

Aplikacija za dodjeljivanje zadataka zaposlenicima

Ristić, Dejana

Undergraduate thesis / Završni rad

2022

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://um.nsk.hr/um:nbn:hr:200:303374>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-28**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**APLIKACIJA ZA DODJELJIVANJE ZADATAKA
ZAPOSLENICIMA**

Završni rad

Dejana Ristić

Osijek, 2021.

SADRŽAJ

1. UVOD.....	1
1.1. Zadatak završnog rada.....	1
2. PREGLED PODRUČJA TEME.....	2
3. KORIŠTENE TEHNOLOGIJE.....	6
3.1. Teorijski pregled korištenih tehnologija.....	6
4. PREGLED IZRADE APLIKACIJE	8
4.1. Stvaranje forme aplikacije korištenjem HTML-a i CSS-a.....	8
4.2. Korištenje JavaScript-a.....	19
4.3. Izrada baze podataka	20
4.4. Povezivanje s bazom podataka korištenjem PHP-a	21
5. ZAKLJUČAK.....	30
LITERATURA.....	31
SAŽETAK	32
ABSTRACT.....	33
ŽIVOTOPIS.....	34

1. UVOD

Razvojem velikih tvrtki s velikim brojem zaposlenika pojavljuje se problem funkcionalne organizacije rada i izrade rasporeda zadataka. Nadređeni pribjegavaju pisanju rasporeda u različitim programima, a često i u Excel tablicama. Ovim završnim radom potrebno je doskočiti tom problemu i stvoriti funkcionalnu aplikaciju koja će značajno olakšati taj posao.

Zadaća ovog završnog rada je proučiti način organizacije rada u tvrtkama i na temelju toga izraditi funkcionalnu aplikaciju pod nazivom *Aplikacija za dodjeljivanje zadataka zaposlenicima*. Ona će omogućiti vođenje funkcionalnog rasporeda unutar određene tvrtke, u ovom slučaju imaginarne trgovine sportske opreme pod nazivom *DO IT*, kojem zaposlenici i nadređeni mogu pristupiti korištenjem korisničkog imena i lozinke.

U prvom poglavlju opisane su neke od aplikacija za izradu rasporeda koje se nalaze na tržištu.

Drugo poglavlje podjeljeno je na dvije cjeline. Prva je posvećena teorijskom dijelu gdje su opisane tehnologije i znanja korištene prilikom izrade, osnovne podatke i pojmove vezane za HTML i CSS, opisana su dva programska jezika: JavaScript i PHP i baze podataka i MySQL. U drugom dijelu nalazit će se opis same aplikacije i postupku izrade. Isječci koda i slike služit će objašnjenju istoga.

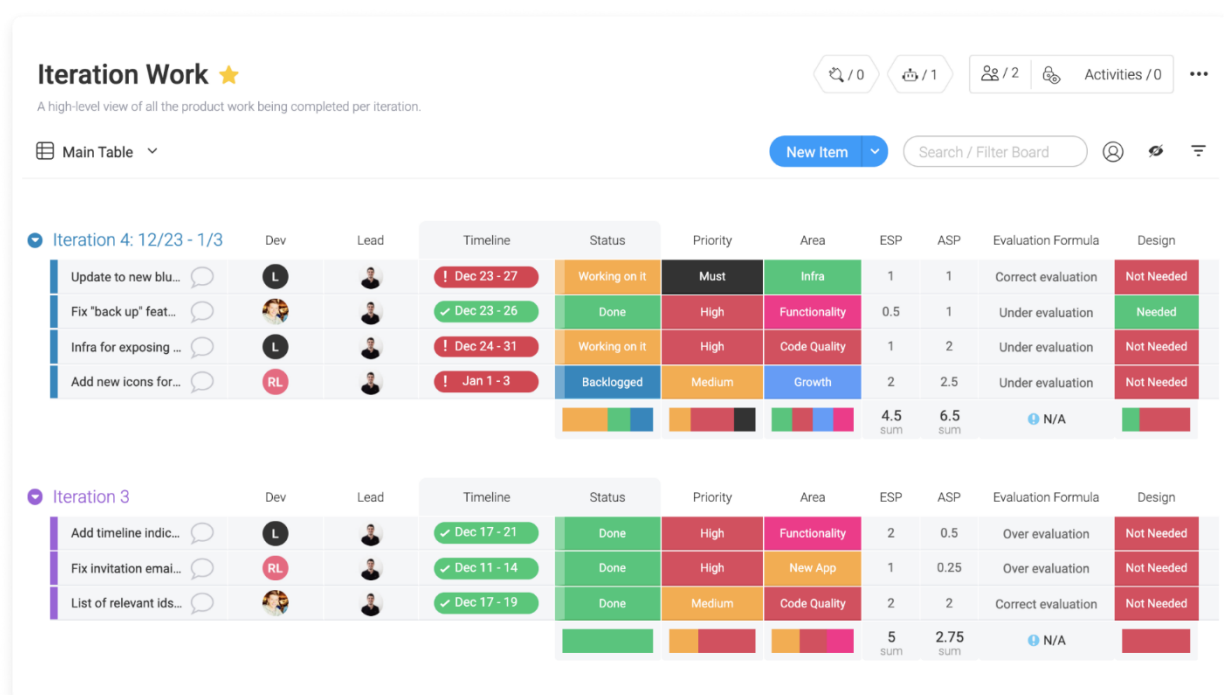
Pregled postignutih rezultata i nedostaci aplikacije, ali i moguće potrebe za nadogradnjom bit će opisani u trećem poglavlju.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je izraditi aplikaciju pod nazivom *Aplikacija za dodjeljivanje zadataka zaposlenicima* koja će omogućiti organizaciju rasporeda unutar određene tvrtke. U našem slučaju to će biti imaginarna tvrtka, trgovina sportske opreme, *DO IT d.o.o.*

2. PREGLED PODRUČJA TEME

Na tržištu postoje deseci sličnih aplikacija za raspoređivanje i organizaciju rada zaposlenika. Jedna od takvih aplikacija je *monday* [1]. Ova aplikacija pomaže planirati, pratiti i organizirati rad zaposlenika. Također, dostupna je na nekoliko jezika što joj omogućuje pristup većem tržištu. Svi se uneseni podaci sinkroniziraju trenutno. U aplikaciju je moguće unijeti različite dokumente, slike, a za sve se može dobiti i odgovarajući povratni odgovor (engl. *feedback*). Moguće je lagano promijeniti status zaposlenikovih zadataka i svi zaposlenici koji rade na tom zadatku dobivaju obavijest. Svi podaci prikupljeni kroz aplikaciju mogu biti prikazani vizualno. Zaposlenici ovu aplikaciju mogu skinuti na svoje iOS i Android uređaje. Veliki bonus ove aplikacije je da se može povezati, tj. integrirati s drugim aplikacijama. Ovo je aplikacija koju koriste velike kompanije kao što su Coca-Cola, deezer, Adobe, BBC studios, L'Oréal Paris i drugi.



The screenshot shows the 'Iteration Work' dashboard in Monday.com. It features two tables representing different iterations of product work. The top table is for 'Iteration 4: 12/23 - 1/3' and the bottom table is for 'Iteration 3'. Each table has columns for Dev, Lead, Timeline, Status, Priority, Area, ESP, ASP, Evaluation Formula, and Design. The tasks are color-coded by status: red for 'Working on it', green for 'Done', and blue for 'Backlogged'. The bottom table also includes a 'sum' row for each iteration.

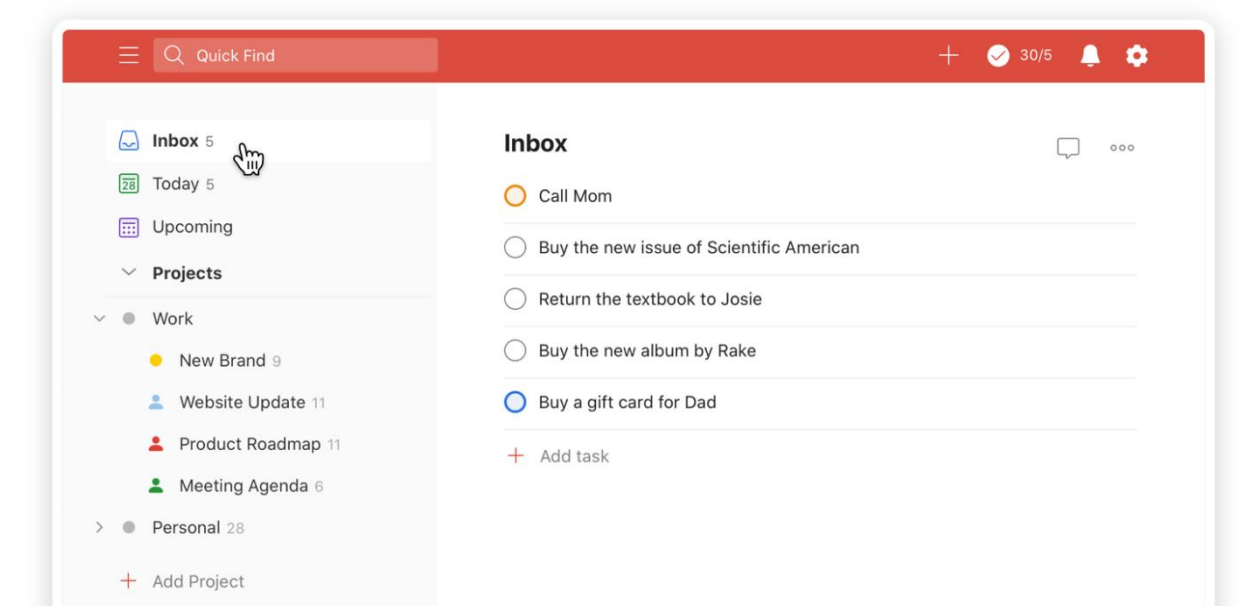
Iteration 4: 12/23 - 1/3	Dev	Lead	Timeline	Status	Priority	Area	ESP	ASP	Evaluation Formula	Design
Update to new blu...	L		! Dec 23 - 27	Working on it	Must	Infra	1	1	Correct evaluation	Not Needed
Fix "back up" feat...			✓ Dec 23 - 26	Done	High	Functionality	0.5	1	Under evaluation	Needed
Infra for exposing ...	L		! Dec 24 - 31	Working on it	High	Code Quality	1	2	Under evaluation	Not Needed
Add new icons for...	RL		! Jan 1 - 3	Backlogged	Medium	Growth	2	2.5	Under evaluation	Not Needed
							4.5 sum	6.5 sum	N/A	

Iteration 3	Dev	Lead	Timeline	Status	Priority	Area	ESP	ASP	Evaluation Formula	Design
Add timeline indic...	L		✓ Dec 17 - 21	Done	High	Functionality	2	0.5	Over evaluation	Not Needed
Fix invitation emai...	RL		✓ Dec 11 - 14	Done	High	New App	1	0.25	Over evaluation	Not Needed
List of relevant ids...			✓ Dec 17 - 19	Done	Medium	Code Quality	2	2	Correct evaluation	Not Needed
							5 sum	2.75 sum	N/A	

Sl. 2.1. Aplikacija *monday.com*

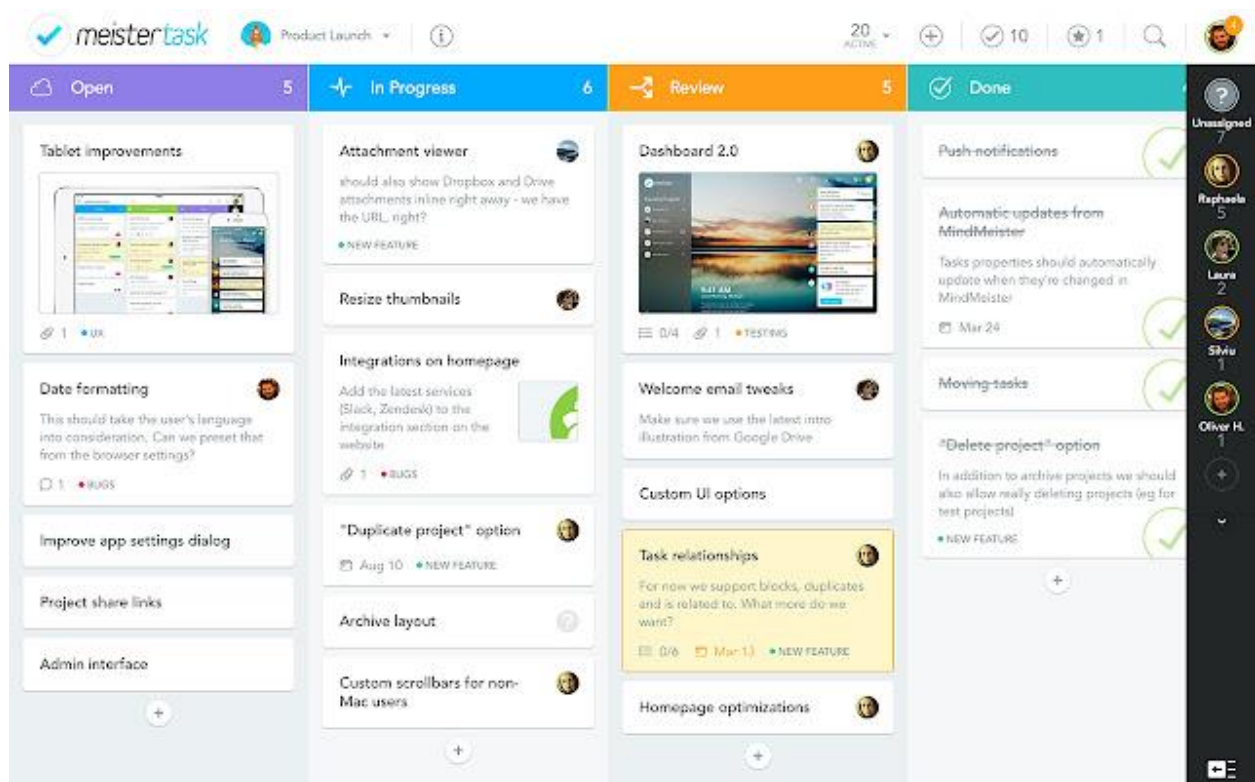
Iduća aplikacije je *todoist* [2]. Ova aplikacija nije strogo namijenjena korištenju u velikim tvrtkama već ju može bilo koja osoba instalirati na svoj mobilni uređaj putem Apple store-a ili Google play-a. Određenim zadacima moguće je odrediti prioritet, mogu se postaviti omiljeni zadaci ili pak na zadatke postaviti podsjetnik. Putem aplikacije zadaci se mogu prebaciti na neku drugu osobu, naravno ako se ta osoba nalazi u sustavu, i na taj način osloboditi raspored korisnika.

Moguće je postavljanje obavijesti kada je neki zadatak odrađen, kada je druga osoba ostavila komentar na zadani zadatak i slično. Moguće je povući zadatke i obveze koje pristižu mailom i automatski ih upisati u aplikaciju, a u slučaju da se zadatke treba premjestiti u neki kalendar po izboru, ova aplikacija omogućuje i to. Korisnik svoje zadatke i obveze može organizirati u grupe, označiti ih različitom bojom i tako samom sebi omogućiti lakše snalaženje. Zadatke je moguće filtrirati na temelju datuma roka, naziva projekta, osobi koja je dodijelila zadatak i slično. Korisniku se omogućuje personalizacija aplikacije različitim stilovima.



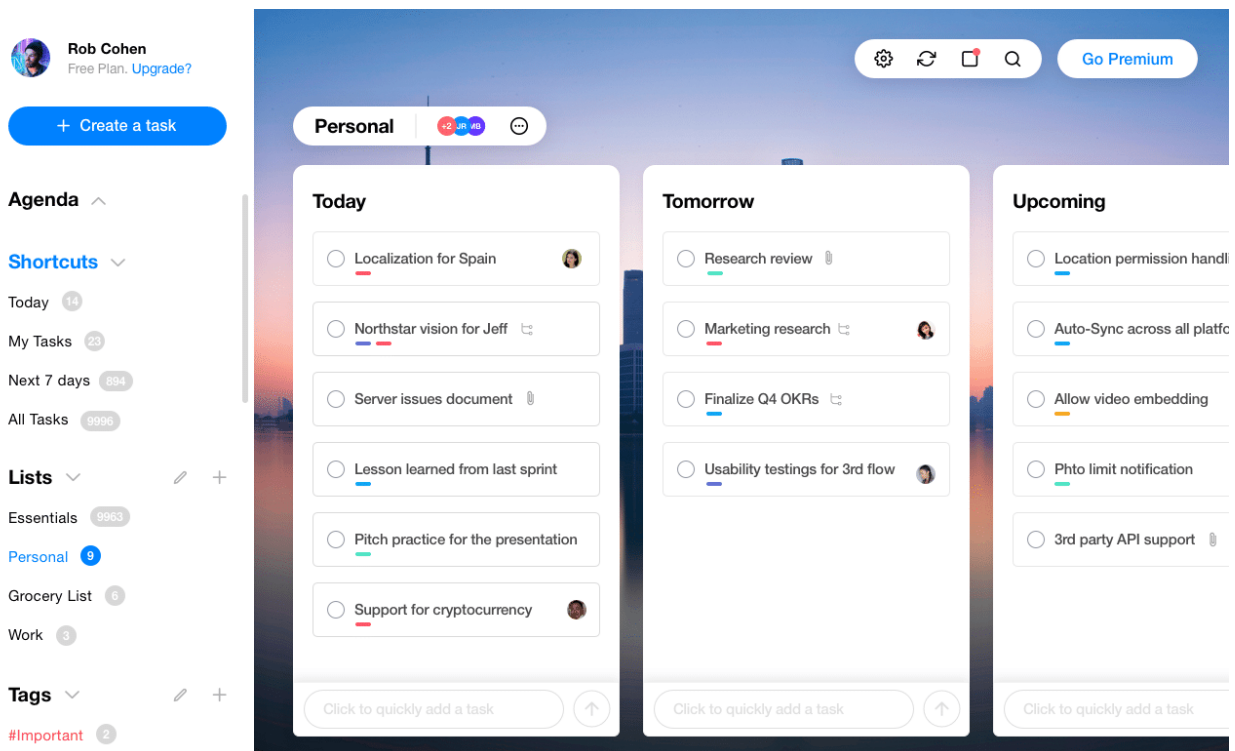
Sl. 2.2. Aplikacija *todoist*

Meister task [3] se može koristiti prilikom organizacije velikog projekta ili digitalizacije rasporeda dnevnih zadataka i poslova. Omogućuje personaliziranu okolinu koja se može prilagoditi potrebama korisnika. Aplikacija funkcionira kao centrala za sve linkove, datoteke i rasprave. Korisnici mogu imati svoju tzv. Agend-u gdje korisnik može ubaciti sve svoje zadatke iz različitih projekata na kojima radi. Nudi se i korištenje timeline-a koji projekt menadžeru pomaže da uskladi sve zaposlenike.



Sl. 2.3. Aplikacija *meister task*

Iduća aplikacija je *Any.do* [4], rješenje koje pomaže u organiziranju zadataka, TO-DO lista i podsjetnika. Olakšava korisnicima organizaciju svojih zadataka u funkcionalne liste, a zadatke se može i sortirati po prioritetu. Korisnici imaju mogućnost kreiranja podzadataka već postojećih zadataka, dodavanja bilješki na zadatke i prilaganje dokumenata. Moguće je dodavanje društvenih događaja, poslovnih sastanaka, a mogu se postavljati i određeni ciljevi. Ovaj kalendar može se koristiti na svim pametnih uređajima i postaviti ga na početni zaslon i na taj način imati pregled dana čak i bez otvaranja aplikacije. Moguće je postavljanje podsjetnika, za svaki od zadataka može se postaviti i lokacija. Osim toga, moguće je dodati popis namirnica za kupnju kojeg je moguće podijeliti s članovima obitelji. Može se reći da je ova aplikacija prilagođenija za osobnu uporabu nego za službenu uporabu u nekoj tvrtci.



Sl. 2.4. Aplikacija *any.do*

Osim ovih aplikacija, postoje još mnoge druge. Neke od njih namijenjene su korištenju u velikim tvrtkama, dok su neke prilagođenije osobnoj uporabi. Različite aplikacije dostupne su i po različitoj cijeni, neke od njih potpuno su besplatne, neke imaju mogućnost nadogradnje koja se naplaćuje, a neke se plaćaju u potpunosti. Aplikacije koje su namjenjene isključivo korištenju u tvrtkama sukladno tome imaju najveću cijenu prilikom kupnje.

3. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju opisane su sve tehnologije korištene prilikom izrade ovog završnog rada. Osim toga, bit će napravljen pregled izgleda ove aplikacije i koda korištenog pri stvaranju.

3.1. Teorijski pregled korištenih tehnologija

HTML (engl. *HyperText Markup Language*) je prezentacijski jezik koji se upotrebljava prilikom izrade web stranica. Kreirao ga je Sir Tim Berners-Lee 1991. godine. Zadaća mu je upućivanje web preglednika na koji način treba prikazati dokument, tj. elemente tog dokumenta, a možemo reći da je jezik pomoću kojega stranice komuniciraju s web preglednicima.

HTML nije programski jezik te zbog toga on ne može obavljati nikakve operacije koje su tipične za programske jezike. Na tržištu postoje mnogobrojni alati koji omogućuju razvoj internetskih stranica čak i kada osoba koja je zadužena za taj posao ne poznaje osnove HTML koda. Takvi programi sami generiraju kod koji podupire vizualni izgled stranice. Za pisanje HTML koda može se koristiti bilo koji uređivač teksta, a za potrebe izrade ove aplikacije korišten je *Visual Studio Code*.

Razvijao se kroz godine, a zadnja značajna verzija HTML-a objavljena je 2012. godine pod nazivom HTML5.

CSS (engl. *Cascading Style Sheets*) predstavlja stilski jezik koji se koristi prilikom vizualnog predstavljanja HTML elemenata. Može se reći da on na svojevrsan način upotpunjuje HTML dokument jer omogućuje dodavanje fontova, margina, boja i ostalih vizualnih elemenata. CSS uvodi revoluciju u internet i internet programiranje. Omogućuje izrađivaču stranice da bude potpuno neovisan o web pregledniku, ali i da tekst na web stranici postaje čitljiviji. Uporabom CSS-a olakšano je i razumijevanje i pisanje HTML koda, a moguće je i izmjenom samo jedne CSS datoteke izmijeniti izgled više HTML stranica. Baš kao i kod HTML-a, za stvaranje CSS-a može se koristiti bilo koji program za uređivanje teksta. Najnovija verzija je CSS 4.

JavaScript je objektno orijentirani skriptni programski jezik koji se koristi na web pregledniku. Izvodi se na osobnom računalu pa zbog toga današnja računala i imaju podršku za njegovu uporabu. Danas je jedan od popularnijih programskih jezika, a baziran je na Javi (imaju sličnu sintaksu). Skriptni programski jezik znači da se prilikom izvršavanja programa, izvršava linija po linija koda. Za razliku od ostalih programskih jezika koji se prije izvođenja prevode u strojni kod, JavaScript se izvodi kako preglednik čita skriptu.

JavaScript omogućuju veću brzinu izvršavanja aplikacija jer nema potrebe za kontaktom s poslužiteljem, pa web preglednik djeluje odmah. Također, omogućuje dinamičko mijenjanje HTML kod-a na stranici.

PHP (engl. *Hypertext Preprocessor*) skriptni je programski jezik orijentiran na C i Perl sintaksi, a namijenjen je programiranju web stranica. Koristi se za podršku baza podataka i internet protokola. Omogućuje stvaranje velikih web aplikacija koje sadrže veliku količinu podataka, što bi bilo vrlo teško i komplicirano kada bi se koristio samo HTML jer bi za svaki podatak bilo potrebno stvoriti novu HTML stranicu. PHP je besplatan (engl. *open-source*), server-side skriptni jezik i omogućuje rad s mnogim bazama podataka. Danas je jedan od najzastupljenijih programskih jezika za programiranje web aplikacija.

MySQL je poslužitelj (engl. *server*) baza podataka kojemu se može pristupiti pomoću korisničkog imena i lozinke. Baze su relacijskog tipa što ih čini prikladnima za korištenje prilikom rukovanja s velikim bazama podataka. Sustav je optimiziran i stabilna te ima podršku različitih programskih jezika kao što je na primjer PHP koji ćemo koristiti prilikom razvoja ove aplikacije. MySQL je također besplatan za korištenje.

Visual Studio Code, Microsoftovo je razvojno okruženje koje se koristi u razvoju širokog spektra računalnih programa kao što su web aplikacije, web stranice i mobilne aplikacije. Pogodan je za korištenje zbog velikog broja ugrađenih jezika kao što su C, C++, C#, JavaScript, HTML, CSS, PHP i drugi. Omogućuje ispravljanje grešaka prilikom pisanja koda, dovršavanje koda, a ima i ugrađeni repozitorij (engl. *git*). Činjenica da je vrlo lako dostupan i besplatan, čini ga jednim od popularnijih razvojnih okruženja. Moguće ga je koristiti na Windows-u, Linux-u i macOS-u.

4. PREGLED IZRADE APLIKACIJE

Zbog opširnosti ove aplikacije i velikog koda koji je korišten prilikom izrade, priloženi su i opisani samo najvažniji dijelovi.

4.1. Stvaranje forme aplikacije korištenjem HTML-a i CSS-a

Prilikom izrade HTML i CSS datoteka unutar aplikacije korišten je Visual Studio uređivač s čijim korištenjem smo se upoznali na fakultetskim kolegijima. Svakako, jedan od važnijih elemenata svakog projekta je i njegov vizualni identitet. CSS omogućuje ne samo to, već i funkcionalno raspoređivanje HTML elemenata unutar stranice. Najprije je potrebno izraditi sve HTML datoteke koje će predstavljati „kostur“ naše aplikacije, a na njega dodajemo CSS.

Header i footer su dva elementa koja se ponavljaju na svakoj stranici naše aplikacije. Header sekcija je kreirana korištenjem html header elements-a. U sebi sadrži sliku i link na početnu stranicu.

Footer sadrži podatke o kompaniji za koju je stranica izrađena i gumb za odjavu korisnika.

```
1 <header class="header_name">
2   
3 </header>
```

Sl. 4.1. Header kod

```
1 <footer class="footer">
2   <p>
3     DO IT d.o.o. trgovina sportske opreme
4   </p>
5   <a href="logout.php" class="link--logout">Log out</a>
6 </footer>
```

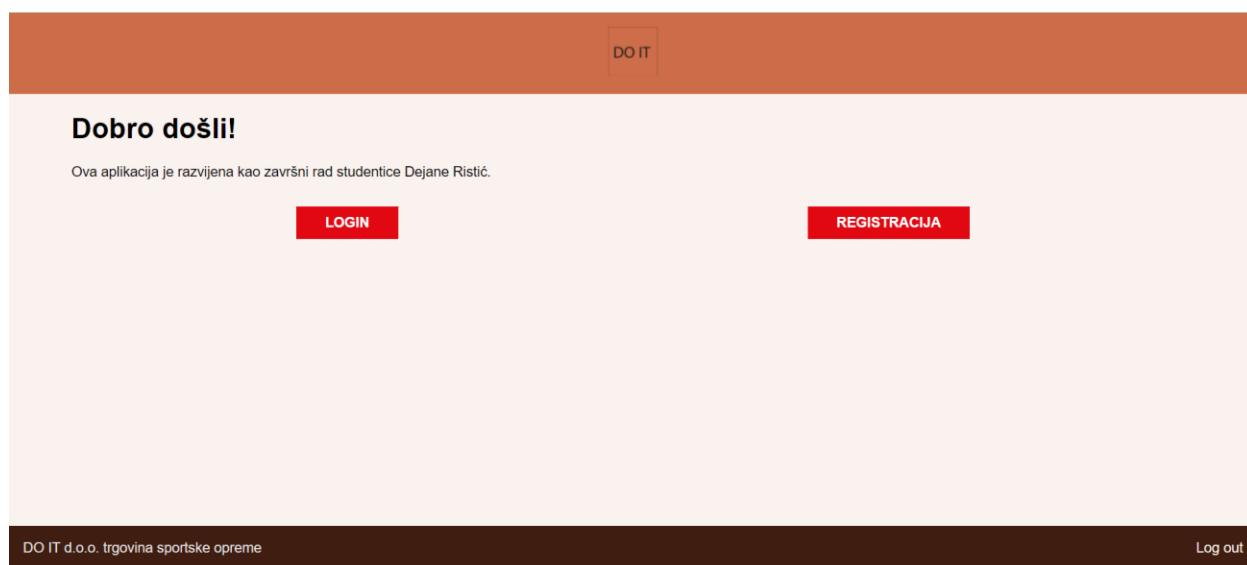
Sl. 4.2. Footer kod

Također, kako bi web aplikacija radila kako treba bilo je potrebno dodati *head* element na svaku stranicu kako bi se kroz njega moglo dohvatiti css i ostale „meta“ podatke.

```
1 <head>
2   <title>Naziv firme</title>
3   <meta charset="utf-8">
4   <link rel="stylesheet" href="./src/css/style.css">
5   <link rel="stylesheet" href="./src/normalize.css">
6 </head>
```

Sl. 4.3. Header kod

Otvaranjem stranice, potrebno je odabrati login stranicu ili pak registraciju. U slučaju da tvrtka ima novog zaposlenika, bit će potrebna registracija tog zaposlenika.



Sl. 4.4. Index stranica

Register page predstavlja obrazac u kojoj novi zaposlenici unose svoje podatke kao što su *Korisničko ime, Lozinka, Potvrda lozinke, Ime i prezime zaposlenika, Odjel i Pozicija*.

The screenshot displays the registration form on the DO IT application. The form is located on a light beige background, with an orange header bar at the top containing the 'DO IT' logo. The form fields are arranged vertically: 'Korisničko ime' (Username) with the value 'ivan' and a green checkmark icon; 'Lozinka' (Password) with masked characters; 'Potvrdi Lozinku' (Confirm Password) also with masked characters; 'Ime i prezime zaposlenika' (Employee Name) with the value 'Ivan Ivić'; 'Odjel' (Department) as a dropdown menu currently showing 'Marketing'; and 'Pozicija u odjelu' (Position in department) with the value 'Voditelj'. At the bottom of the form, there are two buttons: a red 'SUBMIT' button and a light gray 'RESET' button. Below the buttons, a link reads 'Already have an account? Login here.' The dark brown footer bar at the bottom contains the text 'DO IT d.o.o. trgovina sportske opreme' and a 'Log out' link.

Sl. 4.5. Stranica za registraciju

Na stranici za registraciju stvorena je forma u koju će zaposlenici unositi svoje podatke i tako se registrirati u sustav.

Zbog velike količine koda izdvojeni su samo određeni dijelovi na kojima će biti objašnjeno šta taj dio koda radi.

Glavni i najbitniji element ranije spomenute stranice je form element za registraciju zaposlenika. Forma sadrži šest html inputa. Prvi od njih je Korisničko ime. Input se nalazi u elementu. Ovaj element se koristi u HTML-u kada se više elemenata želi organizirati u listu, s tim ovdje je važno napomenuti kako uvijek treba biti unutar ili elementa.

Sva polja za upis podataka (Korisničko ime, Lozinka, Potvrdi lozinku, Ime i prezime zaposlenika) organizirani su na isti način. Oznaka <div> predstavlja dio ili sekciju u html dokumentu u koji je umetnut class selektor kojem je pridružen form-group, pomoću tog class-a može se odabrati točno taj element i CSS-om uređivati točno taj element na stranici. Nakon otvaranja oznake <label> specificira se podatak na koji <label> treba biti povezan, što je u ovom slučaju Username. <input> označava da će postojati forma za unos nekog podatka, koji će u ovom slučaju biti tipa tekst (type="text"), ali može poprimiti različite vrijednosti. Korištenjem PHP-a će se provjeriti je li polje prazno i u slučaju da je, vratit će grešku, to će biti pobliže objašnjeno u PHP poglavlju. je linijski element koji se koristi za označavanje dijela teksta ili dijela dokumenta, a u ovom slučaju se također koristi za izbacivanje pogreške prilikom upisivanja podatka

```
118 <li>
119   <div class="form-group">
120     <label for="username">Korisničko ime</label>
121     <input type="text" name="username" class="form-control" <?php echo (empty($username_err)) ? 'is-invalid' : ''; ?> value="<?php echo $username; ?>"
122     <span class="invalid-feedback"><?php echo $username_err; ?></span>
123   </div>
124 </li>
```

Sl. 4.6. Kod za unos korisničkog imena

Prilikom unošenja odjela u kojem je zaposlenik zaposlen, dobije se padajući izbornik gdje će ponuđene opcije biti *Management, Marketing, Sales i Warehouse*.

U ovom dijelu koda bit će opisano kako je ostvaren padajući izbornik s odjelima tvrtke. <select> element će omogućiti upravo to, padajući izbornik, atribut id="dep_id" koristi se kako bi se odjel povezao s bazom podataka u kojem je odjel zapisan preko svoga id-ja koji u vezi nije tekstualnog već cjelobrojnog tipa. <option> definira moguće opcije za odabir.

```

145 <li>
146   <div class="form-group">
147     <label for="department">Odjel</label>
148     <select name="department" id="dep_id" value="php echo $department; ?">
149       <option value="Managment">Managment</option>
150       <option value="Marketing">Marketing</option>
151       <option value="Sales">Sales</option>
152       <option value="Warehouse">Warehouse</option>
153     </select>
154   </div>
155 </li>

```

Sl. 4.7. Kod za odabir odjela zaposlenika

U kodu ispod ovog teksta može se vidjeti da je za vrijednost „type“ postavljen „submit“. Na taj način input postaje button. Kada se dogodi klik na taj button uneseni podaci bit će poslani na server.

Kod drugog input elementa, vrijednost type-a je „reset“, on služi da kako bi se uneseni podaci poništili, tj. izbrisali.

```

162 <li>
163   <div class="form-group--inline">
164     <input type="submit" class="button--primary button__login" value="Submit">
165     <input type="reset" class="button--secondary" value="Reset">
166   </div>
167 </li>

```

Sl. 4.8. Kod za submit i reset gumb

Nakon što je korisnik potvrdio podatke pritiskom na gumb Submit podaci se koristeći PHP spremati u bazu podataka. Također, u lijevom donjem kutu stranice nalaze se podaci firme, a u donjem desnom Log out koji su definirani u već ranije spomenutom footeru.

U slučaju da je korisnik već registriran, treba se samo logirati, pa se otvoriti *Login page*. Stranica se u ovom slučaju sastoji od obrasca s podacima *Korisničko ime i Lozinka*.

Sl. 4.9. Stranica za logiranje zaposlenika

U gore navedenom HTML kodu može se vidjeti na koji način je strukturirana stranica za registraciju. Potrebno je upisati korisničko ime i lozinku koji su uneseni u bazu prilikom registracije zaposlenika. U slučaju da jedan od tih podataka nije unesen ili točan, sustav će vratiti grešku. Elementi koji se nalaze na ovoj stranici konstruirani su na isti način kao oni na stranici za registraciju.

Ako zaposlenik nema registriran profil, klikom na hipervezu zaposlenik će biti preusmjeren na stranicu za registraciju. HTML element `<p>` koristi se za otvaranje paragrafa, a `<a>` element s href atributom, omogućit će stvaranje hiperveze na stranicu za registraciju.

```
139 <p class="text-style--bold">Ako nemate korisnički račun, <a class="link" href="register_page.php">ovdje ga možete otvoriti</a>.</p>
```

Sl. 4.10. Kod za hiperveza za preusmjerenje na stranicu za registraciju

Nakon što se zaposlenik prijavi, otvara mu se njegov profil. Osmišljeno je da profil sadrži sliku zaposlenika s njegovim osnovnim podacima, a osim toga će zaposlenik na svom profilu vidjeti koje sve zadatke mu je zadao supervizor. Tablica zadataka sastoji se od podataka: *Naziv zadatka*, *Odjel*, *Rok za izvršenje zadatka*, *Opis zadatka*, *Komentar*, gumba *Spremi komentar*, *Trenutni status* i gumba *Potvrdi*.

DO IT

Pozdrav, Ivan!

Ime i prezime: Ivan Ivić
Pozicija: Voditelj
Odjel: Marketing

RESETIRAJ PASSWORD

Naziv zadatka	Odjel	Rok za izvršenje zadatka	Opis zadatka	Komentar	Trenutni status
Pripremiti dizajn za letak	Marketing	09-06-2021	Osmisliti dizajn za letak.	Ovdje upišite vaš komentar SPREMI KOMENTAR	new POTVRDI

DO IT d.o.o. trgovina sportske opreme Log out

Sl. 4.11. Stranica zaposlenika

U jednom od `<div>` elemenata nalazi se slika, za čiji prikaz se koristi element `<img>` uz atribut `src` kojem pridružujemo URL putanju do slike koja se koristi. Trenutno se ovdje nalazi statična slika. U nekim budućim nadogradnjama ovdje bi se moglo omogućiti da svaki korisnik učitava svoju profilnu sliku.

```

26 <div class="column narrow flex-center">
27 | 
28 </div>

```

Sl. 4.12. Kod za prikaz fotografije

U drugi div element bit će potrebno ubaciti `<h3>` element koji se koristi kada se želi tekst koji je upisan unutar oznaka prikazati kao naslov, dok će `<b>` element upisati podatak od iznimne važnosti, a u ovom slučaju to će biti korisničko ime ulogiranog zaposlenika.

`<ul>` element započinje ne sortiranu listu, u koju su ubačeni ubaciti podaci kao što su Ime i prezime, Pozicija i Odjel. Također, zaposlenik ima mogućnost resetiranja svoje lozinke.

```

29 <div class="column">
30 | <div class="company-main__column_employeeinformation">
31 |   <h3>Pozdrav, <b><?php echo htmlspecialchars($_SESSION["username"]); ?></b>!</h3>
32 |
33 |   <ul class="employee-info__list">
34 |     <li>Ime i prezime: <strong><?php echo htmlspecialchars($_SESSION["name"]); ?></strong></li>
35 |     <li>Pozicija: <strong><?php echo htmlspecialchars($_SESSION["position"]); ?></strong></li>
36 |     <li>Odjel: <strong><?php echo htmlspecialchars($_SESSION["department"]); ?></strong></li>
37 |   </ul>
38 |
39 |   <p>
40 |     <a href="reset_password.php" class="button--secondary">Resetiraj password</a>
41 |   </p>
42 |
43 | </div>
44 </div>

```

Sl. 4.13. Kod za prikaz osnovnih podataka zaposlenika

Podaci koje će zaposlenik vidjeti na svojoj stranici vezane za zadatke koje mora obaviti bit će organizirani u tablicu. U redak (`<tr>` element) upisani su naslovi stupca (`<th>` element) Naziv zadatka, Odjel, Datum izvršetka, Opis zadatka, Komentar i Trenutni status.

```

57 <table class="task-list__table">
58 |   <tr>
59 |     <th>Naziv zadatka</th>
60 |     <th>Odjel</th>
61 |     <th>Rok za izvršenje zadatka</th>
62 |     <th>Opis zadatka</th>
63 |     <th>Komentar</th>
64 |     <th>Trenutni status</th>
65 |   </tr>

```

Sl. 4.14. Kod za formiranje tablice zadataka

Radi bolje kontrole zaposlenika i pregleda napretka u obavljanju zadataka uveden je trenutni status zadatka. Zadatak može biti *New*, *In progress* ili *Completed*.

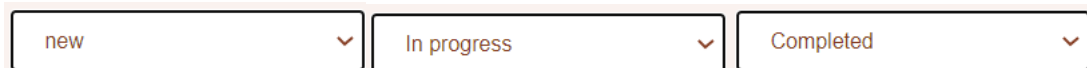

```

87 <form class="form_task-edit" action="task_edit.php">
88   <select name="task_status" id="task_status">
89     <option selected><?php echo $data['task_status']; ?></option>
90     <option value="In Progress">In progress</option>
91     <option value="Completed">Completed</option>
92   </select>
93   <input name="task_id" type="hidden" value="<?php echo $data['task_id']; ?>" />
94   <button class="button--secondary" type="submit">Potvrdi</button>
95 </form>

```

Sl. 4.15. Kod za odabir statusa zadatka

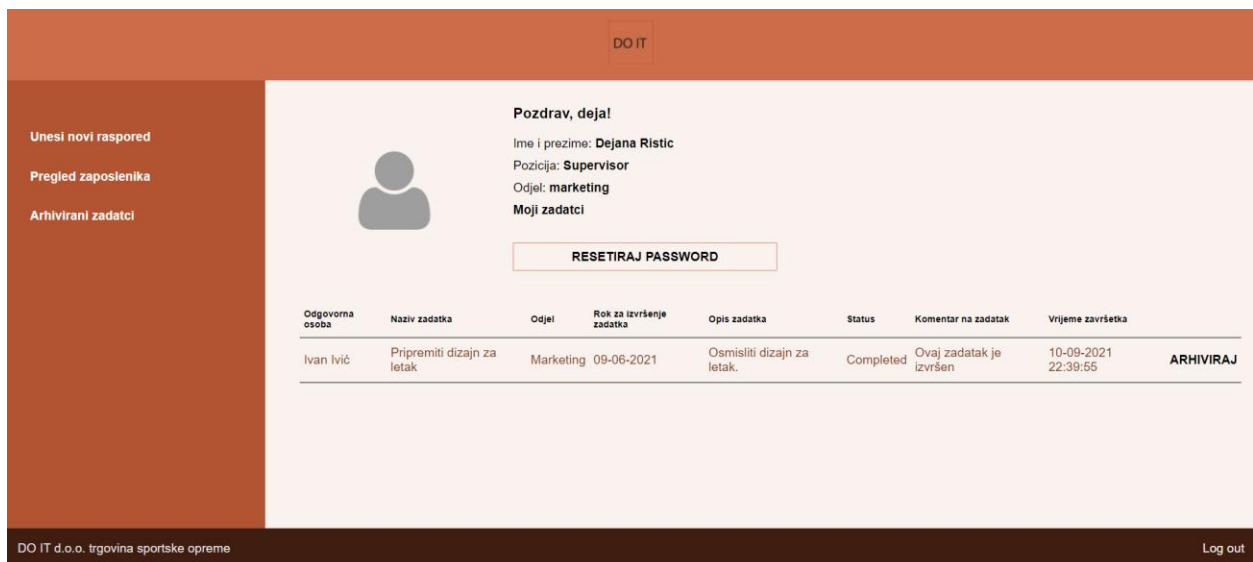
Kako bi se odabrala neka od te tri opcije, ponovno se radi padajući izbornik, a izmjena će se potvrditi s gumbom koji se ostvaruje `<button>` elementom.



Sl. 4.16. Statusi zadataka

Kako bi se ova aplikacija i cjelokupni raspored zaposlenika mogao održavati, potrebno je da se odredi jedan zaposlenik koji će imati povlaštenu pristup bazi. Taj je zaposlenik nazvan supervizor. Nakon što se supervizor ulogira u sustav sa svojim korisničkom podacima, otvorit će se stranica na kojoj se opet nalaze podaci zaposlenika, baš kao i kod zaposlenika koji nema povlaštenu pristup, a jedina razlika će biti hiperveza *Moji zadatci* koji će omogućiti da supervizor otvori vlastiti raspored zadataka i obaveza.

Ispod toga nalazi se popis svih zadataka koji su zadani zaposlenicima tvrtke, *Odgovorna osoba*, *Naziv zadatka*, *Odjel*, *Rok za izvršenje zadatka*, *Opis zadatka*, *Status*, *Komentar na zadatak*, *Vrijeme završetka* i gumb za arhiviranje zadataka. Gumb za arhiviranje će se koristiti onda kada je zaposlenik određeni zadatak označio kao završen (engl. *Completed*), a supervizor taj zadatak želi arhivirati.



Sl. 4.17. Stranica supervizora

Na desnoj vertikalnoj navigacijskoj traci (engl. *side bar*) nalazi se izbornik *Unesi novi raspored*, *Pregled zaposlenika* i *Arhivirani zadatci*. Pregled zaposlenika otvara padajući popis svih zaposlenika upisanih u bazu, Unesi novi raspored preusmjerava supervizora na stranicu za unos ovog zadatka, a Arhivirani zadatci nude vrlo brz i lagan pregled svih do sada obavljenih zadataka.

Vertikalna navigacijska traka formirana je od hiperveze na `supervisor_calendar.php` pod imenom *Unesi novi raspored*, klikom na tu stranicu otvorit će se nova stranica s formom za unos rasporeda.

```
33 <li><a href="supervisor_calendar.php">Unesi novi raspored</a></li>
```

Sl. 4.18. Hiperveza za preusmjerenje na stranicu za unos novog rasporeda

Nakon toga nalazi se pregled svih zaposlenika koji se nalaze u bazi.

```
35 <a id="toggleBtn" class="employee-list_trigger padding-bottom-0" href="#">Pregled zaposlenika</a>
```

Sl. 4.19. Padajući pregled zaposlenika

Na kraju se nalazi hiperveza na stranicu arhiviranih zadataka pod nazivom *Arhivirani zadatci*.

```
62 <li><a href="task_archive.php">Arhivirani zadatci</a></li>
```

Sl. 4.20. Hiperveza za preusmjerenje na stranicu za arhivirane zadatke

Ako se supervisor odluči arhivirati određeni zadatak pritiskom na gumb *Arhiviraj* će se to i učiniti, a ako je uspješno arhiviran kao potvrda će se otvoriti ova stranica.



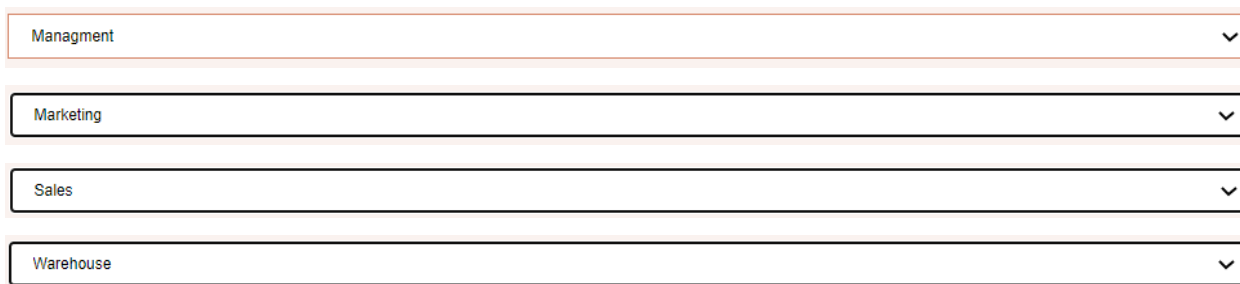
Sl. 4.21. Potvrda uspješno arhiviranog zadatka

Unesi novi raspored je stranica u kojoj supervizor može unijeti podatke za sve zaposlenike, neovisno o tome na kojem odjelu rade.

Sl. 4.22. Forma za unošenje novih zadataka za zaposlenike

Stranica će predstavljati formu u koje supervizor unosi podatke o *Nazivu zadatka*, *Datumu izvišetka zadatka* (predstavlja rok do kojeg zadatak treba biti gotov), *Opis zadatka*, *Odjel* i *Nositelja zadatka*. Prilikom unošenja nositelja zadatka moguće ga je unijeti za više zaposlenika, što znači da će za određeni zadatak biti odgovorno više osoba i taj zadatak će se upisati u njihovim

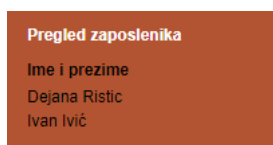
kalendarima. Odjele i Nositelje zadataka moguće je izlistati pomoću padajućeg izbornika. Ponuđeni odjeli bit će: *Managment, Marketing, Sales i Warehouse.*



The image shows four vertically stacked dropdown menus. Each menu has a light blue border and a small downward arrow on the right side. The first menu is labeled 'Managment', the second 'Marketing', the third 'Sales', and the fourth 'Warehouse'.

Sl. 4.23. Izbor odjela u kojem je zaposlenik zaposlen

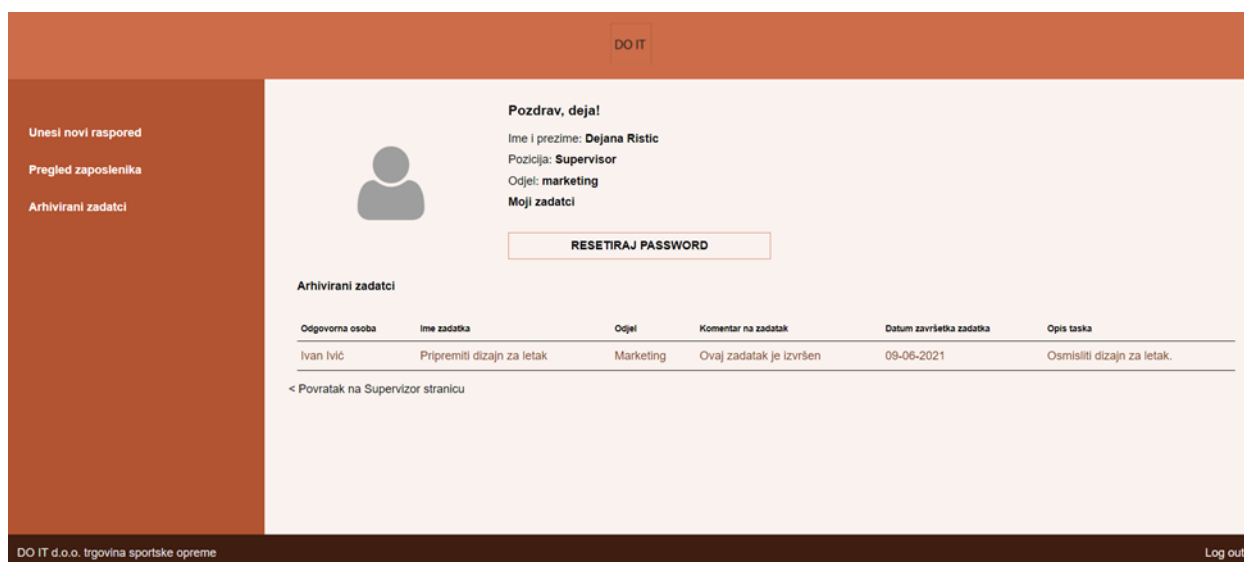
Kod padajućeg izbornika uz Nositelja zadatka, nalazit će se popis svih zaposlenika koji su registrirani u sustavu.



The image shows a small orange rectangular box. Inside the box, the text 'Pregled zaposlenika' is at the top. Below it, the text 'Ime i prezime' is followed by a list of names: 'Dejana Ristic' and 'Ivan Ivić'.

Sl. 4.24. Zadajući pregled zaposlenika na stranici supervizora

Stranica arhivirani zadatci će formom izgledati identično kao i stranica supervizora, jedina razlika je da će se ovdje izlistavati zadatci koje su zaposlenici završili, a supervisor arhivirao.



The image is a screenshot of a web application interface. On the left, there is a dark blue sidebar with three navigation links: 'Unesi novi raspored', 'Pregled zaposlenika', and 'Arhivirani zadatci'. The main content area has a light blue header with the text 'DO IT'. Below the header, there is a user profile section with a placeholder for a profile picture and the following text: 'Pozdrav, deja!', 'Ime i prezime: Dejana Ristic', 'Pozicija: Supervisor', 'Odjel: marketing', and 'Moji zadatci'. Below this, there is a button labeled 'RESETIRAJ PASSWORD'. Underneath the button, there is a section titled 'Arhivirani zadatci' which contains a table with the following columns: 'Odgovorna osoba', 'Ime zadatka', 'Odjel', 'Komentar na zadatak', 'Datum završetka zadatka', and 'Opis taska'. The table has one row of data: 'Ivan Ivić', 'Pripremiti dizajn za letak', 'Marketing', 'Ovaj zadatak je izvršen', '09-06-2021', and 'Osmisliti dizajn za letak.'. Below the table, there is a link '< Povratak na Supervisor stranicu'. At the bottom of the page, there is a dark blue footer with the text 'DO IT d.o.o. trgovina sportske opreme' on the left and 'Log out' on the right.

Sl. 4.25. Arhiva za dataka

Kao što je ranije spomenuto, svaki zaposlenik na svojoj stranici može promijeniti lozinku kojom se prijavljuje u sustav. Pritiskom na gumb *Resetiraj password* otvara se stranica na kojoj se nalazi forma za resetiranje lozinke. Ova forma izrađena je na sličan način kao i forma za registraciju zaposlenika.

Sl. 4.26. Forma za resetiranje lozinke

Zbog velike količine CSS-a, bit će opisan samo princip na koji funkcionira svaki CSS kod, pa tako i ovaj korišten u izradi ove aplikacije, objašnjenje je prikazano na kratkom primjeru.

U HTML-u nalazi se kod za `<button>` kojem je postavljen `class` na vrijednost „`button--primary`“.

```
140      <button class="button--primary" type="submit" value="submit">Spremi zadatak</button>
```

Sl. 4.27. Kod za gumb koji sprema zadatak

U CSS-u se koristi ta vrijednost koja će biti selektor elementa, on će prepoznati koji je to element na HTML stranici i na njega će primijeniti svojstva i vrijednosti koje su upisane u CSS-u. Svojstva su predstavljena nazivima `padding` (unutarnja margina), `display` (prikaz), `text-align` (smještaj teksta), `margin` (margina), `white-space` (prazan prostor), `background` (pozadina), `border` (granica), `color` (boja), `cursor` (kursor), `text-transform` (transformacija teksta), `font-weight` (debljina teksta), `transition` (tranzicija u sklopu animacije promjene boje pozadine). Svako od tih svojstava ima pripadajuću vrijednost koja utječe na način na koji će element biti prikazan.

```

315 .button--primary {
316     padding: 0.5em 2em;
317     display: block;
318     text-align: center;
319     margin: 0 auto;
320     white-space: nowrap;
321     background: #e10811;
322     border: 1px solid #e10811;
323     color: #fff;
324     cursor: pointer;
325     text-transform: uppercase;
326     font-weight: 700;
327     transition: 0.2s ease-in-out background-color;
328 }

```

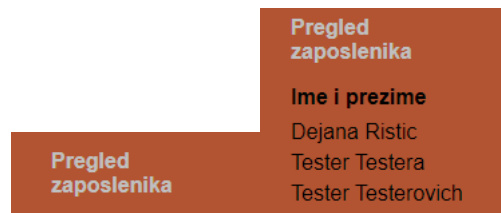
Sl. 4.28. Primjer CSS koda

Ovakav način pisanja CSS-a primjenjuje se za sve elemente HTML-a.

4.2. Korištenje JavaScript-a

Prilikom stvaranja ove aplikacije JavaScript je korišten jedino prilikom izrade stranice supervizora.

Supervisor na svojoj desnoj vertikalnoj traci sadrži Pregled zaposlenika. Prilikom klika, otvara se popis svih zaposlenika koji se nalaze u bazi. Takvu funkcionalnost moguće je ostvariti dodavanjem JavaScripta.



Sl. 4.29. Padajući popis zaposlenika

```

1  $(function() {
2      var self = this,
3      openBtn = $("#toggleBtn"),
4      toggleContainer = $("#toggleContent");
5
6      openBtn.on('click', function (event) {
7          event.preventDefault();
8
9          if (!$('toggleContainer').hasClass('active')) {
10             $(this).addClass('active');
11             toggleContainer.addClass('active');
12         } else {
13             toggleContainer.removeClass('active');
14         }
15     });
16 });

```

Sl. 4.30. JavaScript funkcija koja omogućuje padajući popis zaposlenika

Najprije je potrebno definirati takozvanu „self“ varijablu, kako bi funkcija bila svjesna u kojem se scope-u nalazi. Zatim varijablu openBtn koja će služiti kao okidač i na kraju toggleContainer, varijablu koja će predstavljati element koji se treba sakriti ili pak pokazati.

Klikom na gumb koji je trigger, pokreće se funkcija. U slučaju da na HTML elementu postoji href atribut funkcija preventDefault() omogućit će zaustavljanje preusmjerenja.

Uvjet pod if ispituje ima li kontejner klasu active, u slučaju da nema, dodat će ju na okidač button i na klasu unutar html dokumenta koji sadrži podatke s kojima se želi rukovati, odnosno, one podatke koje treba sakriti ili pokazati. U suprotnom, klasa active će biti uklonjena.

4.3.Izrada baze podataka

Kako bi ova aplikacija bila funkcionalna i radila na pravilan način, potrebno je izraditi bazu podataka u kojoj će se spremati podaci svih zaposlenika, ali i njihovi poslovni zadaci. Za ovaj projekt korišten je XAMPP. Postoje naravno i druge slične razvojne aplikacije, ali ovo je open source program koji je jednostavan za korištenje. Instalacijom XAMPP programa dobijeno je razvojno okruženje. Baza se kreira alatom koji se naziva phpmyAdmin. Za potrebe izrade ove aplikacije izrađena je baza podataka pod nazivom *adzz_01* (skraćeno od Aplikacija za dodavanje zadataka zaposlenicima), a baza podataka se sastoji od tri tablice: *employee_info*, *task* i *task_archive*.

Prva tablica *employee_info* sadrži podatke kao što su *id*, *username*, *password*, *name*, *department*, *position*, *task_id*, *created_at*, *supervisor*. Id predstavlja primarni ključ odnosno šifru koja će biti jedinstvena za svakog zaposlenika, a postavljen je kao AUTO_INCREMENT što omogućuje da se id dodijeli automatski svakom zaposleniku. Supervisor predstavlja jedinstveni atribut koji će poprimiti vrijednost „1“ samo ako je taj zaposlenik supervizor, odnosno osoba koja vozi i unosi raspored zaposlenika u sustav. Ta vrijednost se ručno postavlja u bazi. Izgled baze nalazi se na slici ispod.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	<i>id</i>	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	<i>username</i>	varchar(30)	utf16_general_ci		No	None			Change Drop More
☐ 3	<i>password</i>	varchar(255)	utf16_general_ci		No	None			Change Drop More
☐ 4	<i>name</i>	varchar(50)	utf16_general_ci		Yes	NULL			Change Drop More
☐ 5	<i>department</i>	varchar(30)	utf16_general_ci		Yes	NULL			Change Drop More
☐ 6	<i>position</i>	varchar(30)	utf16_general_ci		Yes	NULL			Change Drop More
☐ 7	<i>task_id</i>	int(6)			Yes	NULL			Change Drop More
☐ 8	<i>created_at</i>	datetime			Yes	current_timestamp()			Change Drop More
☐ 9	<i>supervisor</i>	int(1)			No	None	True if user is supervisor		Change Drop More

Sl. 4.31. Tablica u bazi podataka-*employee_info*

Iduća tablica unutar baze podataka je *task* u koju se pohranjuju svi podaci koji su vezani za zadatke zaposlenika i koji su potrebni da bi bili točno prikazani u aplikaciji, a to su *task*, *date* (datum koji predstavlja rok za završavanje zadatka), *update_date* (datum na koji je zadatak završen, odnosno status zadatka update-an) *description*, *department*, *employee_name*, *task_status*, *employee_id* i *task_id*, *task_comment*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	task	varchar(120)	utf16_general_ci		No	None			Change Drop More
☐ 2	date	varchar(12)	utf16_general_ci		No	None			Change Drop More
☐ 3	update_date	varchar(21)	utf16_general_ci		Yes	NULL			Change Drop More
☐ 4	description	text	utf16_general_ci		No	None			Change Drop More
☐ 5	department	varchar(30)	utf16_general_ci		No	None			Change Drop More
☐ 6	employee_name	varchar(30)	utf16_general_ci		No	None			Change Drop More
☐ 7	task_status	varchar(12)	utf16_general_ci		No	None			Change Drop More
☐ 8	employee_id	int(11)			Yes	NULL			Change Drop More
☐ 9	task_id	int(6)			No	None		AUTO_INCREMENT	Change Drop More
☐ 10	task_comment	text	utf16_general_ci		Yes	NULL			Change Drop More

Sl. 4.32. Tablica u bazi podataka-*task*

Kako bi se podaci mogli arhivirati potrebno je izraditi tablicu *task_archive*. Stranica predstavlja samo arhivu, pa će biti potrebni isti podaci kao i u tablici *task*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	task	varchar(120)	utf16_general_ci		No	None			Change Drop More
☐ 2	date	varchar(12)	utf16_general_ci		No	None			Change Drop More
☐ 3	update_date	varchar(21)	utf16_general_ci		Yes	NULL			Change Drop More
☐ 4	description	text	utf16_general_ci		No	None			Change Drop More
☐ 5	department	varchar(30)	utf16_general_ci		No	None			Change Drop More
☐ 6	employee_name	varchar(30)	utf16_general_ci		No	None			Change Drop More
☐ 7	task_status	varchar(12)	utf16_general_ci		No	None			Change Drop More
☐ 8	employee_id	int(11)			No	None			Change Drop More
☐ 9	task_id	int(6)			No	None			Change Drop More
☐ 10	task_comment	text	utf16_general_ci		Yes	NULL			Change Drop More

Sl. 4.33. Tablica u bazi podataka-*task_archive*

Primarni ključ *id* iz tablice *employee_info* koristit će se kao strani ključ *employee_id* u tablici *task* dok će se primarni ključ *task_id* iz tablice *task* koristiti kao strani u tablici *employee_info*.

4.4. Povezivanje s bazom podataka korištenjem PHP-a

Kombinacijom PHP-a i MySQL-a moguće je stvoriti stranice s bazama podataka u koje se spremaju svi podaci koje je potrebno prikazati ili prikupljati na stranici. Ovaj PHP kod bit će *Client-side*, što znači da će se izvršavati lokalno, na računalu korisnika. Kod je napisan u skriptama s ekstenzijom *php*, međutim tu će se nalaziti i *html* kod. PHP kod od ostatka kod-a odvajamo `<? i ?>` simbolima. Na taj način se poslužitelju govori da se u istoj datoteci nalazi HTML i PHP kod, a prvo će se izvršavati PHP koji će generirati HTML i onda se oni zajedno šalju klijentu.

Kako bi se mogla koristiti baza podataka, potrebno je tu bazu povezati putem PHP-a. To će biti ostvareno unošenjem podatka potrebnih za autentifikaciju i povezivanje s SQL serverom. Ti podaci uključuju ime servera, korisničko ime, lozinku i naziv baze podataka.

Nakon toga, pokušava se povezivanje s MySQL bazom, u slučaju da je povezivanje onemogućeno, sustav će izbaciti grešku.

```
1 <?php
2 define('DB_SERVER', 'localhost');
3 define('DB_USERNAME', 'deya123');
4 define('DB_PASSWORD', 'zavrsnirad123');
5 define('DB_NAME', 'adzz_01');
6
7 try{
8     $pdo = new PDO("mysql:host=" . DB_SERVER . ";dbname=" . DB_NAME, DB_USERNAME, DB_PASSWORD);
9     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10 } catch(PDOException $e){
11     die("ERROR: Could not connect. " . $e->getMessage());
12 }
```

Sl. 4.34. Kod za povezivanje sa serverom

PHP PDO (engl. *PHP Data Objects*) predstavlja sučelje korišteno za pristup raznim bazama podataka u PHP-u. Dobrobiti korištenja PDO-a su sigurnost, lako korištenje i mogućnost ponovnog korištenja (engl. *security, usability, reusability*).

U nastavku ovog poglavlja, bit će analizirani ključni dijelovi PHP koda korištenog u izradi ove aplikacije.

Prilikom registracije zaposlenika, potrebno je podatke unesene u formu spremati u bazu. Najprije će biti stvorene varijable koje će biti inicijalizirane praznim vrijednostima. Varijabla prije inicijalizacije potrebno je dodati znak \$.

```
10 $username = $password = $confirm_password = $name = $department = $position = "";
11 $username_err = $password_err = $confirm_password_err = "";
```

Sl. 4.35. Kod za inicijalizaciju varijabli s praznim vrijednostima

Nakon toga, slijedi kod koji će omogućiti pravilno unošenje username-a u bazu. If petlja provjerit će je li uneseno Korisničko ime (engl. *username*). Elseif provjerava sadrži li username neki drugi znak osim velikih i malih slova, brojeva ili doljnje crte. Super globalna varijabla *\$POST* će prikupiti sve unesene podatke.

```
17 if(empty(trim($_POST["username"]))) {
18     $username_err = "Please enter a username.";
19 } elseif(!preg_match('/^[a-zA-Z0-9_]+$/', trim($_POST["username"]))) {
20     $username_err = "Username can only contain letters, numbers, and underscores.";
```

Sl. 4.36. Kod za provjeru korisničkog imena prilikom registracije

Također, potrebno je utvrditi postoji li taj username u bazi i ako postoji, neće ga se moći odabrati. Koristeći *SELECT* naredbu stvorit će se uvjet s kojim se provjerava uneseni username s onima koji postoje u bazi i sprema se u varijablu *sql*.

```
21 } else{
22     $sql = "SELECT id FROM employee_info WHERE username = :username";
23
24     if($stmt = $pdo->prepare($sql)){
25         $stmt->bindParam(":username", $param_username, PDO::PARAM_STR);
26
27         $param_username = trim($_POST["username"]);
28
29         if($stmt->execute()){
30             if($stmt->rowCount() == 1){
31                 $username_err = "Nažalost već postoji identičan username u bazi.";
32             } else{
33                 $username = trim($_POST["username"]);
34             }
35         } else{
36             echo "Oops! Postoji neki problem, probajte kasnije ponovno.";
37         }
38     }
39     unset($stmt);
40 }
41 }
```

Sl. 4.37. Kod za provjeru korisničkog imena u bazi prilikom registracije

Za svaku od ovih grešaka prilikom unosa Korisničkog imena dobija se odgovarajuća poruka o grešci.

Morat će se provjeravati i lozinka koju zaposlenik unese prilikom registracije. Polje za Lozinku ne smije ostaviti prazno, duljina ne smije biti manja od 6 znakova, u drugim slučajevima lozinka će biti prihvaćena.

```
44 if(empty(trim($_POST["password"]))){
45     $password_err = "Molimo upisite ponovo lozinku.";
46 } elseif(strlen(trim($_POST["password"])) < 6){
47     $password_err = "Lozinka treba biti minimalno 6 znakova.";
48 } else{
49     $password = trim($_POST["password"]);
50 }
```

Sl. 4.38. Kod za provjeru lozinke prilikom registracije

Polje za potvrdu Lozinke također se ne smije ostaviti prazno, ako ostane prazno ili unesena lozinka ne odgovara lozinki u polju iznad, sustav će izbaciti grešku i lozinka neće biti spremljena u bazu.

```

53     if(empty(trim($_POST["confirm_password"]))){
54         $confirm_password_err = "Molimo potvrdite lozinku.";
55     } else{
56         $confirm_password = trim($_POST["confirm_password"]);
57         if(empty($password_err) && ($password != $confirm_password)){
58             $confirm_password_err = "Lozonka ne odgovara.";
59         }
60     }

```

Sl. 4.39. Kod za potvrdu lozinke prilikom registracije

Idući korak je provjera postoji li username error, password error i confirm password error, odnosno, greške prilikom unosa korisničkog imena, lozinke ili potvrde lozinke. Ako ne postoji, vrši se *INSERT* naredba za unošenje podataka u tablicu *employee_info*. Prosljeđuju se svi podaci koji se nalaze u HTML formi, a to su username, password, name, department, position.

Kako bi podaci bilo pravilno prosljeđeni, potrebno je raspisati PDO izraz koji će označiti pravilno svaki uneseni podatak iz forme sa svakom pojedinačnom kolonom u SQL tablici.

```

63 if(empty($username_err) && empty($password_err) && empty($confirm_password_err)){
64
65     $sql = "INSERT INTO employee_info (username, password, name, department, position) VALUES (:username, :password, :name, :department, :position)";
66
67
68
69     if($stmt = $pdo->prepare($sql)){
70         $stmt->bindParam(":username", $param_username, PDO::PARAM_STR);
71         $stmt->bindParam(":password", $param_password, PDO::PARAM_STR);
72         $stmt->bindParam(":name", $param_name, PDO::PARAM_STR);
73         $stmt->bindParam(":department", $param_department, PDO::PARAM_STR);
74         $stmt->bindParam(":position", $param_position, PDO::PARAM_STR);
75
76         $param_username = $username;
77         $param_password = password_hash($password, PASSWORD_DEFAULT);
78         $param_name = trim($_POST["name"]);
79         $param_department = trim($_POST["department"]);
80         $param_position = trim($_POST["position"]);
81
82         var_dump($param_username);
83
84         if($stmt->execute()){
85             header("location: login_page.php");
86         } else{
87             echo "Oops! Something went wrong. Please try again later.";
88         }
89
90         unset($stmt);
91     }
92 }

```

Sl. 4.40. Kod za unos pripremljenih podataka u bazu

Nakon što su raspisani PDO izrazi i varijable koji su potrebne, potrebno je PHP-u reći što će učiniti ako je izraz ispravan, a što ako nije. Ako je ispravan, preusmjerit će korisnika na stranicu za prijavu. Ako pak nije, ispisat će se greška u pretraživaču. Na kraju, poništi se kreirani statement kako ne bi kasnije došlo do problema.

Ovakve provjere rade i prilikom izrade stranice za logiranje. Prilikom prijave radi se provjera za vrijednost supervisor u bazi. U slučaju da je prijavljeni zaposlenik supervisor, on će biti preusmjeren na svoj posebni profil. *\$_SESSION* je varijabla korištena za pohranjivanje podataka na poslužitelja.

```

78     if ($_SESSION["position"] === "Supervisor") {
79         header("location: supervisor_page.php");
80     } else {
81         header("location: employee_page.php");
82     }

```

Sl. 4.41. Kod za provjeru supervizora prilikom prijave

Ako osoba koja nije supervisor pokuša pristupiti supervisor profilu, bit će preusmjeren nazad na svoj profil.

```

20     if ($_SESSION["position"] !== "Supervisor") {
21         header("location: employee_page.php");
22         exit;
23     }

```

Sl. 4.42. Kod za preusmjera vanje neovlaštenog pristupa

Kao što je već spomenuto u poglavlju HTML-a, na vrhu stranice zaposlenika nalazit će se njegovi osnovni podaci, podatke se dohvaća iz baze i ispisuje na za to određena mjesta.

```

34 <li>Ime i prezime: <strong><?php echo htmlspecialchars($_SESSION["name"]); ?></strong></li>
35 <li>Pozicija: <strong><?php echo htmlspecialchars($_SESSION["position"]); ?></strong></li>
36 <li>Odjel: <strong><?php echo htmlspecialchars($_SESSION["department"]); ?></strong></li>

```

Sl. 4.43. Kod koji prikazuje osnovne podatke na stranici zaposlenika

Također, zaposlenik ovdje ima i gumb koji ga vodi na stranicu gdje može promijeniti svoju lozinku.

```

39 <p>
40 | <a href="reset_password.php" class="button--secondary">Resetiraj password</a>
41 </p>

```

Sl. 4.44. Kod za button koji preusmjerava na stranicu za promjenu lozinke

Promjena lozinke bit će odrađena na isti način kao što je i spremanje prve lozinke u bazu podataka. Unijet će se nova željena lozinka, u slučaju da ta lozinka ne zadovoljava iste uvjete kao ona prilikom prijave (ne smijemo ostaviti prazno polje, duljina ne smije biti manja od 6 znakova) bit će odbijena. U polju za potvrdu lozinke treba ponovno unijeti istu lozinku kao i u prvom polju, u slučaju da nije unesena ista lozinka, ona neće biti prihvaćena.

```

89 </li>
90 <div class="form-group">
91 <label>Nova lozinka</label>
92 <input type="password" name="new_password" class="form-control" <?php echo (empty($new_password_err)) ? 'is-invalid' : ''; ?> value="<?php echo $new_password; ?>"
93 <span class="invalid-feedback"><?php echo $new_password_err; ?></span>
94 </div>
95 </li>
96 <li>
97 <div class="form-group">
98 <label>Potvrdi lozinku</label>
99 <input type="password" name="confirm_password" class="form-control" <?php echo (empty($confirm_password_err)) ? 'is-invalid' : ''; ?>
100 <span class="invalid-feedback"><?php echo $confirm_password_err; ?></span>
101 </div>
102 </li>

```

Sl. 4.45. Kod za promjenu lozinke

Gore navedene mogućnosti prikaza podataka nalaze se i na stranici zaposlenika i na stranici supervizora.

Na stranici zaposlenika iz baze se trebaju dohvatiti podaci vezani za zadatak koji će supervizor unositi za zaposlenike tvrtke. Najprije se definira naredba *SELECT* koja se koristi za dohvaćanje podataka iz tablice *task* i ti podaci se prikazuju u tablici sa zadacima.

```

52 <?php
53 $id = (int) $_SESSION['id'];
54 $results = mysqli_query($db,"SELECT * FROM task WHERE employee_id=$id");
55 ?>

```

Sl. 4.46. Kod za *SELECT* naredbu koja omogućuje unos zadatka na određenog zaposlenika

Funkcija *echo* se koristi kada je potrebno ispisati određeni podatak, dok *\$data* dohvaća podatke u bazi podataka ako oni postoje.

```

69 <td><?php echo $data['task']; ?></td>
70 <td><?php echo $data['department']; ?></td>
71 <td><?php echo $data['date']; ?></td>
72 <td><?php echo $data['description']; ?></td>

```

Sl. 4.47. Kod za ispis podataka o zaposleniku iz baze

Svaki zaposlenik može označiti status svoga zadatka. To može biti *New*, *In progress* ili *Completed*.

```

88 <select name="task_status" id="task_status">
89 <option selected><?php echo $data['task_status']; ?></option>
90 <option value="In Progress">In progress</option>
91 <option value="Completed">Completed</option>
92 </select>

```

Sl. 4.48. Kod za odabir statusa zadatka

U kodu ispod prikazano je na koji način je ostvareno da se status zadatka mijenja. Najprije se dohvaća *task_id* zadatka čiji status je potrebno izmijeniti, također je potreban postavljeni status zadatka i datum izmjene. Sve to će se ubaciti u bazu pomoću naredbe *UPDATE*.

```

5  $task_id = $_GET['task_id'];
6  $task_status = $_GET['task_status'];
7  $update_date = date('d-m-Y H:i:s');
8
9
10 mysqli_query($db,"UPDATE task set task_status='$task_status', update_date='$update_date' where task_id='$task_id'");

```

Sl. 4.49. Kod za izmjenu statusa zadatka

Svi podaci koje je supervisor unio u tablicu task u bazi podataka bit će prikazani na supervisorovoj stranici. Način na koji je to izvedeno ne razlikuje se mnogo od tablice na stranici zaposlenika. Tablica će dodatno sadržavati podatke o odgovornoj osobi, vremenu završetka zadatka i mogućnost arhiviranja zadatka.

```

106 <?php
107 $results = mysqli_query($db,"SELECT * FROM task");
108 while($data = mysqli_fetch_array($results)) { ?>
109     <tr>
110         <td><?php echo $data['employee_name']; ?></td>
111         <td><?php echo $data['task']; ?></td>
112         <td><?php echo $data['department']; ?></td>
113         <td><?php echo $data['date']; ?></td>
114         <td><?php echo $data['description']; ?></td>
115         <td><?php echo $data['task_status']; ?></td>
116         <td><?php echo $data['task_comment']; ?></td>
117         <td>
118             <?php if ($data['task_status'] == "Completed") {
119                 echo $data['update_date'];
120             }?>
121         </td>
122         <td><a class="button--remove" href="task_delete.php?task_id=<?php echo $data['task_id']; ?>">Arhiviraj</a></td>
123     </tr>
124 <?php } ?>

```

Sl. 4.50. Kod za prikaz zadataka za prikaz podataka u supervisorovoj tablici

Zadatke koji su označeni za Arhiviranje treba dodati u tablicu *task_archive* naredbom *INSERT*, a izbrisati iz tablice *task* naredbom *DELETE*. Tablica u arhivi izgleda isto kao na supervisor stranici.

```

8  mysqli_query($db,"INSERT INTO task_archive SELECT * FROM task WHERE task_id='$task_id'");
9  mysqli_query($db,"DELETE FROM task WHERE task_id='$task_id'");

```

Sl. 4.51. Kod za premještanje zadataka u arhivu i brisanje iz glavne tablice

Podatke o zadatku koji supervisor zadaje zaposleniku unose se na stranici pod nazivom *supervisor_calendar*. Podaci će se iz te forme prikupljati na isti način kao i prilikom registracije korisnika na stranicu.

Najprije se definiraju varijable koje se inicijaliziraju s praznom vrijednosti.

```

27 $task = $date = $description = $department = $employee_name = $employee_id = $description = $task_status = "";

```

Sl. 4.52. Kod za definiranje varijabli i inicijalizaciju na praznu vrijednost

Zatim, u varijablu `employee` se iz baze dohvaća id zaposlenika i povezuje ga sa imenom usera. To se koristi kako bi se moglo u multi select-u pravilno mapirati koji je zaposlenik izabran za određeni task.

```
35 $employee = $pdo->prepare("SELECT name FROM employee_info WHERE id = :s");
36 $employee->bindParam(":s", $employeeId, PDO::PARAM_STR);
37 $employee->execute();
38 $employee_name = $employee->fetch();
```

Sl. 4.53. Kod za dohvaćanje ID zaposlenika i povezivanje s korisničkim imenom

Naredni kod predstavlja *multi-select* za odabir više zaposlenika kojima će biti uneseni podatci.

```
124 <select multiple name="employee_id[]" id="employee_id">
125     <?php
126     $records = mysqli_query($db,"SELECT name, id FROM employee_info");
127
128     while($data = mysqli_fetch_array($records)) { ?>
129         <option value="<?php echo $data['id']; ?>">
130             <?php echo $data['name']; ?>
131         </option>
132     <?php } ?>
133 </select>
```

Sl. 4.54. Kod za multi-select zaposlenika

Nakon toga podaci se pripremaju za unos u bazu pomoću naredbe *INSERT*. Važno je napomenuti kako će se ovaj dio koda ponoviti za svakog usera koji je odabran u HTML *multi select input*-u. Ako je odabran samo jedan user, logika će se pokrenuti samo jednom i pri tome kreirati samo jedan novi task u bazi.

```
41 $sql = "INSERT INTO task (task, date, description, department, employee_name, task_status, employee_id)
42 VALUES (:task, :date, :description, :department, :employee_name, :task_status, :employee_id)";
```

Sl. 4.55. Priprema podataka za unos u bazu podataka pomoću naredbe *INSERT*

Nadalje, priprema se naredba koji će povezati ili mapirati unose iz HTML forme sa stupcima u *task* SQL tablici. Kada se povezuje ime korisnika, povezuje se polje podataka jer postoji mogućnost kreiranja istog zadatka za više osoba.

```

44 if ($stmt = $pdo->prepare($sql)) {
45     $stmt->bindParam(":task", $param_task, PDO::PARAM_STR);
46     $stmt->bindParam(":date", $sqlDate, PDO::PARAM_STR);
47     $stmt->bindParam(":description", $param_description, PDO::PARAM_STR);
48     $stmt->bindParam(":department", $param_department, PDO::PARAM_STR);
49     $stmt->bindParam(":employee_name", $employee_name["name"], PDO::PARAM_STR);
50     $stmt->bindParam(":task_status", $param_task_status, PDO::PARAM_STR);
51     $stmt->bindParam(":employee_id", $employeeId, PDO::PARAM_STR);
52
53     $param_task = $_POST["task"];
54     $param_data = $_POST['date'];
55     $param_description = trim($_POST["description"]);
56     $param_department = trim($_POST["department"]);
57     $param_task_status = trim($_POST["task_status"]);
58
59     $sqlDate = date("m-d-Y", strtotime($param_data));

```

Sl. 4.56. Kod za mapiranje unosa iz HTML-a da stupcima u tablici baze podataka

Ako su podaci uspješno uneseni u bazu, zaposlenik će biti preusmjeren na supervizor stranicu.

```

63 if ($stmt->execute()) {
64     header("location: supervisor_page.php");
65 } else {
66     echo "Oops! Something went wrong. Please try again.";
67 }

```

Sl. 4.57. Kod za preusmjerenje nakon unosa podataka

U ovom poglavlju sažeto su opisani najznačajniji dijelovi PHP koda korištenog prilikom izrade ove stranice.

5. ZAKLJUČAK

Cilj ovog rada bila je izrada aplikacije koja će omogućiti učinkovito vođenje rasporeda zaposlenika. Objašnjene su tehnologije korištene prilikom izrade kao i programski jezici. Za izradu cijele aplikacije korišten je Visual Studio Code. Ova razvojna okolina omogućila je da se na istom mjestu razviju sve potrebni dijelovi aplikacije. Izgled aplikacije izrađen je koristeći HTML i CSS. JavaScript omogućuje dinamičko funkcioniranje web sučelja kojim korisnik upravlja. Kada JavaScript vrši izmjene na stranici, ne aktivira se web poslužitelj već se akcije izvršavaju na strani korisnika. Drugim riječima, ne koriste se resursi web poslužitelja već resursi korisničkog uređaja. PHP skriptni je programski jezik koji se koristi za podršku bazama podataka i internet protokola, omogućuje stvaranje velikih web aplikacija koje sadrže veliku količinu podataka. Baza podataka za ovu aplikaciju izrađena je koristeći phpMyAdmin koji dolazi u sklopu programa/paketa XAMPP koji se koristi kao lokalno korisničko sučelje MySQL baze podataka. Ova aplikacija mogla bi se poboljšati dodavanjem dodatnih funkcionalnosti kao što su detaljna analiza rada zaposlenika. Aplikacija bi mogla računati sate koje je zaposlenik proveo radeći svoje zadatke, akumulirati ih, zbrajati, a na temelju toga mu obračunavati plaću, određivati broj dana koje može uzeti za godišnji, a koje dane godišnjeg je već iskoristio. Također, nakon što bi se implementirale nove funkcionalnosti, aplikacija bi bila spremna i za vizualno osvježanje. Na primjer, zamijeniti prikaz korisnikovih zadataka u tablici na neki ljepši i dinamički način, a može se i implementirati intuitivnije rješenje za obavijesti akcija za korisnike

LITERATURA

- [1] Product overview, >>monday.com<<. [Mrežno]. Dostupno: <https://monday.com/product/> [12. rujan 2021.].
- [2] Any.do, >>any.do<<. [Mrežno]. Dostupno: <https://www.any.do/> [12. rujan 2021.].
- [3] Mesiter task, >><meister task<. [Mrežno]. Dostupno: <https://www.meistertask.com/> [12. rujan 2021.].
- [4] todoist, >>todoist<<. [Mrežno]. Dostupno: <https://todoist.com/features> [12. rujan 2021.].
- [5] HTML tutorial, >>W3 schools<<. [Mrežno]. Dostupno: <https://www.w3schools.com/html/default.asp> [4. srpanj 2021.].
- [6] CSS tutorial, >>W3 schools<<. [Mrežno]. Dostupno: <https://www.w3schools.com/css/default.asp> [20. kolovoz 2021.].
- [7] JavaScript tutorial, >>W3 schools<<. [Mrežno]. Dostupno: <https://www.w3schools.com/js/default.asp> [12. srpanj 2021.].
- [8] SQL tutorial, >>W3 schools<<. [Mrežno]. Dostupno: <https://www.w3schools.com/sql/default.asp> [3. kolovoz 2021.].
- [9] PHP tutorial, >>W3 schools<<. [Mrežno]. Dostupno: <https://www.w3schools.com/php/default.asp> [12. kolovoz 2021.].
- [10] FAQ: Frequently Asked Questions, >>W3 schools<<. [Mrežno]. Dostupno: <https://www.php.net/FAQ.php> [2. rujan 2021.].
- [11] I. Lukić, M. Kohler, Osnove internet programiranja, Osijek: Elektrotehnički fakultet, 2012.
- [12] HTML: HyperText Markup Language, >>Developer mozilla<<. [Mrežno]. Dostupno: <https://developer.mozilla.org/en-US/docs/Web/HTML> [9. kolovoz 2021.].
- [13] CSS: Cascading Style Sheets, >>Developer mozilla<<. [Mrežno]. Dostupno: <https://developer.mozilla.org/en-US/docs/Web/CSS> [11. srpanj 2021.].
- [14] JavaScript, >>Developer mozilla<<. [Mrežno]. Dostupno: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [2. kolovoz 2021.].

SAŽETAK

Cilj ovog završnog rada je izrada aplikacije koja će se koristiti prilikom izrade rasporeda zadataka zaposlenika. Nakon predstavljanja problema koji se pokušava riješiti ovim završnim radom predstavljena su rješenja koja već postoje na tržištu. Na početku se nalaze teorijski opis tehnologija korištene u izradi. Korišteno je znanje HTML-a , CSS-a i MySQL-a s kojim se upoznaje na fakultetskim kolegijima preddiplomskog studija. Istraživanjem je to znanje nadograđeno, a također je korišteno znanje iz područja PHP-a i JavaScript-a koje je bilo ključno za izradu funkcionalne aplikacije. U ostatku završnog rada priloženi su dijelovi koda uz sažeto objašnjenja onoga što je tim kodom ostvareno. Ostvarena je funkcionalna aplikacija koju je moguće koristiti u tvrkama.

Ključne riječi: internet aplikacija, raspored, zadatci

ABSTRACT

Application for assigning tasks for employees

In the process of making this final paper, the knowledge of HTML, CSS and MySQL that we were introduced to in college courses is used. The research is an upgraded knowledge in the field of PHP and JavaScript that enabled the development of this functional application. The application is intended companies and should enable them to create task schedules for their employees.

Keywords: internet application, schedule, tasks

ŽIVOTOPIS

Dejana Ristić, kći je Miladina i Grozde, rođena 16.12.1999. u Osijeku i mjestom prebivališta u Belom Manastiru gdje je završila osnovnu školu Dr. Franjo Tuđman. Školske godine 2013./2014. upisuje srednju školu Gimnazija Beli Manastir. Nakon završetka srednjoškolskog obrazovanja upisuje preddiplomski sveučilišni studij Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija. Trenutno je studentica 3. godine.

Potpis autora