

ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA
APLIKACIJA ZA NADZOR ZAŠČITNE OPREME

Tematsko področje: Računalništvo

Avtor:

Blaž Mežnar, 4. letnik

Mentor:

Miran Zevnik, univ. dipl. inž. elektrotehnike

Velenje, 2021

Raziskovalna naloga je bila opravljena na ŠC Velenje, Elektro in računalniška šola, 2021

Mentor: Miran Zevnik, univ. dipl. inž. elektrotehnike

Datum predstavitve: april 2021

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD ŠC Velenje, šolsko leto 2020/2021

KG Spletna aplikacija / popisovanje opreme / gasilska društva

AV MEŽNAR Blaž

SA ZEVNIK Miran

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2021

LI 2021

IN **APLIKACIJA ZA EVIDENTIRANJE OSEBNE ZAŠČITNE OPREME**

TD Raziskovalna naloga

OP

IJ SL

JI sl/en

AI Številna gasilska društva v Sloveniji se soočajo s problemom popisovanja gasilske osebne zaščitne opreme. V ta namen sem razvil spletno aplikacijo, ki gasilcem poenostavi ta proces. Društva lahko vanjo vnesejo vso svojo opremo in imajo pregled nad njeno zgodovino (npr. popravila in čiščenja) ter kdo jo ima trenutno v lasti. Za posamezen kos opreme lahko uporabnik, ki jo lasti, tudi napiše poročilo, če je z njo kaj narobe. Uporabniki se delijo na dva ranga: administratorji in navadni uporabniki. Navadni uporabnik ima samo pregled nad svojo opremo ter zanjo lahko napiše poročilo. Administratorji pa imajo poleg prej omenjenih funkcionalnosti še popoln nadzor nad društveno opremo (lahko dodajo novo, spremenijo lastnika opreme ali pa jo izbrišejo), nadzor nad uporabniki (lahko jih izbrišejo, dodelijo/odvzamejo opremo ali pa spremenijo rang) ter pregled vseh poročil (in glede njega ustrezno ukrepajo in označijo, ali se je problem razrešil). Ker pa je spletna aplikacija tudi odzivna, jo lahko uporabljamo na računalniku, tablici ali telefonu, kar pa naredi vse skupaj dosti bolj uporabno in dostopno.

KEY WORDS DOCUMENTATION

ND ŠC Velenje, school year 2020/2021

CX web application / equipment inventory / fire brigades

AU MEŽNAR Blaž

AA ZEVNIK Miran

PP 3320 Velenje, SLO, Trg mladosti 3

PY 2021

TI **APLIKACIJA ZA EVIDENTIRANJE OSEBNE ZAŠČITNE OPREME**

DT Research work

NO

LA SL

AL sl/en

AB Many fire brigades in Slovenia are facing the problem of listing fire brigade personal protective equipment. For this purpose, I have developed a web application that simplifies this process for firefighters. Fire brigades can enter all their equipment into it and have an overview of its history (e.g. repairs and cleanings) and who currently owns it. For each piece of equipment, the user who owns it can also write a report if there is something wrong with it. Users are divided into two ranks: administrators and regular users. The average user only has an overview of their equipment and can write a report for it. In addition to the previously mentioned functionalities, administrators have complete control over the fire brigade equipment (they can add new, change the owner of the equipment or delete it), control users (they can delete, assign / remove equipment or change rank) and review all reports (and take appropriate action and indicate whether the problem has been resolved). However, because the web application is also responsive, it can be used on a computer, tablet or phone, which makes everything much more useful and accessible.

KAZALO VSEBINE

1. UVOD	1
1.1 Hipoteze	1
1.2 Namen in cilj	1
2. PREGLED OBSTOJEČIH OBJAV	2
2.1 Podobne obstoječe rešitve	2
2.1.1 Frontman FIND	2
2.1.2 Inventura OS	2
2.1.3 Sistem OSA	3
3. OGRODJA ZA IZDELAVO APLIKACIJE	4
3.1 Izbira programskega jezika in okolja	4
3.1.1 PHP	4
3.1.2 Atom	5
3.1.3 Toad Data Modeler	5
3.1.4 XAMPP	6
3.1.5 SQL in MySQL	6
3.1.6 PDO	7
3.1.7 Github	7
3.1.7.1 Git	8
3.2 Načrt izdelave aplikacije	8
4. IZDELAVA APLIKACIJE	10
4.1 Kodiranje gesel	12
4.2 Odziven CSS	13
4.3 Filtriranje uporabnikov	14
4.4 Administratorska plošča	15
4.5 Dodajanje novih društev	16
4.6 Funkcija poročila	17
5. DELOVANJE APLIKACIJE	18
5.1 Administratorski del	18
5.2 Uporabniški del	20
6. RAZPRAVA	28
6.1 Pregled hipotez	28
7. ZAKLJUČEK	30
9. ZAHVALA	32
10. VIRI IN LITERATURA	33

10.1 VIRI SLIK	33
11. PRILOGE	34
11.1 PRILOGA A	34

KAZALO SLIK

Slika 1: Logotip Frontman FIND, vir[1]	2
Slika 2: Logotip InveturaOS, vir[2]	3
Slika 3: Osnovni primer PHP kode, lasten vir	5
Slika 4: Primer grafičnega vmesnika Toad Data Modeler-ja, vir[3]	6
Slika 5: Logotip Github-a, vir[4]	8
Slika 6: Koda povezave na podatkovno bazo, lasten vir	10
Slika 7: PHP-koda za registracijo/pisanje uporabnika v bazo podatkov, lasten vir	11
Slika 8: PHP-koda za prikazovanje opreme, ki jo ima prijavljen uporabnik v lasti, lasten vir	11
Slika 9: Preverjanje, če je uporabnik prijavljen in ima dostop do te strani, lasten vir	12
Slika 10: Primer kodiranja in primerjanja gesel, lasten vir	13
Slika 11: Odziven CSS - značka, vir lasten	13
Slika 12: Izsek PHP-kode, ki filtrira uporabnike, lasten vir	14
Slika 13: PHP-koda za prikaz možnosti filtriranja uporabnikov, lasten vir	15
Slika 14: PHP koda za pošiljanje podatkov na email, lasten vir	17
Slika 15: Glavna stran administratorja aplikacije, lasten vir	18
Slika 16: Prikaz vseh administratorjev spletne aplikacije, lasten vir	19
Slika 17: Dodajanje novega administratorja spletne aplikacije, lasten vir	19
Slika 18: Stran za dodajanje novega društva, lasten vir	20
Slika 19: Stran za registracijo uporabnika, lasten vir	20
Slika 20: Prijavna stran za uporabnike, lasten vir	21
Slika 21: Glavna stran aplikacije, lasten vir	21
Slika 22: Pisanje poročila za opremo, lasten vir	22
Slika 23: Seznam mojih poročil, lasten vir	22
Slika 24: Urejanje računa uporabnika, lasten vir	23
Slika 25: Prikaz uporabnikov v društvu, lasten vir	23
Slika 26: Stran za urejanje društvenih podatkov, lasten vir	24
Slika 27: Prikaz vse opreme v društvu, lasten vir	24
Slika 28: Podroben prikaz o opremi, lasten vir	25
Slika 29: Dodajanje čiščenja za opremo, lasten vir	25
Slika 30: Seznam vseh napisanih poročil v društvu, lasten vir	26
Slika 31: Podrobnosti poročila, lasten vir	26

Slika 32: Dodajanje nove opreme, lasten vir.....	27
Slika 33: ER diagram podatkovne baze, lasten vir.....	34

KRATICE

Angl. – polni pomen iz angleškega jezika

DHL – nemško logistično podjetje (angl. Dalsey Hillblom Lynn)

ERP – integrirano upravljanje glavnih poslovnih procesov (angl. Enterprise resource planning)

Ipd. – in podobno

RFID – radiofrekvenčna identifikacija (angl. Radio Frequency IDentification)

PHP – programski jezik (angl. Personal Home Page Tools)

SQL – strukturirani povpraševalni jezik za delo s podatkovnimi bazami (angl. Structured Query Language)

XAMPP – paket rešitev spletnega strežnika (angl. cross-platform Apache MariaDB PHP Perl)

PDO – vmesnik za dostop do baz podatkov (angl. PHP Data Object)

DDL – sintaksa za ustvarjanje in spreminjanje objektov podatkovnih baz (angl. Data Definition Language)

HTTP – metoda za prenos informacij po spletu (angl. HyperText Transfer Protocol)

ANSI/IOS – standard za programski jezik (angl. American National Standards Institute/ International Organization for Standardization)

PDO – (angl. PHP Data Object)

Npr. – na primer

CSS – predloga, ki skrbi za izgled spletne strani (angl. Cascading Style Sheets)

HTML – označevalni jezik za izdelavo spletnih strani (angl. Hyper Text Markup Language)

CRUD – ustvari, beri, posodobi in izbriši (angl. Create, Read, Update and Delete)

ID – identifikacijska številka

Univ. – univerzitetni

Dipl. – diplomirani

Inž. – inženir

1. UVOD

Opazil sem, da se v našem gasilskem društvu soočamo s težavo, to je s popisovanjem osebne zaščitne opreme. Trenutno se to piše v Excel tabele ali pa kar na list papirja, kar pa samo delo precej podaljša in zakomplicira. Prav tako pa je takšen sistem precej nepregleden in težek za posodabljanje. Ker pa so še te datoteke po navadi na lokalnem računalniku, to še dodatno oteži delo. Tako sem prišel do ideje, da bom razvil aplikacijo, ki bo to vse olajšala in poenostavila.

1.1 Hipoteze

- Aplikacija bo poenostavila in izboljšala nadzor osebne zaščitne opreme.
- Aplikacija bo omogočala dostop do podatkov s katerekoli naprave, povezane na internet.
- Hitrost dostopa do podatkov in njihova preglednost bosta večji.

1.2 Namen in cilj

Namen raziskovalne naloge je gasilskim društvom poenostaviti popisovanje osebne zaščitne opreme z izdelavo spletne aplikacije, ki bo povezana s podatkovno bazo, seznam opreme in gasilcev v društvu pa bo lepo pregleden. Aplikacija bi uporabniku nudila preprost pregled in intuitivno uporabo. Prav tako imajo uporabniki možnost vpogleda v zgodovino posamezne opreme (popravila, čiščenja).

Spletno aplikacijo sem predstavil poveljniku v našem gasilskem društvu, ki mi je dal nekaj njegovega mnenja, predloge za izboljšavo in na sploh kaj naj dodam, da bi uporabniki imeli čim več od aplikacija.

2. PREGLED OBSTOJEČIH OBJAV

2.1 Podobne obstoječe rešitve

Trenutno obstaja že nekaj aplikacij za popisovanje oziroma inventuro opreme. Izpostavil bi nekaj večjih, ki so bolj pogosto uporabljena tudi v večjih slovenskih in tudi tujih podjetjih. Bolj kot ne, pa vse te rešitve delujejo na podobnem principu, vendar so bolj optimizirane in prilagojene za delovanje v večjih podjetjih in za popisovanje večjih količin osnovnih sredstev v čim krajšem času. Vendar pa te aplikacije nimajo toliko dodatnih funkcij, kot jih ima moja aplikacija.

2.1.1 Frontman FIND

Frontman FIND je eden izmed najbolj razširjenih sistemov za popis osnovnih sredstev v Sloveniji. Uporabljajo ga podjetja, kot so Akrapovič, Gorenje, DHL in Gen-i. Za popisovanje uporablja prenosni terminal, ki pošilja podatke na ERP- program. Temelji na uporabi črtnih kod, s čimer olajša popisovanje. Omogoča pa tudi izvoz v Excel, prav tako pa je združljiv z večino ERP-programov.



Slika 1: Logotip Frontman FIND, vir[1]

2.1.2 Inventura OS

Inventura OS je podoben sistem kot Frontman FIND. Omogoča hitro in učinkovito popisovanje ter napredno filtriranje podatkov. Prav tako nam omogoča hitrejšo in preglednejšo analizo

sprememb (obdelanost, odpis, slaba nalepka ipd.). Rešitev inventura OS vsebuje ročne terminale, programsko opremo za ročne terminale in računalnike ter nalepke s črtno kodo. Uporabljamo ga lahko tudi za sledenje vrednejše opreme. Uporabljajo ga v številnih šolah, trgovinah, kulturnih ustanovah ipd.



invos

Slika 2: Logotip InveturaOS, vir[2]

2.1.3 Sistem OSA

Sistem OSA prav tako, kot ostala dva, deluje na sistemu črtnih kod, vendar ta sistem lahko uporablja tudi RFID-oznake. Nato se z ročnimi terminali vsa oprema skenira in shrani v sistem. Omogoča tudi vnos dodatnih podatkov (slaba nalepka, predlog za odpis), podatki pa se osvežujejo in uskladijo večkrat na dan.

3. OGRODJA ZA IZDELAVO APLIKACIJE

3.1 Izbira programskega jezika in okolja

Za izdelavo spletne aplikacije sem se odločil, da bom uporabljal programski jezik PHP, ker sem z njim najbolj seznanjen ter smo ga uporabljali v 3. letniku. Kodo sem pisal v urejevalniku besedila Atom, ker ima veliko zastoj paketov, ki olajšajo samo programiranje. Za ustvarjanje baze sem uporabil Toad Data Modeler, ker smo se ga učili uporabljati v šoli in mi je preprost za uporabo, za delo s podatkovno bazo pa sem si izbral jezi SQL in sistem MySQL, ker sem z njim največ delal ter je primeren za uporabo s PHP-jem. Za varnejšo povezavo med spletno aplikacijo in podatkovno bazo sem uporabil PDO, ker mi je bil priporočen. Za razvoj na svojem računalniku pa sem uporabljal XAMPP, ki simulira strežnik na lokalnem računalniku. Za shranjevanje in varnostno kopijo ter zato, ker sem kodo pisal na dveh računalnikih, sem uporabljal Github.

3.1.1 PHP

PHP ('Personal Home Page Tools' ali slovensko orodja za osebne spletne strani) je razširjen odprtokodni programski jezik, ki se uporablja predvsem za razvoj dinamičnih spletnih strani in vsebin. Podoben je običajno strukturiranim programskim jezikom, kot sta C in Perl. Zaradi podobnosti bolj izkušenim programerjem dovoljuje razvijanje zapletenih programov brez dolgega in zapletenega učenja. Napisal ga je dansko-kanadski programer Rasmus Lerdorf, leta 1994. PHP primarno teče na spletnem strežniku, kjer jemlje izvorno PHP-kodo za vhod in generira spletno stran kot izhod. Možnost je tudi zaganjanje skript v ukaznem načinu in kreiranje grafičnih aplikacij.

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

    <?php
      echo "Hello World!";
    ?>

  </body>
</html>
```

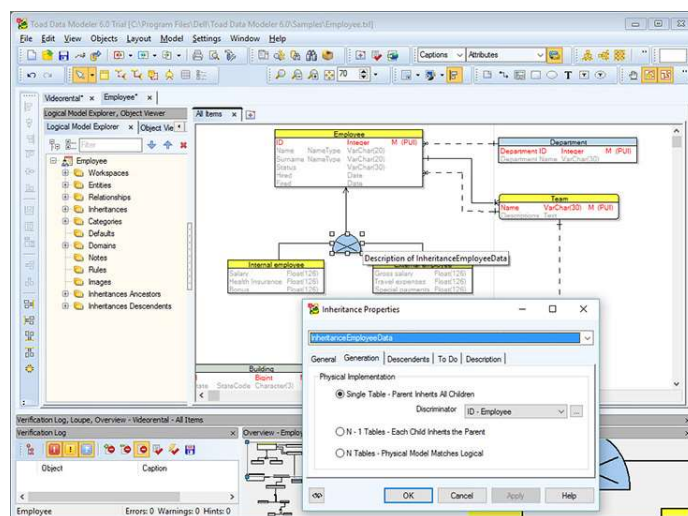
Slika 3: Osnovni primer PHP kode, lasten vir

3.1.2 Atom

Atom je brezplačen odprtokodni urejevalnik besedila za operacijske sisteme Windows, Linux in MacOS. Ima podporo za številne vtičnike, ki so napisani v JavaScriptu in ima vgrajen Git Control. Večina razširitvenih paketov ima licenco za brezplačno programsko opremo in so narejeni in vzdrževani s strani skupnosti. Atom je osnovan na osnovi Electrona (odprtokodni program, ki ga je razvil GitHub, in je namenjen razvoju namiznih grafičnih uporabniških vmesnikov, z uporabo spletnih tehnologij). Po zahvali veliko tem (angl. themes) oziroma motivom in razširitvenih paketov je zelo prilagodljiv uporabniku. Za Atom njegovi ustvarjalci pravijo, da je »hackable« urejevalnik besedila, kar pomeni, da je popolnoma prilagodljiv vsakemu posameznemu uporabniku (od izgleda pa vse do spreminjanja kombinacij tipk). Atom je bil dan v uporabo 25. junija 2015.

3.1.3 Toad Data Modeler

Toad Data Modeler je orodje za izdelavo baz podatkov in omogoča uporabniku, da vizualno ustvari in ureja sistem podatkovnih baz. Uporabljen je za konstrukcijo logičnih in fizičnih podatkovnih modelov, primerja in sinhronizira modele ter generira kompleksne SQL- oziroma DDL-ukaze. Lahko je tudi uporabljen za oblikovanje, vzdrževanje in dokumentacijo podatkovnih baz.



Slika 4: Primer grafičnega vmesnika Toad Data Modeler-ja, vir[3]

3.1.4 XAMPP

XAMPP je brezplačen odprtokodni paket paketov razširitev spletnega strežnika. Je najbolj popularno okolje za razvijanje PHP-kode. Prav tako pa podpira tudi MariaDB, Perl in vključuje Apache HTTP Server. Razvili so ga Apache Friends. Ker pa večina dejanskih serverjev tudi uporablja enake komponente kot XAMPP, pa je tudi prehod iz lokalnega serverja na dejanski, živ server mogoč in kar se da preprost.

3.1.5 SQL in MySQL

SQL (angleško Structured Query Language) je strukturirani povpraševalni jezik za delo s podatkovnimi bazami in je najbolj razširjen in standardiziran povpraševalni jezik za delo s podatkovnimi zbirkami. Določen je z ANSI/ISO SQL-standardom, ki so ga razvili leta 1986. Obstaja več verzij SQL-standarda in so poimenovani z imenom SQL:letnica (npr. SQL:2003 se navezuje na različico iz leta 2003). Za iskanje po zbirki podatkov uporablja programske stavke, ki posnemajo ukaze v naravnem jeziku. Omogoča pa nam, da v podatkovno bazo vpišemo podatke, jih beremo, posodabljammo ali pa brišemo. Za te štiri osnovne operacije je skupno ime CRUD (create, read, update, delete).

MySQL pa je sistem za upravljanje s podatkovnimi bazami in je odprtokodna implementacija relacijske podatkovne baze, ki za delo s podatki uporablja prej omenjen SQL-jezik. Razvija ga podjetje Oracle Corporation. Deluje po sistemu odjemalec – strežnik. Strežnik lahko namestimo kot sistem, porazdeljen na več strežnikov. Obstaja veliko število odjemalcev, zbirk ukazov in programskih vmesnikov za dostop do podatkovne baze MySQL. Napisan je v jezikih C in C++ in deluje na operacijskih sistemih Windows, MacOS in Lunux.

3.1.6 PDO

PHP Data Object (PDO) je lahek dosleden vmesnik za dostop do baz podatkov MySQL v PHP-aplikaciji. Vsak gonilnik podatkovne baze, ki implementira PDO, lahko posebne funkcije, povezane s podatkovno bazo, izpostavi kot običajne razširitvene funkcije. V aplikaciji ne moremo izvesti nobene funkcije PDO same po sebi, zato moramo vključiti datoteko, ki vsebuje povezavo s podatkovno bazo in implementira PDO-vmesnik, da lahko PDO uporabljamo v naši aplikaciji. PDO prav tako ponuja abstrakcijsko plast dostopa do podatkov, kar pomeni, da ne glede na to, katero zbirko podatkov uporabljamo, vedno pišemo iste funkcije za izvajanje poizvedb in pridobivanje podatkov, vendar pa ne more preurejati SQL-a ali posnemati manjkajočih funkcij.

3.1.7 Github

Github je ponudnik, je spletni gostiteljski servis za repozitorije/odlagališča v Git. Ponuja distribuirano upravljanje z izvirno kodo z Git ter nekatere dodatne storitve. Za razliko od Git-a, ki je ukazno orodje, Github uporablja grafični vmesnik za boljšo uporabniško izkušnjo. Svoje osnovne storitve ponuja brezplačno, naprednejše strokovne in podjetniške storitve pa so komercialne. Večina brezplačnih Github računov je uporabljenih za gostovanje odprtokodnih projektov. Omogoča tudi, da ima vsak projekt več sodelavcev, ki skupaj razvijajo kodo.



Slika 5: Logotip Github-a, vir[4]

3.1.7.1 Git

Git je programska oprema, namenjena sledenju kakršnih koli sprememb v katerem koli naboru datotek, ki se običajno uporabljajo za usklajevanje dela med programerji, ki med razvojem programske opreme skupaj razvijajo izvorno kodo. Njeni cilji vključujejo hitrost, celovitost podatkov in podporo porazdeljenim nelinearnim potekom dela (več vzporednih vej). Ustvaril ga je Linus Torvalds za razvoj Linux jedra, s pomočjo drugih razvijalcev jedra. Kot večina sistemov za nadzor in za razliko od večine sistemov odjemalec-strežnik je vsaka Git-ova mapa na vsakem računalniku polno skladišče s popolno zgodovino in možnostjo sledenja zgodovini, neodvisno od internetne povezave.

3.2 Načrt izdelave aplikacije

Najprej sem se pogovoril s poveljnikom v našem gasilskem društvu, da sem dobil približen občutek, kakšne funkcije bi aplikacija imela, kaj vse bi vsebovala, da bi bila čim bolj uporabna ter koristna. Moral sem si zamisliti sistem registracije, saj se v samo aplikacijo ne sme prijaviti vsakdo. Na računalniku sem na začetku napisal, kaj vse hočem, da je v moji aplikaciji, in kako mora delovati, katere vse strani in podstrani bo vsebovala. Ta seznam pa sem sproti ob ustvarjanju aplikacije malo spreminjal in dodeloval, če sem se še spomnil kakšne dobre ideje, ki mi je bila všeč in se mi je zdela primerna.

Čeprav bi aplikacija imela dosti več funkcij kot Excel tabela za popisovanje opreme, pa sem bil odločen, da bo moja glavna naloga, da bo bolj preprostejša za uporabo, kot prejšnji sistemi, ki

pa so bile Excel tabele ali pa kar list papirja. Enostavnejša uporaba pa bi tudi pripomogla k boljšemu beleženju same opreme, saj to ne bo več tako kompliciran proces.

4. IZDELAVA APLIKACIJE

Na začetku sem se lotil izdelovanja modela podatkovne baze v Toad Data Modelerju. Narediti sem moral več verzij baze, saj sem s sprotnim razvijanjem aplikacije našel kakšen problem, da ne deluje tako, kot je potrebno. Nato sem s pomočjo programa izvozil bazo v SQL-datoteko. Sam sem to datoteko uredil tako, da sem še napisal nekaj kod, ki v bazo dodajo nekaj osnovnih podatkov (velikosti, naziv obleke ali opreme, imena proizvajalcev, rangi uporabnikov). Urejeno SQL-datoteko sem potem uvozil v XAMPP phpMyAdmin. Najprej sem moral narediti PHP-datoteko, ki nam omogoča povezavo s podatkovno bazo in nam omogoča uporabo PDO. To datoteko je potrebno vedno vključiti v PHP-kodo, kadar uporabljamo podatkovno bazo.

```
<?php
    $host='localhost';
    $db='firegear';
    $user='root';
    $pass='';
    $charset='utf8mb4';

    $dsn="mysql:host=$host;dbname=$db;charset=$charset";

    $options=[
        PDO::ATTR_ERRMODE            =>PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE=>PDO::FETCH_ASSOC,
        PDO::ATTR_EMULATE_PREPARES  =>false,
    ];

    try{
        $pdo=new PDO($dsn,$user,$pass,$options);
    }
    catch(PDOException $e){
        throw new PDOException($e->getMessage(), (int)$e->getCode());
    }
?>
```

Slika 6: Koda povezave na podatkovno bazo, lasten vir

Nato sem se lotil izdelave strani za registracijo z HTML osnovo in zanjo izdelal izgled s CSS-datoteko. Strani sem dodal še PHP-kodo, ki skrbi za pravilno registracijo uporabnika. Sama registracija je sestavljena iz dveh delov, prvi je kot prijava (uporabniško ime in koda gasilskega društva), drugi pa je dejanska registracija.

```
if($ime!=null && $email != null && $geslo != null &&$geslo2 != null && $datum_roj != null){  
    if($geslo==$geslo2){  
        try{  
            include 'sql.php';  
            $stmt = $pdo->query("SELECT id FROM drustva WHERE username='".$$_SESSION['drustvo']."'");  
            while ($row = $stmt->fetch()) {  
                $drustvo_id=$row['id'];  
            }  
            $stmt = $pdo->query("SELECT * FROM users WHERE email='".$$_email."'");  
            $stetje=0;  
            while ($row = $stmt->fetch()) {  
                $stetje=$stetje+1;  
            }  
            if($stetje==0){  
                $geslohash=password_hash($geslo, PASSWORD_DEFAULT);  
                $sql = "INSERT INTO users (ime,email,pass,birthday,drustvo_id,rank_id) VALUES (?,?,?,?,?,?)";  
                $stmt= $pdo->prepare($sql);  
                $stmt->execute([$ime, $email, $geslohash, $datum_roj, $drustvo_id,2]);  
                $dbh=null;  
                $_SESSION['UserIme']=$ime;  
                $_SESSION['UserPriimek']=$priimek;  
                $_SESSION['UserSlika']=$target_file;  
                header('Location:glavni_meni.php');  
            }  
            else{  
                header('Location:registracija2.php');  
            }  
        }  
    }  
}
```

Slika 7: PHP-koda za registracijo/pisanje uporabnika v bazo podatkov, lasten vir

Za prijavno stran sem lahko prekopiral HTML- in CSS-datoteki, ker sta strani skoraj enaki, in ju malo preuredil. Ustvaril sem PHP-datoteko, ki preveri pravilnost uporabniškega imena in gesla za prijavo in nato preusmeri na glavno ali pa nazaj na prijavno stran. Da sem lahko shranil podatke o prijavljenem uporabniku in jih uporabljal v aplikaciji, pa sem uporabljal sejo (angl. session), ki začasno shranjuje podatke. Naslednja je bila glavna stran, na katero smo preusmerjeni ob prijavi. Izdelal sem PHP-datoteko, v katero sem vstavil osnovne HTML-elemente, nato pa dodal še PHP-kodo, ki bere iz podatkovne baze. V phpMyAdmin sem vstavil nekaj testnih podatkov, da sem lahko preveril, ali aplikacija deluje pravilno. Dodal sem še datoteko CSS, da so podatki lepše prikazani.

```
include 'sql.php';  
//session_start();  
$stmt = $pdo->query("SELECT o.serijska_st AS serijska, n.ime AS naziv, p.ime AS proizvajalec, v.ime AS velikost, o.id AS id FROM oprema AS o  
o INNER JOIN naziv AS n ON n.id=o.naziv_id INNER JOIN proizvajalec AS p ON  
p.id=o.proizvajalec_id INNER JOIN velikosti AS v ON v.id=o.velikost_id INNER JOIN users AS u ON u.id=o.user_id WHERE  
u.id='".$_SESSION['UserId']."'");  
while ($row = $stmt->fetch()) {  
    echo "<tr><td>".$row['serijska']. "</td><td>".$row['naziv']. "</td><td>".$row['proizvajalec']. "</td><td>".$row['velikost']. "</td><td><a  
href='napisi_porocilo.php?id=".$row['id']."'>Napiši poročilo</a></td></tr>";  
}  
$pdo=null;
```

Slika 8: PHP-koda za prikazovanje opreme, ki jo ima prijavljen uporabnik v lasti, lasten vir

Nato sem se lotil izdelovanja vsake strani posebej. Najprej sem naredil stran za dodajanje opreme, da sem lahko to delal v aplikaciji in ne direktno v bazi podatkov. Ker pa sem imel v bazi že shranjene nekatere parametre, kot so proizvajalec, velikost in naziv opreme, sem moral to najprej prebrati iz baze v PHP-ju s SELECT stavki in te podatke prikazati v spustnem meniju. Ko sem uredil, da se ob pravilno vnešenih podatkih oprema shrani v bazo, sem uredil še izgled strani. Vse kar sem naredil do sedaj, se na naslednjih straneh v bistvu ponavlja, vendar ima vsakič drugačen namen. Potem pa sem se lotil še ostalih strani, kot so urejanje svojega računa, pregled svojih poročil, urejanje podatkov društva (uporabniško ime in koda društva), vse člane, ki so prijavljeni v društvu, vsa poročila in oprema iz društva ter dodajanje nove opreme. Vse te strani so bile narejene po istem postopku kot prejšnje, torej sem najprej naredil HTML-stran, dodal sem jim PHP-kodo, da po potrebi bere podatke iz baze podatkov, ter na koncu sem dodal še CSS-datoteko za izgled. Za boljšo uporabniško izkušnjo sem dodal tudi nekaj Bootstrap elementov, kot sta spustni meni in pojavno okno. Za dodatno funkcijo sem dodal tudi filtriranje uporabnikov. Ker pa sem ugotovil, da na nekatere strani ne sme dostopati vsak uporabnik, ampak samo administrator v vsakem društvu, pa sem na strani dodal tudi preverjanje položaja (angl. rank) uporabnika.

```
session_start();
if((!isset($_SESSION['UserId'])) || ($_SESSION['UserRank']!=1)){
    header("Location:index.php");
}
```

Slika 9: Preverjanje, če je uporabnik prijavljen in ima dostop do te strani, lasten vir

4.1 Kodiranje gesel

Če se v kašnem primeru slučajno zgodi, da pride v našo podatkovno bazo kakršna koli tretja oseba, ki ne bi smela videti podatkov, ki jih imamo shranjene, in so gesla zapisana v navadnem zapisu, točno tako, kot jih mi zapišemo v aplikacijo (npr. »geslo123«), s tem puščamo občutljive podatke, so kot na dlani roke tej tretji osebi, ki tega ne bi smela videti. Torej puščamo občutljive podatke, ki so nam jih uporabniki zaupali. Da pa to preprečimo, uporabljamo funkcije, ki geslo enosmerno zakodirajo. Ker pa to dandanes ni več dovolj, pa geslu dodamo tudi sol – to je nek niz znakov, ki se dodajo geslu in se potem to oboje skupaj zakodira (primer kodiranja gesla, če je originalno geslo: »geslo123«, se je v podatkovno bazo v tem primeru shranilo kot: »\$2y\$10\$zdR22w/rX6n9MEej1qGcj.oMeumCJeymzkTR9b/EGAV0i3.l0/3HW«). Namen soli

pa je, da te predračunane dekodeerje kodiranih gesel naredijo neuporabne, saj zaradi dodane soli v geslu vrnejo neko popolnoma drugo geslo. Prav zaradi tega sistema pa ni mogoče gesel obnoviti, ampak jih lahko samo spremenimo. PHP ima za to funkcije `sha1()`, ki ni več tako varen, `md5()`, `password_hash()` in še nekaj drugih. Jaz sem uporabljal funkcijo `password_hash()`, ki geslu avtomatsko doda sol ter ga zakodira, da pa lahko preverimo pravilnost gesla pri prijavi, pa uporabimo funkcijo `password_verify()`, ki nam vrne `true`, če se gesli ujemata in `false`, če se ne.

```
$gesloHash=password_hash("geslo123", PASSWORD_DEFAULT);

if(password_verify("geslo123",$gesloHash)==true){
    //gesli se ujemata
}
else{
    //gesli se ne ujemata
}
```

Slika 10: Primer kodiranja in primerjanja gesel, lasten vir

4.2 Odziven CSS

Ker dandanes uporabljamo številne različne naprave za ogled spletnih vsebin, kar je tudi ta spletna aplikacija, sem moral v aplikacijo vključiti odziven CSS, ki mi omogoči, da imajo elementi na strani različne attribute glede na velikost zaslona, na katerem gledamo. Za pravilno delovanje moramo najprej v HTML-ju v `<head>` značko dodati `<meta name="viewport" content="width=device-width, initial-scale=1.0">`, kar nam omogoči, da vemo, kakšne dimenzije je zaslon, na katerem smo. Nato pa v CSS-datoteki uporabimo pravilo `@media`, ki vključi CSS-atribute znotraj njega samo, kadar je pogoj zraven pravilen (npr. kadar je minimalna širina zaslona 600 pikslov z oznako `min-width:600px`, kot prikazuje slika spodaj).

```
@media screen and (min-width: 600px){
    .logo{
        width:30%;
    }
    .container{
        width:60%;
        margin-top:20px;
    }
}
```

Slika 11: Odziven CSS - značka, vir lasten

4.3 Filtriranje uporabnikov

Za dodatno funkcijo sem dodal tudi filtriranje uporabnikov, ki mi jo je predlagal poveljnik društva. Glede na vnesene parametre pa nam prikaže vse uporabnike, ki imajo/nimajo (glede na to, kaj izberemo) določene opreme. Ta funkcija v bistvu preveri vse, ki imajo v lasti izbrano opremo iz društva in jih primerja z vsemi uporabniki v društvu in nam rezultat izpiše v tabeli, kot prikaz vseh uporabnikov, ki ustrezajo izbranim kriterijem. Ta funkcija lahko precej skrajša čas pri iskanju, katera oprema nam še manjka, da bodo vsi uporabniki opremljeni z vso potrebno opremo.

```
if($_GET['enacaj']=='nienako'){
    $stmt = $pdo->query("SELECT u.ime AS ime, u.email AS email ,u.birthday AS bday, r.ime AS rankN ,u.id AS uid FROM users AS u INNER JOIN ranki AS r
    ON r.id=u.rank_id WHERE u.drustvo_id=".$drustvo_id." AND u.id NOT IN (SELECT u.id FROM users AS u INNER JOIN ranki AS r ON r.id=u.rank_id INNER
    JOIN oprema AS o ON o.user_id=u.id WHERE u.drustvo_id=".$drustvo_id." AND o.naziv_id=".$_GET['oprema'].");");
    while ($row = $stmt->fetch()) {
        if($row['email']==$_SESSION['UserEmail']){
            echo "<tr><td>".$row['ime']. "</td><td>".$row['email']. "</td><td>".date("d/m/Y", strtotime($row['bday'])). "</td><td>".$row['rankN']. "</td><td><td><a href='uredi_user.php?id=".$row['uid']."'>Uredi</a></td></tr>";
        }
        else{
            echo "<tr><td>".$row['ime']. "</td><td>".$row['email']. "</td><td>".date("d/m/Y", strtotime($row['bday'])). "</td><td>".$row['rankN']. "</td><td><a href='user_izbris.php?id=".$row['uid']."'>Izbris</a></td><td><a href='uredi_user.php?id=".$row['uid']."'>Uredi</a></td></tr>";
        }
    }
    $pdo=null;
}
```

Slika 12: Izsek PHP-kode, ki filtrira uporabnike, lasten vir


```
<select class="" name="enacaj">
  <?php
  if(isset($_GET['enacaj'])&&isset($_GET['oprema']))){
    if($_GET['enacaj']=='enako'){
      echo "<option value='enako' selected='selected'>imajo</option>";
      echo "<option value='nienako'>nimajo</option>";
    }
    else{
      echo "<option value='enako'>imajo</option>";
      echo "<option value='nienako' selected='selected'>nimajo</option>";
    }
  }
  else{
    echo "<option value='' disabled selected='selected'></option>";
    echo "<option value='enako'>imajo</option>";
    echo "<option value='nienako'>nimajo</option>";
  }
  ?>
</select>
<select class="" name="oprema">
  <?php
  include 'sql.php';
  $stmt = $pdo->query("SELECT id,ime FROM naziv");
  if(isset($_GET['oprema'])&&isset($_GET['enacaj']))){
    while ($row = $stmt->fetch()) {
      if(isset($_GET['oprema']) && $row['id']==$_GET['oprema']){
        echo "<option value='".$row['id']."' selected='selected'>".$row['ime']."</option>";
      }
      else{
        echo "<option value='".$row['id']."'>".$row['ime']."</option>";
      }
    }
  }
  else{
    echo "<option value='' disabled selected='selected'></option>";
    while ($row = $stmt->fetch()) {
      echo "<option value='".$row['id']."'>".$row['ime']."</option>";
    }
  }
  ?>
</select>
```

Slika 13: PHP-koda za prikaz možnosti filtriranja uporabnikov, lasten vir

4.4 Administratorska plošča

Hitro po začetku izdelave aplikacije sem prišel do spoznanja, da moja aplikacija potrebuje še neko administracijsko ploščo oziroma prostor, kjer bodo administratorji celotne spletna aplikacije lahko upravljali z gasilskimi društvi. Naredil sem preprosto prijavo, ampak nobene registracije, saj se ne sme nihče kar tako registrirati. Tukaj imajo seznam vseh društev v sistemu in lahko vidijo, kdo so njihovi administratorji oziroma superuporabniki ter njihove e-mail

naslove, da jih lahko po potrebi kontaktirajo. Vsako društvo pa lahko tudi izbrišejo. Administrator ima tudi možnost dodajanja več teh administratorjev, da jih lahko več opravlja isto delo. Vidi lahko seznam vseh, ki so administratorji spletne aplikacije. In pa seveda glavna funkcija te administratorske plošče je dodajanje novih gasilskih društev, kar pa je bil nov problem.

4.5 Dodajanje novih društev

Pri ustvarjanju novega društva pride do problema, da se morata administrator spletne aplikacije in poveljnik oziroma tisti, ki bo administrator v društvu, dogovoriti, kaj bo uporabniško ime društva in geslo. To samo po sebi ni precej zahteven proces, vendar sem ga vseeno olajšal. Administrator spletne aplikacije mora za dodajanje društva vnesti samo tri parametre. To so ime društva (npr. PGD Šoštanj-mest), uporabniško ime društva, ki je lahko nastavljeno po želji uporabnika po dogovoru, ali pa administrator spletne aplikacije sam nastavi preprosto uporabniško ime (npr. pgd.sostanj) ter elektronski naslov poveljnika ali administratorja v gasilskem društvu. Ko so ti podatki vnešeni, se preveri, če že slučajno kateri izmed njih ne obstaja v podatkovni bazi, in če ne, se ti podatki shranijo v bazo. Koda za društvo in geslo za uporabnika pa sta ob ustvaritvi društva avtomatsko zgenerirana. Poveljnik oziroma administrator gasilskega društva dobi na e-mail sporočilo, da je bil račun uspešno ustvarjen. Zraven dobi uporabniško ime društva in zgenerirano kodo ter svoj e-mail in geslo za prijavo (administrator društva je edini uporabnik, ki se mu ni potrebno registrirati). Z razvijanjem aplikacije pa sem naletel na nov problem, in sicer komu bo pripadala oprema, če nima lastnika oziroma je trenutno nihče ne lasti? Če ima oprema vrednost 'lastnik_id=null', je ne bomo imeli prikazane, saj ni preko uporabnika povezana do našega društva. Če pa v bazi ustvarimo uporabnika, ki ima ime 'Ni dodeljeno', ki ne pripada nobenemu društvu, pa naletimo na isti problem, razen če malo spremenimo kodo, v tem primeru pa nam prikaže opremo vseh društev, tudi tisto, ki je mi dejansko nimamo. Zato je bilo potrebno ustvariti še enega uporabnika z imenom 'Ni dodeljeno', ki pripada svojemu društvu.

```
$alphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890';
$pass = array();
$alphalength = strlen($alphabet) - 1;
for ($i = 0; $i < 8; $i++) {
    $n = rand(0, $alphalength);
    $pass[] = $alphabet[$n];
}
$pass2=array();
for ($i = 0; $i < 8; $i++) {
    $n = rand(0, $alphalength);
    $pass2[] = $alphabet[$n];
}
$geslo=implode($pass);
$adminGeslo=implode($pass2);

$prejemnik = $_POST['mail'];
$naslov = "Ustvarjen FireGear račun";
$vsebina = "Zdravo!\n Ustvarjeno je bil račun društva ".$društvo.". Prosimo, da se na spletni strani prijavite.
V aplikacijo se prijavite s tem mailom, vaše geslopa je: ".$adminGeslo."\n Podatki društva potrebni za registracijo drugih uporabnikov:\n
Uporabniško ime društva: ".$usernameDruštva."\n Koda društva: ".$geslo."\n Prosimo, da zaradi dodatne varnosti
takoј po prijavi spremenite svoje geslo, posodobite svoje podatke in spremenite kodo društva. Za več informacij nas kontaktirajte.\n
Na Pomoč!";
```

Slika 14: PHP koda za pošiljanje podatkov na email, lasten vir

4.6 Funkcija poročila

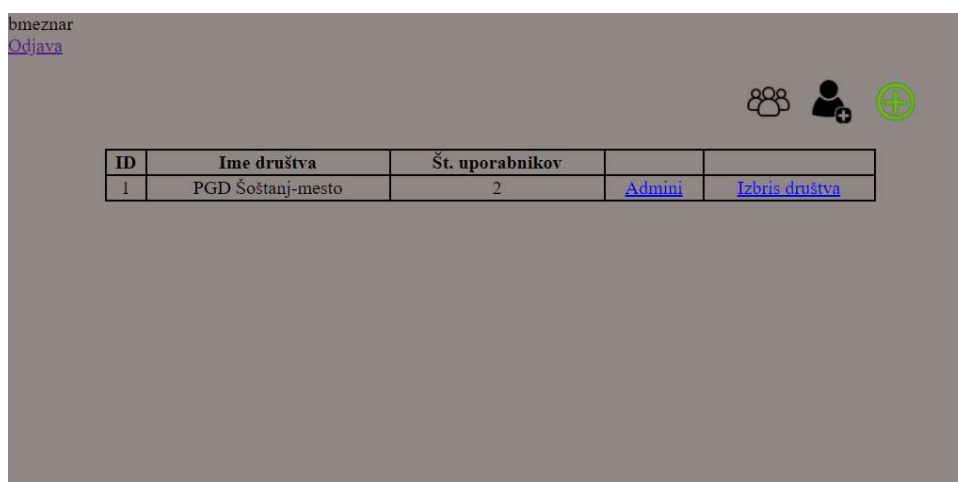
Ko se oprema redno uporablja, se seveda tudi kdaj poškoduje ali pa obrabi in jo je potrebno popraviti ali pa zamenjati. V ta namen sem dodal funkcijo poročila. Deluje tako, da lahko uporabnik, ki ima opremo v lasti, napiše poročilo za določen kos opreme, če je z njo kaj narobe ali kaj podobnega. Napisati more točno, v čem tiči problem ali v čem je napaka, da ni treba cele opreme »obrniti na glavo«, da najdemo napako. Vsa poročila, ki jih uporabniki pišejo, pa vidi administrator društva. Iz podatkov, ki jih je dobil, lahko preveri, v kakšnem stanju je kos opreme, za katerega je bilo napisano poročilo, in se odloči, ali je napaka dovolj velika, da gre na popravilo ali se zamenja. Ko je napaka odpravljena, pa lahko v aplikaciji obkljuka, da je bilo poročilo razrešeno, kar pa lahko vidi tudi uporabnik, ki je poročilo napisal.

5. DELOVANJE APLIKACIJE

Aplikacija je deljena na dva dela, in sicer del za administratorja spletne aplikacije, ki upravlja z gasilskimi društvi, in del za uporabnike/gasilce oziroma gasilska društva, ki popisujejo svojo osebno zaščitno opremo.

5.1 Administratorski del

Administrator spletne aplikacije ima ločeno prijavo od navadnih uporabnikov aplikacije. Ko se administrator pravilno prijavi, je preusmerjen na glavno stran administratorske plošče, kjer ima seznam vseh društev, ki obstajajo, in vse povezave, ki jih potrebuje. Za vsako društvo ima prikazan njegov ID, ime ter število uporabnikov. Če klikne na link »Admini«, pa lahko vidi seznam vseh, ki so administratorji v izbranem društvu (zraven ima tudi njihove e-mail naslove, da jih lahko, če je potrebno, kontaktira). Sama administratorska plošča je precej osnovna in nima toliko poudarka na uporabniški izkušnji, saj je njen primarni namen funkcionalnost, ki jo ponuja. Prav tako pa bodo z njo upravljali le izkušeni, ki imajo dostop, tako da ni potrebno, da je tako izpopolnjena glede izgleda in uporabniške izkušnje.



Slika 15: Glavna stran administratorja aplikacije, lasten vir

V zgornjem desnem kotu ima administrator tri povezave do drugih strani administratorske plošče. Prva povezava (od leve proti desni) je seznam vseh administratorjev spletne aplikacije,

ki lahko dostopajo do te administracijske plošče. Tu se mu v tabeli izpišejo z uporabniškim imenom in geslom.



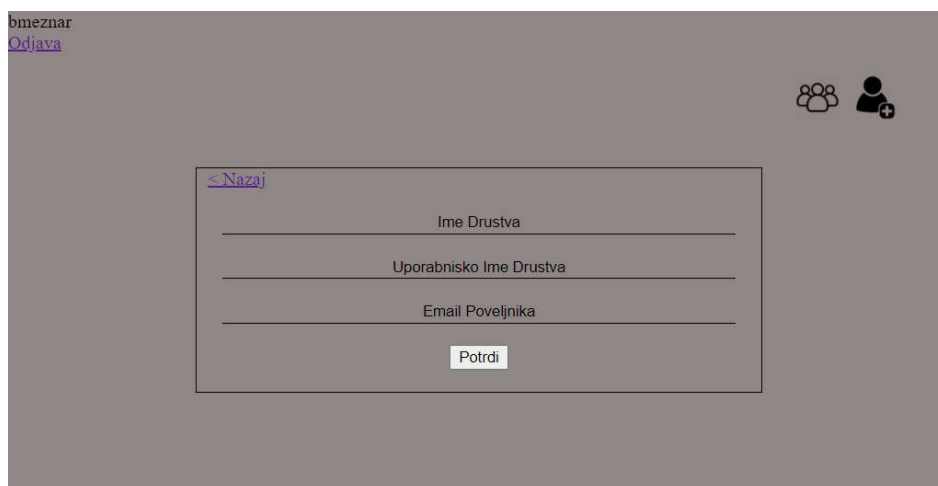
Slika 16: Prikaz vseh administratorjev spletne aplikacije, lasten vir

Naslednja povezava (druga od leve proti desni) pripelje administratorja do druge strani, kjer pa lahko doda novega administratorja za upravljanje s spletno aplikacijo. Doda ga samo z vpisanim uporabniškim imenom ter geslom.



Slika 17: Dodajanje novega administratorja spletne aplikacije, lasten vir

Tretja povezava (tretja od leve proti desni) pa uporabnika pripelje do glavne funkcije te administratorske plošče, to je dodajanje novih društev. Tukaj administrator vnese ime gasilskega društva, nato pa še njegovo uporabniško ime (lahko ga izbere poveljnik v posameznem društvu) ter elektronski naslov poveljnika oziroma tistega, ki bo upravljal z opremo v tistem gasilskem društvu.



bmeznar
[Odjava](#)

[< Nazaj](#)

Ime Društva

Uporabniško Ime Društva

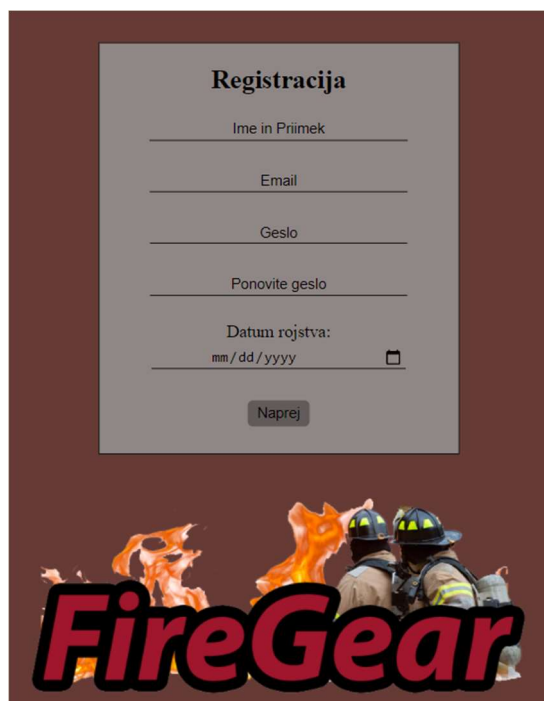
Email Poveljnika

Potrdi

Slika 18: Stran za dodajanje novega društva, lasten vir

5.2 Uporabniški del

Vsak uporabnik se mora najprej registrirati, za kar potrebuje uporabniško ime in kodo društva, v katerem je (dobi jo od poveljnika oziroma tistega, ki je administrator v društvu). Nato vnese svoje podatke, in če so vsi pravilni, je uporabnik registriran, nato pa se mora še enkrat prijaviti, da lahko vstopi v aplikacijo.



Registracija

Ime in Priimek

Email

Geslo

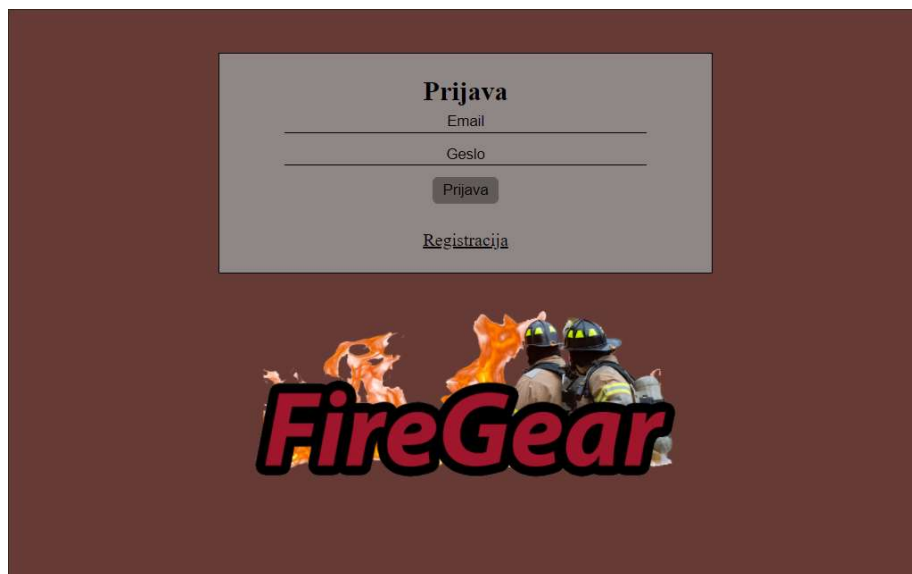
Ponovite geslo

Datum rojstva:
mm/dd/yyyy

Naprej

FireGear

Slika 19: Stran za registracijo uporabnika, lasten vir



Slika 20: Prijavna stran za uporabnike, lasten vir

Ko se navaden uporabnik prijavi v aplikacijo, ga takoj preusmeri na glavno stran, kjer je tabela, v kateri ima prikazano vso opremo, ki jo ima trenutno v lasti. Od vsakega kosa opreme ima prikazane osnovne lastnosti, kot so serijska številka, kateri kos opreme je to (jakna, škornji, hlače ...), proizvajalec, velikost ter povezava do strani, da napiše poročilo za opremo.



Slika 21: Glavna stran aplikacije, lasten vir

Če uporabnik klikne na povezavo »Napiši poročilo«, je preusmerjen na stran, kjer napiše naslov oziroma na kratko, v čem je problem, ter v opisu problema problem natančneje opiše, da ima administrator v društvu čim več informacij.



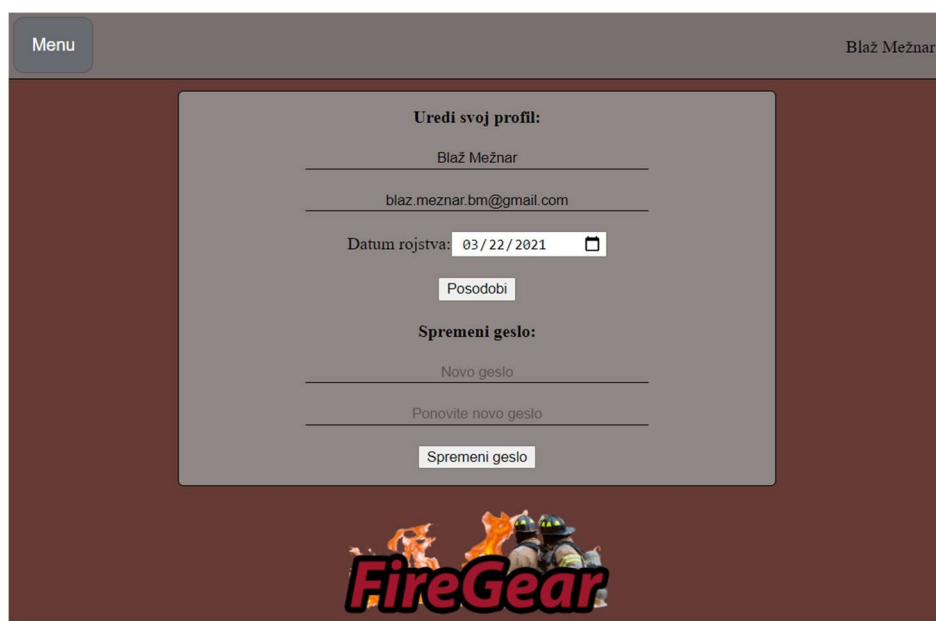
Slika 22: Pisanje poročila za opremo, lasten vir

Vsak uporabnik ima seznam vseh poročil, ki jih je napisal, kdaj jih je napisal, za katero opremo, kaj je bil problem in zraven znak, ali je bil problem že razrešen ali ne, če pa klikne povezavo več, pa prikaže še vse podatke in tudi daljši opis problema na novi strani.

Moja poročila				
Datum	Oprema	Naslov	Razrešeno	Informacije
30/03/2021	Čelada	Odtrgan pas	✖	Več

Slika 23: Seznam mojih poročil, lasten vir

Vsak uporabnik pa lahko seveda tudi spremeni svoje podatke, kot so ime in priimek, e-mail naslov ali pa datum rojstva. Prav tako si lahko zamenja geslo. Osebni podatki in geslo se posodablja ločeno, torej če uporabnik spremeni svoje osebne podatke in geslo, se bodo posodobili tisti podatki, pod katerimi klikne posodobi.



Slika 24: Urejanje računa uporabnika, lasten vir


To so vse funkcije, ki jih lahko navaden uporabnik uporablja. V vsakem društvu pa je tudi vsaj en administrator ali superuporabnik, ki ima več funkcij, kot ostali uporabniki. Prva izmed teh je pregled nad vsemi uporabniki in njihov filtriran prikaz. Če imamo v polju za filtriranje obe polji prazni, imamo prikazane vse uporabnike, če ne, pa samo tiste, ki ustrezajo iskanim kriterijem. Če poleg uporabnika klikne na povezavo »Več«, lahko uporabniku spremeni spremeni položaj (angl. rank) ali pa mi doda kakšen kos opreme.



Ime	Email	Datum rojstva	Rank	
Blaž Mežnar	blaz.meznar.bm@gmail.com	22/03/2021	admin	Uredi
Janez Koren	janez@gmail.com	10/03/2021	uporabnik	Izbrisi Uredi

Slika 25: Prikaz uporabnikov v društvu, lasten vir

Administrator v društvu ima prav tako možnost, da spremeni društvene podatke, to so uporabniško ime in koda, ki sta potrebna za novo registracijo uporabnika.



Slika 26: Stran za urejanje društvenih podatkov, lasten vir

Dodano ima tudi stran, na kateri lahko administrator društva vidi, katero vso opremo imajo in njene osnovne informacije, kot so naziv opreme, proizvajalec, model, velikost, serijska številka in lastnik. Če pa klikne na povezavo več, pa se mu prikažejo vse informacije o opremi, tudi katera vsa poročila so bila že zanjo napisana ter njeno zgodovino čiščenja, lahko pa se tudi doda zapis, da je bila obleka na čiščenju. Tu pa lahko tudi zamenjamo uporabnika te opreme.

Vsa oprema						
Naziv	Proizvajalec	Model	Velikost	Serijska številka	Uporabnik	
Jakna-ognjevarna	S-gard	Ranger	S1	456172831	Blaž Mežnar	Več
Škornji	Rosenbauer	/	43	721378124	Ni dodeljeno	Več
Čelada	Gallus	/	52-64	Ni vidno	Janez Koren	Več

Slika 27: Prikaz vse opreme v društvu, lasten vir

The screenshot shows a web application interface with a dark red background. At the top left is a 'Menu' button, and at the top right is the user name 'Blaž Mežnar'. The main content area is a light gray box containing the following details for a piece of equipment:

- Naziv: Jakna-ognjevarna
- Proizvajalec: S-gard
- Model: Ranger
- Oznaka: GO12
- Velikost: S1
- Serijska številka: 456172831
- Leto izdelave: 2020
- Uporabno do: Nima roka uporabe
- Uporabnik: Blaž Mežnar (dropdown menu)

Below these details are two buttons: 'Posodobi' and 'Dodaj čiščenje'. At the bottom of the gray box are two links: 'Vsa poročila' and 'Vsa čiščenja'. At the very bottom of the interface are two links: '< Nazaj na vso opremo' and 'Izbrši opremo'.

Slika 28: Podroben prikaz o opremi, lasten vir

The screenshot shows a form titled 'Novo čiščenje' with a close button (X) in the top right corner. The form contains the following elements:

- Four checkboxes for cleaning types: 'Dekontaminacija', 'Dezinfekcija', 'Mokro pranje', and 'Impregnacija'.
- A date field labeled 'Datum čiščenja:' with a placeholder 'mm/dd/yyyy' and a calendar icon.
- A text input field labeled 'Številka potrdila'.
- A text input field labeled 'cena'.
- A large text area labeled 'Opombe'.
- A 'Dodaj' (Add) button at the bottom.

Slika 29: Dodajanje čiščenja za opremo, lasten vir

Seveda pa mora tudi administrator imeti zbrana vsa poročila na enem mestu. Zato ima na eni strani seznam vseh poročil, ki so jih napisali uporabniki v društvu. Zanje ima napisano, kdaj je bilo poročilo napisano, naslov poročila ter majhen znak, ali je bil problem že razrešen ali ne.

Vsa poročila			
Datum	Naslov	Razrešeno	Info
30/03/2021	Odtrgan pas	✖	Več
29/03/2021	Strgan rokva	✖	Več
29/03/2021	Odtргana zadrѓa	✓	Več

Slika 30: Seznam vseh napisanih poročil v društvu, lasten vir

Če klikne na povezavo »Več«, se prikažejo vsi podatki o tem specifičnem poročilu in če še ni bilo razrešeno, lahko obkljuka, da je bilo, in shrani spremembe. Potem se na seznamu vseh poročil prikaže tudi zelena puščica, da je bilo razrešeno.

Datum poročila: 29/03/2021

Strgan rokva

Oprema: Jakna-ognjevarna S-gard S1 Serijska št:456172831

Uporabnik: Blaž Mežnar

Poročilo:
Jakna ima na desnem rokavu pri komolcu manjšo luknjo.

☐ Razrešeno

[< Nazaj](#)

Slika 31: Podrobnosti poročila, lasten vir

Seveda pa mora administrator v društvu tudi imeti možnost dodajanja nove opreme. To je v aplikaciji precej preprosto in intuitivno, saj je sistem zelo preprost in nas opozori, če smo katero od potrebnih informacij pozabili vnesti.

The screenshot shows a web application interface for adding new equipment. At the top, there is a dark grey header bar with a 'Menu' button on the left and the user's name 'Blaž Mežnar' on the right. The main content area has a dark red background. In the center, there is a light grey rectangular form titled 'Dodaj novo opremo:'. The form contains several input fields: a dropdown menu for 'Jakna-ognjevarna' (set to 'Jakna-ognjevarna'), a dropdown for 'Proizvajalec:' (set to 'Z. Ptuj'), a text field for 'Model opreme', a dropdown for 'Velikost:' (set to 'S1'), a text field for 'Serijska številka' with a checkbox for 'Ni vidno', a text field for 'Oznaka', a text field for 'Leto izdelave', and a text field for 'Uporabno do (letnica)' with a checkbox for 'Nima življ. dobe'. At the bottom of the form is a 'Potrdi' button.

Slika 32: Dodajanje nove opreme, lasten vir

6. RAZPRAVA

Ob ustvarjanju te spletne aplikacije sem poglobil svoje znanje o uporabi programskega jezika PHP. Prav tako pa sem utrdil znanje HTML- in CSS-jezika. Ta dva mi večinoma nista delala preglavic, če pa je prišlo do kakšnega problema, pa sem ga s pomočjo svetovnega spleta hitro rešil. Nekaj več težav mi je delal PHP, ampak ne sam jezik, temveč bolj to, kako bi rešil problem, ki ga imam v aplikaciji. Seveda sem si lahko zopet pomagal z svetovnim spletom, ampak je bilo nekatere rešitve malo težje najti, saj so bile precej specifične. Sem se pa iz vsega tega naučil, da programiranje samo po sebi ni težko. Zahteven je tisti del, kjer moraš sam najti napake v kodi, probleme v delovanju in to rešiti. Prav tako je velik del tega na svetovnem spletu, saj lahko z njegovo pomočjo najdemo veliko odgovorov in rešitev za naše napake in probleme pri programiranju, če le znamo pravilno iskati.

Če bi primerjali mojo aplikacijo z obstoječimi rešitvami, bi ugotovili, da se kar dosti razlikujejo, saj imajo obstoječe aplikacije poudarek na hitrosti popisovanja večjih količin osnovnih sredstev in imajo tudi funkcije, prilagojene za te namene. Medtem ko pa ima moja aplikacija poudarek na evidentiranju osebne zaščitne opreme, predvsem v gasilskih društvih. V ta namen ima tudi funkcije, prilagojene za to delo in s tem olajšajo nadzor nad osebno zaščitno opremo.

6.1 Pregled hipotez

Po končani raziskovalni nalogi in izdelavi spletne aplikacije za evidentiranje osebne zaščitne opreme sem imel dovolj informacij, da sem lahko potrdil oziroma ovrgel zadane hipoteze.

1. Aplikacija bo poenostavila in izboljšala nadzor osebne zaščitne opreme.

To hipotezo lahko le delno potrdim, saj aplikacije v tem trenutku še nisem dal nikomur za uporabo, so pa tisti v našem gasilskem društvu, ki sem jim aplikacijo testno pokazal, rekli, da bi ta aplikacija precej poenostavila evidenco osebne zaščitne opreme, saj trenutno uporabljajo Excel tabele, ki pa zahtevajo dosti več dela, da vanjo vstavimo malce več kosov opreme in zanjo napišemo vse parametre, še posebno, če jih nismo precej večji uporabljati.

2. Aplikacija bo omogočala dostop do podatkov s katerekoli naprave, povezane na internet.

Izdelal sem odzivno spletno aplikacijo, ki je povezana s svojo podatkovno bazo. Ker je aplikacija tudi na živem serverju, lahko do nje dostopam kjerkoli in kadarkoli, bodisi iz računalnika bodisi iz telefona, in zaradi odzivnosti aplikacije bodo podatki vedno razumljivo prikazani. Zato lahko to hipotezo brez dvoma potrdim.

3. Hitrost dostopa do podatkov in njihova preglednost bosta večji.

Dostop do podatkov je lahko precej hiter, saj moramo samo obiskati spletno stran in se vanjo prijaviti. Povprečen čas za to bi bil 1 minuta. Če bi to primerjal z odpiranjem datoteke na računalniku, je vsekakor moja rešitev počasnejša. Do večjih razlik pride, če je ta datoteka na računalniku, ki ga nimamo pri roki. Res je, da lahko naložimo to datoteko na katero od oblčnih storitev, kot je Google Drive, ampak potem je hitrost dostopa do podatkov enako dolga kot pri moji aplikaciji, lahko tudi daljša, če je dokument večji. Preglednost podatkov pa je boljša, kot v Excel preglednicah in podobnih programih, saj niso vsi podatki zbrani na enem mestu in so lepše razporejeni čez celo stran. Prav tako ima aplikacija nekaj več funkcij, kot sama Excel tabela. To hipotezo lahko potrdim.

7. ZAKLJUČEK

Z izdelavo spletne aplikacije sem naletel na številne probleme, zaradi katerih sem lahko še poglobil svoje znanje programiranja v PHP in uporabo HTML- in CSS-jezika. Svoje znanje sem nadgrajeval s pomočjo svetovnega spleta in mentorja. V tej raziskovalni nalogi je bilo logično razmišljanje ključnega pomena, še posebej zato, da je aplikacija vse zahtevane podatke prikazala tako, kot sem jih jaz želel.

Ker ima ta spletna aplikacija poudarek na shranjevanju podatkov, je bila torej podatkovna baza velikega pomena. S sprotnim razvijanjem sem moral podatkovno bazo dostikrat spremeniti, da je ustrezala mojim željam in potrebam aplikacije. S tem sem tudi poglobil svoje znanje v podatkovnih bazah in njihovi uporabi.

Prav tako sem se naučil precej novega glede evidentiranja opreme in osnovnih sredstev v gasilskem domu. Videl sem, kaj vse je potrebno narediti, da evidenca teče tako, kot je treba.

8. POVZETEK

Številna gasilska društva v Sloveniji se soočajo s problemom popisovanja gasilske osebne zaščitne opreme. Ta naloga je še posebej pomembna, da lahko zagotovijo pravilno delovanje in funkcionalnost opreme, ter da ne odpove takrat, ko jo najbolj potrebujemo.

V ta namen sem razvil spletno aplikacijo, ki gasilcem poenostavi ta proces. Društva lahko vanjo vnesejo vso svojo osebno zaščitno opremo in imajo pregled nad njeno zgodovino (npr. popravila in čiščenja) ter kdo jo ima trenutno v lasti. Za vsak posamezen kos opreme lahko uporabnik, ki jo lasti, tudi napiše poročilo, če je z njo kaj narobe. Uporabniki se delijo na dva ranga: administratorji in navadni uporabniki. Navadni uporabnik ima omejeno število funkcij, kot so pregled opreme v lasti, pisanje poročil zanje in urejanje svojega profila. Administratorji društva pa imajo poleg funkcije navadnega uporabnika še funkcije za upravljanje z vso društveno opremo, uporabniki v društvu in pregled nad vsemi napisanimi poročili. Aplikacija vsebuje tudi administracijsko ploščo za administratorje spletne aplikacije, kjer pa lahko dodajajo nova društva.

Izgled spletne aplikacije je izdelan s pomočjo HTML-ja in CSS-ja, za pravilno delovanje aplikacije pa skrbi programski jezik PHP. Vsi podatki pa so shranjeni v MySQL podatkovni bazi. Za razvijanje na lokalnem računalniku pa sem uporabljal orodje XAMPP.

Trenutno že obstajajo številne aplikacije za evidentiranje osnovnih sredstev oziroma inventuro, ampak so prilagojena za večja podjetja in čim večjo hitrost, nimajo pa številnih funkcij, kot jih ima moja spletna aplikacija (npr. pisanje poročila, lastnik opreme – uporabniki, prikaz in filtriranje uporabnikov ipd.).

9. ZAHVALA

Rad bi se zahvali za vso pomoč pri razvijanju in nastajanju raziskovalne naloge: mentorju Miranu Zevniku, univ. dipl. inž. elektrotehnike, za spodbudo in pomoč pri programiranju, Lidiji Šuster, prof., za lektoriranje besedila, mag. Vlasti Leban za lektoriranje angleškega povzetka, recenzentom raziskovalne naloge, staršem in vsem ostalim, ki so pripomogli k razvoju te raziskovalne naloge in mi stali ob strani.

10. VIRI IN LITERATURA

1. https://leoss.si/prodajni_program/celovite_resitve/inventura_osnovnih_sredstev/150/inventura_osnovnih_sredstev/
2. <https://www.spica.si/popis-osnovnih-sredstev>
3. <https://www.logistar.si/resitve/popis-osnovnih-sredstev/>
4. <https://www.php.net/manual/en/book.pdo.php>
5. https://www.w3schools.com/css/css_rwd_intro.asp
6. <https://apl.gasilec.net/Vulkan>
7. <https://stackoverflow.com/questions/6101956/generating-a-random-password-in-php>
8. <https://www.php.net/manual/en/function.mail.php>
9. <https://www.quest.com/products/toad-data-modeler/>
10. <https://www.php.net/manual/en/preface.php>
11. <https://atom.io/>
12. <https://dev.mysql.com/doc/refman/8.0/en/>
13. <https://github.com/features>

10.1 VIRI SLIK

Vir[1] <https://www.spica.si/popis-osnovnih-sredstev> (28. 3. 2021)

Vir[2]

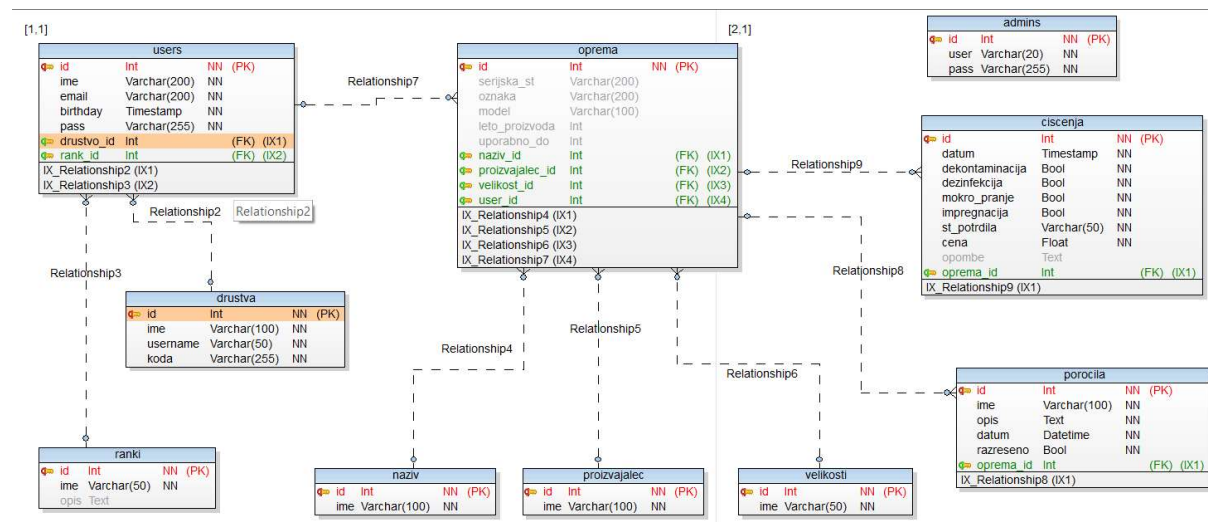
https://leoss.si/prodajni_program/celovite_resitve/inventura_osnovnih_sredstev/150/inventura_osnovnih_sredstev/ (28. 3. 2021)

Vir[3] <https://www.quest.com/products/toad-data-modeler/> (29. 3. 2021)

Vir[4] <https://github.com/> (29. 3. 2021)

11. PRILOGE

11.1 PRILOGA A



Slika 33: ER diagram podatkovne baze, lasten vir