Aplikacija za upravljanje autoškolom

Jović, Mateo

Undergraduate thesis / Završni rad

2022

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:088256

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Preddiplomski Stručni studij Računarstva

APLIKACIJA ZA UPRAVLJANJE AUTOŠKOLOM

Završni rad

Mateo Jović

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 18.09.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	Mateo Jović
Studij, smjer:	Preddiplomski stručni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AR 4727, 19.07.2019.
OIB Pristupnika:	47229271094
Mentor:	Robert Šojo, mag. ing. comp.
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Marina Peko, dipl. ing.
Član Povjerenstva 1:	Robert Šojo, mag. ing. comp.
Član Povjerenstva 2:	mr.sc. Željko Štanfel
Naslov završnog rada:	Aplikacija za upravljanje autoškolom
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Aplikacija omogućava vođenje autoškole u svakoj etapi polaznikova napredovanja prema završnom ispitu. Postoje različite uloge u aplikaciji za različite zaposlenici u autoškoli (administrator aplikacije, nastavnik teorijskog dijela, nastavnik prve pomoći, instruktor vožnje, komisija). Administrator nadgleda korake prijave ispita te potvrđuje njihove realizacije, kreira uloge u aplikaciji, ima uvid u konačni uspjeh polaznika u svakom dijelu polaganja, kreira polaznike. Nastavnik teorijskog dijela i nastavnik prve pomoći potvrđuje nove polaznike, evidencija dolazaka,
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	18.09.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 22.09.2022.

Ime i prezime studenta:	Mateo Jović
Studij:	Preddiplomski stručni studij Računarstvo
Mat. br. studenta, godina upisa:	AR 4727, 19.07.2019.
Turnitin podudaranje [%]:	13

Ovom izjavom izjavljujem da je rad pod nazivom**: Aplikacija za upravljanje autoškolom**

izrađen pod vodstvom mentora Robert Šojo, mag. ing. comp.

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. POSTOJEĆE APLIKACIJE 3
2.1. Autoškola 3
2.2. Total Drive 4
2.3. Auto Ecole 5
3. KORIŠTENE TEHNOLOGJIE PRI IZRADI WEB PORTALA 7
3.1. HTML 7
3.2. CSS 10
3.3. Bootstrap 11
3.4. JavaScript 12
3.5. PHP 14
3.6. Visual Studio Code 16
3.7. Firebase 16
4. FUNKCIONALNOSTI APLIKACIJE 17
4.1. Struktura aplikacije 17
4.2. Struktura baze podataka 18
4.3. Početna stranica web aplikacije 19
4.4. Registracija 20
4.5. Login 22
4.6. Pregled registriranih korisnika 23
4.7. Pregled polaznika 24
4.8. Odabir Polaznika 25
4.9. Prikaz prihvačenih polaznika 29
4.10. Pregled dolazaka i položenog ispita 29
4.11. Prikaz Rasporeda 32
4.12. Komisija Pregled 36
4.13. Uspjeh polaznika 36
5. ZAKLJUČAK 37
LITERATURA 38
SAŽETAK 39
ABSTRACT 40

1. UVOD

U ovom završnom radu objašnjena je funkcionalnost i metode korištene pri izradi web aplikacije za upravljanje autoškolom. Upravljanje autoškolom je interaktivna aplikacija koja zahtjeva korisničku aktivnost. Unutar aplikacije postoji 6 različitih uloga: nastavnik teorije, nastavnik prve pomoći, instruktor B kategorije, instruktor C kategorije, komisija i admin. Admin ima najveću ulogu, ima pregled na sve registrirane korisnike i mogućnost dodavanja uloga korisnicima. Dodaje nove polaznike i vozila, ima pregled svih polaznika i potvrđuje prolazak ispita. Nastavnik teorije i nastavnik prve pomoći odabiru polaznike, kreiraju raspored predavanja, evidentiraju njihove dolaske i rezultate ispita. Polaznik da bi mogao izaći na ispit iz teorije mora odslušati barem 30 sati predavanja, a za ispit prve pomoći mora odslušati barem 10 sati predavanja. Instruktor može odabrati polaznike tek nakon što im se potvrdi položena teorija iz vožnje i položena prva pomoć od strane admina. Definira raspored predavanja i unosi polaznike u slobodne termine za vožnju. Nakon što polaznik odvozi 35 sati instruktor ga prijavljuje na glavnu vožnju i dodaje komisiju. Komisija je registrirana osoba s ulogom komisija, ima uvid na polaznike koji su mu dodani i unosi rezultate glavne vožnje. Admin potvrđuje prolazak glavne vožnje i ima uvid na sve polaznike koji su riješili sve ispite, njihove nastavnike i broj izlazaka potreban za prolaz odgovarajućeg ispita.

U drugom poglavlju predstavljena su slična rješenja kroz pregled područja izrađena u drugim tehnologijama. U trećem poglavlju su objašnjene tehnologije koje se koriste za razvoj web aplikacije opisane u radu. U četvrtom poglavlju su detaljno opisane funkcionalnosti aplikacije i metode koje se koriste.

1.1. Zadatak završnog rada

Aplikacija omogućava vođenje autoškole u svakoj etapi polaznikova napredovanja prema završnom ispitu. Postoje različite uloge u aplikaciji za različite zaposlenike u autoškoli (administrator aplikacije, nastavnik teorijskog dijela, nastavnik prve pomoći, instruktor vožnje, komisija). Administrator nadgleda korake prijave ispita te potvrđuje njihove realizacije, kreira uloge u aplikaciji, ima uvid u konačni uspjeh polaznika u svakom dijelu polaganja, kreira polaznike. Nastavnik teorijskog dijela i nastavnik prve pomoći potvrđuje nove polaznike, evidencija dolazaka, njihova napretka, prijava na ispit, definira raspored nastave. Instruktor

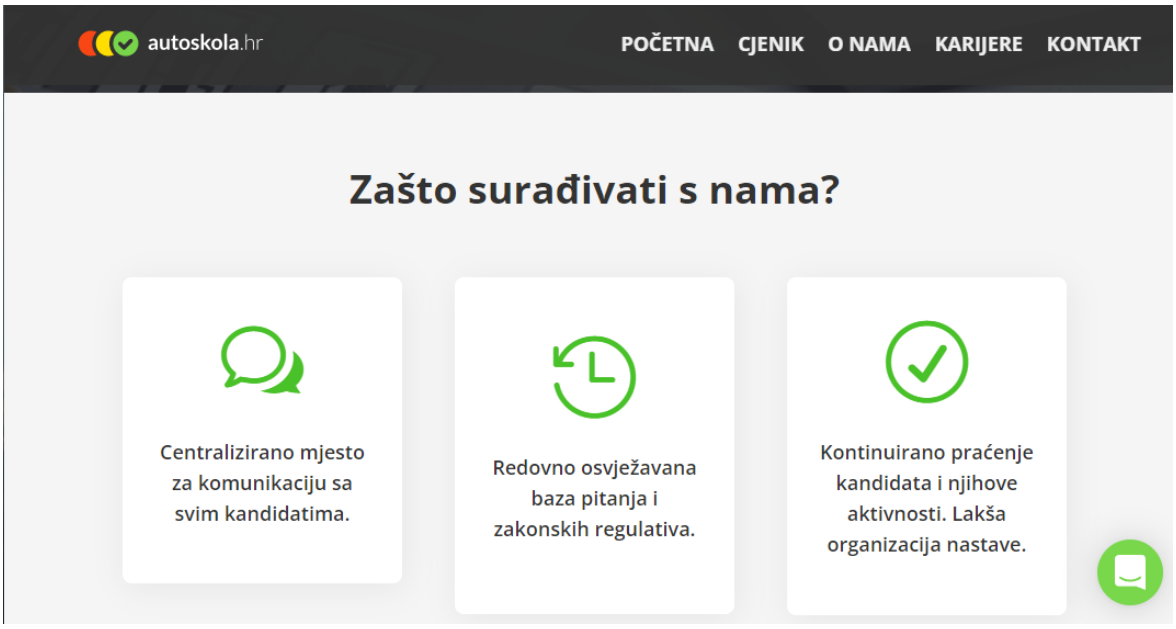
vožnje odabire polaznike, odabire vozilo, uvid u položene prethodne ispite s uspjehom, evidencija dolazaka, evidencija napretka, definiranje termina ispita, definira raspored vožnje, definiranje komisije, mogućnost dodavanja dodatnih sati. Komisija ima uvid u položene prethodne ispite s uspjehom, potvrda uspješno položenog ispita vožnje, komentari na vožnju. Student treba razviti funkcionalnu aplikaciju koristeći različite tehnologije.

2. POSTOJEĆE APLIKACIJE

U ovom poglavlju dan je prikaz sličnih aplikacija za prikupljanje i prikaz određenih podataka vezanih za upravljanje autoškolom.

2.1. Autoškola

Glavni cilj platforme autoskola.hr je olakšati autoškolama poslovanje i pomoći im da lakše i kvalitetnije pripreme kandidate za polaganje ispita iz prometnih propisa i prometnih pravila. Autoškola hr. nudi jednostavnu evidenciju kandidata i grupa, uvid u aktivnosti svakog kandidata, simulacija ispita iz prometnih propisa i sigurnosnih pravila. Moguće je koristiti postojeće ali i stvaranje vlastitih domaćih zadaća. Kako bi polaganje teorijskog ispita prošlo jednostavno nudi 1100+ redovito ažuriranih pitanja. Poslovanje bez ugovorne obveze, uvijek dostupna korisnička podrška za kandidate i autoškole. Slika 2.1. prikazuje početnu stranicu Autoškole. prikazuje početnu Prilikom prijave kao autoškola potrebno je plaćati određeni mjesečni iznos ovisno o tome koliko postoji kandidata godišnje [1].



Sl. 2.1. *Autoškola.hr.*

2.2. Total Drive

Total Drive predstavlja mobilnu aplikaciju za instruktore, autoškole s više automobila i njihove učenike. Omogućava upravljanje svojim dnevnikom, evidencijom učenika, napretkom učenika i porukama na jednom mjestu, s digitalnim zapisima bez papira [2].

Pogodnosti instruktora:

1. Prilagodljiva aplikacija za instruktorski dnevnik
2. Dodavanje pojedinačne lekcije, intenzivne tečajeve i tjedna ponavljanja
3. Odabir boja svog dnevnika kako bi odgovaralo Vašem načinu rada
4. Klikom na mrežu dnevnika može se brzo i jednostavno unositi lekcije
5. Besplatni podsjetnici lekcija poslani 24/48 sati unaprijed
6. Dnevnik rada potpuno offline
7. Sinkronizacija s Google i Apple kalendarom

Zapisi učenika:

1. Kartica lekcija s budućim i prošlim lekcijama
2. Nastavni plan i program napretka u vožnji
3. Praktični pokušaji testiranja i stopa uspjeha
4. Reflektivni zapisnici, privatne bilješke i sažeci lekcija
5. Kontaktni podaci učenika, dozvola i provjera vida

Pogodnosti za autoškole:

1. Pregled svih rezervacija, učenika i statistike na jednom mjestu
2. Tjedna analiza isporučenih i rezerviranih sati
3. Postotak rezerviranih za sljedeću lekciju
4. Postotak prolaznosti
5. Broj na listi čekanja i aktivnih učenika
6. Dugovanja vezana uz plaćanje
7. Status položene teorije Vaših učenika
8. Pregled instruktora

Pogodnosti za polaznike:

1. Lako pregledavanje rezerviranih i prošlih lekcija
2. Primanje automatske podsjetnike za lekcije
3. Iskorištavanje dodatnih lekcije kroz sustav praznina

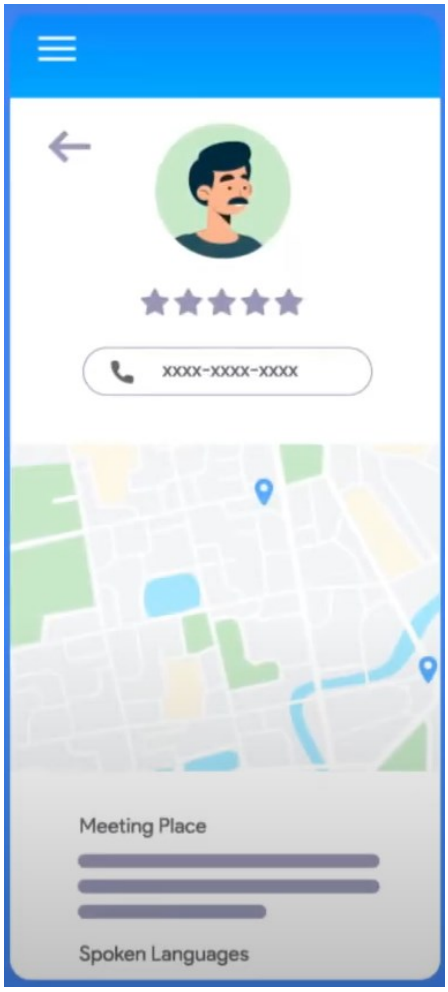
Slika 2.2. predstavlja raspored unutar aplikacije

Sl. 2.2. *Prikaz rasporeda.*

2.3. Auto Ecole

Aplikacija Auto Ecole izrađena je kako bi potencijalni polaznici autoškola mogli vidjeti autoškole blizu njih, instruktore, vozila autoškola te ostavljene komentare vezane za autoškolu. Instruktori na popisu aplikacije su potvrđeni. Putem aplikacije moguće je rezervirati i plaćati sate vožnje. Auto Ecole nudi praćenje vlastitog napretka do dobivanja vozačke dozvole. Aplikacija centralizira sve vaše rezervacije s vašom autoškolom ili instruktorom, pojedinosti svakog tečaja i

podatke za kontakt vašeg instruktora. Aplikacija je besplatna za sve osobe koje polažu vozački ispit [3]. Slika 2.3. prikazuje profil instruktora.



Sl. 2.3. *Prikaz profila instruktora.*

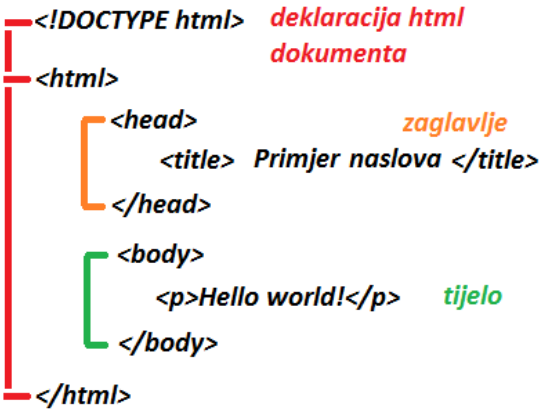
3. KORIŠTENE TEHNOLOGJIE PRI IZRADI WEB PORTALA

Web aplikacija je računalni program koji koristi web preglednik i web tehnologiju za obavljanje zadataka putem interneta. Web aplikacije mogu biti dizajnirane za široku paletu namjene i mogu ih koristiti svi, od organizacije do pojedinca iz brojnih razloga. Princip povezanosti na kojem radi je klijent-server te se klijentu pruža grafičko sučelje koje je definirano na serveru. Koristi kombinaciju skripti na strani poslužitelja PHP (engl. *Hypertext Preprocessor*) i ASP (engl. *Active Server Pages*) za rukovanje pohranom i dohvaćanjem informacija, a skripte na strani klijenta JavaScript i HTML (engl. *HyperText Markup Language*) kako bi prezentirali informacije korisnicima. Omogućuju zaposlenicima izradu dokumenata, razmjenu informacija, suradnju na projektima i rad na zajedničkim dokumentima bez obzira na lokaciju ili uređaj. Web aplikacije obično imaju kratke razvojne cikluse i mogu se izraditi s malim razvojnim timovima. Nekim web aplikacijama može pristupiti samo određeni preglednik, no većina je dostupna bez obzira na preglednik. Poznate web aplikacije su: Gmail, Microsoft 365, Google Apps.

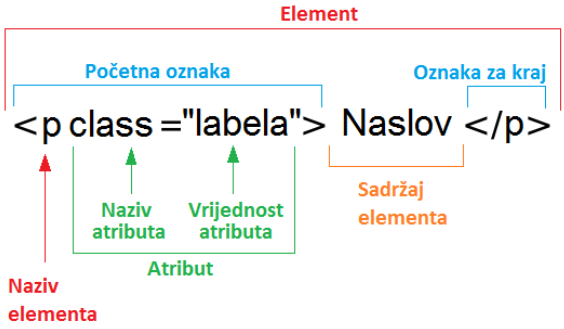
3.1. HTML

HypterText Markup Language ili HTML jezik je koji se koristi prilikom izarade webstranica. Prvu verziju HTML-a napisao je Tim Berners-Lee 1993. i otada je postajalo mnogo različitih verzija HTML-a. Najraširenija verzija tijekom 2000-ih bila je HTML 4.01, koja je postala službeni standard u prosincu 1999. godine[4]. Trenutno najraširenija verzija je HTML5 kojeg preglednici i drugi uređaji koji podržavaju web već široko podržavaju. Dolaskom do velike popularnosti Web-a HTML je poboljšan kako bi zadovoljio zahtjeve za pristupom multimediji, veću kontrolu izgleda i fontova te podršku za interaktivne aplikacije. Od 2018. za prikaz sadržaja i podršku interakcije na webu koriste se: HTML za strukturno označavanje temeljnog sadržaja, CSS (engl. *cascading style sheets*) za primjenu oblikovanja na taj sadržaj i *JavaScript* za podršku u interakciji i kao temelj okvira za upravljanje sadržajem koji dinamički sastavljaju stranice iz dijelova sadržaja [5].

Na slici 3.1. nalazi se struktura HTML-a dokumenta. Svaki HTML dokument počinje s <html> i završava s </html>. Unutar tih oznaka nalazi se dvije grupe, <head> i <body>. Unutar head-a se unosi naslov (<title>), povezuju se vanjske datoteke CSS i/ili JavaScript datoteke. Unutar <body> nalaze se sve ostale informacije kao što su tekstovi, slike, videozapisi. Prikaz elementa HTML-a nalazi se na slici 3.2.



Sl. 3.1. *Struktura HTML-a.[6]*



Sl. 3.2. *Element HTML-a.*

HTML element sastoji se od naziva elemta, oznake za početak, oznake za kraj, atributa i sadržaja. Atributi se koriste kako bi browser-u dali dodatne informacije vezane uz pojedini element. Atributi kao što su ID i Class služe za povezivanje s CSS-om i JavaScript-om. Neki od tekstualnih elemenata su: heading, paragraf, link. Koriste se kako bi stvorili organizacijsku strukturu, poboljšali vizualni doživljaj stranice i pomogli kod grupacije dijelova sadržaja. U HTML-u je moguće postaviti 6 posebnih razina dokumenta koji se određuju naslovima. <h1>. <h1> bi bio najviše istaknut naslov dok bi <h6> bio najmanje istaknut naslov. Unutar svakog elementa mogu se koristit oznake: <b> - podebljanje teksta, <i> - kosi tekst i <u> - podvučena crta. Povećanje preglednost sadržaja postiže se unošenjem slika, listi i tablica. Liste se dijele na uređenu i neuređenu. Uređenu listu koristimo kada želimo prebrojati informacije i koriste se numeričke oznake. Neuređenu listu koristimo kada redosljed po kojem su izlistane stavke nije nužan za razumijevanje infromacija koje se prikazuju. Slika 3.3 prikazuje ispit rezultata HTMl elemenata.

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

<h1>H1</h1>
<p>Prvi paragraf.</p>

<h2>H2</h2>
<p>Drugi paragraf.</p>

<h3>H3</h3>
<p>Treći paragraf.</p>

<h4>H4</h4>
<p>Četvrti paragraf.</p>

<h5>H5</h5>
<p>Peti paragraf.</p>

<h6>H6</h6>
<p>Šesti paragraf.</p>

</body>
</html>
```

H1

Prvi paragraf.

H2

Drugi paragraf.

H3

Treći paragraf.

H4

Četvrti paragraf.

H5

Peti paragraf.

H6

Šesti paragraf.

Sl. 3.3. *Primjer HTML elemenata i njihovi rezultati.*

3.2. CSS

CSS predstavlja jednostavan mehanizam za dodavanje stilova (npr. fontova, boja, razmaka) web dokumentima [7]. CSS je kreiran kako bi omogućio web dizajnerima da mijenjaju izgled, boje i fontove svojih web stranica[9]. Izvorno, web stranice su bile namijenjene samo istraživačima, tako da ukras nije bio važan. Kako su web stranice postale raširene, rasla je potreba da izgledaju lijepo. CSS danas podržavaju gotovo svi preglednici, a i podržavaju ga i druge aplikacije. Prilikom korištenja CSS-a dovoljan je ureživač teksta, ali mogu se koristiti i drugi alati koji to čine lakšim.

Korištenjem CSS obrasca mogu se kontrolirati bilo koji dijelovi segmenta na stranici:

1. Podešavanje pozadine (boja, slike)
2. Uređivanje margina
3. Kontrola veličine elemenata
4. Uređivanje tablica
5. Uređivanje teksta

HTML omogućava povezivanje CSS-a na tri načina:

- Unutar jednog HTML elementa, putem atributa „style“, (linijski CSS),
- Unutar <head> dijela među oznakama <style>, (unutarnji CSS),
- U posebnoj CSS datoteci, (vanjski CSS).

Moguće je referencirati višestruke CSS dokumente unutar jednog HTML dokumenta. Najčešći način korištenja CSS-a je vanjski način. Korištenjem tog načina uštedi se najviše vremena, sav stil se odredi unutar vanjske CSS datoteke, poziva se putem <link> oznake. Slike 3.4, 3.5 i 3.6 prikazuju načine povezivanja CSS-a.

```
<head>
<link rel="stylesheet" type="text/css" href="stil.css">
</head>
```

Sl. 3.4. *Povezivanje CSS-a, vanjski način.*

```
<p style="color:green;font-size:20px;">Ovo je paragraf</p>
```

Sl. 3.6. *Povezivanje CSS-a, linijski način.*

```
<head>
<style type="text/css">
body {background-color:red;}
p {color:blue;}
</style>
</head>
```

Sl. 3.5. *Povezivanje CSS-a, unutarnji način.*

3.3. Bootstrap

Bootstrap je snažan frontend alat pun značajki. Pomoću bootstrapa može se doći od prototipa do proizvodnje - za nekoliko minuta. Bootstrap koristi mnogo važnih globalnih stilova i postavki, od kojih su svi gotovo isključivo usmjereni na normalizaciju stilova među preglednicima [10]. Bootstrap je prvobitno razvijen za mobilne uređaje, strategija u kojoj prvo optimiziramo kod za mobilne uređaje, a zatim po potrebi povećavamo komponente pomoću CSS medijskih upita. Izvorno stvoren od strane dizajnera i programera na Twitteru, Bootstrap je postao jedan od najpopularnijih front-end okvira i open source projekata na svijetu [11]. Bootstrap je stvoren na Twitteru sredinom 2010. godine. Najnovije izdanje Bootstrap 5, usredotočeno je na poboljšanje kodne baze v4 sa što manje velikih promjena. Poboljšava postojeće značajke i komponente, uklonjena podrška za starije preglednike, izbačen jQuery za obični JavaScript i prihvaćanje tehnologije koje su prilagođenije budućnosti kao što su CSS

prilagođena svojstva kao dio alata. Bootstrapov stil i izgled mogu se primijeniti na širok raspon struktura za označavanje. Bootstrap card kod je prikazan na slici 3.7.

```
<div class="card" style="width: 18rem;">
 <img src="image.jpg" class="card-img-top" alt="…">
 <div class="card-body">
   <h5 class="card_title">Card title</h5>
   <p class="card-text">Some quick example text
    to build on the card title and make up the bulk
    of the card's content.</p>
   <a href="#" class="btn btn-primary">Go somewhere</a>
 </div>
</div>
```

Sl. 3.7. *Bootstrap card primjer*

3.4. JavaScript

JavaScript (JS) je lagan, interpretiran ili kompajliran pravovremeno programski jezik s prvoklasnim funkcijama. Iako je najpoznatiji kao skriptni jezik za web stranice, koriste ga i mnoga okruženja koja nisu preglednici, kao što su Node.js, Apache CouchDB i Adobe Acrobat [12]. JavaScript je prototipski, višeparadigmatski, jednonitni, dinamički jezik koji podržava objektno orijentirane, imperativne i deklarativne (npr. funkcionalno programiranje) stilove. Deklaracija varijabli se vrši uz ključnu riječ var, npr: var nazivAutomobila. Varijable se dijele na znakovne, logičke, cjelobrojne i decimalne.

JavaScript se najčešće koristi za:

1. Dodavanje interaktivnosti na web stranicu čime se dobiva dinamična web lokacija
2. Razvoj mobilnih aplikacija
3. Izradu igara temeljenih na web pregledniku
4. Pohranu i učitavanje informacija o korisnikovom računalu
5. Provjeru ispravnosti i vjerodostojnosti podataka prije nego što se pošalju na server.

Funkcija za provjeru unosa imena nalazi se na slici 3.8.

```
<script>
 function provjeri() {
  let ime = document.getElementById(„ime“).value;
  if(ime.length == 0) {
   alert(„Molimo, unesite ime“);
  }else if(ime.length < 3) {
   alert(„Potrebno je unijeti barem 3 znaka.“);
  }
 }
</script>
```

Sl. 3.8. *Primjer funkcije za provjeru unosa imena.*

Naziv JavaScript proizašao je iz Netscapeove podrške za Java aplete unutar njegovog preglednika [13]. JavaScript je stvoren za tumačenje tijekom izvođenja, što ga čini mnogo dinamičnijim. JavaScript ima okidače ili pokretače događaja koji pokreću određene funkcije. Koriste se za izradu interaktivnih dijelova web stranice kao što su gumbi koji reagiraju na klik miša, izbornici, izmjenjivi sadržaji i drugo. Primjer promjene teksta prikazan je na slici 3.9.

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Drugi
tekst.";
}
</script>
</head>
<body>

<h2>JavaScript primjer</h2>

<p id="demo">Prvi tekst.</p>

<button type="button" onclick="myFunction()">Try
it</button>

</body>
</html>
```

JavaScript primjer

Drugi tekst.

Try it

Sl. 3.9. *Primjer promjene teksta paragrafa pritiskom na gumb.*

3.5. PHP

Dizajniran je za web razvoj i programiranje opće namjene 1994. Godine. S više od dva desetljeća razvoja, PHP je doživio i uspone i padove. Sada PHP-om upravlja The PHP Group i on je u stalnom razvoju. PHP je skraćenica za *Hypertext Preprocessor* koji je promijenjen u odnosu na početno ime "*Personal Home Page*"[15]. PHP se uglavnom koristi s HTML kodom, sustavom za upravljanje web sadržajem, sustavima web predložaka i drugim popularnim web okvirima. PHP jezik obrađuje poslužitelj. U svakom slučaju, može se koristiti za stvaranje nevjerojatne web aplikacije u kojoj se PHP kod uvijek izvršava na strani poslužitelja. Nadalje, može se koristiti i za implementaciju grafičke aplikacije koja je samostalne prirode. Najnovija PHP verzija trenutno je PHP 7.4. Donosi puno novih značajki, dodataka sintakse i popravaka. PHP datoteke mogu sadržavati tekst, HTML, CSS, JavaScript i PHP kod. PHP kod se izvršava na poslužitelju, a rezultat se vraća pregledniku kao običan HTML [16].

PHP može:

- Generirati dinamičan sadržaj stranice
- Stvarati, otvarati, čitati, pisati, brisati i zatvarati datoteke na poslužitelju
- Prikupljati podatke obrasca
- Slati i primati kolačiće

- Dodavati, brisati, mijenjati podatke u Vašoj bazi podataka
- Kontrolirati pristup korisnika
- Šifrirati podatke

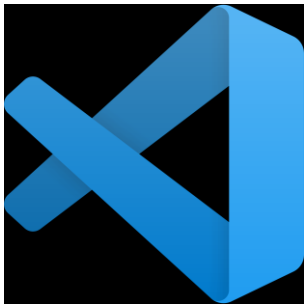
PHP radi na raznim platformama (Windows, Linux, Unix, Mac OS X, itd.). PHP je komplatibilan s gotovo svim poslužiteljima koji se danas koriste (Apache, IIS, itd.). Podržava šitok raspon baza podataka, besplatan je. Php unutar html-a prikazan je na slici 3.10.

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


<?php
Class Auto {
 public $boja
 public $model
 public function __construct($boja, $model){
  $this->boja = $boja;
  $this->model = $model;
 }
 public function poruka(){
  return „Moj auto je „ . $this->boja . „ „ . $this->model . „!“;
 }
}

$mojAuto = new Auto(„crni“, „Volvo“);
echo $mojAuto-> poruka();
echo „<br>“;
$mojAuto = new Auto(„crveni“, „Toyota“);
echo $mojAuto -> poruka();
?>

</body>
</html>
```

Sl. 3.10. *PHP unutar html dokumenta.*

3.6. Visual Studio Code

Visual Studio Code je lagan, ali moćan uređivač izvornog koda i dostupan je za Windows, macOS i Linux. Dolazi s ugrađenom podrškom za JavaScript, TypeScript i Node.js i ima bogat ekosustav proširenja za druge jezike (kao što su C++, C#, Java, Python, PHP, Go) i vremena izvođenja (kao što je .NET) [17]. Značajke za podršku su: *debugging*, *snippet*, refaktoriranje koda i ugrađeni Git. Visual Studio Code je jedan od najpopularnijih razvojnih alata. Slika 3.11. prikazuje ikonu.



Sl. 3.11. *Visual Studio Code ikona.*

3.7. Firebase

Firebase je platforma za razvoj aplikacija koja pomaže u izradi i razvoju aplikacija i igara za korisnike. Firebase ima podršku Google-a i povjerenje milijuna tvrtki širom svijeta. Firebase autentifikacijska i sigurnosna pravila središnja su za sigurnu aplikaciju. Za otkrivalje zlonamjernog prometa, kao što su napadi uskraćivanja usluge (DOS), potrebno je postaviti nadzor i upozorenje za Cloud Firestore, bazu podataka u stvarnom vremenu i pohranu u oblaku. Firebase koristi API ključeve samo za identifikaciju Firebase projekta aplikacije na Firebase uslugama, ne za kontrolu pristupa bazi podataka (to se radi pomoću Firebase sigurnosnih pravila). Firebase A/B testiranje pomaže optimizirati iskustvo aplikacije olakšavajući pokretanje, analizu i skaliranje eksperimenata proizvoda i marketinga [20]. Nudi usluge isprobavanja prilike s korisničkim sučeljem, značajkama ili marketinškim kampanjama svoje aplikacije kako bi vidjeli jesu li one drugačije u mjernim podacima od važnosti, prije nego što se uvedu u ostatak svijeta. Odluke se mogu podržavati stvarnim podacima, a ne samo slutnjama.

4. FUNKCIONALNOSTI APLIKACIJE

U ovom poglavlju predstavljen je postupak izrade funkcionalnosti vezane uz upravljanje autoškolom. Objašnjene su metode izrade dijelova aplikacije. Opisana je baza i način korištenja podataka.

4.1. Struktura aplikacije

U izrađenoj aplikaciji upravljanje autoškolom osmišljeni su i napravljeni dnosi koji se trebaju razumjeti kako bi se moglo služiti aplikacijom. Aplikacija je namjenjena za više korisnika s različitim ulogama. Postojeće uloge u aplikaciji su: Administrator, Nastavnik teorije, Nastavnik prve pomoći, Instruktor kategorije B, Instruktor kategorije C, Komisija i Gost. Gost je neprijavljeni korisnik koji unutar aplikacije može pristupiti samo osnovnim podacima vezanim za Autoškolu.

Administrator ima najveću ulogu. Administrator ima uvid na registrirane korisnike te im može dodavati ili brisati uloge. Administrator dodaje nove polaznike i ima uvid u njihov uspijeh prilikom polaganja ispita do dobivanja vozačke dozvole za određenu kategoriju. Administrator potvrđuje prolaznost svakog pojedinog ispita, te se bez njegove potvrde ne mogu nastaviti izvoditi polaznikovi koraci do vozačke dozvole. Administrator tijekom bilo kojeg ciklusa polaznikovog puta do vozačke dozvole može obrisati polaznika.

Korisnik s postavljenom ulogom nastavnika teorije, može odabirati polaznike. Definira vlastiti raspored predavanja, unosi pojedino za svakog polaznika evidenciju dolaznosti, te nakon odslušanih 30 sati teorije polaznik je spreman za ispit iz teorije. Nastavnik teorije unosi rezultate ispita teorije, te vezano uz rezultate polaznik prelazi na prvu pomoć ili ponovno izlazi na teorijski ispit.

Nakon administratorove potvrde o prolasku teorije, korisnik s ulogom nastavnika prve pomoći odabire polaznike koji su položili teoriju vezanu za vožnju. Nastavnik prve pomoći ima iste mogućnosti kao i nastavnik teorije samo vezane za prvu pomoć. Ukoliko bi polaznik položio prvu pomoć i administrator to prihvatio došao bi na red instruktor vožnje.

Instruktoru vožnje prikazuju se samo oni polaznici s položenim i potvrđenim ispitima iz teorije i prve pomoći. Prikazani su mu polaznici namjenjeni za njegovu kategoriju. Instruktor si definira raspored i unosi termine za svakog prihvaćenog polaznika. Unosi evidenciju polaznika, komentare na vožnju, te može prijaviti glavnu vožnju i odabrati osobu za komisiju. Ukoliko polaznik položi glavnu vožnju instruktor ga više ne može unositi u raspored za vožnju.

Korisnik s ulogom komisija ima uvid na polaznike kojima je on dodjeljen kao komisija. Unosi rezultat vožnje i komentar na vožnju. Na samom kraju Administrator potvrđuje položenu glavnu vožnju polazniku.

4.2. Struktura baze podataka

Pri izradi aplikacije korištena je platforma Firebase. Potrebno je instalirati composer kako bi se mogao koristiti Firebase. Za kreiranje i modeliranje baze podataka nakon instaliranja composera potrebno je postaviti bazu unutar projekta . Inicijalizacijom baze podataka i korisnika postavljeno je sučelje za rad. Pokretanjem XAMPPA potrebno je uključiti Apache web server. Aplikacija se sastoji od 14 zasebnih tablica. Svaka tablica prikazuje podatke vezane za polaznikov put do dobivanje vozačke dozvole, neizostavni je dio aplikacije. Određene tablice su povezane zbog lakšeg i preglednijeg korištenja podataka iz tablica. Id određenog polaznika bio bio unesen unutar atributa polaznikuid u tablici raspored te bi se tako održala konekcija između polaznika i rasporeda, te kako ne bi došlo do greške prilikom ispisa podataka. Slikom 4.1. prikazana je php datoteka za spajanje s firebaseom. Slika 4.2. prikazuje jednog elementa iz „dolazakTeorija“ i na koji način je dodana veza na polaznika.

dbcon.php

```php
<?php

require __DIR__.'/vendor/autoload.php';

use Kreait\Firebase\Factory;
use Kreait\Firebase\Contract\Auth;

$factory = (new Factory)
    ->withServiceAccount('upravljanje-autoskolom-firebase-adminsdk-75hv4-4f209f0e47.json')
    ->withDatabaseUri('https://upravljanje-autoskolom-default-rtdb.firebaseio.com/');

$database = $factory->createDatabase();
$auth = $factory->createAuth();



?>
```

Sl. 4.1. *dbcon.php.*

```
dolazakTeorija
    -NC-y6FOhhsM4GEi36ED
        brojSati: "7"
        komentar: "Polaznik prati na satu te aktivno sudjeluje"
        polaznikuid: "-NC-x8QlYkwOzQgIziBC"
```

Sl. 4.2. *Prikaz polaznikovog Id-a unutar atributa polaznikuid.*

4.3. Početna stranica web aplikacije

Početna stranica neprijavljenom korisniku pruža uvid što je potrebno odslušati i položiti da bi se došlo do vozačke dozvole B i C kategorije. Na vrhu se nalazi navigacijska traka koja je izrađena u odvojenoj datoteci i koja se uključuje na svaku sranicu unutar skripte php-a: <?php include('navbar.php);?> i poziva se unutar datoteke header.php. Dizajn navigacijske trake napravljen je pomoću Bootstrap 5.1.3 razvojnog okvira. Unutar tagova <head></head> naveden je Bootstrap i css korišten za aplikaciju. Na temelju korisnikove uloge unutar aplikacije prikazuju mu se zadani linkovi na odgovarajuće php dokumente. Primjer uvjeta koji to omogućava nastavniku teorije da se samo njemu prikaže navigacija traka za teoriju je ?php if(isset($_SESSION($_SESSION['verified_nastavnikT])) : ?>. Primjer navigacijske trake za neprijavljenog korisnika prikaza je na slici 4.3. Nakon korisnikove prijave prikazuje mu se u gornjem desnom kutu „Logout“. Ispod navigacijske trake nalazi se carousel koji je zapravo Bootstrapova komponenta dijaprojekcije za kruženje kroz elemente, odnosno slike ili slajdove teksta. Za web aplikaciju korištene su dvije slike koje se izmjenjuju. Prikaz carousela nalazi se na slici 4.4.

Sl. 4.3. *Navigacijska traka za neprijavljenog korisnika.*

Sl. 4.4. *Carousel.*

4.4. Registracija

Obrazac za registraciju korisnika unutar web apliakcije nalazi se na navigacijskoj traci s desne strane. Prilikom klika na registraciju otvara se sučelje u koje korisnik unosi vlastite podatke. Unosi se puno ime i prezime, zatim email, te lozinka. Sva polja su obavezna prilikom popunjavanja te se ne može kreirati korisnički račun ukoliko nisu popunjena. Ukoliko korisnik ne popuni sva polja ili ih ne popuni po odgovarajućim pravilima stranica će se samo ponovno očitati te će se ispisati poruka što treba promjeniti kako bi se mogao registrirati. Ukoliko korisnik već ima kreirani postojeći račun, ispod gumba „Register“ nalazi se tekst „Login here“ koji vodi na sučelje za prijavu. Prilikom registracije uvedena su brojna ograničenja kako bi se zaštitilo korisnike, a i kako ne bi došlo do rušenja aplikacije. Prilikom pritiska na gumb provjera se jesu li sva polja unesena, ako jesu provjerava se je li puno ime i prezime dugačko barem 6 znakova, nakon toga provjerava se dužina šifre, ukoliko je manja od 6 prekida se regisracija, slika 4.5. Ako su ove stavke ispunjene gleda se dalje u kakvom je formatu email adresa, te se na kraju pregledava postoji li već neki korisnik s tom istom email adresom, slika 4.6. Ukoliko su sve stavke ispunjene i email adresa nije već iskorištena, kreira se novi korisnik i šalje ga se na login stranicu preko koje se zatim prijavljuje. Kod za registraciju nalazi se u code.php datoteci.

```php
if(isset($_POST['register_btn'])){

    $fullname = $_POST['full_name'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    if(empty($fullname) || empty($email) || empty($password)){
        $_SESSION['status'] = "Unesite vrijednosti";
        header('Location: register.php');
        exit();
    }
    if(strlen($fullname)<6){
        $_SESSION['status'] = "Potrebno barem 6 znakova za ime i prezime";
        header('Location: register.php');
        exit();
    }
    if(strlen($password)<6){
        $_SESSION['status'] = "Lozinka mora sadrzavati barem 6 znakova ";
        header('Location: register.php');
        exit();
    }

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {

        $_SESSION['status'] = "Krivi email format";
        header('Location: register.php');
        exit();
    }
```

Sl. 4.5. *Kod za registraciju 1.dio.*

```php
    $users = $auth->listUsers();

    foreach ($users as $user)
    {
        if($user->email == $email){
            $_SESSION['status'] = "Email već postoji u bazi, potreban drugi";
            header('Location: register.php');
        }
    }

    $userProperties = [
        'email' =>  $email,
        'emailVerified' => false,
        'password' => $password,
        'displayName' => $fullname,
    ];

    $createdUser = $auth->createUser($userProperties);

    header('Location: login.php');


}
```

Sl. 4.6. *Kod za registraciju 2.dio.*

4.5. Login

Da bi se korisnik prijavio treba unutar navigacijske trake odabrati stavku login. Klikom na login otvara se obrazac za prijavu koji se sastoji od emaila i lozinke. Kao i kod registracije postoje određeni uvjeti kako bi se prijavio korisnik. Korisnik mora unijeti oba parametra, email mora biti u ispravnom formatu i mora postojati u bazi podataka, te lozinka mora odgovara prijavljenom emailu. Nakon što se ispune uvjeti provjerava se korisnikova uloga te se pohranjuju 3 podatka unutar sesije, slika 4.7. Podaci se pohranjuju na način: $_SESSION['verified_admin'] = true; $_SESSION['verified_user_id'] = $uid; $_SESSION['idTokenString'] = $idTokenString;, primjer za admina, ista stvar se obavlja i za ostale uloge ovisno o prijavljenom korisniku.

```
try {
    $verifiedIdToken = $auth->verifyIdToken($idTokenString);
    $uid = $verifiedIdToken->claims()->get('sub');

    $claims = $auth->getUser($uid)->customClaims;
    if(isset($claims['admin']) == true)
    {
        $_SESSION['verified_admin'] = true;
        $_SESSION['verified_user_id'] = $uid;
        $_SESSION['idTokenString'] = $idTokenString;
    }elseif(isset($claims['nastavnikT']) == true){
        $_SESSION['verified_nastavnikT'] = true;
        $_SESSION['verified_user_id'] = $uid;
        $_SESSION['idTokenString'] = $idTokenString;
    }
    elseif(isset($claims['nastavnikPP']) == true){
        $_SESSION['verified_nastavnikPP'] = true;
        $_SESSION['verified_user_id'] = $uid;
        $_SESSION['idTokenString'] = $idTokenString;
    }
    elseif(isset($claims['instruktorB']) == true){
        $_SESSION['verified_instruktorB'] = true;
        $_SESSION['verified_user_id'] = $uid;
        $_SESSION['idTokenString'] = $idTokenString;
```

Sl. 4.7. *Pohranjivanje korisnikovih podataka unutar sesije.*

4.6. Pregled registriranih korisnika.

Pregled registriranih korisnika ima pravo samo admin, njemu se prilikom logiranja prikazuje u navigacijskoj traci stavke „Registrirani Korisnici“ i „Polaznici“. Pomoću php dokumenta „admin_auth.php“ u kojem je striktno definirano ograničenje samo za admina, uključivanjem tog dokumenta stranici se ne može pristupiti ni preko apsolutne putanje ukoliko korisnik nije prijavljen kao admin. Stranica „Registrirani korisnici“ ispisuje listu prijavljenih registriranih korisnika i njihove paremetre: „Puno ime i prezime“, „Email“ „Uloga“ i „Edit“. Klikom na edit otvara se za odabranog registriranog korisnika nova stranica unutar koje mu se može promjenit uloga u aplikaciji. POST metoda za dodjeljivanje uloge prikazana je slikom 4.8. i 4.9.

```
if(isset($_POST['user_claims_btn']))
{
    $uid = $_POST['claims_user_id'];
    $roles = $_POST['role_as'];

    if($roles == 'admin'){
        $auth->setCustomUserClaims($uid, ['admin' => true]);
        $msg = "Korisnik - Admin";

    }elseif($roles == 'nastavnikT'){
        $auth->setCustomUserClaims($uid, ['nastavnikT' => true]);
        $msg = "Korisnik - Nastavnik Teorije";

    }elseif($roles == 'nastavnikPP'){
        $auth->setCustomUserClaims($uid, ['nastavnikPP' => true]);
        $msg = "Korisnik - Nastavnik Prve Pomoći";

    }elseif($roles == 'instruktorB'){
        $auth->setCustomUserClaims($uid, ['instruktorB' => true]);
        $msg = "Korisnik - Instruktor B kategorije";

    }elseif($roles == 'instruktorC'){
        $auth->setCustomUserClaims($uid, ['instruktorC' => true]);
        $msg = "Korisnik - Instruktor C kategorije";

    }elseif($roles == 'komisija'){
        $auth->setCustomUserClaims($uid, ['komisija' => true]);
        $msg = "Korisnik - Komisija";
```

Sl. 4.8. *Kod za dodjeljivanje uloga 1.dio.*

```
    }else{
        $auth->setCustomUserClaims($uid, null);
        $msg = "Korisnik - nije admin";
    }

    if($msg){
        $_SESSION['status'] = "$msg";
        header("Location: edit.php?id=$uid");
    }else{
        $_SESSION['status'] = "$msg";
        header("Location: edit.php?id=$uid");
    }
}
```

Sl. 4.9. *Kod za dodjeljivanje uloga 2.dio.*

4.7. Pregled polaznika

Klikom na stavku „Polaznici“ kojoj može pristupiti samo admin, otvara se stranica sa listom unesenih polaznika autoškole. Unutar stranice se ispisuju vrijednosti polaznika za: „Ime“, „Prezime“, „Godina“, „Kategorija“, „PolozenaT“, „PotvrdaT“, „PolozenaPP“, „PotvrdaPP“, „PolozenaGV i „PotvrdaGV“. Kada bi polaznik prošao određenu stavku na putu do vozačke dozvole, to bi se unjelo u tablicu i čekala bi se potvrda admina kako bi se ostali koraci aplikacije vezani uz polaznika mogli izvršavati. U gornjem desnom kutu nalaze se dva gumba, „Dodaj polaznika“ koji vodi na stranicu za unošenje polaznika i „Dodaj vozilo“ koje vodi na stranicu za unošenje vozila. Nakon dodavanja polaznika, automatski se učita unutar stranice „Polaznici“. Način na koji se dohvaćaju polaznici je: $ref_table = 'polaznik'; $fetchdata = $database->getReference($ref_table)->getValue();. Nakon toga se polaznici i njihovi podaci prikazuju pomoću foreach petlje. Slika 4.10. prikazuje formu za dodavanje novog polaznika.

```
<form action="code.php" method="POST">

    <div class="form-group mb-3">
        <label for="">Ime</label>
        <input type="text" name="first_name" class="form-control">
    </div>
    <div class="form-group mb-3">
        <label for="">Prezime</label>
        <input type="text" name="last_name" class="form-control">
    </div>
    <div class="form-group mb-3">
        <label for="">Broj godina</label>
        <input type="text" name="old" class="form-control">
    </div>
    <div class="form-group mb-3">
        <label for="">Kategorija</label>
        <select name="category" class="form-control" required>
                <option value="">Odaberi</option>
                <option value="B">B</option>
                <option value="C">C</option>

        </select>
    </div>
    <div class="form-group mb-3">
        <button type="submit" name="save_member" class="btn btn-primary">Add</button>
    </div>

</form>
```

Sl. 4.10. *Forma za dodavanje novog polaznika.*

4.8. Odabir Polaznika

Odabir polaznika autoškole omogućen je određenim registriranim korisnicima unutar autoškole. Kako bi prijavljen korisnik mogao birati polaznike mora mu biti dodjeljena jedna od uloga: nastavnik teorije, nastavnik prve pomoći ili instruktor vožnje. Prvi prijavljen korisnik koji dolazi na red je nastavnik teorije. Prikazuje mu se lista polaznika, no samo oni polaznici kojima već prethodno nije unesen nastavnik teorije. $uidNastavnikT = $_SESSION['verified_user_id'] dohvaća se ID prijavljenog nastavnika teorije te pritiskom na gumb „Prihvaćam“ šalju se preko forme svi podaci polaznika, uključujući njegov ID, te se šalje i $uidNastavnikT. Preko poslanog ID-a polaznika vrši se njegovo ažuriranje tako da mu za atribut „nastavnikT“ unesemo spremljeni ID prijavljenog nastavnika teorije. Nakon što se unese nastavnikov ID , ni jedan drugi nastanvik teorije ne može vidjeti tog polaznika te ga iz tog razloga ne može niti odabrati. Prikazivanje se ivršava ukoliko je polazniku za atributr „nastavnikT“ postavljena vrijednost ' '. Opciju odabir polaznika također imaju i nastavnik teorije i instruktor vožnje. Prikazivanje polaznika za nastanvika prve pomoći vrši se na temelju provjere je li polazniku potvrđen prolaz teorije od strane admina i ukoliko mu je polje „nastanvikPP“ prazno, prijavljeni nastavnik prve pomoći ga može odabrati. Instruktor vožnje da bi odabrao polaznika, polaznik mora imati

potvrđenu teoriju i prvu pomoć od strane admina, prazno polje „instruktorB“ ili „instruktorC“ ovisi za koju kategoriju je prijavljen korisnik i polaznik mora imati postavljenu pripadajuću kategoriju. Slika 4.11. prikazuje dohvaćanje polaznika iz baze, te ispisivanje samo onih polaznika kojima nije unesen nastavnik teorije. Slika 4.12. prikazuje izgled stranice za odabir polaznika.

```
$ref_table = 'polaznik';
$fetchdata = $database->getReference($ref_table)->getValue();
$uidNastavnikT =  $_SESSION['verified_user_id'];
$brojac = 0;
if($fetchdata > 0){
    $i=1;
    foreach($fetchdata as $key => $row)
    {
        if($row['nastavnikT'] == ""){
```

Sl. 4.12. *Prikaz polaznika koji kao vrijednost „nastavnikT“ imaju „ „.*

Popis polaznika

B.	Ime	Prezime	Godina	Kategorija	Nastavnik T	Prihvati
1	Gabrijel	Leh	21	B		Prihvacam
2	Ante	Antic	25	C		Prihvacam
3	Franciska	Rebic	22	B		Prihvacam

Sl. 4.11. Izgled stranice za odabir polaznika za nastavnika teorije

Slika 4.13 i 4.14 prikazuju POST metodu za unos nastavnika teorije polazniku.

```
if(isset($_POST['dodjela_nastavnikT']))
{
    $key = $_POST['key'];
    $ime = $_POST['ime'];
    $prezime = $_POST['prezime'];
    $godina= $_POST['godina'];
    $kategorija = $_POST['kategorija'];
    $nastavnikT = $_POST['nastavnikT'];
    $nastavnikPP = $_POST['nastavnikPP'];
    $instruktorB = $_POST['instruktorB'];
    $instruktorC = $_POST['instruktorC'];
    $komisija = $_POST['komisija'];
    $polozenaT = $_POST['polozenaT'];
    $izlazakT = $_POST['izlazakT'];
    $potvrdaT = $_POST['potvrdaT'];
    $polozenaPP = $_POST['polozenaPP'];
    $izlazakPP = $_POST['izlazakPP'];
    $potvrdaPP = $_POST['potvrdaPP'];
    $polozenaGV = $_POST['polozenaGV'];
    $izlazakGV = $_POST['izlazakGV'];
    $potvrdaGV = $_POST['potvrdaGV'];
    $terminGV = '';
    $komentar = '';
```

Sl. 4.13. *Dodjela nastavnika teorije 1.dio.*

```
    $updateData = [
        'ime' => $ime,
        'prezime' => $prezime,
        'godina' => $godina,
        'kategorija' => $kategorija,
        'nastavnikT' => $nastavnikT,
        'nastavnikPP' => $nastavnikPP,
        'instruktorB' => $instruktorB,
        'instruktorC' => $instruktorC,
        'komisija' => $komisija,
        'polozenaT' => $polozenaT,
        'izlazakT' => $izlazakT,
        'potvrdaT' => $potvrdaT,
        'polozenaPP' => $polozenaPP,
        'izlazakPP' => $izlazakPP,
        'potvrdaPP' => $potvrdaPP,
        'polozenaGV' => $polozenaGV,
        'izlazakGV' => $izlazakGV,
        'potvrdaGV' => $potvrdaGV,
        'terminGV' => $terminGV,
        'komentar' => $komentar,
    ];

    $ref_table = "polaznik/".$key;
    $updatequery = $database->getReference($ref_table)->update($updateData);

    header('Location: nastavnikTOdabir.php');
}
```

Sl. 4.14. *Dodjela nastavnika teorije, 2.dio*

4.9. Prikaz prihvačenih polaznika

Stranicu o prikazu prihvaćenih polaznika koriste nastavnik teorije, nastavnik prve pomoći i instruktor vožnje. Predstavlja jednostavnu tablicu koja prikazuje polaznike kojima je zadani parametar (npr. nastavnikT) postavljen kao id trenutnog prijavljenog korisnika. Unutar tablice postoji kolona Pregled, sastoji se od linka za svakog polaznika koja vodi na njegovu evidenciju, prikaže se na način da se unutar linka postavi polaznikov id. Na slici 4.15 nalazi se grafičko sučelja za instruktora vožnje i prikazuje prihvaćene polaznike.

Popis uzetih polaznika

B.	Ime	Prezime	Godina	Kategorija	PolozenaGV	Dolasci
1	Borna	Jovic	18	B	da	Pregled
2	Marin	Maric	23	B	da	Pregled
3	David	Biuk	23	B		Pregled

Sl. 4.15. *Prikaz prihvaćenih polaznika.*

4.10. Pregled dolazaka i položenog ispita

Stranica za evidenciju dolazaka i položenog ispita za uloge nastavnika teorije, nastavnika prve pomoći i instruktora sastoji se od 3 tablice. Jedna tablica prikazuje podatke o odabranom polazniku, druga prikazuje dolaske polaznika i komentare, a treća prikazuje podatke vezane za ispit. Korisnik s odgovarajućom ulogom može dodavati dolaske za polaznika. Kako bi polaznik teorije mogao izaći na ispit, mora prvo zadovoljiti uvjet od odslušanih 30 sati teorije, nakon toga se prikazuje gumb „Dodaj ispit“ u kojem nastavnik teorije unosi broj izlaska, bodove i je li polaznik osvario prolaz ili ne. Ukoliko polaznik ne uspije položiti ispit , polaznik može pohađati dodatnu nastavnu, koju nastavnik teorije evidentira. Ukoliko polaznik sljedeći put položi ispit, nastavnik to evidentira te se gumbovi „Dodaj dolazak“ i „Dodaj ispit“ više ne prikazuju. Nastavnik prve pomoći ima jednak prikaz, samo što je za unos ispita i njegovog rezultata potrebno odslušati minimalno 10 sati. Instruktor vožnje ima nešto drugačiji prikaz i unos ispita. Instruktor vožnje definira termin ispita polazniku, nakon što polaznik ispuni dolaznost od 35 sati. Polazniku se dodjeljuje datum, vrijeme i broj izlaska na glavnu vožnju. Nakon toga dodjeljuje

mu se komisija. Instruktor odabire jednu osobu iz liste registriranih korisnika kojima je uloga „komisija“ i toj osobi dodjeljuje ocjenjivanje polaznikove glavne vožnje. Slika 4.16. prikazuje formu za dodavanje dolazaka. Na slici 4.17. nalazi se kod pomoću kojeg se unosi dolazak.

```
<form action="code.php" method="POST">

    <div class="form-group mb-3">
        <label for="">Broj odlsušanih sati</label>
        <select name="brojSati" class="form-control" required>
                <option value="">Odaberi broj sati</option>
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
                <option value="6">6</option>
                <option value="7">7</option>
                <option value="10">10</option>
                <option value="12">12</option>
                <option value="15">15</option>
        </select>
    </div>
    <div class="form-group mb-3">
        <label for="">Komentar</label>
        <input type="text" name="komentar" class="form-control">
    </div>
    <?php
        $key_child = $_GET['id'];

    ?>
    <input type="hidden" name="polaznikuid" class="form-control" value=<?=$key_child;?>>
    <div class="form-group mb-3">
        <button type="submit" name="save_dolazakT" class="btn btn-primary">Add</button>
    </div>

</form>
```

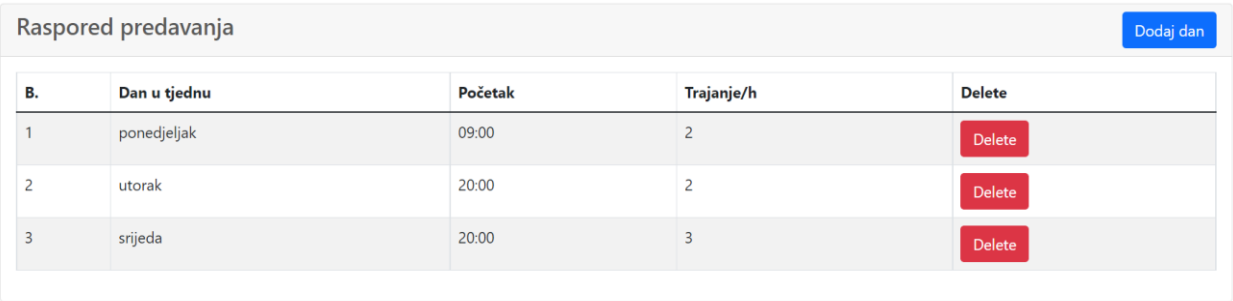
Sl. 4.16. *Prikaz forme za dodavanje dolaska za teoriju.*

```php
if(isset($_POST['save_dolazakT']))
{

    $brojSati = $_POST['brojSati'];
    $komentar = $_POST['komentar'];
    $polaznikuid = $_POST['polaznikuid'];

    $postData = [
        'brojSati' => $brojSati,
        'komentar' => $komentar,
        'polaznikuid' => $polaznikuid,
    ];

    $ref_table = "dolazakTeorija";
    $postRef_result = $database->getReference($ref_table)->push($postData);

    header('Location: dolasciTeorija.php?id='.$polaznikuid);


}
```

Sl. 4.17. *Unos dolaska na predavanje iz teorije.*

4.11. Prikaz Rasporeda

Stranicu prikaz rasporeda koriste korisnici s ulogama: nastavnik teorije, nastavnik prve pomoći i instruktor vožnje. Raspored predavanja za nastavnika teorije i prve pomoći je identičan. Postoji gumb „Dodaj dan“ u kojem se dodaje dan u tjednu između ponedjeljka i petka, definira se vrijeme početka predavanja za taj dan i predviđeno trajanje predavanja u satima. Ukoliko nastavnik teorije ili prve pomoći ne želi više npr. ponedjeljkom imati predavanja u 20:00h, on može obrisati ponedjeljak i dodati novi ponedjeljak s željenim terminom i trajanjem predavanja. Raspored za instruktora B ili C kategorije je osmišljen drugačije. Raspored se sastoji od stupaca: datum, pregled dana i obriši dan. Ispod stupaca „pregled dana“ i „obrisi dan“ nalazi se za svaki pojedini datum gumb za pregledati ili obrisati dan. Klikom na „obrisi dan“ šalje se id datuma kako bi se obrisao točno traženi datum. Klikom na „pregled dana“ otvara se nova stranica koja prikazuje tablicu s imenima stupaca. Stupci se sastoje od datuma i termina vožnje, počevši od 08:00 do 15:00h (termini su po jedan sat) i stupcem unesi polaznike s linkom koji vodi na unošenje polaznika. Ispod svakog sata je prikazano ime i prezime osobe koja ima termin vožnje postavljen u to vrijeme. Ukoliko nije nitko unesen za određeno vrijeme, polje za to vrijeme će biti prazno. Klikom na unesi polaznike otvara se nova php stranica u kojoj se ispisuje datum i sati kao nazivi stupaca, te se ispisuju svi prihvaćeni polaznici instruktora koji još nisu položili glavnu vožnju. Ispod sata u tablici, ukoliko nitko nije već upisan za to vrijeme nalazit će se gumb „add“ koji unosi za taj određeni sat polaznika koji se nalazi u redu tablice u kojoj je pritisnut gumb „add“. Ukoliko je već netko rezerviran za određeni sat, sva polja ispod stupca za sat će biti prazna te neće postojati gumb za unos. Grafički prikaz za nastavnika teorije i prve pomoći na slici 4.18.



Sl. 4.18. *Raspored predavanja za nastavnika teorije i prve pomoći*

Na slici 4.19. je prikaz rasporeda za određen datum instruktora.If se koristi za provjeru svakog pojedinog sata.

```
$uidInstruktorB =  $_SESSION['verified_user_id'];

if(isset($_GET['id'])){
    $i=1;
    $key_child = $_GET['id'];
    $ref_table = 'rasporedB';
    $polaznik = 'polaznik';
    $getdata = $database->getReference($ref_table)->getChild($key_child)->getValue();

    if($getdata > 0){
        ?>
        <td>
            <?=$i;?>
        </td>
        <td>
            <?=$getdata['datum'];?>
        </td>
        <td>
        <?php
            if($getdata['sat1'] == ''){
        ?>
            <?=$getdata['sat1'];?>
        <?php
        }else{
            $id = $getdata['sat1'];
            $naziv = $database->getReference($polaznik)->getChild($id)->getValue();
            ?>
                <?=$naziv['ime'];?> <?=$naziv['prezime'];?>
            <?php
        }
        ?>
        </td>
```

Sl. 4.19. *Prikaz rasporeda za određeni datum instruktora.*

Slika 4.20. prikazuje formu za unos polaznika za određeni termin prilikom definiranja dnevnog rasporeda vožnje instruktora.

```
<td>
    <?php
        if($getdata['sat1'] == ''){
    ?>
    <form action="Bcode.php" method="POST">
        <input type="hidden" id="key" name="key" value=<?=$key_dolazak;?>>
        <input type="hidden" name="instruktorB" class="form-control" value=<?=$uidInstruktorB?>>
        <input type="hidden" name="datum" class="form-control" value=<?=$getdata['datum'];?>>
        <input type="hidden" name="sat1" class="form-control" value=<?=$key?>>
        <input type="hidden" name="sat2" class="form-control" value=<?=$getdata['sat2'];?>>
        <input type="hidden" name="sat3" class="form-control" value=<?=$getdata['sat3'];?>>
        <input type="hidden" name="sat4" class="form-control" value=<?=$getdata['sat4'];?>>
        <input type="hidden" name="sat5" class="form-control" value=<?=$getdata['sat5'];?>>
        <input type="hidden" name="sat6" class="form-control" value=<?=$getdata['sat6'];?>>
        <input type="hidden" name="sat7" class="form-control" value=<?=$getdata['sat7'];?>>
        <input type="hidden" name="sat8" class="form-control" value=<?=$getdata['sat8'];?>>
        <button type="submit" name="save_sat" class="btn btn-primary">Add</button>
    </form>
    <?php
        }
    ?>

</td>
```

Sl. 4.20. *Forma za unos polaznika u slobodni termin u rasporedu.*

Slika 4.21. prikazuje spremanje polaznika za određeni termin u raspored za B kategoriju.

```php
if(isset($_POST['save_sat']))
{
    $key = $_POST['key'];
    $instruktorB = $_POST['instruktorB'];
    $datum = $_POST['datum'];
    $sat1 = $_POST['sat1'];
    $sat2 = $_POST['sat2'];
    $sat3 = $_POST['sat3'];
    $sat4 = $_POST['sat4'];
    $sat5 = $_POST['sat5'];
    $sat6 = $_POST['sat6'];
    $sat7 = $_POST['sat7'];
    $sat8 = $_POST['sat8'];

    $updateData = [
        'instruktorB' => $instruktorB,
        'datum' => $datum,
        'sat1' => $sat1,
        'sat2' => $sat2,
        'sat3' => $sat3,
        'sat4' => $sat4,
        'sat5' => $sat5,
        'sat6' => $sat6,
        'sat7' => $sat7,
        'sat8' => $sat8,
    ];

    $ref_table_polaznik = "rasporedB/".$key;
    $updatequery = $database->getReference($ref_table_polaznik)->update($updateData);

    header('Location: BUnosRaspored.php?id='.$key);
}
```

Sl. 4.21. *POST metoda za unos polaznika u raspored B kategorije.*

4.12. Komisija Pregled

Komisija predstavlja korisnika s dodjeljenom ulogom komisija, prikazuju mu se polaznici koji su položili teoriju i prvu pomoć i imaju potvrdu admina za to. Korisniku se prikazuju samo oni polaznici koji su mu dodjeljeni od strane instruktora za glavnu vožnju. Tablica prikazuje podatke vezane za korisnika i termin glavne vožnje i gumb „Dodaj rezultat“. Klikom na dodaj rezultat otvara se forma za unos prolaza ili pada glavne vožnje i komentara na glavnu vožnju. Ukoliko je prolaz s tim polaznikom nema više nikakve interakcije. Ako se dogodi da polaznik padne, upisuje mu se u prolaznost ne. Instruktor mu dodjeljuje novi termin glavne vožnje te će onda komisija moći ponovno unijeti rezultate vožnje. Slika 4.22. prikazuje uvjet za pojavljivanje gumba za unos rezultata glavne vožnje.

```
<td><?php
    if($row['komentar'] == '' && ($row['polozenaGV'] == ''|| $row['polozenaGV'] == 'ne')){
    ?>
    <a href="KaddVoznja.php?id=<?=$key?>" class="btn btn-primary">Dodaj Rezultat</a>
    <?php
    }
    ?>
</td>
```

Sl. 4.22. *Pojavljivanje gumba za unos rezultata glavne vožnje*

4.13. Uspjeh polaznika

Stranici Uspjeh polaznika može pristupiti samo korisnik aplikacije s ulogom admin. Uspjeh polaznika prikazuje evidenciju polaznika s dobivenom vozačkom dozvolom. Sortiranje za prikaz se vrši na način da se kroz foreach petlju provjeravaju podaci polaznika i jesu li mu potvrđeni prolazi iz teorije, prve pomoći i glavne vožnje. Stranica prikazuje podatke polaznika kao što su ime, prezime, godina, kategorija, redni broj izlaska koji mu je bio potreban da bi prošao određeni ispit i nastavnike koji su ga podučavali u procesu dolaska do vozačke dozvole.

5. ZAKLJUČAK

Web aplikacija upravljanje autoškolom predstavlja aplikaciju koja prati polaznikove korake od početka upisa u autoškolu do samog završetka polaganja autoškole. Napravljen je prikaz 3 slična riješenja, odnosno aplikacija s istom tematikom. Aplikacija je izrađena pomoću tehnologija koje se većinom, gotovo svaka od njih, koriste za izradu web aplikacija i stranica. Temelj za izradu „kostura“ aplikacije je HTML koji je korišten za glavne dijelove. Za uređenje dizajna korišteni su CSS i Bootstrap 5 kako bi stranica izgledala ljepše i preglednije korisniku. Svi podaci koji su uneseni korištenjem aplikacije spremljeni su na firebasu. U aplikaciji najviše je korišten PHP. Radnje i mogućnosti aplikacije su detaljno opisani, kao i tehnologija korištena pri izradi. Aplikacija se sastoji od 6 različitih uloga. Svaka uloga ima svoje mogućnosti, ali i ograničenja. Prijavljeni korisnici s postavljenom ulogom vode evidenciju polaznika, njihove dolaske i uspjeh kroz polaganje ispita. Izradom ove aplikacije postigao se cilj prikaza i kontroliranja dolaznosti i uspjeha polaznika na putu do vozačke dozvole. Poboljšanja ove aplikacije su moguća na način da se doda uloga ocjenjivača teorijiskog ispita, te bi osoba s ulogom ocjenjivača teorijskog ispita mogla unositi komentare i rezultate ispita iz teorije. Izradom aplikacije za upravljanje autoškolom može se zaključiti da je to jedna kompleksna i velika baza podataka koja olakšava pregled polaznika i njihovog uspjeha.

LITERATURA

[1] https://autoskola.hr/ 05.06.2022.

[2] https://totaldrive.co.uk/ 16.06.2022.

[3] https://driving-school.app/public 16.06.2022.

[4] https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html 16.06.2022.

[5] https://www.loc.gov/preservation/digital/formats/fdd/fdd000475.shtml 16.06.2022.

[6] http://www.webtech.com.hr/html.php 16.06.2022.

[7] https://www.w3.org/Style/CSS/Overview.en.html 16.06.2022.

[8] http://www.webtech.com.hr/html.php 16.06.2022.

[9] https://www.bu.edu/lernet/artemis/years/2020/projects/FinalPresentations/HTML/historyofcss.html 16.07.2022.

[10] https://getbootstrap.com/docs/5.2/getting-started/introduction/ 17.06.2022.

[11] https://getbootstrap.com/docs/5.2/about/overview/ 17.06.2022.

[12] https://developer.mozilla.org/en-US/docs/Web/JavaScript 17.06.2022.

[13] https://www.springboard.com/blog/data-science/history-of-javascript/ 17.06.2022.

[14] https://www.w3schools.com/js/js_whereto.asp 17.06.2022.

[15] https://educationecosystem.com/guides/programming/php/history 18.06.2022.

[16] https://www.w3schools.com/php/php_intro.asp 18.06.2022.

[17] https://code.visualstudio.com/docs 19.06.2022.

[18] https://firebase.google.com/ 19.06.2022.

[19] https://firebase.google.com/support/guides/security-checklist 19.06.2022.

[20] https://firebase.google.com/docs/engage 19.06.2022.

SAŽETAK

Web aplikacija pod nazivom Upravljanje autoškolom izrađena je pomoću tehonologija: HTML, CSS, Bootstrap 5, JavaScript , Php i Firebase. Ovisno o postavljenoj ulozi registriranog korisnika, mogu se izvršavati različite radnje i to su: unos polaznika, unos vozila, odabir polaznika, kreiranje rasporeda, dodavanje sati, prijavljivanje ispita i dodavanje rezultata. Podaci se mogu brisati i ažurirati u ovisnosti o ulozi prijavljenog korisnika.

Ključne riječi: autoškola, baza podataka, upravljanje i vođenje, web aplikacija

ABSTRACT

Title: Driving school management application

The web application called Driving School Management was created using the following technologies: HTML, CSS, Bootstrap 5, JavaScript, PHP and Firebase. Depending on the set role of the registered user, different actions can be performed and those are: entry of participants, entry of vehicles, selection of participants, creation of schedules, addition of hours, registration of exams and addition of results. Data can be deleted and updated depending on the role of the logged-in user.

Keywords: dana base, driving school, management, web application