KOMUNALNI ASSISTENT

Tomašić, Trpimir

VELEUČILIŠTE U VIROVITICI

Preddiplomski stručni studij Računarstva

TRPIMIR TOMAŠIĆ

KOMUNALNI ASISTENT

ZAVRŠNI RAD

VIROVITICA, 2021.

VELEUČILIŠTE U VIROVITICI

Preddiplomski stručni studij Računarstva

KOMUNALNI ASISTENT

ZAVRŠNI RAD

Predmet: Programiranje informacijskih sustava

Mentor:

dr.sc. Oliver Jukić, prof. v. š.

Student:

Trpimir Tomašić

VIROVITICA, 2021.

Veleučilište u Virovitici

Preddiplomski stručni studij Računarstva - Smjer Programsko inženjerstvo

OBRAZAC 1b

ZADATAK ZAVRŠNOG RADA

Student/ica: **TOMAŠIĆ TRPIMIR** **JMBAG:** **1312103516**

Imenovani mentor: dr. sc. Oliver Jukić, prof. v. š.

Imenovani komentor: -

Naslov rada:

Komunalni asistent – *citizen wizard*

Puni tekst zadatka završnog rada:

Sustav *Komunalni asistent* ima zadaću pomoći građanima u evidentiranju komunalnih problema na području grada. Jedan dio sustava, a ujedno i zadatak Vašeg završnog rada, je web aplikacija kojom građanin može prijaviti komunalni problem bilo koje vrste, pri čemu se prijavom u sustav prosljeđuje kratki opis problema, lokacija problema (adresa, latituda i longituda) te opcionalno dodatni opis koji može biti tekst i/ili više fotografija. Nakon što je građanin prijavio problem, unutar sustava koristimo terminologiju da je građanin otvorio *ticket* (zapis).

Nakon otvaranja *ticketa* građanin će dobiti identifikacijski broj *ticketa* (*ticket* ID).

U bilo kojem trenutku, u posebnom dijelu aplikacije, unosom *ticket ID*-ja građanin se može informirati o statusu *ticketa* te pregledati detalje o njemu, a što uključuje i pregled fotografija..

Sustav naslanja se na centralnu bazu podataka u kojoj su evidentirani *ticketi.*

U sklopu završnog rada morate kreirati bazu podataka na platformi MSSQL server, koja će biti osnova za izradu aplikacije. Potrebno je kreirati osnovne tablice, za pojedina polja koja imaju predefinirane vrijednosti potrebno je kreirati i šifrarnike. U osnovnim tablicama potrebno je definirati primarne ključeve i indekse. Bazu je potrebno napuniti testnim podacima. Nakon kreiranja baze podataka potrebno je nekim od formalnih jezika unutar jezičnog prostora napraviti specifikaciju aplikacije koja će biti zatim implementirana kroz završni rad. Za pristup aplikacije podacima u bazi potrebno je realizirati sučelje koje može biti korišteno i za druge svrhe, u bilo kojoj od tehnologija.

Datum uručenja zadatka studentu/ici: 28.07.2021.
Rok za predaju gotovog rada: 06.09.2021.

Mentor:

dr. sc. Oliver Jukić, prof. v. š.

Dostaviti:
1. Studentu/ici
2. Povjerenstvu za završni rad - tajniku

KOMUNALNI ASSISTENT

CITIZEN WIZZARD

SAŽETAK – *Cilj ovoga rada bio je napraviti JavaScript/PHP aplikaciju koja korisniku omogućava prijavu komunalnih problema na području grada. Za izradu rada korišten je Visual Studio Code kao uređivač koda, a JavaScript i PHP programski jezici kao tehnologije za izradu aplikacije. Komunalni assistent je aplikacija tipa web stranica koja pruža grafičko korisničko sučelje pomoću HTML opisnog jezika. Komunalni assistent omogućuje korisniku dodavanje i provjeru statusa ticketa. Dodavanjem ticketa otvara se Bootstrap modal koji sadrži polja za popunjavanje: status, slike (opcionalno), datum, adresa i google mapu pomoću koje s jednostrukim klikom na nju dodaje latitudu i longitudu adrese. Nakon popunjenih svih polja korisnik treba jednostrukim kliknuti na gumb "Dodaj ticket" te se zatim taj ticket sprema u bazu podataka i aplikacija korisniku vraća identifikacijski broj njegovog ticketa u obliku modala. S tim identifikacijskim brojem korisnik ima mogućnost provjeravanja statusa svoga ticketa. Status ticketa se provjerava tako da korisnik upiše identifikacijski broj svog ticketa u search bar i pritiskom tipke enter na tipkovnici otvara se modal sa svim informacijama o traženom ticketu. Za bazu podataka korišten je MS SQL server, a za rad nad bazom podataka korišten je Microsoft sql server management studio 18.*

Ključne riječi: *JavaScript, PHP, Bootstrap modal, HTML, aplikacija*

SUMMARY *- The aim of this work was to create a JavaScript / PHP application that allows the user to report utility problems in the city area. Visual Studio Code was used to create the work as a code editor, and JavaScript and PHP programming languages were used as technologies for creating the application. A utility assistant is a web-type application that provides a graphical user interface using an HTML descriptive language. The utility assistant allows the user to add and check the status of the ticket. Adding a ticket opens the Bootstrap modal which contains fields to fill in: status, images (optional), date, address and a google map with which you can add the latitude and longitude of the address with a single click on it. After filling in all the fields, the user should single-click on the "Add ticket" button and then the ticket is saved in the database and the application returns to the user the identification number of his ticket in the form of a modal. With this identification number, the user has the opportunity to*

check the status of his ticket. The status of the ticket is checked by the user entering the identification number of their ticket in the search bar and pressing the enter key on the keyboard opens a module with all the information about the requested ticket. MS SQL server was used for the database and Microsoft sql server management studio 18 was used to work on the database.

Keywords: *JavaScript, PHP, Bootstrap modal, HTML, application*

SADRŽAJ

1.UVOD 1
2. JAVASCRIPT 2
2.1 .Varijable i konstante 2
2.2 .Tipovi podataka 3
2.3. Operatori 4
2.4 Kontrola toka programa 4
2.5. Objekti 5
2.6. Funkcije 5
2.7. jQuery 6
3. PHP 7
3.1. Varijable 7
3.2. Tipovi podataka 8
3.3. Operatori 8
3.4. Kontrola toka programa 9
3.5. Objekti 9
3.6. Funkcije 10
4. DODATNE KORIŠTENE TEHNOLOGIJE I ALATI 11
4.1. HTML 11
4.1.1. Button 11
4.1.2. Form 12
4.2. CSS 12
4.3. jQuery Ajax 13
4.4. Microsoft SQL Server Management Studio 14
4.5. Bootstrap 15
4.5.1. DateTimePicker 15

4.5.2. Modal 15
4.6. Google maps 16
5. ARHITEKTURA APLIKACIJE 17
5.1. Dodavanje ticketa 19
5.2. Provjera statusa ticketa 23
6. ZAKLJUČAK 26

1.UVOD

Svrha aplikacije je kako bi građani mogli prijaviti komunalni problem bilo koje vrste na vrlo brz i efikasan način bez kreiranja korisničkog računa ili prijavljivanja na stranicu. Kod prijavljivanja komunalnog problema postoji mogućnost unosa opisa, adrese što uključuje latitudu i longitudu, datuma i dodavanje slika. Kada se prijavi komunalni problem, korisnik zauzvrat dobiva identifikacijski broj s kojim poslije može provjeriti status problema da li je on otvoren, riješen ili odbačen. U prva četiri poglavlja rada opisane su korištene tehnologije koje su korištene u izradi završnog rada i dani su primjeri. Od tehnologija opisani su : JavaScript, PHP, HTML, CSS, jQuery, jQuery AJAX, Microsoft SQL Server. U drugom dijelu rada detaljno se opisuje rad aplikacije.

2. JAVASCRIPT

JavaScript je skriptni jezik koji nam omogućuje kreiranje dinamičkih web stranica. "Dinamičke web stranice poput JavaScripta omogućuju međudjelovanje s korisnikom, upravljanje web preglednikom. Izvorno je bio zamišljen kao skriptno sučelje između web stranice i aplikacije na poslužitelju. JavaScript se izvodi na klijentskom računalu i može se izvoditi na svim modernim internet preglednicima gdje se koristi za upravljanjem sadržaja web stranice."[1] Kreirao ga je američki programer Brendan Eich 1995 godine. JavaScript datoteke možemo prepoznati po ekstenziji .js.

"Prednosti korištenja JavaScripta:

- manja potreba komunikacije s poslužiteljem, može se provjeriti ispravnost podataka prije slanja stranice na poslužitelja
- smanjuje se promet prema poslužitelju
- korisnik ne treba čekati da se stranica ponovno učita kako bi vidio da je zaboravio unijeti neki podatak"[2]

2.1 .Varijable i konstante

Varijable su kontejneri za spremanje podataka koje mogu promijeniti vrijednost u tijeku izvršavanja programa. Varijabla se kreira ključnom riječ var i jedinstvenim imenom. Jedinstveno ime se zove identifikator. "Identifikatori mogu biti kratki nazivi (poput x i y) ili više opisnih imena (dob, zbroj, ukupni volumen)."[3] "Konstanta je vrsta varijable čiju vrijednost ne možemo mijenjati. Konstanta se kreira s ključnom riječi const i pri njenoj deklaraciji mora se dodijeliti vrijednost."[4]

---

1 dr.sc. Oliver Jukić, prof.v.š: Skriptni programski jezici(Laboratorijske vježbe 1), Veleučilište u Virovitici, 2017./2018.

2 dr.sc. Oliver Jukić, prof.v.š: Skriptni programski jezici(SPJ_P3_Javascript-Uvod-TipoviPodataka-Operatori), Veleučilište u Virovitici, 2017./2018.

3 https://www.w3schools.com/js/js_variables.asp (9.08.2021.)

4 https://www.w3schools.com/js/js_const.asp (9.08.2021.)

Slika 1. Primjer JavaScript varijabli

```
var prazna;
var ime = "Trpimir";
var x = 5;
const PI = 3.141592653589793;
```

Izvor: autor

2.2 .Tipovi podataka

"Tip podataka (eng. *Data type*) varijable je tip podataka koja varijabla trenutno sadrži i kako ga interpretira skriptni mehanizam JavaScripta."[5]

Varijable niza sadrže niz znakova, numerička varijabla sadrži brojčanu vrijednost itd. "Za razliku od mnogih drugih jezika, u JavaScriptu ista varijabla može sadržavati različite tipove podataka, sve unutar iste aplikacije. Ova koncepcija je poznata po izrazima slobodno zadavanje tipa (eng. *Loose typing*) i dinamičko zadavanje tipa (eng. *Dynamic typing*). JavaScript je "dinamic typing" programski jezik."[6]

Slika 2. Primjer JavaScript tipova podataka

```
var Ime = "Trpimir";
var Broj = 9;
var Osoba = {ime:"Trpimir", prezime:"Tomašić", adob:26};
```

Izvor: autor

Varijabla ime je tip podatka tipa string, broj je tipa number i osoba je tipa objekt.

5dr.sc. Oliver Jukić, prof.v.š: Skriptni programski jezici(SPJ_P3_Javascript-Uvod-TipoviPodataka-Operatori), Veleučilište u Virovitici, 2017./2018.

6 dr.sc. Oliver Jukić, prof.v.š: Skriptni programski jezici(SPJ_P3_Javascript-Uvod-TipoviPodataka-Operatori), Veleučilište u Virovitici, 2017./2018.

2.3. Operatori

JavaScript uključuje operatore isto kao i drugi programski jezici. Operatori služe za izvršavanje matematičkih i logičkih operacija. Operater izvodi neke operacije na jednom ili više operanda (vrijednost podataka) i proizvodi rezultat. Na primjer: 1 + 2, znak + jer operator, 1 je operand s lijeve strane, a 2 s operand s desne. Operator + izvodi zbrajanje dvije numeričke vrijednosti i vraća rezultat.

Slika 3. Primjer korištenja operatora

```
var x = 5;
var y = 2;
var z = x + y;
```

Izvor: autor

Na slici 2 prikazan je primjer deklaracije tri varijable x, y i z. Varijabli x pridodajemo vrijednost pet, varijabli y pridodajemo vrijednost dva i varijabli z pridodajemo vrijednost rezultata zbrajanja varijabla x i y.

2.4 Kontrola toka programa

Kontrola toka se koristi za izvođenje različitih radnji na temelju različitih uvjeta. JavaScript koristi različite naredbe za kontrolu toka, a uz for petlju, while i do-while petlje, tu su if, else, else if i switch. If upotrebljavamo ukoliko želimo izvršiti nešto na temelju nekog uvjeta. Else upotrebljavamo kada želimo izvršiti kod svaki puta kada if uvjet ima vrijednost false. Naredba else mora slijediti naredbu if ili else if. Else if koristimo kada želimo postaviti dodatan uvjet nakon if naredbe. Switch je uvjetni izraz poput izraza if. Koristan je kada želimo izvršiti jedan od više blokova koda na temelju povratne vrijednosti navedenog izraza.

Slika 4. Primjer if naredbe za kontrolu toka

```
if (sati < 18) {
  pozdravi = "Dobar dan";
  }
```

Izvor: autor

2.5. Objekti

"Objekt je osnovni gradivni element JavaScript-a koji se sastoji od niza svojstava koje možemo sami definirati. Svako svojstvo se sastoji od naziva i vrijednosti. Vrijednost svojstva može biti ugrađeni tip podatka (string, number, array, bool...) funkcija ili objekt. JavaScript objekti su dinamični - svojstva se mogu dodavati, uređivati i brisati. Svaka vrijednost u JavaScript-u, true, false, null ili undefined je objekt."[7]

Slika 5. Primjer objekta

```
var osoba = {ime : "Trpimir",
  prezime  : "Tomašić",
  dob      : 26,
  bojaOčiju  : "zelena"
```

Izvor: autor

Slika 5 prikazuje jedan JavaScript objekt. Osoba je ime objekta, a ime, prezime, dob i boja očiju su svojstva dok Trpimir , Tomašić , 26 i zelena su vrijednosti tih svojstava.

2.6. Funkcije

JavaScript pruža funkcije slične većini skriptnih i programskih jezika. U JavaScriptu funkcija omogućava da definirate blok koda, date mu ime, a zatim ga izvršite koliko god puta želite. JavaScript funkcija se definira pomoću ključne riječi function. Funkcija u sebi može posjedovati jedan ili više parametra zavisno o potrebi. Parametri u funkcijama mogu se

7 dr.sc. Oliver Jukić, prof.v.š: Skriptni programski jezici(Laboratorijske vježbe 2), Veleučilište u Virovitici, 2017./2018.

proslijediti i mogu se koristiti unutar funkcije. JavaScript je jezik dinamičkog tipa pa parametar funkcije može imati vrijednost bilo koje vrste podataka.

Slika 6. Kreiranje i pozivanje funkcije

```
function PrikaziPoruku()
{
    alert("Pozdrav!");
}

PrikaziPoruku();
```

Izvor: autor

Sa slike broj šest zadnja linija koda je poziv funkcije, a u tijelu funkcije: alert(“Pozdrav!”); je što će funkcija izvršiti kada ju se pozove.

2.7. jQuery

JQuery je dinamična, mala i bogata JavaScript biblioteka uključena u jednu js. datoteku. JQuery olakšava život web developerima jer pruža mnoge ugrađene funkcije pomoću kojih se može jednostavno i brzo izvršavati razne zadatke. Neke od prednosti korištenja jQuerya:

- lako ga je naučiti jer podržava isto kodiranje u JavaScript stilu
- temelji se na činjenici : “write less do more”. Nudi bogat skup značajki koje povećavaju produktivnost programera pisanjem manjeg i čitljivog koda
- pruža izvrsnu online API dokumentaciju
- oslanja se na osnove znanja iz : HTML, CSS , JavaScript

Slika 7. jQuery metoda append

```
$("p").append(" <b>Dodaj tekst</b>.");
```

Izvor: autor

JQuery metoda append umeće sadržaj “Dodaj tekst” na kraj HTML “p” elementa.

3. PHP

PHP (eng. *PHP: Hypertext Preprocessor*) je programski jezik koji web programerima omogućuje stvaranje dinamičkog sadržaja koji je u interakciji s bazama podataka. Izvršava se na strani poslužitelja. "Njegova uloga je upravljanje dinamičkim sadržajem, praćenje sesija, manipuliranje stvarima koje su bitne web developerima.

Jedna od njegovih bitnih značajki je da upravlja i bazama podataka, tako da je integriran s brojnim popularnim bazama kao što su MySQL, Oracle, MS SQL server i druge."[8]

Poslužitelj obrađuje naredbe PHP jezika i šalje rezultat pregledniku. Osmislio ga je Rasmus Lerdof 1994. Godine. Neke od prednosti korištenja PHP-a:

- PHP radi na različitim platformama: Windows, Linux, Unix, MAC OS X itd...
- kompatibilan je gotovo sa svim poslužiteljima koji se danas koriste (Apache, IIS itd.)
- podržava širok raspon baze podataka
- besplatan je
- jednostavan za učenje i učinkovito radi na poslužiteljskoj strani

3.1. Varijable

U PHP-u varijable se deklariraju znakom dollar ($), a zatim proizvoljnim imenom varijable. Varijable u PHP-u mogu sadržavati osam tipova podataka. "Vrijednost varijable se može zamijeniti drugom vrijednošću drugog tipa. Tip varijable se ne provjerava ni u vrijeme provođenja ni u vrijeme izvođenja.".[9]

Slika 8. Primjer PHP varijabli

```
$txt = "Trpimir!";
$x = 9;
```

Izvor: autor

8 https://repozitorij.vuv.hr/islandora/object/vsmti%3A549(11.08.2021.)

9 dr.sc. Oliver Jukić, prof.v.š: Web programiranje na strani poslužitelja(WPSP_P7_PHP-Uvod-Varijable-Nizovi-Polja), Veleučilište u Virovitici, 2017./2018

3.2. Tipovi podataka

Varijable mogu pohranjivati podatke različitih vrsta, a različite vrste podataka mogu činiti različite stvari. PHP podržava sljedeće tipove podataka:

- cjelobrojne vrijednosti
- brojevi s pokretnim zarezom
- nizovi znakova
- logičke vrijednosti
- polja
- objekti
- ersurs
- NULL

Slika 9. Primjer PHP tipova podataka

```
$txt = "Trpimir!";
$x = 9;
$y = 10.5;
$z = 10.365;
$t = true;
$cars = array("Volvo","BMW","Toyota");
```

Izvor: autor

3.3. Operatori

Operatori se koriste za izvođenje operacija nad varijablama i vrijednostima. PHP dijeli operatore na sljedeće grupe:

- aritmetički operatori
- operatori dodjeljivanja
- operatori usporedbe
- operatori povećanja/smanjenja
- logički operatori
- operatori uvjetnog dodjeljivanja
- operatori polja

– operatori znakova

Slika 10. Primjer PHP operatora

```
$x = 10;
$y = 6;
echo $x + $y;
```

Izvor: autor

3.4. Kontrola toka programa

Kontrola toka se koristi za izvođenje različitih radnji na temelju različitih uvjeta. PHP koristi četiri različite naredbe za kontrolu toka, a to su if, else, else if i switch. If upotrebljavamo ukoliko želimo izvršiti nešto na temelju nekog uvjeta. Else upotrebljavamo kada želimo izvršiti kod svaki puta kada if uvjet ima vrijednost false. Naredba else mora slijediti naredbu if ili else if. Else if koristimo kada želimo postaviti dodatan uvjet nakon if naredbe. Switch je uvjetni izraz poput izraza if. Koristan je kada želimo izvršiti jedan od više blokova koda na temelju povratne vrijednosti navedenog izraza.

Slika 11. Primjer kontrole toka if naredbom

```
$vrijeme = date("H");

if ($t < "20") {
  echo "Ugodan dan vam želim!";
} else {
  echo "Laku noć!";
}
```

Izvor: autor

Na slici 10 je prikazana varijabla vrijeme koja ima vrijednost trenutnog vremena. U if naredbu smo kao uvjet postavili ako je trenutno vrijeme manje od 20h tada će ispisati “Ugodan dan vam želim” inače će se ispisati “Laku noć”.

3.5. Objekti

U PHP-u objekt je složeni tip podataka (zajedno s poljima). Vrijednosti više vrsta mogu se zajedno pohraniti u jednu varijablu. Objekt je instanca ugrađene ili korisnički definirane

instance. PHP tretira objekte na isti način kao reference što znači da svaka varijabla sadrži referencu objekta, a ne kopiju cijelog objekta. Na slici 12 programer je klasa, a Trpimir je novi objekt klase programer.

Slika 12. Primjer kreiranja klase i objekta

```
class Programer
{

}

$Trpimir = new Programer;
```

Izvor: autor

3.6. Funkcije

PHP jezik je poznat po vrlo moćnim i raznolikim tipova funkcija. PHP funkcije slične su drugim programskim jezicima. Funkcija je dio koda koji uzima još jedan ulaz u obliku parametra i vrši neku obradu i vraća vrijednost. Funkcija se deklarira s ključnom riječi function. Sav PHP kod trebao bih se staviti unutar {} zagrada. PHP daje mogućnost prosljeđivanja parametara unutar funkcije. Ti parametri rade kao varijable unutar funkcije.

Na slici 13 prikazano ja kreiranje i pozivanje funkcije Poruka. Kada se funkcija pozove, na ekranu se ispisuje “Pozdrav”.

Slika 13. Primjer kreiranja i pozivanja funkcije

```
function Poruka()
{
  echo "Pozdrav!";
}

writeMsg();
```

Izvor: autor

4. DODATNE KORIŠTENE TEHNOLOGIJE I ALATI

4.1. HTML

HTML (eng. *Hypertext Markup Language*) je opisni jezik koji se upotrebljava za izradu i prikaz web stranica. "HTML dokument stvara se pomoću html jezika kojim se oblikuje sadržaj i stvaraju hiperveze između HTML dokumenta. HTML nije programski jezik i služi isključivo kao skup uputa ili oznaka koje upućuju web preglednike kako prikazati opisani HTML dokument. HTML datoteke su obične tekstualne datoteke, a ekstenzije su im .html ili .htm te ih se može stvoriti u bilo kojem uređivaču tekstualnih datoteka. HTML elementi mogu sadržavati atribute koji definiraju neka dodatna svojstva. Smješteni su unutar početne oznake elementa i sastoje se od dva dijela: naziva atributa i vrijednosti atributa."[10] Prvu verziju HTML-a napisao je Tim Berners-Lee 1993. godine.

Slika 14. Primjer HTML elementa

```
<html lang="hr">
```

Izvor: autor

Html je naziv elementa, lang je naziv atributa, a hr je vrijednost lang atributa.

4.1.1. Button

HTML element button predstavlja gumb koji se može kliknuti i obično se koristi za slanje obrazaca ili da izvrši određeni dio koda bilo gdje u dokumentu. Gumbi su važan element web dizajna. Oni mogu pomoći privući pažnju posjetitelja, pružiti im važne informacije ili ih nagovoriti da poduzmu mjere na web lokaciji. Shvaćajući važnost ovog elementa dizajna, front-end razvojni alat Bootstrap pruža gotove predloške za gumbe. Oznaka <button> definira HTML element gumb na koji se može kliknuti. Klasa gumba Bootstrap je .btn.

10 dr.sc. Oliver Jukić, prof.v.š: Osnove web programiranja(Laboratorijske vježbe 1), Veleučilište u Virovitici, 2017./2018

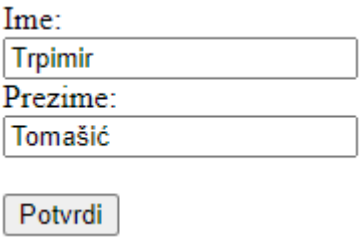
Slika 15. Primjer dva gumba

Izvor: autor

4.1.2. Form

HTML obrazac (eng. *Form*) koristi se za prikupljanje podataka unosa korisnika. Korisnički se unos najčešće šalje na poslužitelj radi obrade. Element <form> spremnik je za različite vrste ulaznih elemenata, kao što su: tekstualna polja, potvrdni okviri, radio gumbi, gumbi za slanje itd.

Slika 16. Primjer forme



Izvor: autor

4.2. CSS

"CSS (eng. *Cascading Style Sheets*) opisni je jezik koji se svojom primjenom oslanja na HTML, a služi za veću slobodu nad detaljima oblikovanja i dizajna web stranica. Tako je moguće mijenjati boju teksta, precizno smještati slike ili tekstove bilo gdje na stranici, sakrivati elemente i sl."[11] Na slici 13 upotrijebili smo CSS kako bi HTML elementu p dodali vrstu, boju i veličinu fonta.

11 dr.sc. Oliver Jukić, prof.v.š: Osnove web programiranja(laboratorijske vježbe 1), Veleučilište u Virovitici, 2017./2018

Slika 17. Primjer uporabe CSS-a

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

Izvor: autor

4.3. jQuery Ajax

"Ajax je skup tehnika web razvoja koji koristi različite web tehnologije na strani klijenta za stvaranje asinkronih web aplikacija. S Ajaxom, web aplikacije mogu slati i dohvatiti podatke s web poslužitelja asinkrono (u pozadini) bez ometanja prikaza i ponašanja postojeće web stranice. Odvajanjem sloja za razmjenu podataka od prezentacijskog sloja Ajax omogućuje web stranicama i web aplikacijama da dinamički mijenjaju sadržaj bez potrebe za ponovnim učitavanjem cijele stranice."[12] U današnjici za razmjenu podataka često se koristi JSON (eng. *JavaScript Object Nation*) kao format za razmjenu podataka. "Razne popularne JavaScript biblioteke, uključujući jQuery, uključuju apstrakcije koje pomažu u izvršavanju Ajax zahtjeva."[13] Na slici 15 mijenjamo text div elementa pomoću Ajax zahtjeva.

12 http://ommolketab.ir/aaf-lib/4cdbqz68f1sdz49yauhpc6z1k1bxau.pdf
13 http://ommolketab.ir/aaf-lib/4cdbqz68f1sdz49yauhpc6z1k1bxau.pdf

Slika 18. Primjer uporabe jQuery Ajax

```
<head>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
      $("#div1").html(result);
    }});
  });
});
</script>
</head>
<body>
<div id="div1"><h2>Dopusti jQuery AJAX da promjeni ovaj text</h2></div>
<button>Dohvati vanjski sadržaj</button>
</body>
```

Izvor: autor

4.4. Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio (SSMS) je integrirano okruženje za upravljanje SQL infrastrukturom. SSMS pruža alate za upravljanje, praćenje i konfiguriranje instancama bazama podataka i SQL servera. SSMS je korišten u ovom radu kao alat za upravljanje nad bazom podataka MS SQL serverom.

Slika 19. Povezivanje na bazu podataka

```
<?php
    $sHost = "193.198.57.183";
    $sUsername = "pin";
    $sPassword = "Vsmti1234!";
    $sDatabase = "STUDENTI_PIN";

    try
    {
        $oDbConnector = new PDO("sqlsrv:Server=$sHost;Database=$sDatabase;ConnectionPooling=0", "$sUsername", "$sPassword");
    }
    catch(PDOException $e)
    {
        echo "Error" . $e;
    }
?>
```

Izvor: autor

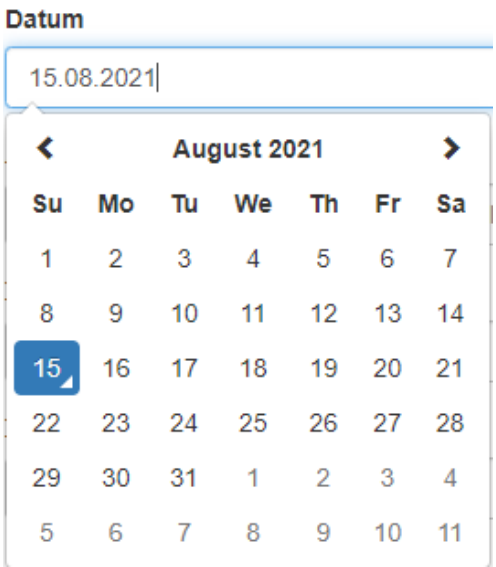
4.5. Bootstrap

Bootstrap je besplatni CSS okvir otvorenog koda napisan u HTML, JavaScriptu i CSS-u. Bootstrap omogućuje programerima i web-dizajnerima da na vrlo brz, lagan i efikasan način izgrade potpuno responzivne web stranice. Poznat je po svojim obrascima, gumbima, datetimepickerima, navigacijama itd.

4.5.1. DateTimePicker

DateTimePicker je Bootstrapov widget (hr. *Naprava*) za grafičko sučelje preko kojeg korisnik ima mogućnost odabira godine, mjeseca i dana. U ovome projektu korišten je DateTimePicker kao alat s kojim će korisnik unijeti datum prijave komunalnog problema.

Slika 20. DateTimePicker



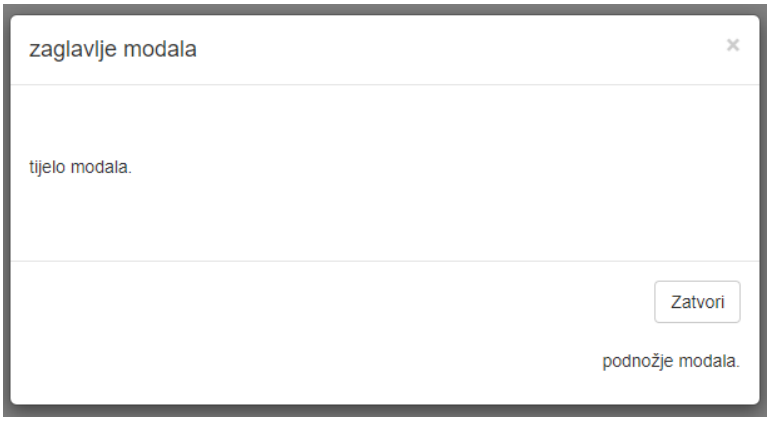
Izvor: autor

4.5.2. Modal

Bootstrapov skočni prozor (eng. *Modal*) jedan od najkorisnijih jQuery Bootstrap dodataka. Modali nude lagani, višenamjenski skočni prozor koji je prilagodljiv i odaziv. Može se koristiti za prikaz skočnih prozora upozorenja, videozapisa i slika na web mjestu. Web stranice temeljene na Bootstrapu mogu koristiti Bootstrap modale za prikazivanje, na primjer, uvjeta i odredbi (kao dio procesa registracije), videozapisa ili čak widgeta za društvene medije.

Bootstrap modali podijeljeni su u tri primarna odjeljka: zaglavlje, tijelo i podnožje. Svaki ima svoju ulogu i stoga ga treba koristiti u skladu s tim.

Slika 21. Primjer Bootstrap modala

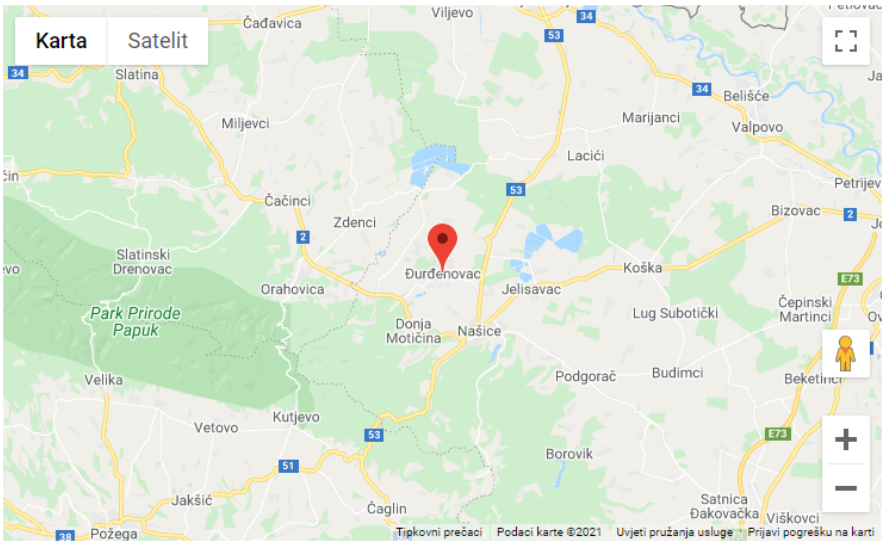


Izvor: autor

4.6. Google maps

Google karte (eng. *Google maps*) su web-usluga koja pruža detaljne informacije o zemljopisnim regijama i web mjestima širom svijeta. Osim konvencionalnih mapa cesta, Google karte nude zračna i satelitska snimanja mnogih mjesta. U nekim gradovima Google karte nude prikaze ulica koje uključuju fotografije snimljene iz vozila. Sučelje aplikacijskog programa (eng. *API*) Google karata omogućuje administratorima web stranica ugradnju Google karata u vlasničko mjesto, poput vodiča za nekretnine ili stranice usluga zajednice. Google mapa je korištena u ovom radu kao grafičko korisničko sučelje za odabir adrese komunalnog problema.

Slika 22. Google maps



Izvor: autor

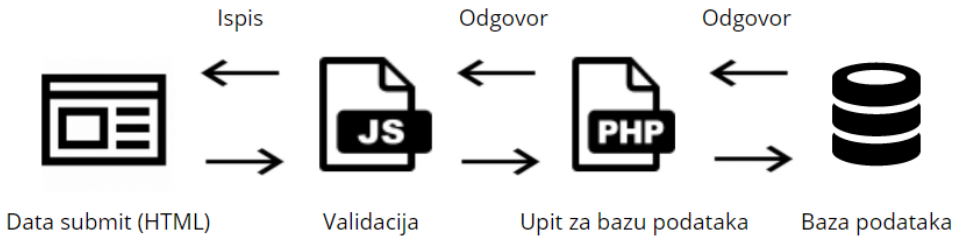
5. ARHITEKTURA APLIKACIJE

U nastavku rada bit će opisano kako web aplikacija radi, detaljna arhitektura te programske funkcije koje su korištene unutar slojeva aplikacije. Web aplikacija je vrsta računalnog programa. Koristi mrežnu tehnologiju (uključujući preglednike) za izvršavanje velikog broja različitih zadataka.

Web aplikacija podijeljena je na dva sloja. Sloj za dohvaćanje, obrađivanje i pohranjivanje podataka odrađujemo pomoću poslužiteljskog programskog jezika PHP, dok klijentski programski jezik JavaScript predstavlja prezentacijski sloj na korisničkom sučelju. Kao baza podataka korišten je MS SQL server, a za rad nad bazom korišten je Microsoft sql server management studio 18. Sve što je potrebno za pristup web aplikaciji je internetska veza. Za povezivanje s aplikacijom koristi se web preglednik kao što su Google Chrome, Mozilla Firefox ili Opera.

Web aplikacija služi za prijavljivanje komunalnih problema, kao što su palo drvo na cestu, srušen prometni znak, pokidana klupa za sjediti, prevrnut kontejner, na području gradova Republike Hrvatske.

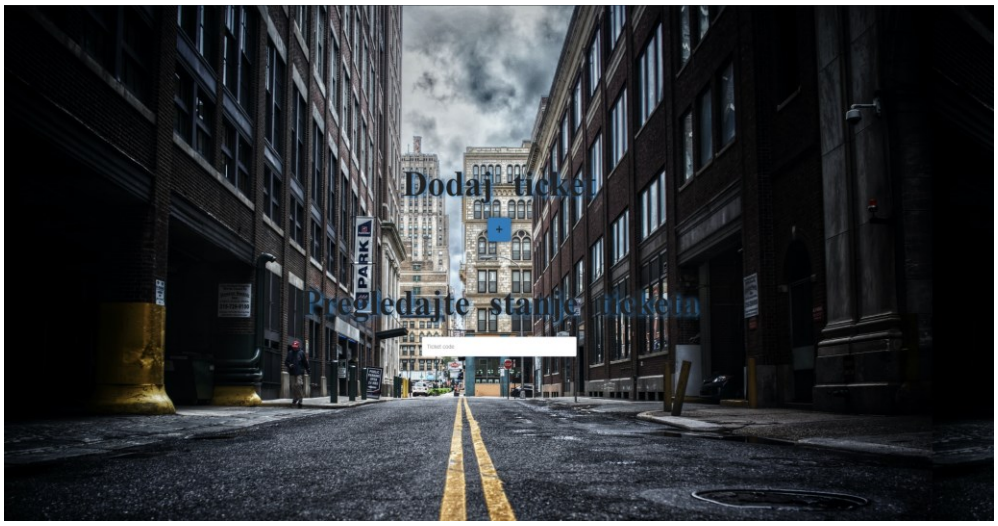
Slika 23. Prikaz arhitekture aplikacije



Izvor: autor

Nakon što se pokrene web preglednik, u adresnu traku (također traka lokacije ili URL traka) upisuje se http://student.vsmti.hr/ttom/ što je web lokacija aplikacije Komunalni asistent. Nakon unesene navedene adrese i pritiskom tipke enter otvara se početna stranica web aplikacije.

Slika 24. Početna stranica aplikacije



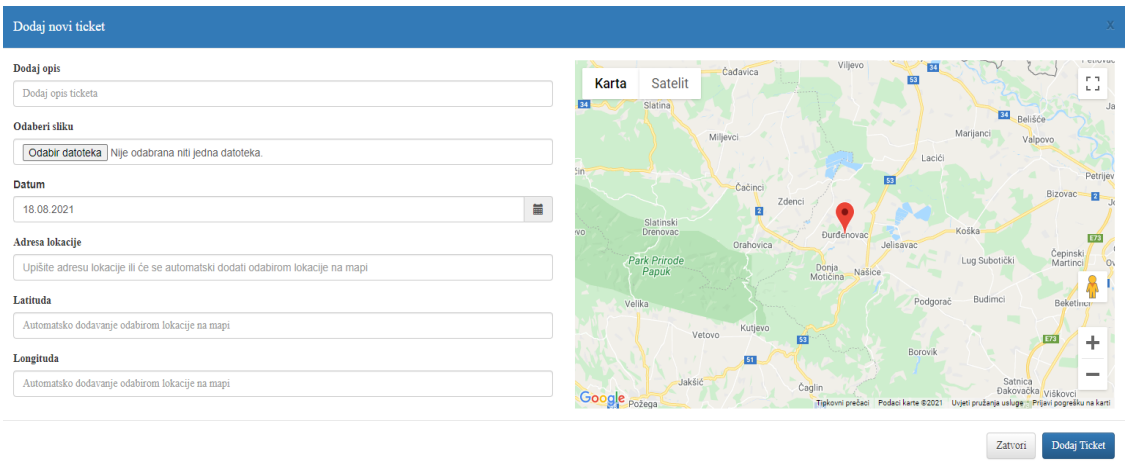
Izvor: autor

Početna i jedina stranica web aplikacije sadrži dva html <h1> elementa. Oznake <h1> do <h6> koriste se za definiranje HTML naslova. <h1> element definira najvažniji i najveći naslov, a <h6> element definira najmanje važan naslov i vizualno najmanji. Ispod naslova

„Dodaj ticket“ nalazi se Bootstrapov gumb (eng. *Button*) sa simbolom „+“. Jednostrukim klikom miša na gumb „+“ otvara se Bootstrapov modal (hr. *Skočni prozor*) .

5.1. Dodavanje ticketa

Slika 25. Modal „Dodaj ticket“



Izvor: autor

Modal je podijeljen na tri primarna odjeljka: zaglavlje, tijelo i podnožje. U lijevom kutu zaglavlja postavljen je <h4> element s natpisom „Dodaj novi ticket“ što označava koja je svrha modala, a u samom desnom kutu je gumb „x“ koji jednostrukim klikom na njega zatvara modal. Tijelo modala podijeljeno je na dva stupca pomoću HTML Bootstrap stila „style="display: flex;flex-direction: row“ i dva div elementa „<div class="container" style="width: 50%;"“. Oznaka definira podjelu ili odjeljak u HTML dokumentu. U prvom stupcu nalazi se šest obrazaca (eng. *Form*) za popunjavanje podataka o komunalnom problemu sa svojim label (hr. *Oznaka*) i input (hr. *Ulaz*) elementima. Element <label> koristi se za povezivanje tekstualne oznake s poljem <input>. Oznaka se koristi kako bi korisnicima rekla vrijednost koju treba unijeti u pridruženo polje za unos. Svako input ima svoj atribut placeholder. Placeholder (hr. *Rezervirano mjesto*) navodi kratki savjet koji opisuje očekivanu vrijednost polja za unos. Atribut Placeholder funkcionira sa sljedećim vrstama unosa: tekst, pretraživanje, url, tel, e - pošta i lozinka. Input „Opis“ ima atribut tipa text što definira jednoredno tekstualno polje. U input „Opis“ upisuje se kratki opis komunalnog problema na primjer: palo drvo na cestu, srušen prometni znak itd. „Odaberi sliku“ je input element tipa file (hr. *Datoteka*) kojem je zadaća uploadanje slika komunalnog problema.

Korisnik je u mogućnosti odabrati više slika, a ta mogućnost ostvarena je pomoću atributa „multiple“. Dodavanje slika je opcionalno tako da u slučaju da korisnik ne unese sliku neće biti nikakih problema sa aplikacijom. Datum korisnik unosi pomoću Bootstrap widgeta (hr. *Naprava*) DateTimePicker-a. DateTimePicker je Bootstrapov widget za grafičko sučelje preko kojeg korisnik ima mogućnost odabira godine, mjeseca i dana. Adresa zajedno sa latitudom i longitudom lokacije ispunjava se preko metoda jednostrukog klika mišem ili povlačenja markera na Google mapi koja se nalazi na desnoj strani tijela modala. Obje metode su u programskom kodu zasebne funkcije i obje su postignute pomoću JavaScript metoda addEventListener. Metoda addEventListener () pridružuje događaj navedenom elementu. Događaj je važan dio JavaScripta. Web stranica odgovara u skladu s događajem. Neke događaje generiraju korisnici, a neke API -je. Slušač događaja je postupak u JavaScriptu koji čeka da se događaj dogodi. Jednostavan primjer događaja je korisnik koji klikne mišem ili pritisne tipku na tipkovnici. AddEventListener () je ugrađena funkcija u JavaScriptu koja zahtijeva događaj za osluškivanje i drugi argument koji se poziva svaki put kada se opisani događaj pokrene. U nastavku na slici 24. prikazana je funkcija koja omogućava korisniku klikanje po Google mapi pomoću JavaScript metode addEventListener. Funkcija ujedno uzima vrijednosti adrese, latitude i longitude.

Slika 26. Funkcija addListener

```
74    google.maps.event.addListener(map, 'click', function(event) {
75
76      var address;
77      var geocoder = new google.maps.Geocoder();
78
79      geocoder.geocode( { latLng: event.latLng }, function ( result, status ) {
80      if ( 'OK' === status )
81      {
82        console.log(result[0]);
83        address = result[0].formatted_address;
84        console.log(address);
85        AdressaElement.value = address;
86      }
87      else
88      {
89        console.log( 'Geocode was not successful for the following reason: ' + status );
90      }
91    });
92      document.getElementById("DodajLatitudu").value = event.latLng.lat();
93      document.getElementById("DodajLongitudu").value = event.latLng.lng();
94      marker.setPosition(event.latLng);
95    });
96
97  }
```

Izvor: autor

U početku funkcije kreirana je prazna varijabla „address“ i varijabla „geocoder“ nad kojom se poziva funkcija „Geocode“ i rezultat se sprema u tu varijablu. Geocode je proces pretvaranja adresa (poput Ulica kralja Zvonimira 9, 31511, Đurđenovac, Hrvatska) u geografske koordinate (poput latitude 45,54569606362175 i longitude 18,04717631346955). Zatim se „address“ varijabli pridodaju vrijednosti formatirane adrese u obliku (Kralja Zvonimira 9, 31511, Đurđenovac, Hrvatska) i izjednačava se sa AdressaElement što je input polje adrese u modalu. Elementi modala latituda i longituda se popunjavaju tako da pomoću motode getElementById dohvatimo te elemente i postavimo im vrijednosti putem event.latLng() funkcije.

Nakon popunjenih svih polja, u podnožju modala smještena su dva gumba „Zatvori“ i „Dodaj ticket“. Ako korisnik jednostrukim klikom klikne na gumb „Zatvori“, modal će se zatvoriti, a ako korisnik klikne na gumb „Dodaj ticket“, a nije popunio sva polja na ekran će se pomoću JavaScript metode alert izbaciti poruka „Niste popunili sva polja“. Metoda alert u JavaScriptu koristi se za prikaz virtualnog okvira upozorenja. Uglavnom se koristi za davanje poruka upozorenja korisnicima. Prikazuje dijaloški okvir upozorenja koji se sastoji od neke navedene poruke i gumba OK. Kad se pojavi dijaloški okvir, moramo pritisnuti "OK" za nastavak. Ako je korisnik popunio sva polja i kliknuo na gumb „Dodaj ticket“, poziva se funkcija „DodajNoviTicket“ i modal se zatvara.

Slika 27. Funkcija za dodavanje ticketa

```
function DodajNoviTicket() {

 var Opis = $('#DodajOpis').val();
 var Slike = Images;
 var Status = 2;
 var Datum = $('#datetimepickerAdd').data("DateTimePicker").date().format('YYYY-MM-DD');
 var Adresa = $('#DodajAdresa').val();
 var Latituda = $('#DodajLatitudu').val();
 var Longituda = $('#DodajLongitudu').val();

 if(Opis=="" || Datum=="" || Adresa=="" || Latituda=="" || Longituda=="" )
 {
     alert("Niste ispunili sva polja!");
 }
 else
 {
   $( ".inner" ).html("");
   $(".inner").append("ID vašeg ticketa je: ");

     var tiket = {
         opis: Opis,
         slika: Slike,
         status: Status,
         datum_prijave: Datum,
         adresa: Adresa,
         latituda: Latituda,
         longituda: Longituda
     };
     console.log(tiket);

     $.ajax({
           type: "POST",
           url: 'action.php?action=dodajTicket',
           data:

           {
               ti_opis: tiket.opis,
               ti_putanja: tiket.slika,
               status_st_id: tiket.status,
               ti_datum: tiket.datum_prijave,
               ti_adresa: tiket.adresa,
               ti_latituda: tiket.latituda,
               ti_longituda: tiket.longituda
           },
           success: function (oData)
           {
               console.log("OVO ISPISUJE: " + oData);
               console.log("VALJA");
               $('#modalDodajTicket').modal('hide');
               $( ".inner" ).append( "<p>" + oData + "</p>" );
               $('#modalBrojTicketa').modal('show');
               $('#myModal').modal('show');
               $('#DodajOpis').val('');
               $('#DodajSliku').val('');
               $('#DodajAdresa').val('');
               $('#DodajLatitudu').val('');
               $('#DodajLongitudu').val('');
           },
           error: function (XMLHttpRequest, textStatus, exception) {
               console.log("Ajax failure\n");
           },
           async: true
       });
 }
```

Izvor: autor

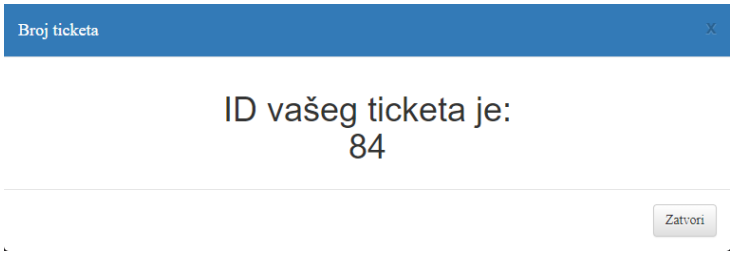
Na samom početku funkcije kreiraju se varijable i dodijeljene su im vrijednosti iz modal inputa. U if naredbi za kontrolu toka pod uvjet je stavljen ako su polja za popunjavanje informacija o komunalnom problemu prazna da se ispiše poruka „Niste popunili sva polja“, a ukoliko uvjet nije zadovoljen kreira se objekt „tiket“ i pridodaju mu se vrijednosti varijabli popunjenih iz modala. Pomoću jQuery Ajax funkcije tipa post (hr. *Poslati*) koja koristi query

(zahtjev za podacima ili informacijama iz tablice baze podataka) „dodajTiket“ kreiraju se objekti sa (key, value) vrijednostima preko kojih će se u action.php skripti dohvaćati vrijednosti. U action.php skripti kreira se query koji pomoću „insert into“ metode dodaje podatke u tablicu baze podataka koja se nalazi na MS SQL serveru.

5.2. Provjera statusa ticketa

Nakon što korisnik uspješno prijavi komunalni problem, aplikacija u obliku skočnog prozora vrati korisniku id (identifikacijski broj) njegovoga prijavljenog problema.

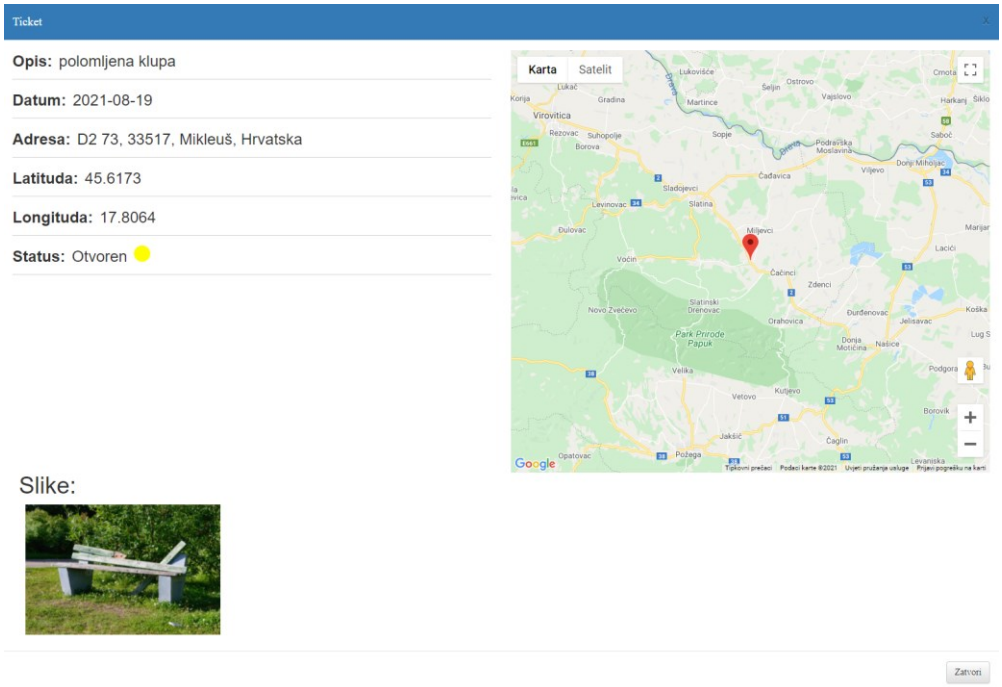
Slika 28. Modal broj ticketa



Izvor: autor

Pomoću identifikacijskog broja korisnik ima mogućnost provjere statusa svog prijavljenog komunalnog problema. Provjera se vrši putem search bara u koji se upisuje id i pritiskom tipke enter na tipkovnici pojavljuje se skočni prozor za informacijama o traženom ticketu.

Slika 29. Modal provjerenog ticketa



Izvor: autor

Na slici 27. prikazan je žuti kružić u redu „Status: otvoren“. Ako je status ticketa riješen kružić će biti zelene boje, a ako je odbačen bit će crvene. Na početku funkcije „PronađiTicket“ deklarira se varijabla Code i pridružujemo joj vrijednost identifikacijskog broja koja je upisana u search bar. U if naredbi za kontrolu toka pod uvjet je stavljen ako je polje za upis identifikacijskog broja o komunalnom problemu prazno da se ispiše poruka „Polje ne smije biti prazno!“, a ukoliko uvjet nije zadovoljen prikazuje se modal na kojem će se u nastavku funkcije pridodati vrijednosti iz baze podataka. Pomoći jQuery ajax funkcije tipa post koja koristi query „pronadiTiket“ kreiraju se objekti sa (key, value) vrijednostima preko kojih će se u action.php skripti dohvaćati vrijednosti. U dodatnu if naredbu postavljen je uvjet ukoliko u bazi podataka nema traženog identifikacijskog broja ispisat će se „Nema rezultata za traženi code!“. U protivnom, podaci koji su dobiveni uspješnom pretragom iz baze podataka, bit će prikazani u modalu.

Slika 30. Funkcija za pronalazak ticketa

```
function PronadiTicket() {
  var Code = $('#input_search_code').val();
  if(Code=="")
  {
      alert("Polje ne smije biti prazno!");
  }
  else
  {
    $('#modalTicketa').modal('show');
    $.ajax({
      type: "POST",
      url: 'action.php?action=pronadiTicket',
      data:{
          ti_code: Code
      },
      success: function (oData)
      {
        var object;
        data = JSON.parse(oData);
        data.forEach(function(ticket)
        {
          object = {
            id: ticket.Ti_id,
            opis: ticket.Ti_opis,
            lat: ticket.Ti_lat,
            lng: ticket.Ti_lng,
            status: ticket.Status_st_id,
            datum: ticket.Ti_datum,
            adresa: ticket.Ti_adresa
          };
          console.log("AEEEEEEEEEEE: " + object.opis);
        });

        if (oData === undefined || oData.length == 2) {
          $(".no_match").show();
          $(".ticket_info").hide();
        }
        else {
          $(".no_match").hide();
          $(".ticket_info").show();

          $( "#opis" ).append(
            object.opis
          );
          $( "#datum" ).append(
            object.datum
          );
          $( "#adresa" ).append(
            object.adresa
          );
          $( "#lat" ).append(
            object.lat
          );
          $( "#lng" ).append(
            object.lng
          );
          $( "#status" ).append(
            object.status
          );

          $( "#slikee" ).html("");

          var myLatlng = {lat:Number(object.lat),lng:Number(object.lng)}
          var map = new google.maps.Map(document.getElementById('map_canvasT'),
          {
            zoom: 10,
            center: myLatlng,
          });

          marker = new google.maps.Marker({
            position: myLatlng,
            map: map
          });
        }
      },
      error: function (XMLHttpRequest, textStatus, exception) {
          console.log("Ajax failure\n");
      },
      async: true
      });
  }
}
```

Izvor: autor

6. ZAKLJUČAK

Kako bi ova web aplikacija, kao i aplikacija bilo kojeg tipa, bila kvalitetno napravljena, bitno je detaljno osmisliti koncept rada i arhitekturu aplikacije. Osim toga, potrebno je opširno upoznavanje s programskim jezicima i alatima u kojima će aplikacija biti kreirana. Komunalni asistent je jednostavna aplikacija kojoj se može pridodati nekoliko zanimljivih i korisnih funkcionalnosti koje bi aplikaciju proširile i olakšale korisniku njezino upotrebljavanje. Primjerice, može se kreirati administracijski dio gdje korisnik ima mogućnost otvaranja svog računa, a o rješenju komunalnog problema korisnik bi mogao biti obaviješten putem e-maila. Nadalje, moguće je kreirati tablicu gdje korisnici mogu filtrirati komunalne probleme po vrijednostima poput datuma, statusa, lokacije i slično. Izrada ove aplikacije pridonijela je osviještenost koliko su JavaScript i PHP moćni i prošireni jezici. Iskustvo projektiranja ovog rada autoru je obogatilo percepciju i znanje za buduće projekte s kojima će se susretati.

7. LITERATURA

Knjige:

David S. McFarland (2012). JavaScript & jQuery: The Missing Manual: O'Reilly Media, Inc.

Douglas Crockford (2008). JavaScript: The Good Parts: The Good Parts: : O'Reilly Media, Inc.

Marijn Haverbeke (2018). Eloquent JavaScript: A Modern Introduction to Programming: No Starch Press.

Mark Myers (2017). A Smarter Way to Learn JavaScript: CreateSpace Independent Publishing Platform.

Robin Nixon (2014). Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites 3rd Edition: O'Reilly Media, Inc.

Internetski izvori:

https://www.w3schools.com/js/js_variables.asp (2.08.2021.)

https://www.w3schools.com/js/js_const.asp (3.08.2021)

https://repozitorij.vuv.hr/islandora/object/vsmti%3A549 (6.08.2021)

http://ommolketab.ir/aaf-lib/4cdbqz68f1sdz49yauhpc6z1k1bxau.pdf (9.08.2021)

Laboratorijske vježbe:

dr.sc. Oliver Jukić, prof.v.š., Skriptni programski jezici, Laboratorijske vježbe broj 1.

dr.sc. Oliver Jukić, prof.v.š., Skriptni programski jezici, Laboratorijske vježbe broj 2.

dr.sc. Oliver Jukić, prof.v.š., Osnove web programiranja, Laboratorijske vježbe broj 1.

Predavanja:

dr.sc. Oliver Jukić, prof.v.š., Skriptni programski jezici, JavaScript-Uvod-Tipovi_Podataka-Operatori.

dr.sc. Oliver Jukić, prof.v.š., Web programiranje na strani poslužitelja, Uvod-Varijable-Nizovi-Polja.

8. POPIS ILUSTRACIJA

Slike:

1. Primjer JavaScript varijabli
2. Primjer JavaScript tipova podataka
3. Primjer korištenja operatora
4. Primjer if naredbe za kontrolu toka
5. Primjer objekta
6. Kreiranje i pozivanje funkcije
7. JQuery metoda append
8. Primjer PHP varijabli
9. Primjer PHP tipova podataka
10. Primjer PHP operatora
11. Primjer kontrole toka if naredbom
12. Primjer kreiranja klase i objekta
13. Primjer kreiranja i pozivanja funkcije
14. Primjer HTML elementa
15. Primjer dva gumba
16. Primjer forme
17. Primjer uporabe CSS-a
18. Primjer uporabe jQuery Ajax
19. Povezivanje na bazu podataka
20. DateTimePicker
21. Primjer Bootstrap modala
22. Google maps
23. Prikaz arhitekture aplikacije
24. Početna stranica aplikacije
25. Modal “Dodaj ticket”
26. Funkcija addListener
27. Funkcija za dodavanje ticketa
28. Modal broj ticketa
29. Modal provjerenog ticketa
30. Funkcija za pronalazak ticketa

Veleučilište u Virovitici

OBRAZAC 5

IZJAVA O AUTORSTVU

Ja, Trpimir Tomašić

izjavljujem da sam autor/ica završnog/diplomskog rada pod nazivom

Komunalni asistent - citizen wizard

Svojim vlastoručnim potpisom jamčim sljedeće:

- da je predani završni/diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korištene bilješke i bibliografija,
- da su radovi i mišljenja drugih autora/ica, koje sam u svom radu koristio/la, jasno navedeni i označeni u tekstu te u popisu literature,
- da sam u radu poštivao/la pravila znanstvenog i akademskog rada.

Potpis studenta/ice

Trpimir Tomašić

Veleučilište u Virovitici

OBRAZAC 6

ODOBRENJE ZA POHRANU I OBJAVU ZAVRŠNOG/DIPLOMSKOG RADA

Ja Tripimir Tomašić

dajem odobrenje za objavljivanje mog autorskog završnog/diplomskog rada u javno dostupnom digitalnom repozitoriju Veleučilišta u Virovitici te u javnoj internetskoj bazi završnih radova Nacionalne i sveučilišne knjižnice bez vremenskog ograničenja i novčane nadoknade, a u skladu s odredbama članka 83. stavka 11. Zakona o znanstvenoj djelatnosti i visokom obrazovanju (NN 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15, 131/17).

Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog završnog/diplomskog rada. Ovom izjavom, kao autor navedenog rada dajem odobrenje i da se moj rad, bez naknade, trajno javno objavi i besplatno učini dostupnim:

a) široj javnosti
b) studentima i djelatnicima ustanove
c) široj javnosti, ali nakon proteka 6 / 12 / 24 mjeseci (zaokružite odgovarajući broj mjeseci).

Potpis studenta/ice

Tripimir Tomašić

U Virovitici, 20. 8. 2021.

**U slučaju potrebe dodatnog ograničavanja pristupa Vašem završnom/diplomskom radu, podnosi se pisani obrazloženi zahtjev.*