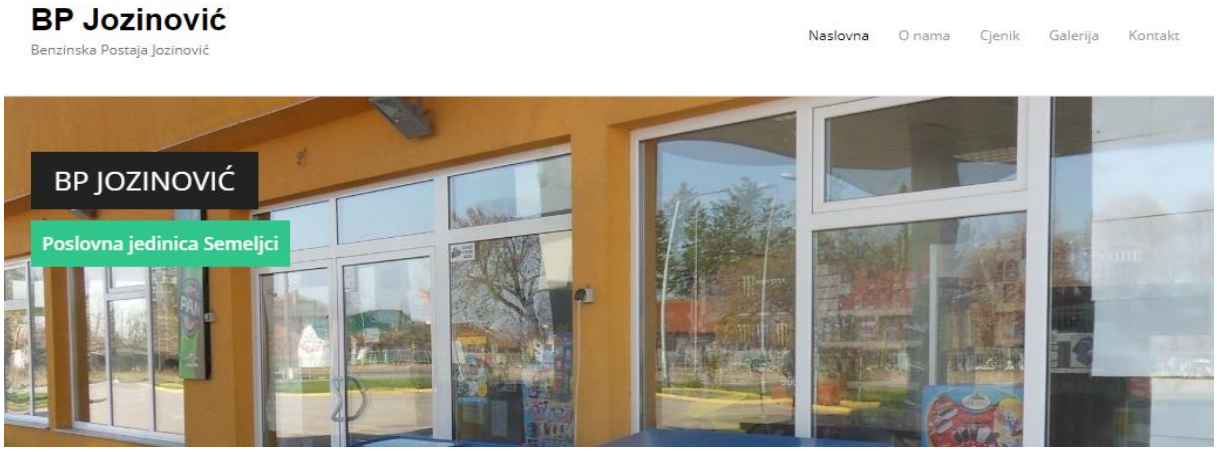
1.1.Zadatak završnog rada

Web aplikacija omogućava prijavu administratora (upravitelja benzinske postaje) i djelatnika benzinske postaje. Administrator ima mogućnost kreiranja djelatnika, uvid u smjene i satnicu svih djelatnika, te rad benzinske postaje. Djelatnik se treba prijaviti kako bi imao uvid u rad benzinske postaje i svoju statistiku radnih sati. Web aplikacija automatski prati stanice za pumpanje guma, LPG/UNP, benzina i dizela, praćenje količine derivat, statistika upotrebe, zauzeća, korištenje pojedine stanice, te sve to sprema u bazu podataka, također i potrebno obavještavanje nadležne osobe za punjenje i održavanje pojedine stanice. Stanica prikazuje trenutne cijene derivata i mogućnost izmjene.

2. PREGLED PODRUČJA

U ovome poglavlju napravit će se pregled dvije stranice od već postojećih benzinskih postaja na području Slavonije: Benz Nuštar i BP Jozinović.

U današnje vrijeme važnost web aplikacija je iznimno velika te može donijeti bitnu prevagu kod savladavanja tržišta ili unapređenja poslovanja poduzeća. Benzinska postaja Benz Nuštar i Benzinska postaja Jozinović na svojim službenim web aplikacijama imaju osnovan sadržaj koji služi samo kao prikaz informacija korisniku kao što su adresa, cijena derivata i kontakt. Ukoliko bi se web aplikacija unaprijedila na način kao što je prikazan u ovome radu poslovanje i vođenje posla bi se značajno pojednostavilo. Putem web aplikacije svaki radnik bi imao točan uvid u svoju smjenu, poznavanje količine dostupnih artikala te lakšu komunikaciju s radnim kolegama zbog mogućnosti izmjena poruka putem web aplikacije. Većina poduzeća svoju komunikaciju provodi putem grupa na društvenim mrežama što i nije poželjno raditi ukoliko želimo podići poslovanje na višu razinu. Web aplikacija pruža mogućnost rada od kuće te svaki zaposlenik ukoliko poželi saznati bilo kakvu vrstu informacije vezane za posao ne mora se fizički pojaviti u poduzeću. Takav pristup informacijama štedi vrijeme i novac što je ključ uspjeha kod poslovanja poduzeća. Na slici 2.1. prikazana je web stranica benzinske postaje Jozinović. Na slici 2.2. prikazana je web stranica benzinske postaje Benz Nuštar.



Slika 2.1. *Web stranica benzinske postaje Jozinović.*

Slika 2.2. *Web stranica benzinske postaje Benz Nuštar.*

3. KORIŠTENE TEHNOLOGIJE

Zbog složenosti zahtjeva web aplikacija se izrađuje pomoću više programskih jezika i alata. Svaki od korištenih alata služi za izvršenje pojedinog dijela aplikacije te skupa tvore funkcionalnu web aplikaciju. Jezici i alati koji su korišteni u ovome radu su:

1. HTML
2. CSS
3. Javascript
4. Node.js
5. MongoDb
6. Atom

Svaka od navedenih tehnologija je detaljnije objašnjena u sljedećim poglavljima.

3.1.HTML

Pomoću HTML-a (engl. *Hyper Text Markup Language*) jezika postavljamo strukturu i sadržaj web aplikacije. Svaka web stranica i aplikacija posjeduje HTML kôd koji je predstavljen kao obična tekstualna datoteka što omogućava jednostavnu izradu HTML koda (bilo koje tekstualno sučelje). Pomoću HTML-a nije moguće izvršavati nikakvu logičku zadaću pa čak niti osnovne matematičke operacije. Glavna svrha je opisivanje web pregledniku na koji će se način određeni elementi pojaviti u web pregledniku. HTML datoteke sadrže proširenje .html ili .htm. Prva inačica je nastala 1993. godine i sadržaj je bio poprilično ograničen dok se u današnje vrijeme koristi HTML5 koji donosi brojne nove mogućnosti kao što je reprodukcija videozapisa, upravljanje pomoću tipkovnice i opcijama za bilo koju vrstu manipulacije te kao ostali brojni novi elementi [1]. Slika 3.1. prikazuje jednostavan HTML kôd.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Slika 3.1. *Prikaz jednostavnog HTML kôda sa osnovnom strukturom.*

3.1 CSS

CSS (engl. *Cascading Style Sheet*) je jezik koji služi za oblikovanje izgleda web stranica. HTML i CSS skupa definiraju cjelokupnu strukturu, sadržaj i izgled web stranica. Glavna je ideja CSS-a odvajanje prezentacijskog kôda u zasebne datoteke i njegovo definiranje pomoću jednostavnih pravila koja se mogu odnositi na više elemenata odjednom. Prva inačica CSS-a definirana je krajem 1996. no do pojave u web preglednicima prolazi dosta dugo vremena. Razvojem CSS-a bavi se Word Wide Web te je trenutačna inačica CSS 3. Korištenjem CSS kôda uređujemo stil stranice nad već ranije definiranim HTML elementima te na taj način poboljšavamo kvalitetu prikaza sadržaja web stranice. Postoje tri načina implementacije CSS koda nad HTML elementima a to su inline CSS, internal CSS i external CSS. Glavna razlika između navedena tri tipa CSS-a je područje definiranja CSS kôda te je najpoželjnije i praktičnije koristiti external CSS (koristiti zasebnu datoteku za stil web stranice)[2]. Slika 3.2. prikazuje CSS kôd.

```css
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

Slika 3.2. *Prikaz CSS kôda.*

3.3 JavaScript

JavaScript je jednostavan, interpretiran programski jezik namijenjen ponajprije razvoju interaktivnih HTML stranica. Jezgra JavaScripta uključena je u većinu današnjih preglednika (Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Safari i drugi).

JavaScript omogućuje izvršavanje određenih radnji u inače statičnim HTML dokumentima, npr. interakciju s korisnikom, promjenu svojstava korisnikovog prozora ili dinamičko stvaranje HTML sadržaja. JavaScript se može pisati u bilo kojem uređivaču teksta (editoru) koji podržava standard ASCII. Sučelje za pisanje (Notepad) je prisutno u svim inačicama operacijskog sustava Windows pa se nameće kao pogodno i za programiranje u JavaScriptu dok zbog složenosti zadatka web aplikacije koristit će se programski alat Atom radi lakšeg snalaženja unutar koda. JavaScript se može koristiti za animacije, navigaciju, jednostavne igre, kalkulatore, validaciju formi te za brojne druge korisničke zahtjeve. Pri izradi web aplikacije preporuka je odvajati (HTML, CSS, JavaScript) dokumente zasebno zbog neovisnosti i lakšeg održavanja kôda.

JavaScript funkcije se mogu pozvati unutar <script> elemenata ili kada nastane određeni događaj. Na primjer: onClick, onMouseDown, onMouseUp, onKeyDown, onKeyUp, onSubnit i mnogi drugi. Dok se JavaScript standard još uvijek koristi za obavljanje osnovnih client-side funkcija, mnogi web dizajneri preferiraju korištenje JavaScript biblioteka poput jQuery-a za dodavanje naprednijih i dinamičnijih elemenata na web stranice [3]. Slika 3.3. prikazuje dohvaćanje HTML elementa putem JavaScripta kôda.

```
<!DOCTYPE html>
<html>
<body>

<h2>Što JavaScript može učiniti?</h2>

<p id="demo">JavaScript može promijeniti HTML sadržaj. </p>

<button type="button" onclick='document.getElementById("demo").innerHTML =
"Tekst promijenjen!"'>Pritisni me!</button>

</body>
</html>
```

Slika 3.3. *Prikaz dohvaćanja HTML elementa putem JavaScript-a i izmjena vrijednosti elementa.*

3.4 Node.js

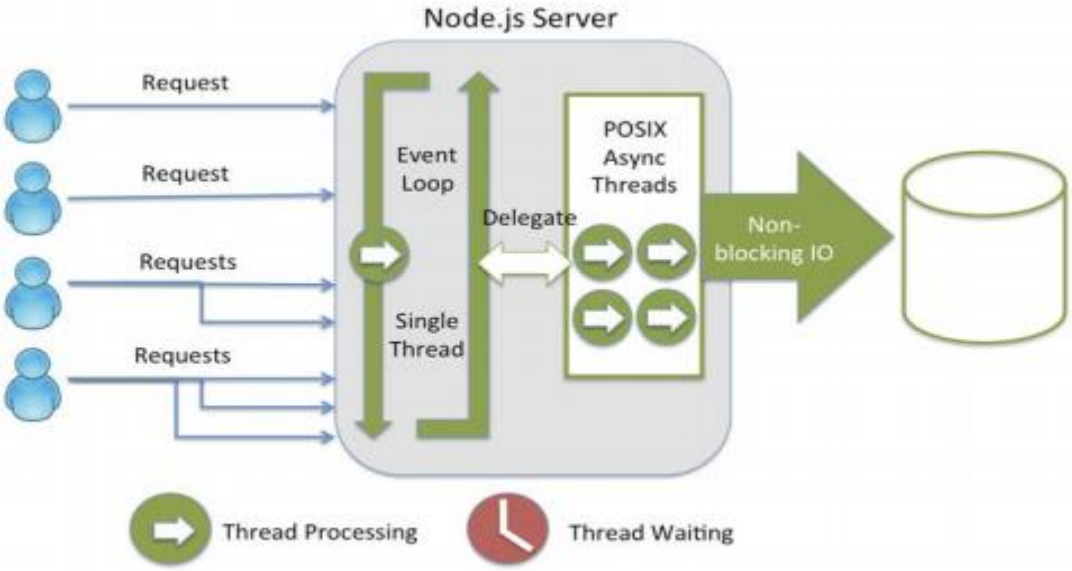
Node.js je JavaScript skalabilna programska platforma koja se odvija na serverskoj strani (engl. *server side*), a za svoj rad koristi JavaScript programski jezik. Razvio ju je Ryan Dahl 2009. godine. Osnovna ideja node.js platforme je implementacija ne blokirajućeg sustava signalizacije čime se postiže skalabilnost i brzina, korištenje novih i bržih načina rada s ulazno-izlaznim sustavima te rad u jednoj petlji sa signalizacijom događaja. Jezgra Node.js-a napisana je koristeći module koji se automatski uključuju u svaki program napisan za ovo okruženje. Node.js sadrži veliki broj već ugrađenih modula s osnovnim, podrazumijevanim funkcionalnostima kao što su pristup datotečnom sistemu, HTTP serveru i slično. Node.js također pruža bogatu biblioteku različitih Javascript modula koji olakšavaju razvoj web aplikacija [4]. Slika3.4. prikazuje Node.js kôd. Arhitektura Node.js-a je prikazana slikom 3.5.

```
app.use(session({
    secret: 'fusionauth',
    resave: false,
    saveUninitialized: true,
}))

app.get('/', function(req, res) {
    res.send('Hello World!');
});

app.get('/logout', function(req, res) {
    req.session.destroy()
    res.send("Successfully logged out");
});
```

Slika 3.4. *Prikaz Node.js kôda.*



Slika 3.5. *Node.js arhitektura.*[5]

3.5 MongoDB

MongoDB je snažna, fleksibilna i skalabilna baza podataka opće namjene. Osim kreiranja, čitanja, ažuriranja i brisanja podataka (CRUD metode) pruža sljedeće jedinstvene značajke:

1. Pohrana podataka - podržava protokol koji je jednostavan za korištenje a služi za pohranjivanje velikih datoteka i njihovih metapodataka.

2. Agregacija - MongoDB dopušta izradu kompleksnih agregacija iz jednostavnih dijelova i dozvoljava bazi da ih optimizira (engl. *aggregation pipeline*).
3. Indeksiranje - podržava sekundarne indekse što omogućuje razne brze upite i pruža jedinstvene, složene i geoprostorne mogućnosti indeksiranja.

Dokument (engl. *document*) je osnovna jedinica podataka u MongoDB-u. Dokument je ugrubo ekvivalent redu (engl. *row*) u relacijskom sustavu za upravljanje bazom podataka. Dokument se sastoji od jednog ili više polja (engl. *fields*), koje možemo usporediti s stupcima (engl. *column*s) u relacijskom svijetu. Slično, kolekcija (engl. *collection*) se može shvatiti kao tablica s dinamičnom shemom. Dakle, kolekciju čini grupa dokumenata. Kolekciju se identificira prema njenom nazivu. Naziv može biti bilo koji UTF-88 niz znakova s nekoliko ograničenja – ne smije biti prazan (" "), ne smiju sadržavati znak \0 (engl. *null*), ne smije počinjati s riječi system. jer je ona rezervirana za interne kolekcije, ne smije sadržavati znak $. Osim što se dokumenti grupiraju u kolekcije, kolekcije se grupiraju u baze podataka. Jedna instanca MongoDB-a može posluživati više nezavisnih baza podataka, od koje svaka može imati svoje kolekcije. Svaki dokument ima specijalni ključ. "_id", koji je jedinstven unutar kolekcije[6]. Tablica 3.1. prikazuje tipove podataka koji su korišteni u MongoDB bazi podataka.

Tablica 3.1. *Tipovi podataka korišteni u MongoDB bazi podataka.*

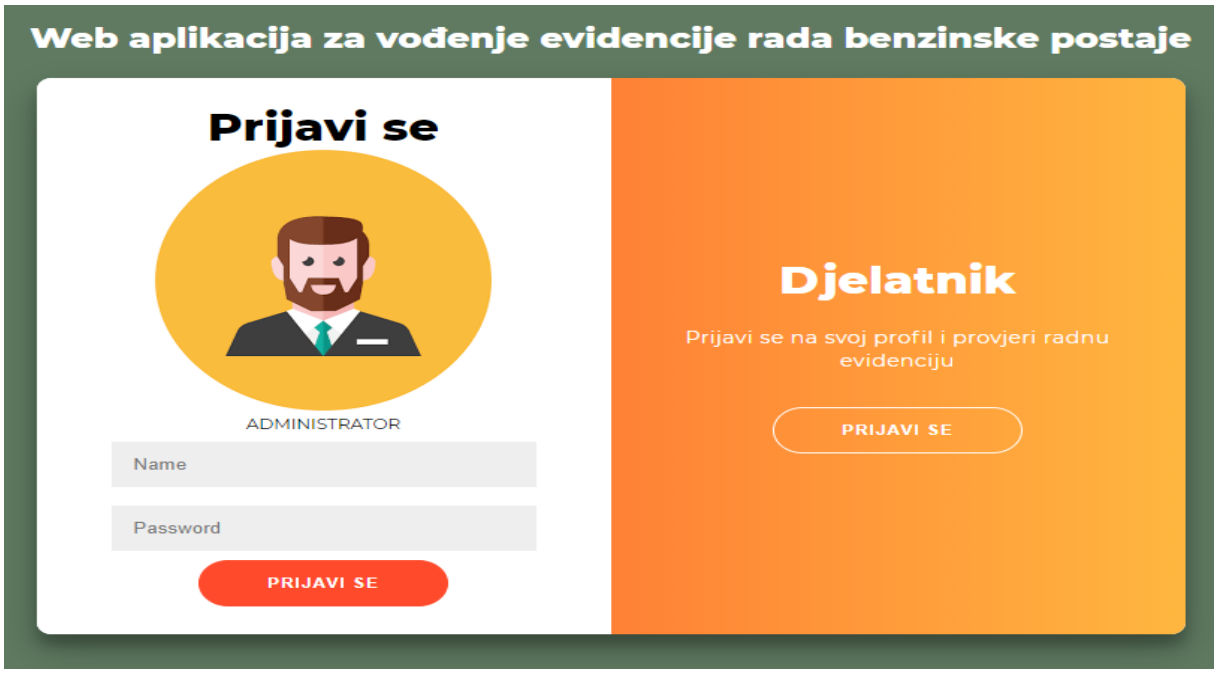
Tip podatka	Primjer korištenja
Number	{"x" : 3.14}
Date	{"x" : new Date()}
regular expression	{"x" : /foobar/i}
Array	{"x" : ["a", "b", "c"]}
embedded document	{"x" : {"foo" : "bar"}}
object id	{"x" : ObjectId()}
binary dana	/
Code	{"x" : function() { /* ... */ }}

4. KORIŠTENJE APLIKACIJE

Većina poslodavaca za svoj posao odabiru web aplikacije upravo zbog svoje pristupačnosti u bilo koje vrijeme s bilo kojeg mjesta. Internet aplikacije polako preuzimaju tržište jer su najbliže i najlakše za funkcionirati s kupcima odnosno potrošačima. Danas je gotovo nezamislivo imati poduzeće a da ne postoje internet aplikacije ili stranice koje služe u svrhu prodaje ili promoviranja posla tog poduzeća.

Ideja web aplikacije za nadzor rada benzinske postaje je da u svakom trenutku bilo koja osoba povezana sa radom na benzinskoj postaji ima uvid u podatke poslovanja, rasporeda, robe, stanja derivata itd. Sve benzinske postaje imaju svoje softvere dok niti jedna od tih postaja nije direktno povezana s internet preglednikom što znači da ako se poželi vidjeti bilo kakvo stanje ili informaciju vezanu za točno određenu benzinsku postaju trebalo bih se fizički pojaviti na benzinskoj postaji i zatražiti potrebne informacije s računala koji posjeduje softver za poslovanje. Ukoliko se izradi web aplikacija koju će moći koristiti radnici i poslovođa benzinske postaje, poslovanje i rad će biti mnogo jednostavnije.

Aplikacija se sastoji od dvije korisničke prijave (Prikazano na slici 4.1.). Prva prijava biti će administratorska (voditelj benzinske postaje) jer će tu prijavu koristiti upravitelj benzinske postaje dok će druga prijava biti namijenjena djelatnicima benzinske postaje. Svaka korisnička prijava se sastoji od različitih mogućnosti upotrebe. Administratorska prijava posjeduje sva prava web aplikacije, uvid u svaki podatak te pravo na dodavanje, brisanje ili nadograđivanje podataka unutar aplikacije. Zaposlenik kroz svoju prijavu ima pravo na uvid u sve podatke potrebne za rad te mu stižu obavijesti putem aplikacije koje služe u svrhu dobro obavljenog posla. Zaposlenička prijava ne može posjedovati sva prava uvida upravo iz svrhe očuvanja poslovnih tajni i sredstava.



Slika 4.1. *Prikaz početnog korisničkog sučelja.*

4.1. Zahtjevi administratora

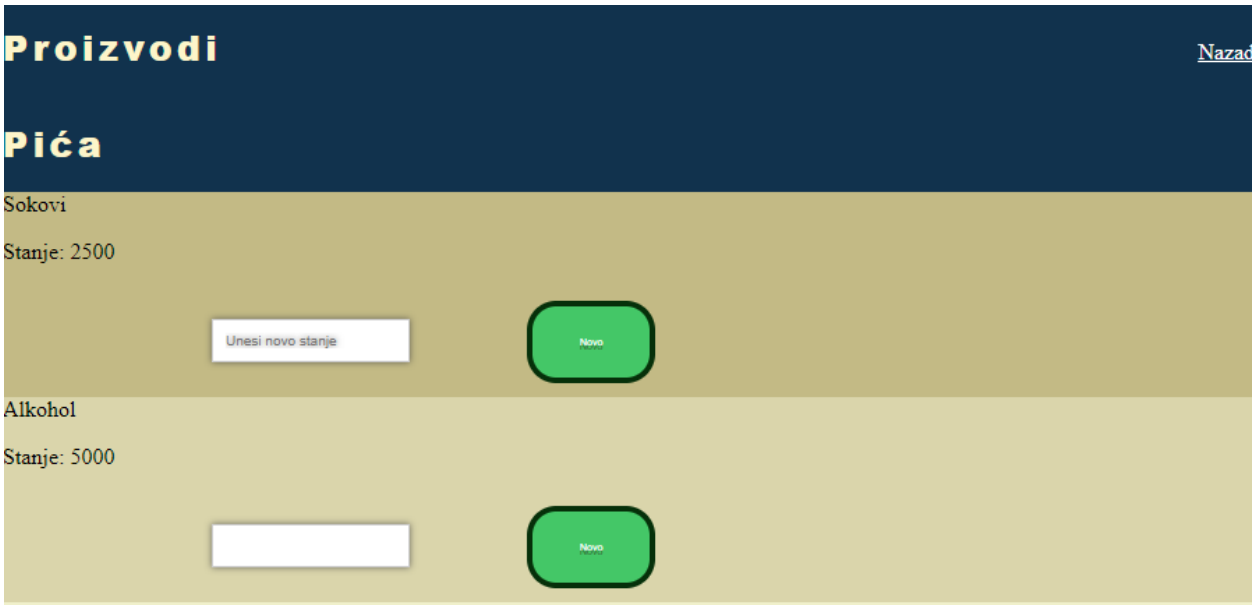
Prilikom administratorske prijave provjerava se da li je korisnik prijavljen te da li mu je uloga (engl. *Role*) administratorska. Ukoliko su uvjeti zadovoljeni otvara se sučelje od administratora, prikazano slikom 4.2. te administrator odabire željenu uslugu.

Slika 4.2. *Prikaz administratorskog sučelja.*

Nakon što se pokrene početno administratorsko sučelje aplikacija će obavijestiti administratora ukoliko je stanje pojedinog agregata manje od 250 litara. Slika 4.3. prikazuje administratorsku obavijest.

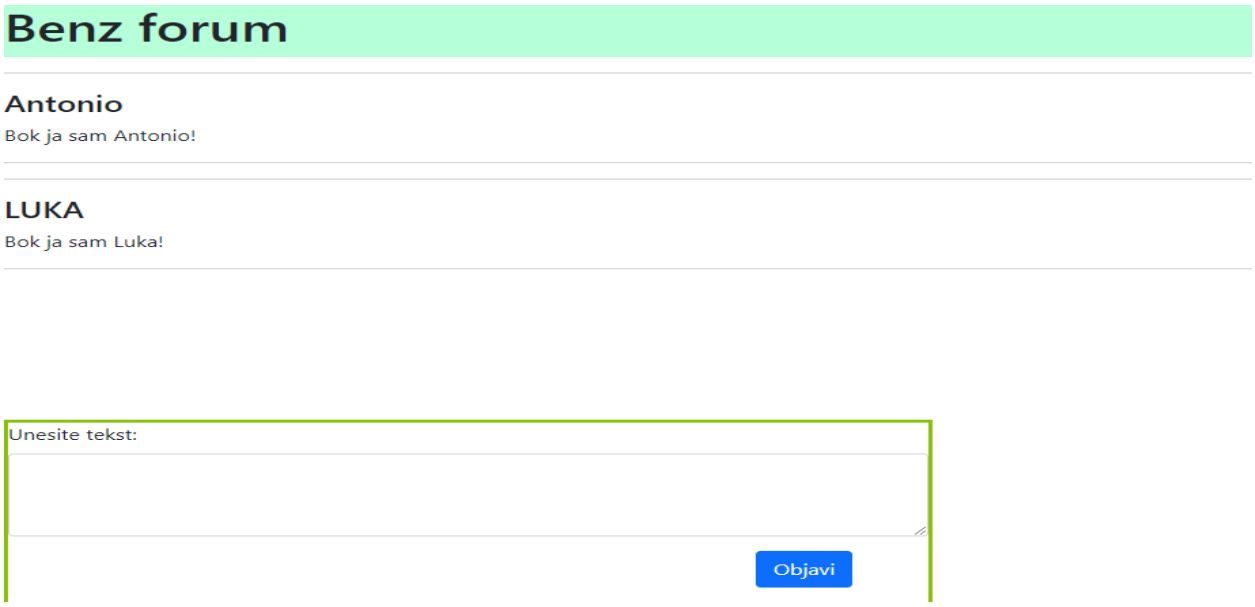
Slika 4.3. *Prikaz obavijesti za EuroSuper95 Class Plus.*

Ukoliko administrator odabere opciju Proizvod otvoriti će se sučelje prikazano slikom 4.4. u kojem administrator posjeduje uvid trenutnog stanja svih proizvoda kao i mogućnost izmjene stanja.



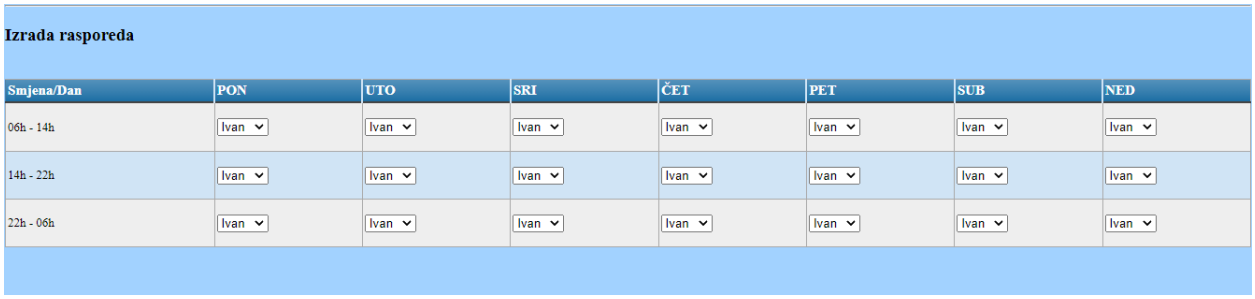
Slika 4.4. *Prikaz administratorskog sučelja opcije Proizvod.*

Prilikom ulaska na opciju Forum otvoriti će se sučelje prikazano slikom 4.5. u kojem se nalaze poruke od svih zaposlenika benzinske postaje. Iznad poruka se nalazi ime korisnika koji je napisao poruku. Administrator ima mogućnost kreiranja nove poruke koja je vidljiva svim djelatnicima benzinske postaje unutar opcije Forum.



Slika 4.5. *Prikaz sučelja Forum.*

Unutar opcije Raspored administrator posjeduje mogućnost kreiranja novog tjednog rasporeda koji je vidljiv svim djelatnicima benzinske postaje. Prilikom izrade rasporeda administratoru su ponuđeni svi trenutni zaposlenici benzinske postaje te pritiskom na opciju Potvrdi kreira se novi tjedni raspored. Slika 4.6. prikazuje sučelje za izradu tjednog rasporeda.



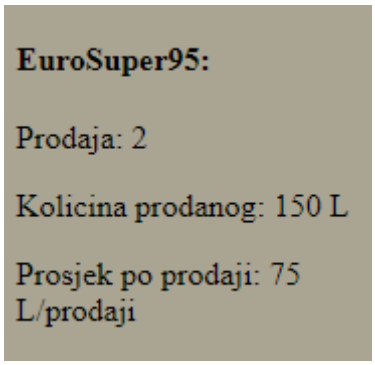
Slika 4.6. *Prikaz sučelja za izradu tjednog rasporeda.*

Opcija Kreiraj služi za kreiranje novog djelatnika benzinske postaje. Sučelje za kreiranje novog djelatnika je prikazano slikom 4.7.

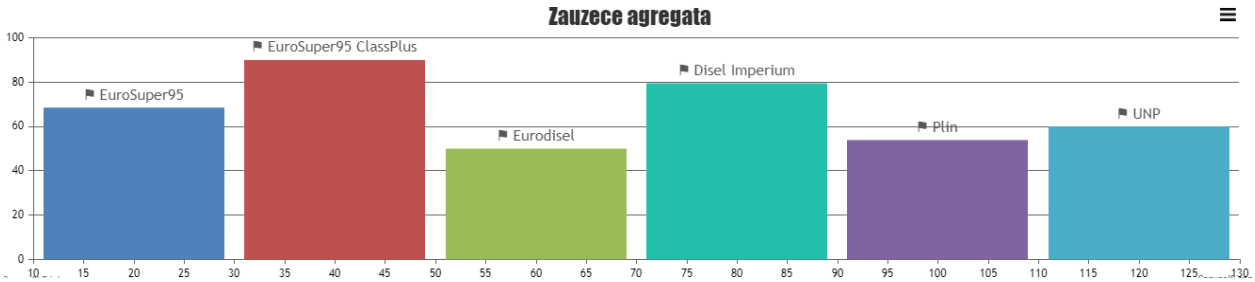
Slika 4.7. *Prikaz sučelja za izradu novog djelatnika.*

Unutar opcije Agregati administrator posjeduje mogućnost izmjene cijene i stanja svakog agregata. Također posjeduje mogućnost postavljanja zauzeća agregata u dva stanja. Prvo stanje je "U pogonu" što omogućuje daljnju upotrebu agregata i postavlja zeleno svijetlo na agregatu. Drugo stanje je "U kvaru" te predstavlja agregat koji nije u pogonu. Ukoliko je agregat postavljen na stanje "U kvaru" svijetlo na agregatu je crveno. Ukoliko je na agregatu postavljeno crveno svijetlo djelatnik nije u mogućnosti prodavati derivat sa toga agregata. Slika 4.8. prikazuje agregat koji predstavlja EuroSuper95 vrstu goriva. Ispod agregata se nalazi statistika prodaje svakog od postojećih agregata kao i grafički prikaz stanja svakog agregata. Slika 4.9. prikazuje statistiku prodaje EuroSuper95 agregata. Slika 4.10. prikazuje statistički prikaz stanja svih agregata.

Slika 4.8. *Prikaz EuroSuper95 agregata.*



Slika 4.9. *Prikaz statistike prodaje EuroSuper95 agregata.*



Slika 4.10. *Statistički prikaz stanja agregata.*

Pritiskom na opciju Djelatnici otvara se novo sučelje koje je prikazano slikom 4.11. Unutar sučelja se ispisuju svi trenutni djelatnici benzinske postaje te administrator ima mogućnost brisanja djelatnika.

Slika 4.11. *Prikaz sučelja Djelatnici.*

4.2 Zahtjevi djelatnika

Prilikom djelatničke prijave provjerava se da li je djelatnik prijavljen te da li mu je uloga (engl. *Role*) djelatnička. Ukoliko su uvjeti zadovoljeni otvara se sučelje od djelatnika (Prikazano slikom 4.12.) te odabire željenu uslugu.

Slika 4.12. *Prikaz djelatnikovog sučelja.*

Ukoliko djelatnik odabere opciju Proizvod otvoriti će se novo sučelje koje je prikazano slikom 4.13. Unutar sučelja posjeduje mogućnost uvida u stanje svih proizvoda unutar benzinske postaje.

Slika 4.13. *Prikaz sučelja Proizvod.*

Unutar opcije Forum djelatnik posjeduje ista prava kao i administrator. Svaki komentar koji je objavljen u sučelju Forum vidljiv je svim zaposlenicima benzinske postaje.

Raspored koji je izrađen unutar administratorskog sučelja vidljiv je svim djelatnicima unutar opcije "Raspored". Slika 4.14. prikazuje djelatnikovo sučelje rasporeda.

Slika 4.14. *Prikaz tjednog rasporeda unutar opcije Raspored.*

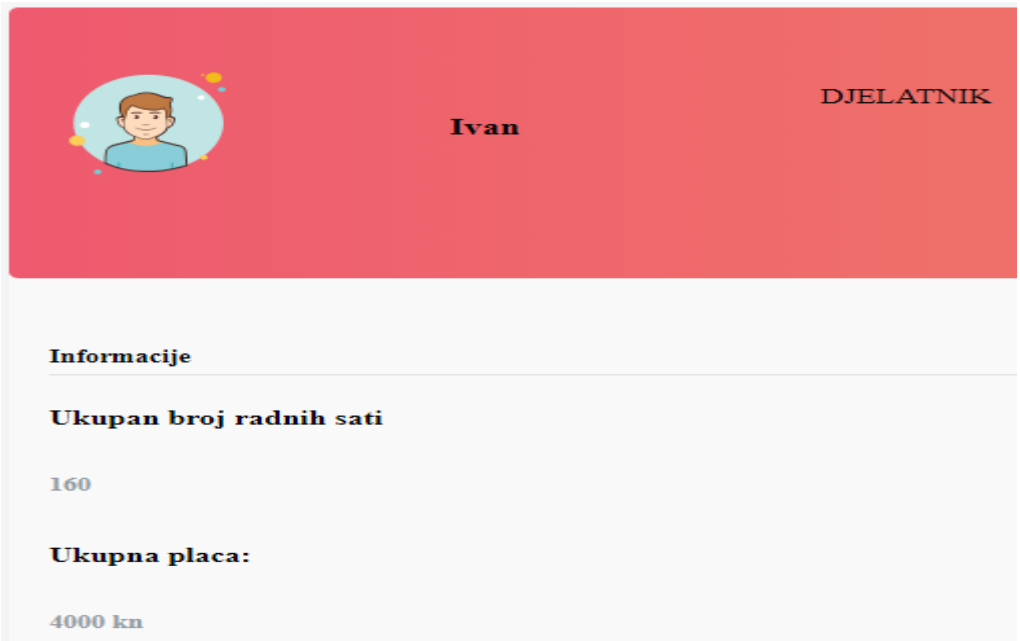
Prilikom korištenja opcije "Prodaja" djelatnik unosi količinu koja se prodala u odgovarajuće sučelje. Ukoliko djelatnik unese veću količinu od stanja proizvoda kojeg prodaje, prodaja će biti onemogućena zbog nedostatka proizvoda na stanju. Prodaja derivata će biti omogućena samo u slučaju narančastog svijetla na agregatu koji se prodaje. U slučaju da se na agregatu nalazi zeleno ili crveno svijetlo prodaja neće biti moguća zbog toga što jedino narančasto svijetlo predstavlja natočenu količinu. Zeleno svijetlo predstavlja agregat koji je u funkciji i spreman je za upotrebu dok crveno svijetlo predstavlja agregat u kvaru. Slika 4.15. prikazuje sučelje za djelatnikov unos prodaje.

Slika 4.15. *Prikaz djelatnikovog sučelja Prodaja.*

Unutar opcije "Agregati" djelatnik posjeduje uvid u cijenu i stanje agregata koji su postavljeni putem administratorskog sučelja. Djelatnik također ima uvid u zauzeće agregata. Ukoliko je svijetlo zauzeća zeleno djelatnik ima mogućnost postaviti svijetlo na narančasto ili crveno. Nakon što se zauzeće postavi na narančasto svijetlo potrebno je prodati natočenu količinu unutar opcije "Prodaja" da bih se svjetlo vratilo na zeleno. Ukoliko djelatnik pritisne crveno svijetlo na agregatu postavlja agregat u stanje kvara te je moguće vratiti zeleno svijetlo isključivo putem administratorskog sučelja. Agregat u stanju kvara nema mogućnost prodaje sve dok administrator ne postavi da je agregat ponovo u pogonu. Slika 4.16. prikazuje Disel Imperium agregat sa kojeg se može iščitati trenutno stanje, cijena i zauzeće agregata (agregat je postavljen u stanje kvara što nam prikazuje crveno svijetlo).

Slika 4.16. *Prikaz Disel Imperium agregata.*

Prilikom odabira na opciju "Moj profil" otvoriti će se sučelje prikazano slikom 4.17. u kojem djelatnik ima pravo uvida u svoju statistiku radnih sati tekućeg tjedna te ukupnu zaradu na osnovu radnih sati u tjednu.



Slika 4.17. *Prikaz djelatnikovog profila.*

5. IMPLEMENTACIJA APLIKACIJE

U ovome poglavlju je predstavljen način izrade web aplikacije za nadzor rada zaposlenika benzinske postaje. Web aplikacija je podijeljena na dva osnovna korisnička slučaja (administrator i djelatnik). Web aplikacija je rađena u programu Atom koji je zapravo jedan od brojnih drugih uređivača teksta no vrlo jednostavan za primjenu i korištenje u razvoju web aplikacija sa mogućnošću korištenja već unaprijed deklariranih biblioteka.

5.1 Web aplikacija

Prilikom izrade web aplikacije ponajprije se izrađuju HTML dokumenti vezani za strukturu stranice. U HTML kodu deklariraju se svi elementi koji se žele prikazati na odabranoj stranici. Unutar oznaka <html> se nalazi HTML kôd dok se unutar <head> zaglavlja nalaze sve poveznice meta podataka koje želimo koristiti za bolju manipulaciju nad HTML elementima. Željene elemente postavljamo unutar <body> oznake te im se mogu pridružiti različite klase i Id-ovi radi lakšeg dohvaćanja željenog elementa. Slika 5.1. prikazuje dio HTML kôda iz web aplikacije.

```
<h2>Web aplikacija za vođenje evidencije rada benzinske postaje</h2>
<div class="container" id="container">
  <div class="form-container sign-up-container">
    <form action="/user" method="post">

      <h1>Prijavi se</h1>
      <div class="slikaDjelatnika">
        <img src="Dje.png" alt="">
      </div>

      <span>DJELATNIK</span>
      <input type="text" name="username" placeholder="Name" />
      <input type="password" name="password" placeholder="Password" />
      <button>Prijavi se</button>
    </form>
```

Slika 5.1. *Prikaz HTML kôda iz web aplikacije.*

Tijekom izrade web aplikacije potrebno je kvalitetno i ugodno rasporediti sadržaj stranice te ga adekvatno urediti zbog lakšeg korisničkog snalaženja i ugodnijeg iskustva. Upravo iz toga razloga koristi se CSS za uređivanje sadržaja web aplikacije. Unutar <head> oznaka u HTML kodu pridružuje se CSS dokument s HTML dokumentom te se dohvaćaju svi podaci nad kojima se mogu vršiti razne manipulacije. Slika 5.2. prikazuje dio CSS kôda iz web aplikacije.

```
body {
        background-color: #5F7A61;
        display: flex;
        justify-content: center;
        align-items: center;
        flex-direction: column;
        font-family: 'Montserrat', sans-serif;
        height: 100vh;
        margin: -20px 0 50px;
}

h1 {
        font-weight: bold;
        margin: 0;
```

Slika 5.2. *Prikaz CSS kôda iz web aplikacije.*

Zbog specifičnosti zahtjeva web aplikacije potrebno je podijeliti aplikaciju na dva osnovna dijela (administrator i djelatnik). Aplikacija prepoznaje korisnikovu ulogu zbog dodatnog atributa "role" (uloga) dodijeljenog u MongoDB bazi podataka. Postoje dvije vrste uloga to su "admin" i "user". Slika 5.3. prikazuje kôd koji provjerava pravo pristupa.

```
app.get("/kreiraj", function(req, res) {
  if (req.isAuthenticated() && req.user.role === "admin"){
    res.render("kreiraj");
  } else {
    res.redirect("/");
  }
});
```

Slika 5.3. *Prikaz provjere prava pristupa stranici kreiraj.*

U početku rada aplikacije potrebno je unutar Node.js-a unaprijed deklarirati administratorski profil jer web aplikacija nema mogućnost kreiranja novog administratora. Broj djelatnika na početku iznosi nula pa je upravo zbog toga moguć samo jedan pristup stranici i to je administratorski. Tek nakon prijave administratora postoji mogućnost kreiranja novog djelatnika te na taj način povećati broj zaposlenika benzinske postaje koji imaju pravo pristupa web aplikaciji. Slika 5.4. prikazuje kreiranje administratorskog profila unutar kôda.

```
User.register({username: "admin", role: "admin"}, "PASSWORD", function(err, user) {
  if (err) {
    console.log(err);
  } else {
    passport.authenticate("local") (req, res, function() {
        res.redirect("/");
    });
  }
});
```

Slika 5.4. *Kreiranje administratorskog profila.*

Nakon što se administrator prijavi u svoje radno sučelje putem JavaScript-a i Node.js-a, provjerava se stanje svakog agregata. Ukoliko je stanje agregata manje od 250 litara na ekranu , putem JavaScript-a ispisuje se obavijest o kojem je agregatu riječ te obavještava administratora da je potrebno novo punjenje agregata. Administrator ima mogućnost izmjene cijene i stanja svakog agregata. Slika 5.5. prikazuje način promjene stanja i cijene agregata.

```
app.post("/benzin", function(req, res) {
  const stanje = req.body.stanje;
  const cijena = req.body.cijena;

  Agregati.updateMany({ime:"Benzin"}, {stanje: stanje, cijena: cijena}, function(err){
    if (!err){
      res.redirect("/agregati");
    } else {
      res.send("Nesto je poslo po krivu. Probajte ponovno.");
    }
  });
});
```

Slika 5.5. *Primjer promjene cijene i stanja agregata (Eurosuper95).*

Unutar korisničke opcije "Agregati" potrebno je razviti logiku zauzeća agregata. Postoje tri vrste zauzeća. Prvo zauzeće se označava zelenom bojom i predstavlja agregat u pogonu slobodan za korištenje. Ukoliko se agregat započne koristiti zauzeće se postavlja na drugu vrijednost predstavljena sa narančastom bojom. Agregat nije moguće vratiti sa narančastog stanja u zeleno stanje sve dok se u korisničkoj opciji "Prodaja" ne unese odgovarajuća količina i izvrši se prodaja derivata. Treća vrijednost zauzeća predstavljena je crvenom bojom i označava agregat u kvaru. Ukoliko je zauzeće označeno crvenom bojom samo administrator posjeduje ovlasti vraćanja na zelenu vrijednost zauzeća agregata. Slika 5.6. prikazuje kôd za pritisak na zelenu tipku zauzeća.

```
app.post("/provjeraZaBenzinZelena", function(req, res) {
  Zauzece.findOne({
    agregat: "Benzin"
  }, function(err, foundAgregat) {
    if (foundAgregat.value === "orange") {
      res.send("Prodaj prvo");
    } else if (foundAgregat.value === "red") {
      Zauzece.updateMany({
        agregat: "Benzin"
      }, {
        value: "red"
      }, function() {
        res.redirect("/djelatnik");
      })
    } else if (foundAgregat.value === "green") {
      Zauzece.updateMany({
        agregat: "Benzin"
      }, {
        value: "green"
      }, function() {
        res.redirect("/djelatnik/agregati");
      })
    }
  })
```

Slika 5.6. *Kôd koji predstavlja logiku za pritisak na zelenu tipku zauzeća* .

U korisničkoj opciji "Proizvodi" administrator koristiti dvije metode (*post* i *get*) dok djelatnik koristi samo *get* metodu. Djelatnik putem *get* metode dohvaća bazu podataka "Proizvod" pomoću koje ispisuje stanja svakog proizvoda. Administrator uz *get* metodu koristi i *post* metodu pomoću koje se dohvaća unesen broj i mijenja se stanje proizvoda pritiskom na gumb.

Opcija "Forum" služi za lakšu komunikaciju između zaposlenika benzinske postaje. Pristup forumu imaju svi registrirani korisnici web aplikacije. Korisnik u donjem lijevom dijelu ekrana unosi svoj komentar te nakon što potvrdi unos (pritiskom na gumb) komentar se sprema na forum.

Komentar je vidljiv svim zaposlenicima. Svaki put kada se postavlja komentar na forum dohvaća se ime djelatnika koji ga je napisao i ispisuje se iznad komentara. Način objave komentara prikazan je slikom 5.7.

```
app.post("/forum", function(req, res) {
  const noviKomentar = new Forum({
    ime: req.user.username,
    komentar: req.body.komentar
  });
  noviKomentar.save();

  res.redirect("forum");
});
```

Slika 5.7. *Forum post metoda.*

Kod izrade tjednog rasporeda u opciji "Raspored" administrator ima dvije tablice kreirane u HTML-u. Jedna tablica je kreirana u svrhu izrade rasporeda dok je druga tablica prikaz trenutnog rasporeda. Pritiskom na bilo koji dan u prvoj tablici otvara se lista svih djelatnika benzinske postaje i administrator postavlja željenog radnika za odabranu smjenu. Nakon što administrator potvrdi promjene pritiskom na gumb "Potvrdi", mijenja se stanje druge tablice. Djelatnik ima pravo uvida samo u drugu tablicu jer je to tablica koja je kreirana od strane administratora te ona predstavlja trenutni raspored. Slika 5.8. prikazuje način dohvaćanja svih djelatnika prilikom izrade rasporeda.

```
<td>
  <form action="/raspored" method="post">
    <select id="User" name="pon1">
      <%foundUsers.forEach(function(user) {%>
        <option ><%= user.ime %></option>
      <%});%>
    </select>


</td>
```

Slika 5.8. *Dohvaćanje svih djelatnika za odabir radnika.*

Jedna od glavnih funkcija djelatnika je prodaja proizvoda i agregata. To je jedina stranica na koju administrator nema pristup jer on nije zadužen za prodaju. Djelatnik ima mogućnost unosa prodane količine za bilo koji proizvod ili agregat. Postoje dva glavna uvjeta za izradu prodaje. Prvi uvjet

je da unesena količina nije veća od trenutnog stanja jer benzinska postaja onda nije u mogućnosti ispostaviti unesenu količinu. Drugi uvjet se odnosi na prodaju derivata. Ukoliko se određeni derivat prodaje potrebno je da zauzeće toga agregata bude postavljeno na narančastu vrijednost zbog toga što narančasta vrijednost predstavlja natočeni agregat spreman za naplatu. Slika 5.9. prikazuje primjer kôda za prodaju derivata.

```
Zauzece.findOne({
  agregat: "Plin"
}, function(err, foundAgregat) {
  if (foundAgregat.value === "orange") {
    Agregati.findOne({
      ime: "plin"
    }, function(err, pronadenProizvod) {
      const x = pronadenProizvod.stanje;
      if (x >= prodano) {
        Agregati.updateMany({
          ime: "plin"
        }, {
          stanje: x - prodano
        }, function(err) {
          if (!err) {
            Zauzece.updateMany({
              agregat: "Plin"
            }, {
              value: "green"
            }, function() {
```

Slika 5.9. *Primjer kôda za prodaju plina.*

U nastavku istog kôda obavljamo funkcionalnost provođenja statistike prodaje. Svaki put kada se prodaja derivata uspješno provede, broj unutar baze podataka se uvećava za 1 dok se količina uvećava za prodani iznos. Slika 5.10. prikazuje primjer provođenja statistike prodaje za agregat plina.

```
  Statistika.findOne({
    agregat: "Plin"
  }, function(err, agr) {
    const broj = agr.broj;
    const kolicina = agr.kolicina;
    Statistika.updateMany({
      agregat: "Plin"
    }, {
      broj: broj + 1,
      kolicina: kolicina + +prodano
    }, function() {
      res.redirect("/djelatnik/prodaja");
    })
  })
});
```

Slika 5.10. *Primjer kôda za statistiku prodaje plina.*

Opcija "Moj Profil" služi primarno za prikaz vlastitih podataka i statistiku radnih sati. Kroz bazu podataka od rasporeda možemo pretražiti svakog djelatnika te izračunati koliko puta se pojavljuje u rasporedu. Koliko god puta da se radnik pojavio taj broj će se pomnožiti s osam (jer jedna smjena ima osam sati) i dobit će se ukupan broj radnih sati tekućeg tjedna. Također se ispisuje djelatnikova zarada toga tjedna na osnovu radnih sati. Slika 5.11. prikazuje funkcije za izračun plaće i radnih sati.

```
function ispisRadnihSati(req, res, next) {
  res.locals.radniSati = foundUser * 8;
  next();
};

function placa(req, res, next) {
  res.locals.placa = (foundUser * 8) * 25;
  next();
}
```

Slika 5.11. *Funkcije za izračun radnih sati i plaće.*

Unutar opcije "Djelatnici" ispisujemo sve (trenutne) djelatnike benzinske postaje. Administrator posjeduje mogućnost brisanja djelatnika. Kôd za brisanje djelatnika je prikazana slikom 5.12.

```
app.post("/deleteDjelatnik", function(req, res) {
  User.deleteOne({
    ime: req.body.button
  }, function(err) {
    res.redirect("/statistika");
  });
});
```

Slika 5.12. *Kôd za brisanje djelatnika.*

5.2 Struktura MongoDB baze podataka

Zbog više korisničkih zahtjeva i različitih mogućnosti usluge web aplikacije potrebno je stvoriti bazu podataka koja je postavljena prema potrebama korisnika aplikacije. Baza podataka pruža uslugu kreiranja, brisanja, dohvaćanja i promijene podataka koji su već postojeći u bazi. Za izradu strukture baze podataka u MongoDB potrebno je izraditi shemu podatka te mu pridružiti njegove potrebne atribute. Tablica 5.2. prikazuje sheme korištenih podataka unutar MongoDB-a.

Tablica 5.2. *Prikaz sheme korištenih podataka u MongoDB*.

MongoDB struktura podataka	
Korisnik	Ime: String Username: String Password: String Role: String
Agregati	Ime: String Stanje: Number Cijena: Number
Proizvodi	Ime: String Stanje: Number
Forum	Ime: String Komentar: String
Raspored	Prva: String Druga: String Treca: String
Statistika prodaje	Agregat: String Broj: Number Količina: Number
Zauzeće	Agregat: String Vrijednost: String

6. ZAKLJUČAK

Web aplikacija za vođenje evidencije rada benzinske postaje uvodi jednostavnije rješenje za vođenjem posla benzinske postaje. Aplikacija omogućuje pristup administratoru i djelatniku benzinske postaje, kreiranje djelatnika, uvid u stanje artikala te njihovu promjenu, kreiranje rasporeda, promjenu stanja i cijene derivata, statistiku radnih sati i forum (zbog lakše komunikacije među zaposlenicima). Aplikacija također obavještava administratora ukoliko je stanje pojedinog agregata manje od dopuštenog. Programsko rješenje web aplikacije je izvedeno pomoću HTML-a, CSS-a, JavaScript-a, Node.js-a, MongoDB baze podataka te programski alat Atom kao razvojno okruženje. Unaprjeđenju web aplikacije bi uvelike pomoglo kada bi web aplikacija bila direktno spojena na sustav benzinske postaje radi što točnijeg i efikasnijeg obrađivanja podataka. Pomoću web aplikacije za vođenje evidencije rada svi zaposlenici imaju točan uvid u cjelokupan rad benzinske postaje u bilo kojem trenutku što rezultira sa većom mobilnošću i lakšim pristupom podacima.

LITERATURA

[1] HTML- https://www.w3schools.com/html/default.asp (5.8.2021)

[2] CSS- https://www.w3schools.com/css/default.asp (5.8.2021)

[3] JavaScript- https://www.w3schools.com/js/default.asp (30.7.2021)

[4] NodeJs - https://books.goalkicker.com/NodeJSBook/ (22.7.2021)

[5] NodeJs arhitektura- https://maxprog.net.pl/wp-content/uploads/2015/11/threading_node.png.pagespeed.ce_.WVP8haZ_BX-1024x612.png (3.8.2021)

[6] MongoDB- https://books.goalkicker.com/MongoDBBook/ (8.8.2021)

[7] Uvod u web programiranje (Carles Mateu) -https://archive.org/details/ost-computer-science-fta-m4-web_development (28.7.2021)

[8] HTML,CSS i JavaScript literaturahttps://www.programming-books.io/ (2.8.2021)

SAŽETAK

Tema: Web aplikacija za nadzor rada benzinske postaje

Web aplikacija za nadzor rada benzinske postaje je jednostavna aplikacija za evidenciju poslovanja benzinske postaje. Aplikacija se sastoji od dva osnovna dijela. Prvi dio je administratorski koji služi kao sučelje za upravitelja benzinske postaje. Drugi dio aplikacije je zaposlenički te služi kao sučelje za zaposlenika benzinske postaje. Administrator ima najveće ovlasti, a to su: kreiranje zaposlenika, kreiranje rasporeda, promjena cijene derivata, promjenu stanja artikala te uvid u cjelokupan rad benzinske postaje. Zaposlenik ima manje ovlasti od administratora te svoje sučelje koristi za prodaju artikala, prodaju derivata, provjeru tjednog rasporeda, stanje derivata, uvid u statistiku radnih sati i stanje artikala. Aplikacija omogućava zaposlenicima benzinske postaje jednostavniji način vođenja posla i lakšu komunikaciju među svim zaposlenicima.

Ključne riječi: Internet tehnologije, Baza podataka, Benzinska postaja, Evidencija rada.

ABSTRACT

Tittle: Web application for gas station monitoring

The web application for gas station monitoring of gas station employees is a simple application for keeping records of the gas station business. The application consists of two basic parts. The first part is the admin which serves as the interface for the gas station manager. The second part of the application is employee and serves as an interface for the gas station employee. The administrator has the greatest powers, which are: creating employees, creating a schedule, changing the price of aggregate, changing the condition of items and insight into the entire operation of the gas station. The employee has less authority than the administrator and uses his interface to sell items, sell aggregates, check the weekly schedule, the status of aggregates, insight into working hours statistics and the status of items. The app allows gas station employees a simpler way of running a business and easier communication among all employees.

Keywords: Internet technologies, Database, Gas station, Work records.

ŽIVOTOPIS

Bruno Spajić rođen 25. studenog 1998. godine u Vinkovcima. Osnovnu školu završio je u Osnovnoj školi Vladimira Nazora u Vinkovcima 2013. godine. Po završetku osnovne škole, upisuje se u srednju Tehničku školu Ruđera Boškovića u Vinkovcima u kojoj završava smjer Mehatronika 2018. godine. Dvotjednu praksu odrađuje u Dilj d.o.o. gdje se prvi put susreće sa električnim postrojenjima. Obrazovanje nastavlja 2017 godine na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Također 2020. godine odrađuje praksu u Mono Osijek gdje se upoznao s timskim radom na projektima. Stekao je znanje iz C# i baza podataka, zna se služiti programskim jezicima: C++, JavaScript, NodeJs, Python.

PRILOZI

1. Prilog 1-Tablicom 4.1. su opisani svi administratorski zahtjevi koji su implementirani u ovome radu.

Tablica 4.1. *Prikaz svi zahtjeva administratorske stranice.*

Zahtjevi administratora	
Prijava / odjava	Omogućuje administratoru jedinstven pristup sučelju i pruža jednostavnu odjavu administratora iz radnog sučelja
Raspored	Aplikacija pruža uslugu izrade tjednog rasporeda koji se sastoji od tri smjene po danu. Administrator ima mogućnost odabira bilo kojega (postojećeg) zaposlenika za bilo koju smjenu te ima mogućnost spremanja rasporeda u MongoDB bazu podataka.
Agregati	Odabirom na opciju "Agregati" otvara se sučelje koje prikazuje stanje, cijenu i zauzeće svih agregata. Administrator ima u uvid sedam agregata (Eurosuper95, Eurosuper95 Class Plus, Eurodisel, Disel Imperium, Plin, UNP, Stanica za pumpanje guma) te ima pravo promjene stanja, cijene kao i mogućnost promjene zauzeća svakog od navedenih agregata.
Proizvodi	Opcija "Proizvodi" omogućuje ispis svih artikala benzinske postaje te administrator ima mogućnost promjene stanja odabranog artikla.
Djelatnici	Odabirom na opciju "Djelatnici" otvara se sučelje u kojem su ispisani svi (postojeći) zaposlenici benzinske postaje. Administrator ima uvid u podatke zaposlenika te mogućnost brisanja zaposlenika.

Kreiraj	Omogućuje administratoru unos podataka novog zaposlenika te njegovo kreiranja. Nakon kreiranja novog djelatnika, djelatnik ima mogućnost prijave u aplikaciju putem djelatničke prijave.
Forum	Odabirom na opciju "Forum" otvara se novo sučelje u kojem svi zaposlenici mogu komunicirati. Svaki komentar ostavljen na forum stranici je vidljiv svim zaposlenicima te iznad komentara piše i naziv osobe koja je napisala komentar.

2. Prilog 2 - Tablicom 4.2. su opisani svi djelatnički zahtjevi koji su implementirani u ovome radu.

Tablica 4.2 *Prikaz svih zahtjeva djelatnika benzinske postaje.*

Zahtjevi djelatnika	
Prijava / odjava	Omogućuje djelatniku jedinstven pristup sučelju i pruža jednostavnu odjavu administratora iz radnog sučelja
Raspored	Djelatnik pritiskom na opciju "raspored" ima pravo uvida na tjedni raspored kojeg je administrator preko svoga sučelja spremio u bazu podataka.
Agregati	Odabirom na opciju "Agregati" otvara se sučelje koje prikazuje stanje, cijenu i zauzeće svih agregata. Djelatnik ima uvid u sedam agregata (Eurosuper95, Eurosuper95 Class Plus, Eurodisel, Disel Imperium, Plin, UNP, Stanica za pumpanje guma). Djelatnik nema mogućnost izmjene stanja i cijene agregata. Unutar opcije "Agregati" djelatnik postavlja stanje zauzeća

	agregata koje može biti zeleno, crveno i narančasto.
Proizvodi	Opcija "Proizvodi" omogućuje ispis svih artikala benzinske postaje i njihova trenutna stanja. Djelatnik nema mogućnost izmjene stanja.
Moj profil	Odabirom na opciju "Moj profil" djelatniku se pruža usluga ispisa radne statistike trenutnog tjedna te ukupna zarada na odrađen broj sati (ispis ukupnog broja radnih sati u tjednu).
Prodaja	Omogućuje djelatniku benzinske postaje unos prodanih artikala. Nakon što djelatnik unese prodanu količinu određenog artikla ta količina se smanjuje sa stanja. Nakon što djelatnik potvrdi prodaju, svi djelatnici mogu vidjeti novo stanje prodanog artikla. djelatnik ne može prodati veću količinu od trenutnog stanja željenog artikla za prodaju. Ukoliko djelatnik prodaje derivate potrebno je narančasto svijetlo na agregatu koji se prodaje. Ako je crveno ili zeleno svijetlo na agregatu djelatnik nije u mogućnosti izvršiti prodaju derivata.
Forum	Odabirom na opciju "Forum" otvara se novo sučelje u kojem svi zaposlenici mogu komunicirati. Svaki komentar ostavljen na forum stranici je vidljiv svim zaposlenicima te iznad komentara piše i naziv osobe koja je napisala komentar.