# Web aplikacija za evidenciju utrka Formule 1

**Šarčević, Luka**

**Undergraduate thesis / Završni rad**

**2021**

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

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:200:774782

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-05-12**

*Repository / Repozitorij:*

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

**Sveučilišni studij**

# Web aplikacija za evidenciju utrka Formule 1

**Završni rad**

**Luka Šarčević**

**Osijek,2021.**

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

**Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

**Osijek, 25.08.2021.**

**Odboru za završne i diplomske ispite**

# Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

| | |
|---|---|
| **Ime i prezime studenta:** | Luka Šarčević |
| **Studij, smjer:** | Preddiplomski sveučilišni studij Računarstvo |
| **Mat. br. studenta, godina upisa:** | R4276, 26.07.2018. |
| **OIB studenta:** | 09812877327 |
| **Mentor:** | Izv. prof. dr. sc. Ivica Lukić |
| **Sumentor:** | |
| **Sumentor iz tvrtke:** | |
| **Naslov završnog rada:** | Web aplikacija za evidenciju utrka Formule 1 |
| **Znanstvena grana rada:** | **Informacijski sustavi (zn. polje računarstvo)** |
| **Predložena ocjena završnog rada:** | Izvrstan (5) |
| **Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:** | Primjena znanja stečenih na fakultetu: 3 bod/boda<br>Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda<br>Jasnoća pismenog izražavanja: 3 bod/boda<br>Razina samostalnosti: 3 razina |
| **Datum prijedloga ocjene mentora:** | 25.08.2021. |
| **Datum potvrde ocjene Odbora:** | 08.09.2021. |

| | |
|---|---|
| Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija: | Potpis: |
| | Datum: |

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

# IZJAVA O ORIGINALNOSTI RADA

**Osijek, 08.09.2021.**

| | |
|---|---|
| **Ime i prezime studenta:** | Luka Šarčević |
| **Studij**: | Preddiplomski sveučilišni studij Računarstvo |
| **Mat. br. studenta, godina upisa:** | R4276, 26.07.2018. |
| **Turnitin podudaranje [%]:** | 8% |

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za evidenciju utrka Formule 1**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Ivica Lukić

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. POSTOJEĆA RJEŠENJA ..... 2
2.1 Službena stranica Formule 1 ..... 2
2.2 Internet stranica BBC.co.uk ..... 3
2.3 Internet stranica ESPN.com ..... 3
2.4 Internet stranica f1calendar.com ..... 4
2.5 Internet stranica racefans.net ..... 5
3. TEHNOLOGIJE KORIŠTENE ZA IZRADU ..... 6
3.1 Visual Studio Code ..... 6
3.2 HTML ..... 6
3.3 CSS ..... 7
3.4 Bootstrap ..... 8
3.5 XAMPP ..... 9
3.5.1 PHP ..... 9
3.5.2 MySQL ..... 10
4. IZRADA APLIKACIJE ..... 11
4.1 Izrada baze podataka ..... 11
4.1.1 Opis potreba baze podataka ..... 11
4.1.2 Definiranje entiteta i atributa baze podataka ..... 11
4.1.3 Konceptualno oblikovanje ..... 12
4.1.4 Fizičko oblikovanje ..... 12
4.2 Izrada korisničkog sučelja aplikacije ..... 14
4.3 Izrada serverskog sučelja ..... 20
5. PREGLED ZAVRŠENE APLIKACIJE ..... 26
6. ZAKLJUČAK ..... 34

LITERATURA ..... 35
SAŽETAK ..... 37
ABSTRACT ..... 38
ŽIVOTOPIS ..... 39
PRILOZI ..... 40

# 1. UVOD

Užurbani raspored Formule 1 za neke obožavatelje može biti zahtjevan za praćenje. Iako se neke utrke održavaju u razmaku od tri tjedna, svaka utrka se održava u drugoj državi što zna biti zbunjujuće. Zbog tog razloga postoji potražnja za intuitivnim rasporedom utrka, podacima o utrkama, stazama na kojima se utrkuje, vozačima i timovima, te o poretku timova i vozača u pojedinom dijelu sezone. Sve te informacije teško je naći na jednom mjestu.

Cilj završnog rada je realizirati tu ideju, tj. napraviti web aplikaciju koja će prikazati sve utrke koje se odvijaju u sezoni Formule 1, najbitnije informacije o stazama na kojima se utrka odvija, timovima i vozačima tih timova, te poredak timova i vozača u dijelu sezone kada taj poredak zanima posjetitelja stranice.

U drugome poglavlju prikazana su slična rješenja koja također rješavaju zadani problem. U trećem poglavlju prikazane su tehnologije korištene za izradu aplikacije. U četvrtom poglavlju obuhvaćen je postupak stvaranja aplikacije koristeći navedene tehnologije. Peto poglavlje prikazuje gotovu aplikaciju i njene mogućnosti. Zadnje poglavlje predstavlja zaključak završnog rada u kojemu se opisuju dobiveni rezultati.

## 1.1 Zadatak završnog rada

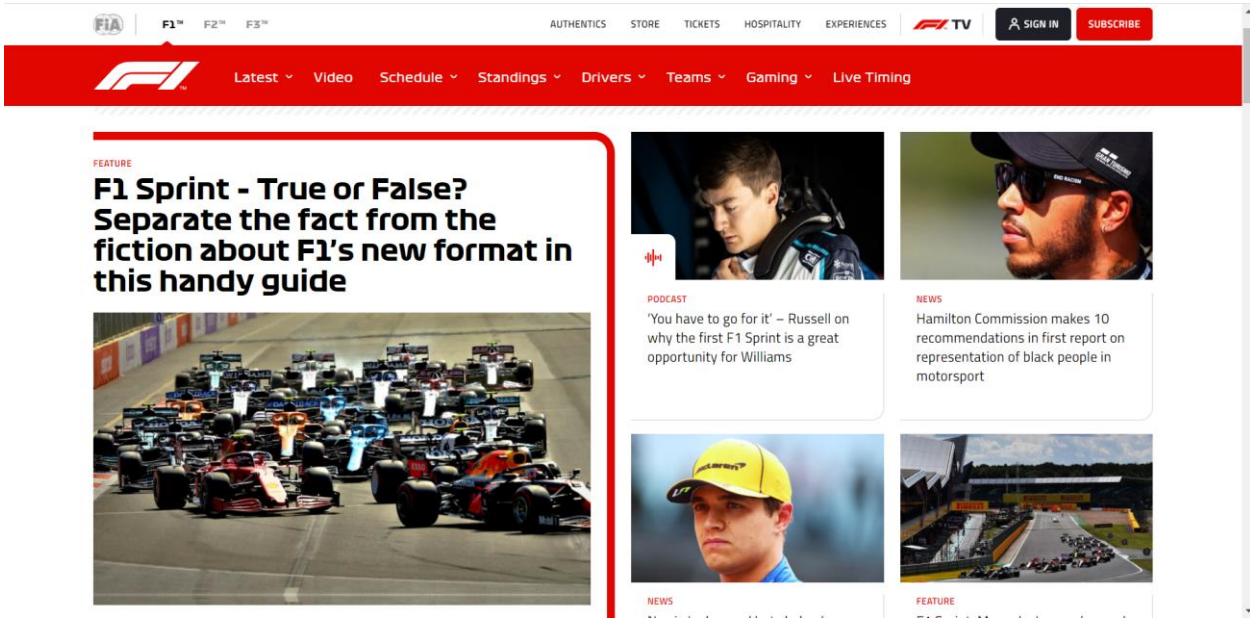
Zadatak završnog rada je napraviti web aplikaciju za evidenciju Formula 1 utrka, a uz to i evidenciju cijele Formula 1 sezone jer osim utrka evidentiraju se timovi, vozači za pojedine timove i poredak timova i vozača.

# 2. POSTOJEĆA RJEŠENJA

S obzirom na popularnost i opseg Formule 1, postoje rješenja za evidenciju utrka i poretka vozača i timova. U nastavku je nabrojano pet najpoznatijih rješenja.

## 2.1 Službena stranica Formule 1

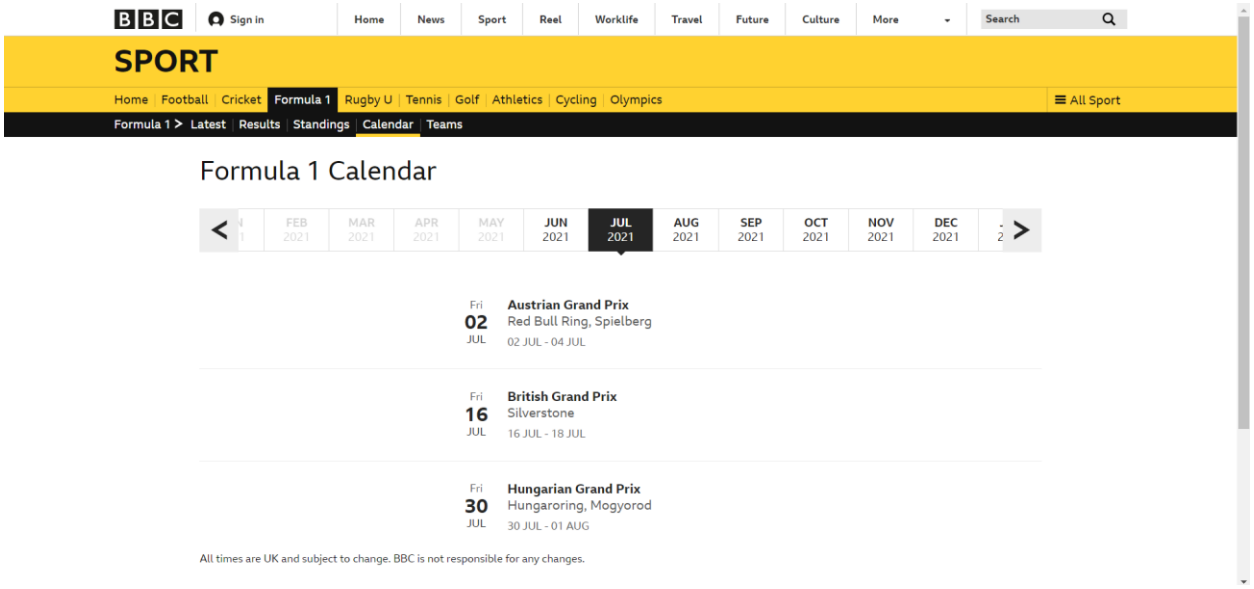
Kao što je prikazano na slici 2.1. službena stranica Formule 1 prvo je i glavno odredište za ljubitelje Formule 1. Stranica je puna informacija za one koji žele saznati novosti o sportu, mijenjanju rasporeda utrka, potpisivanju novih ugovora i slično. Obilje informacija je i negativna strana stranice jer su te informacije važne samo najvećim obožavateljima sporta. Za povremenog gledatelja utrka potrebna je jednostavnija stranica bez pretrpavanja početne stranice medijskim člancima. Stranica treba biti intuitivna za korištenje, što službena stranica Formule 1 nije. Ako se želi provjeriti tablični poredak timova ili vozača u sezoni se treba stisnuti „Standings“ gumb, ali nova stranica ne prikazuje ukupni poredak, nego poredak po pojedinoj utrci. Za ukupni poredak potreban je dodatni stisak gumba „Drivers“ ili „Teams“ ako se želi pogledati njihov poredak, što je nepotrebno.




Sl. 2.1. Početna stranica službene stranice Formule 1

## 2.2 Internet stranica BBC.co.uk

BBC-eva stranica za raspored utrka Formule 1 prikazana je na slici 2.2. BBC-eva internet stranica također nudi informacije o svemu što je vezano za sport Formule 1 ali ne guši posjetitelja sa medijskim člancima, nego odmah učitava kalendar utrka. Negativna strana BBC-evog kalendara

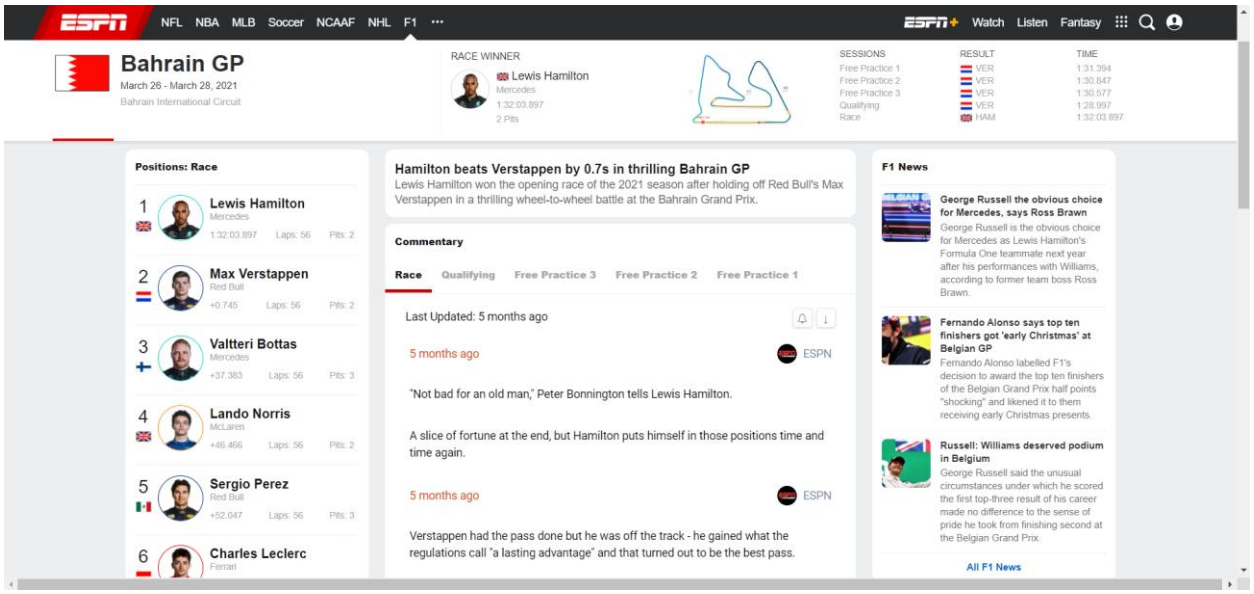



Sl. 2.2. Početna stranica BBC.co.uk

je što utrke koje su završene nisu prikazane. Ako se želi saznati rezultate pojedine utrke koja je završena, treba se kliknuti na gumb „Results“ te onda na pojedinu utrku o kojoj posjetitelj želi više saznati. Stranica također ne nudi detaljne informacije o timovima, vozačima i stazama na kojima se utrke održavaju.

## 2.3 Internet stranica ESPN.com

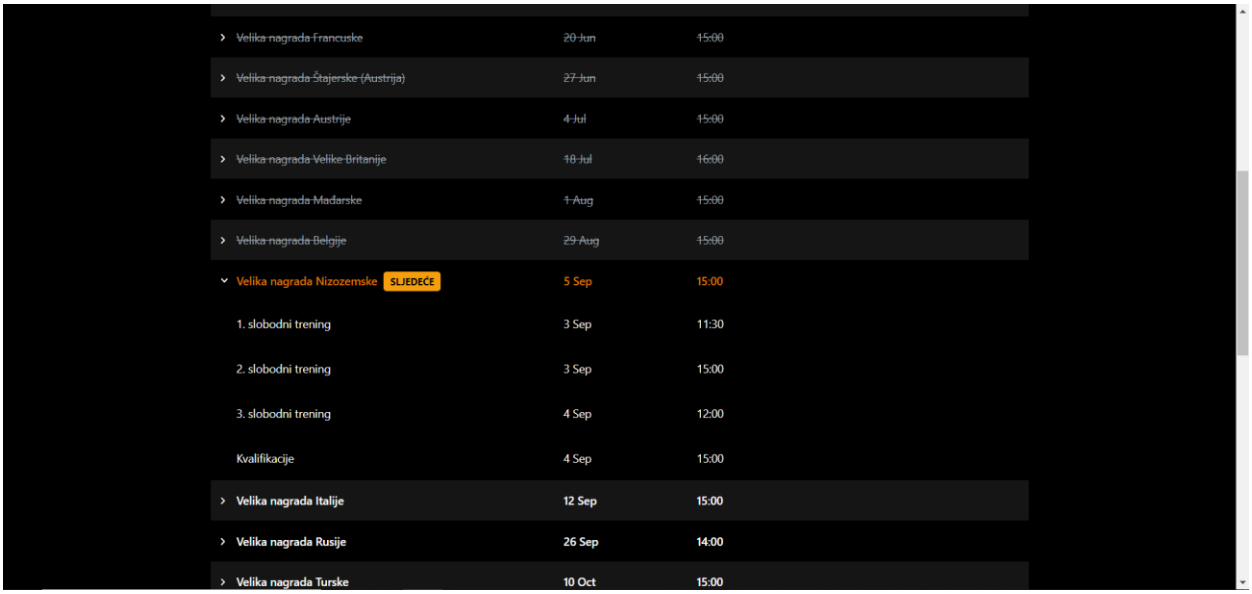
ESPN-ova stranica nudi više informacija od BBC-eve ali manje od službene stranice Formule 1. Za razliku od BBC-eve stranice, ESPN-ova nudi informacije o utrkama koje su završene, a sličnost između dvije stranice je da također ne guši posjetitelja sa medijskim člancima. Velika mana stranice je što rezultate svih sjednica prikazuje na neintuitivan način kao što je prikazano na slici 2.3. Poredak je zguran u lijevi kut, dok su u sredinu smješteni komentari ESPN-ovih novinara. Poredak bi trebao biti u sredini ekrana gdje pogled posjetitelja odmah pada pri učitavanju stranice.




Sl. 2.3. Podaci o pojedinoj utrci na stranici ESPN.com/f1

## 2.4 Internet stranica f1calendar.com

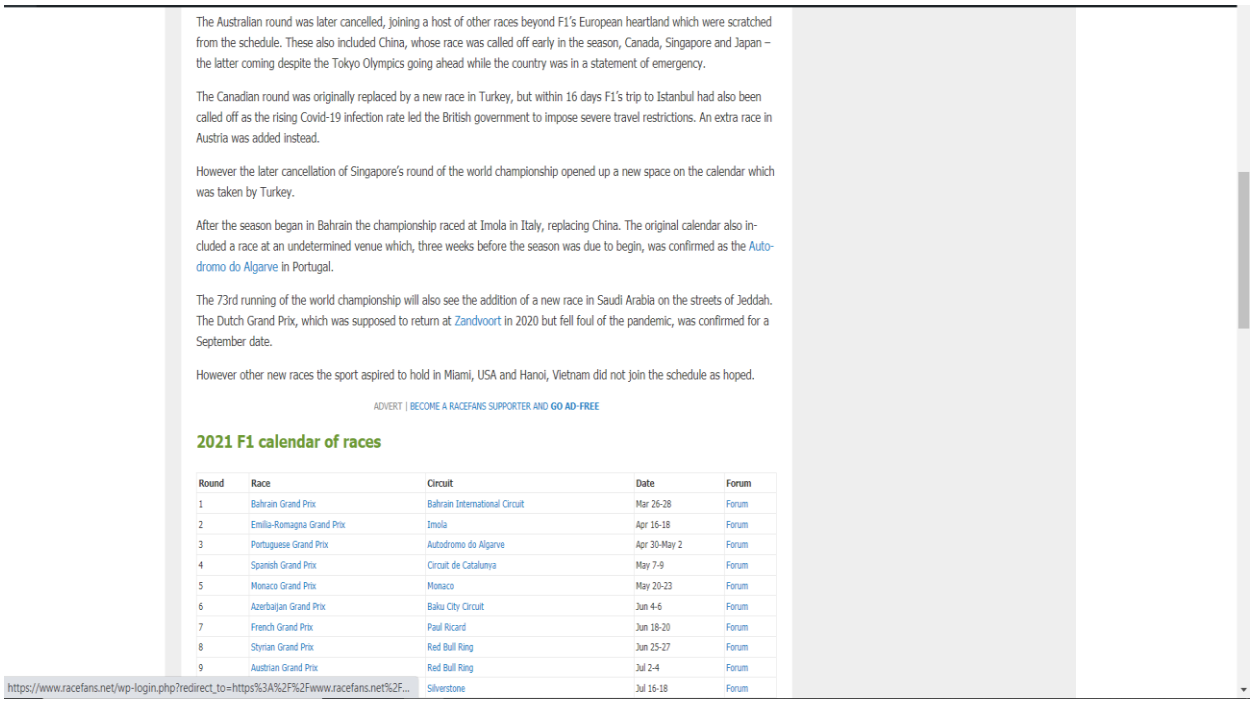
Internet stranica f1calendar.com je vrlo jednostavna stranica koja prikazuje samo nadolazeće utrke Formule 1 kao što je prikazano na slici 2.4. Stranica ne nudi informacije o utrkama koje su završene. Također, na stranici ne postoje informacije o timovima i vozačima koji sudjeluju u utrkama. Stranica samo prikazuje ime utrke i datume kada će se pojedina sjednica u 'rundi' dogoditi. Jedna velika prednost ove stranice je što je prevedena na hrvatski jezik. Osim toga, ne postoji razlog zašto odabrati ovu stranice nad već navedenim stranicama.




Sl. 2.4. Početna stranica f1calendar.com

## 2.5 Internet stranica racefans.net

Internet stranica racefans.net, slično kao si službena stranica Formule 1, pri ulasku u stranicu prikazuje medijske članke, što je povremenom gledatelju nepotrebno. Stranica nije jednostavnija za korištenje. Klikom na gumb „Calendar“ raspored utrka nije postavljen na vrh stranice, nego tekst koji objašnjava što je u kalendaru, a nakon toga je napokon prikazana tablica sa rasporedom utrka, kao što se vidi na slici 2.5. Informacije iznad tablice su nepotrebne jer se sve može saznati iz nje. Klikom na pojedinu utrku u tablici prikazuju se vremena sjednica u 'rundi', ali se ne prikazuju rezultate tih sjednica. Manjak važnih informacija o sjednicama u 'rundi' poništava razlog posjeta ovoj stranici.




Sl. 2.5. Prikaz nebitnih informacija iznad tablice rasporeda utrka na stranici racefans.net

# 3. TEHNOLOGIJE KORIŠTENE ZA IZRADU

Svaku aplikaciju koja prikazuje veliku količinu informacija koje se konstantno mijenjaju poželjno je napraviti pomoću baze podataka i programa koji komunicira s bazom podataka i automatski ih mijenja u aplikaciji kada se promijene u bazi podataka. Za izradu pristupnog sučelja (engl. *front-end*) aplikacije, tj. djela s kojim korisnik ima interakciju, korišten je HTML, CSS i Bootstrap. Za pozadinski sustav (engl. *back-end*) aplikacije tj. interakciju baze podataka i aplikacije koja je korisniku nevidljiva, odnosno nepoznata, korišteni su XAMPP i PHP. Sav kôd pristupnog sučelja aplikacije i dio pozadinskog sustava koji je napisan u PHP-u napisani su u Visual Studio Code-u.

## 3.1 Visual Studio Code

Visual Studio Code besplatni je uređivač izvornog kôda kojeg je proizveo Microsoft. Visual Studio Code nudi ugrađenu podršku za JavaScript, TypeScript i Node.js. Također omogućuje veliki broj proširenja za druge jezike kao što su C++, Python, PHP, C# i slično. Osim proširenja, omogućuje brzu navigaciju kroz program i njegovo uređivanje, refaktoriranje i ispravak grešaka u programu te njegovo nadopunjavanje i davanje savjeta pri pisanju programa. [1]

## 3.2 HTML

HTML (engl. *Hyper Text Markup Language*) najkorišteniji je jezik za označavanje koji se koristi za izradu internet stranica. Najkorišteniji je zato jer je besplatan, jednostavan za korištenje i lako se pokreće na bilo kojem uređaju.

HTML koristi oznake (engl. *tag*) i atribute koji internet pregledniku govore kako je stranica strukturirana. [2] Oznake se koriste za pozicioniranje elemenata na stranici. Kako bi oznaka funkcionirala ona se treba otvoriti i zatvoriti. Oznaka se otvara sa znakovima 'manje' i 'veće' u kojima se nalazi ime oznake, a zatvara se istom oznakom ali je ispred imena oznake kosa crta. Otvorena i zatvorena oznaka čini jedan element u HTML dokumentu. U HTML-u postoje 'block' i 'inline' elementi. 'Block' elementi uvijek počinju u novom redu, dok 'inline' elementi zauzimaju onoliko mjesta na ekranu koliko u njima ima sadržaja. Atributi se koriste za dodavanje dodatnih informacija elementima. Svi elementi mogu imati atribute. Atributi se uvijek dodjeljuju u oznaci za otvaranje elementa. Primjerice, u oznaku <img> treba se postaviti atribut *src* i njegova

vrijednost označava izvor slike te atribut *alt* koji u slučaju da preglednik nije u mogućnosti prikazati sliku daje tekst za opis slike. [3]

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Naslov za dokukent</title>
</head>
<body>
    <h1>Dobrodošli!</h1>
    <p>Ovo je HTML primjer.</p>
</body>
</html>
```

Sl. 3.1. Primjer HTML dokumenta

Na slici 3.1.. vidi se da svaki HTML dokument počinje sa oznakom <!DOCTYPE html> koji se navodi jednom prije svih oznaka. Ova oznaka govori da se radi o HTML5 dokumentu. Nakon <!DOCTYPE html> slijedi <html> element. On sadrži cijeli sadržaj stranice i svi elementi trebaju biti unutar njega. Element <html> se dalje dijeli na <head> i <body> elemente. Element <head> služi kao kontejner za sve što se uključuje u dokument. To može biti naslov, poveznica na CSS dokument i slično. U <body> element se stavlja elemente koji će se prikazati na internet stranici. Na slici 3.1. je prikazano da se u tijelu dokumenta nalaze jedan naslov i jedan odlomak (engl. *paragraph*). [4]

## 3.3 CSS

CSS (engl. *Cascading Style Sheets*) je jezik kojemu je cilj pojednostaviti uređivanje HTML dokumenata. Pomoću CSS-a HTML elementi se mogu brzo i jednostavno rasporediti na stranici, promijeniti boju i veličinu teksta, vrstu fonta i slično. [5] Prednosti CSS-a nad stiliziranjem direktno u HTML-u je što se .css dokumenti mogu odvojiti od .html dokumenta tj. lakše je za održavanje i izmjenu stilova, kompatibilno je sa raznim uređajima i jednu .css datoteku se može povezati u više .html datoteka. Mana vanjskog CSS-a je što je rad klijenta usporen jer se mora generirati novi HTTP zahtjev.

Stilska pravila CSS-a sastoje se od dva dijela: selektora (engl. *selector*) i deklaracije (engl. *declaration*). Selektor predstavlja element u HTML-u koji želimo stilizirati, a u deklaraciji se

stiliziraju odabrani element. Selektori mogu biti HTML elementi, klase, *id*-evi i slično. Deklaracija se piše unutar vitičastih zagrada. Najvažniji tipovi selektora su *element* selektori koji označava pojedini HTML element, *class* selektor koji označava više elemenata koji imaju isto ime klase, *id* selektor koji označava jedinstveni element u dokumentu te univerzalni selektor koji se označava sa '*'. Univerzalni selektor odabire sve elemente u dokumentu i primjenjuje pravila za uređivanje. [6]

```
1   *{
2       padding: 30px;
3   }
4
5   .container{
6       margin: 0px;
7   }
8
9   #jedinstveni:hover{
10      color: brown;
11  }
12
13  table{
14      border: 1px solid brown;
15  }
```

Sl. 3.2. Primjer CSS dokumenta

Na slici 3.2. vide se sva četiri navedena selektora. Prvi (univerzalni) na sve elemente unutar .html dokumenta stavlja *padding* na 30 piksela, drugi tj. klasni marginu stavlja na 0 piksela na sve elemente koji imaju klasu 'container'. Klasni selektor započinje sa točkom pa nakon točke slijedi ime klase. Treći selektor je *id* selektor. On započinje sa ljestvicama, nakon čega slijedi ime atributa *id* u elementu. On sadrži i pseudo selektor *:hover*, koji ne odabire element, nego određeni dio elementa ili element u određenom kontekstu. Kada se mišem prijeđe preko tog elementa, pseudo selektor mijenja boju elementa u smeđu kojemu atribut *id* ima vrijednost „jedinstveni“. Zadnje, imamo selektor koji odabire sve tablice u HTML dokumentu i postavlja crni rub na tablicu širine 1 piksel.

## 3.4 Bootstrap

Bootstrap je jedan od najpopularnijih razvojnih okvira za jednostavan dizajn korisničkog sučelja (engl. *front-end framework*). Koristan je za brzi dizajn *responzivnih* stranica i sadrži gotove

komponente koje se koriste za izgradnju funkcionalne aplikacije kao što su gumbi, navigacijske trake, prezentaciju slika i slično. Njega sačinjavaju CSS i JavaScript datoteke. [7] *Responzivna* web stranica je stranica koje se automatski prilagođava veličini ekrana na kojemu se prikazuje.

## 3.5 XAMPP

XAMPP (engl. *Cross platform – X, Apache – A, MySQL – M, PHP – P, Pearl - P*) je najpopularniji, besplatan paket softvera (engl. *software stack*) za jednostavno spajanje baze podataka i aplikacije. Izradili su ga Apache Friends. [8] XAMPP nudi Apache server na kojemu programeri mogu lagano testirati baze podataka i aplikacije koje o tim bazama podataka ovise. [9] PhpMyAdmin je alat koji je dio XAMPP-a i koristi se za administraciju MySQL-a te nudi intuitivan prikaz baze podataka.

### 3.5.1 PHP

PHP (engl. *PHP:Hypertext Preprocessor*) je skriptni programski jezik nalik C-u koji se koristi za razvoj dinamičnih internet aplikacija. Pomoću PHP-a razvija se serversko sučelje (engl. *back-end*) aplikacije koje omogućuje prikaz podataka iz baze podataka u aplikaciji. Pomoću PHP-a, bazi podataka mogu se slati SQL upiti, manipulirati dobiveni podaci i što je najbitnije .php datoteka prepoznaje HTML i može ga generirati što ga čini sjajnim za razvoj internet aplikacija. PHP dokument započinje sa oznakom <?php i završava sa oznakom ?>. Između tih oznaka nalazi se skriptni kôd. Izvan tih oznaka može se pisati HTML kôd s kojim se stvara stranica na poslužitelju prije nego se pošalje klijentu. Pomoću oznaka <?php i ?> poslužitelju se govori da se unutar njih nalazi PHP program koji se mora izvršiti. Ako unutar tih oznaka postoji HTML kôd, on se klijentu šalje nakon izvršavanja. PHP je široko korišten u svijetu jer je nalik C-u, lako proširiv, besplatan i slično. Sve varijable u PHP-u označavaju se sa znakom '$' ispred imena varijable. Slika 3.3.

```php
<?php
    function connectToDB (){
        $conn = mysqli_connect("localhost","root","","formula1");
        if($conn->connect_error){
            die("Connection failed: " . $conn->connect_error);
        }
        return $conn;
    }
?>
```

Sl. 3.3. Primjer PHP programa

prikazuje jednostavan PHP program. Definirana je funkcija *connectToDB()* koja se spaja na bazu podataka pomoću ugrađene funkcije *mysqli_connect()* i ako je spajanje neuspješno, pomoću funkcije *die()* ispisuje „Connection failed“ i izlazi iz .php datoteke.

### 3.5.2 MySQL

MySQL je besplatan sustav za upravljanje relacijskim bazama podataka koji je ugrađen unutar XAMPP okruženja. [11]

Tablice se u relacijskim bazama podataka nazivaju relacije. Tablice se sastoje od redova i stupaca. Atribut je ime za pojedini stupac dok je n-torka jedan redak tablice, tj. jedna vrijednost od svakog atributa u redu. MySQL koristi standardni SQL (engl. *Structured Query Language*) za stvaranje, brisanje i manipuliranje tablica i podataka.

```
1 SELECT name AS 'Driver name'
2 FROM driver
```

Sl. 3.4. Primjer SQL upita

Slika 3.4. prikazuje jednostavan SQL upit. Naredba SELECT dohvaća podatke iz tablice. Poslije naredbe SELECT određuje se koje podatke se želi odabrati iz pojedine tablice. To su uglavnom atributi iz tablica, ali unutar SELECT naredbe može biti još jedna SELECT naredba i slično. Odabran je atribut '*name*' i u rezultatu su dobiveni podaci predstavljeni pod imenom „*Driver name*“ pomoću ključne riječi AS. Poslije toga trebamo naredbom FROM odabrati iz koje tablice se podaci dohvaćaju, u ovom slučaju to je tablica '*driver*'.

# 4. IZRADA APLIKACIJE

U slijedećim potpoglavljima prikazani su koraci izrade aplikacije. Prvo će biti prikazana izrada baze podataka, nakon toga izrada korisničkog sučelja, a na kraju izrada serverskog sučelja aplikacije.

## 4.1 Izrada baze podataka

Kako bi se izradila baza podataka potrebno je saznati što baza podataka obuhvaća.

### 4.1.1 Opis potreba baze podataka

Sezona Formule 1 dijeli se na tzv. 'runde' koje se odvijaju vikendom. Jedna sezona sadrži 23 'runde'. Jedna runda sadrži tri sjednice vježbanja, jednu sjednicu kvalifikacija, te jednu utrku koja se još naziva i „Velika nagrada …“ države u kojoj se utrka odvija, kao na primjer „Velika nagrada Monaca“. Prve dvije sjednice vježbanja odvijaju se u petak, treća vježba i kvalifikacije održavaju se subotom, a utrka uvijek nedjeljom. Sve sjednice u 'rundi' održavaju se na jednoj stazi.

U svakoj 'rundi' sudjeluje 10 timova i 20 vozača, 2 vozača za svaki tim. Svaka sjednica ima svoj poredak nakon završetka sjednice.

### 4.1.2 Definiranje entiteta i atributa baze podataka

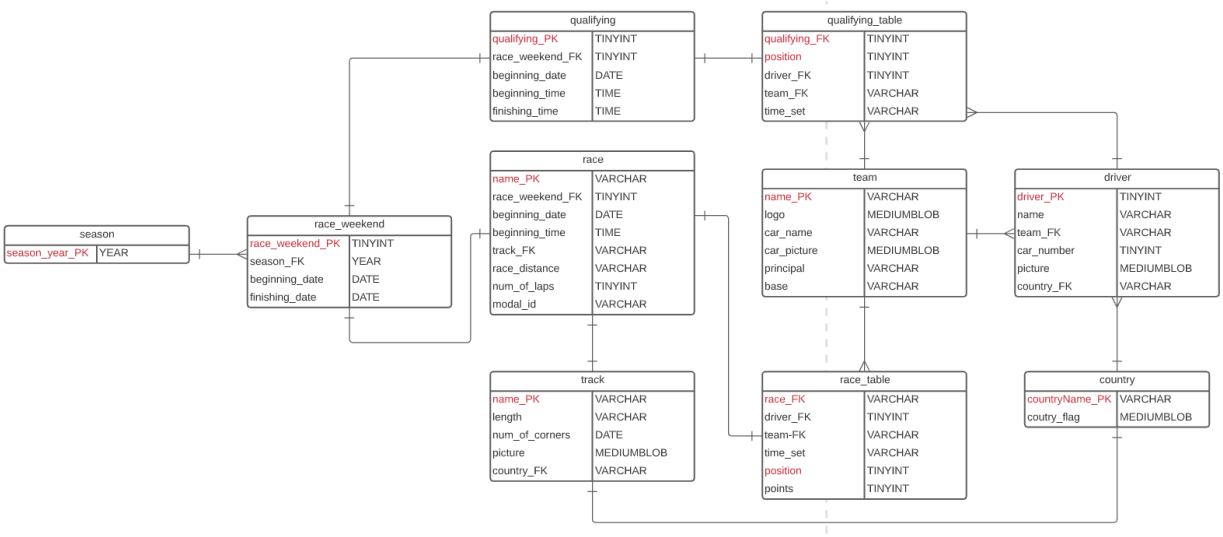
U tablici 4.1. su definirani svi entiteti i atributi koji pripadaju tim entitetima.

Tablica 4.1. Definirani entiteti i atributi baze podataka

| Entitet | Atributi |
|---|---|
| *sesaon* | *seasonYear_PK* |
| *race_weekend* | *race_weekend_FK, sesaon_FK, beginnind_date, finishing_date* |
| *race* | *name_PK, race_weekendFK, begnning_date,beginning_time, track_FK, race_distance, num_of_laps, modal_id* |
| *qualifying* | *qualifying_PK, race_weekend_FK, beginning_date. Beginning_time, finishing_time* |
| *race_table* | *race_FK, driver_FK,team_FK. time_set, points, position* |
| *qualifyng_table* | *qualifying_FK, driver_FK, team_FK, time_set, position* |
| *driver* | *driver_PK, name, team_FK, car_number, picture, country_FK* |
| *team* | *name_PK, logo, car_name, car_picture, principal, base* |
| *track* | *name_PK, length, num_of_corners, picture, country_FK* |
| *country* | *countryName_PK, country_flag* |

### 4.1.3 Konceptualno oblikovanje

Nakon analize zahtjeva i definicije svih entiteta, atributa i veza, dobiven je E/R dijagram na slici 4.1. na kojemu se vide veze između entiteta te pojedini atributi koji pripadaju određenim entitetima.




Sl. 4.1. E/R dijagram

Veza koja ima više entiteta je prikazana sa tri crtice na kraju veze dok strana veze koja ima jednu okomitu crticu na vezu označava kako postoji samo jedan takav entitet. Crvenom bojom su označeni primarni ključevi entiteta. Entiteti „*qualifying_table*“ i „*race_table*“ imaju složeni primarni ključ od dva entiteta. U E/R dijagramu staza i zemlja u kojoj je staza smještena imaju jednostruku vezu na oba kraja dok se u aplikaciji pojavljuju dvije utrke u Italiji u dva grada. Razlog tome je zbog ograničenja uzrokovanih COVID-19 virusom, Međunarodna automobilistička federacija (franc. *Federation Interationale de l'automobile*, skraćeno FIA) je odlučila da se smiju održati dvije ili više utrka u istoj državi iako je to protiv uobičajenih pravila.

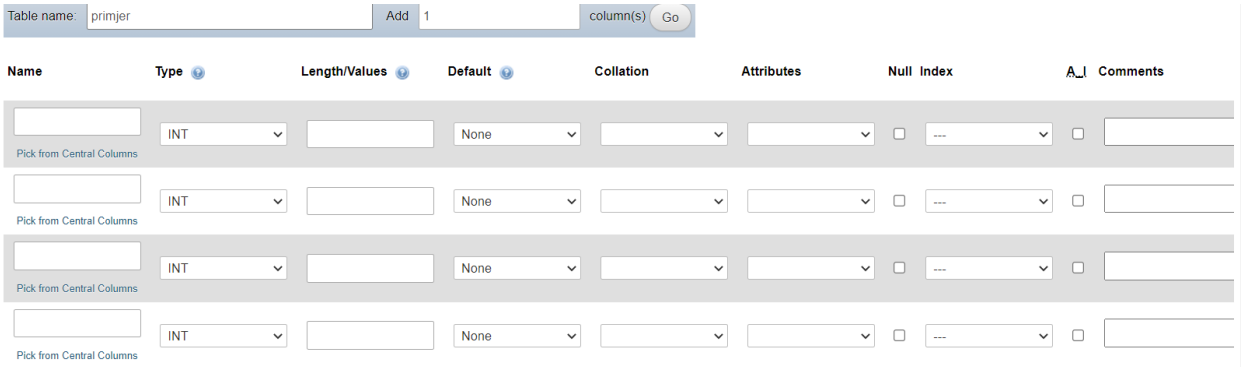
### 4.1.4 Fizičko oblikovanje

Za fizičko oblikovanje upotrijebljen je phpMyAdmin koji dolazi kao dio XAMPP okruženja. Pomoću phpMyAdmin-a lako se dodaju tablice u bazu podataka i podaci u tablice. Na slici 4.2. prikazane su sve tablice u bazi podataka „formula1“. Na slici 4.3. prikazano je sučelje za dodavanje tablica u bazu. U istom sučelju lako je dodati i atribute tj. naziv atributa, tip atributa,

veličinu tipa, učiniti ga primarnim ključem i slično. U tablice se kroz isto sučelje unose podaci, dodaju i uklanjaju atributi te ključevi.

| | Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|---|
| ☐ | country | Browse Structure Search Insert Empty Drop | 20 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| ☐ | driver | Browse Structure Search Insert Empty Drop | 20 | InnoDB | utf8mb4_general_ci | 2.5 MiB | - |
| ☐ | qualifying | Browse Structure Search Insert Empty Drop | 23 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| ☐ | qualifying_table | Browse Structure Search Insert Empty Drop | 180 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| ☐ | race | Browse Structure Search Insert Empty Drop | 23 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| ☐ | race_table | Browse Structure Search Insert Empty Drop | 180 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| ☐ | race_weekend | Browse Structure Search Insert Empty Drop | 23 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| ☐ | season | Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| ☐ | team | Browse Structure Search Insert Empty Drop | 10 | InnoDB | utf8mb4_general_ci | 1.5 MiB | - |
| ☐ | track | Browse Structure Search Insert Empty Drop | 21 | InnoDB | utf8mb4_general_ci | 2.5 MiB | - |
| | 10 tables | Sum | 501 | InnoDB | utf8mb4_general_ci | 6.8 MiB | 0 B |

Sl. 4.2. Prikaz svih tablica u bazi podataka




Sl. 4.3. Prikaz sučelja za dodavanje tablica u bazu

Kroz sučelje se i postojeći podaci mogu mijenjati, brisati i dodavati.

## 4.2 Izrada korisničkog sučelja aplikacije

Za korisničko sučelje korišteni su HTML, CSS i Bootstrap. Prvi korak u izradi korisničkog sučelja je kopiranje CSS i JavaScript mapa, koji su dio Bootstrapa, u mapu u kojoj se nalaze dokumenti izvornog kôda kako bi se mogli eksterno uključiti u .php dokument i kako bi Bootstrap ispravno radio. Ovaj korak je potreban kako bih Bootstrap funkcionirao. Sve datoteke osim .css datoteka imaju nastavak .php jer interpreter u XAMPP-u prepoznaje HTML.

```
7   <!DOCTYPE html>
8   <html lang="en">
9   <head>
10      <meta charset="UTF-8">
11      <meta http-equiv="X-UA-Compatible" content="IE=edge">
12      <link rel="stylesheet" href="css/bootstrap.css">
13      <link rel="stylesheet" href="styles/mainPage.css">
14      <link rel="stylesheet" href="styles/navBar.css">
15      <link rel="stylesheet" href="styles/footer.css">
16      <meta name="viewport" content="width=device-width, initial-scale=1.0">
17      <title>2021 Formula 1 Season</title>
18  </head>
```

Sl. 4.4. <head> element

Na slici 4.4. prikazan je <head> element. U njemu se uključuju svi CSS dokumenti: „mainPage.css“, „navBar.css“, „footer.css“ i „bootstrap.css“. JavaScript dokument uključen je na kraju <body> elementa. CSS dokumenti uključuju se pomoću <link> elementa, gdje atribut *rel* treba biti vrijednosti „*stylesheet*“ dok atribut *href* ima vrijednost lokacije CSS dokumenta na poslužitelju. Budući da je mapa „*styles*“, koja sadržava sve CSS datoteke, na istoj razini kao i datoteka glavne stranice aplikacije, u <link> elementu prvo nabrajamo mapu „*styles*“, nakon toga kosa crta koja označava da tražimo datoteku unutar te mape i onda slijedi ime datoteke unutar mape „*styles*“ koju se želi umetnuti unutar HTML datoteke. Element <title> sadrži naslov dokumenta. Datoteka „navBar.css“ sadrži CSS kôd koji uređuje navigacijsku traku, „footer.css“ kôd koji uređuje podnožje te „mainPage.css“ kôd koji uređuje ostali sadržaj na stranici.

Nakon <head> elementa slijedi tijelo stranice tj. ono što preglednik učitava na ekranu. Za tu svrhu koristimo <body> element. Prva stvar potrebna za funkcionalnu stranicu je navigacijska traka. Bootstrap omogućuje lako dodavanje elementa <nav> koji ima gotove stilove, gumbe i responzivan je. Navigacijska traka postavljena je u  element koji predstavlja zaglavlje aplikacije. Zaglavlje je fiksirano za vrh stranice pomoću dodane klase „fixed-top“, pa ako sav

sadržaj ne stane na ekran, možemo listati kroz stranicu, a navigacijska traka ostaje vidljiva na vrhu ekrana. (Sl. 4.5.)

```
<div class="header fixed-top">
    <nav class="navbar navbar-expand-lg navbar-light">
        <div class="container-fluid">
        <a class="navbar-brand" href="#"><img src="f1_logo.png" alt="F1 logo" id="f1_logo"></a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-ex
            <span class="navbar-toggler-icon"></span>
        </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link" href="pages/standingsPage.php">Standings</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="pages/teamsAndDriversPage.php">Teams/Drivers</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
</div>
```

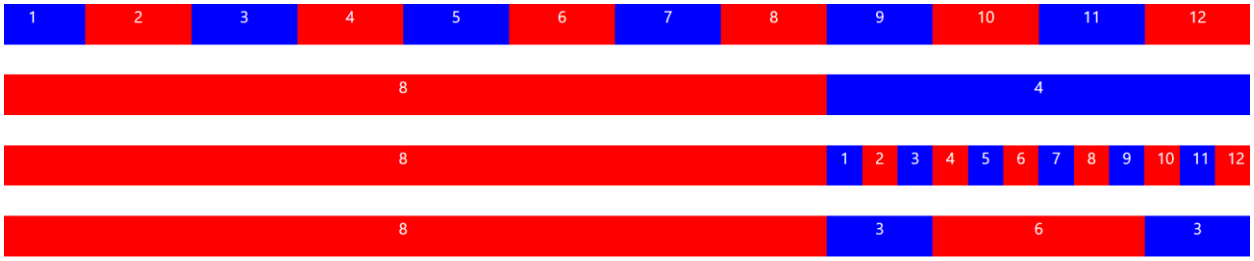
Sl. 4.5. Kôd za navigacijsku traku

Slika 4.5. također prikazuje kako su implementirani gumbi u navigacijskoj traci. Prvi *link* predstavlja logo stranice, u ovom slučaju logo Formule 1 koji se ponaša kao gumb za povratak na početnu stranicu. Slijedeći element predstavlja gumb za padajući izbornik koji se pojavljuje samo kada je ekran veličine pametnoga telefona. Zadnji dio navigacijske trake je neporedana lista koja predstavlja poveznice na druge stranice aplikacije. Oni su na velikim ekranima prikazani na desnoj strani navigacijske trake, dok su na manjim zaslonima dio padajućeg izbornika.

```
    <div class="row site_info">
        <div class="col-md-8">
            <h1 id="schedule_page_title">FORMULA 1 SCHEDULE</h1>
        </div>

        <div class="col-md-4 upcomming_race">
            <div class="row">
                <div class="col-3">
                    Next up:
                </div>
                <div class="col-6">
                    <?php
                        $upcomming_races_data = array();
                        $upcomming_races_data = getUpcommingRaces();
                        $upcomming_race = $upcomming_races_data[0];
                        echo $upcomming_race['Race name'];
                    ?>
                </div>
                <div class="col-3">
                    <?php
                        echo '<img src="'.base64_encode($upcomming_race['Country flag']).'" height="50px" alt="...">';
                    ?>
                </div>
            </div>
        </div>
    </div>
```

Sl. 4.6. Kôd za traku informacija početne stranice u datoteci „mainPage.php“

Ispod navigacijske trake nalazi se traka sa naslovom stranice i informacija koja je utrka slijedeća. Slika 4.6. prikazuje implementaciju trake za informacije. Bootstrap ima *grid* sustav koji je sačinjen od redaka (engl. *row*) i svaki redak je sačinjen od 12 stupaca (engl. *column*) iste širine. Zbroj stupaca u retku uvijek treba biti dvanaest kao što je prikazano u prvom redu na slici 4.7. Pomoću *row* i *col* klasa lagano se može stvoriti raspored informacija na stranici. Prvo je stvoren jedan redak u kojemu imamo dva velika stupca, jedan širine osam stupaca, a drugi ostala četiri stupca što ukupno čini dvanaest originalnih stupaca, kao što je prikazano u drugom retku na slici 4.7. U širi stupac postavljen je naslov stranice dok je u drugi (uži) stupac postavljena obavijest za slijedeću utrku. Bootstrap nam dozvoljava ugnježđivanje redaka i stupaca. U drugom (užem)




Sl. 4.7. Kôd za traku informacija početne stranice u datoteci „mainPage.php“

stupcu smješten je redak koji je sačinjen od novih dvanaest stupaca, kao što je prikazano u trećem stupcu na slici 4.7. Ti stupci su podijeljeni u tri velika stupca unutar novog retka koji nalikuju na tablicu. Rubni stupci tablice široki su tri stupca novostvorenoga reda, a srednji šest, što opet čini dvanaest stupaca, što se vidi u zadnjem redu na slici 4.7. Dohvaćanje informacija o slijedećim utrkama će biti naknadno objašnjeno.

```
93              <div class="col">
94                  <div class="card h-100">
95                      <?php echo '<img src="'.base64_encode($datas['Country flag']).'" class="card-img-top" alt="...">'?>
96                      <div class="card-body">
97                          <h5 class="card-title"><?php echo $datas["Beggining Date"];?> - <?php echo $datas["Finishing Date"];?></h5>
98                          <p class="card-text"><?php echo $datas["Race name"];?></p>
99                      </div>
100                     <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#<?php echo $datas["Race name PK"];?>">
101                         More detail
102                     </button>
103                 </div>
104             </div>
```

Sl. 4.8. Kôd za karticu na početnoj stranici u datoteci „mainPage.php“

Početna stranica ujedno predstavlja i kalendar utrka za cijelu sezonu Formule 1. Utrke su prikazane pomoću kartica koje su implementirane u Bootstrapu. Kartice su podijeljene tako da u

redu bude najviše 5 kartica (na velikom ekranu), na ekranu srednje veličine (npr. tablet) prikazati će se 3 kartice u redu, dok će na mobilnim uređajima biti samo jedna kartica u redu.

Slika 4.8. prikazuje implementaciju kartice u aplikaciji. Na vrhu kartice nalazi se slika koja predstavlja zastavu države u kojoj se utrka odvija. Ispod slike postavljen je datum utrke te ispod datuma ime utrke. Na dnu kartice postavljen je gumb koji otvara *modal*.

*Modal* je element Bootstrapa koji se pozicionira iznad sadržaja stranice i nudi dodatne informacije o utrci za koju se posjetitelj zanima. *Modali* se automatski ažuriraju ovisno o tome je li utrka završena ili se još nije dogodila. Ako je utrka završena u *modalu* će biti prikazana tablica imena, prezimena i slika prva tri plasirana vozača u utrci, potpuni poredak utrke i kvalifikacija te informacije o stazi. (Sl. 4.9.) Ako je posjetitelj zainteresiran za dodatne informacije ispod tablice su postavljena dva *accordion* elementa, također dio Bootstrapa. *Accordion* element dozvoljava čist način prikazivanja samo potrebnih informacija. On se ponaša kao padajuća zavjesa, na prvu je zatvoren, ali ako kliknemo na njega, proširi se prema dolje i prikazuje informacije unutar zavjese. Također se klikom može i zatvoriti tj. uvući.

```
113                 <div class="modal-body">
114                 <h3>Podium</h3>
115                     <table class="table">
116                         <tbody>
117
118                         <?php ···
122                         ?>
123
124                         <tr>
125                             <td>2nd</td>
126                             <td>1st</td>
127                             <td>3rd</td>
128                         </tr>
129                         <tr>
130                             <td><?php echo '<img src="'.base64_encode($data_second['Driver picture']).'" alt="...">'?></td>
131                             <td><?php echo '<img src="'.base64_encode($data_first['Driver picture']).'" alt="...">'?></td>
132                             <td><?php echo '<img src="'.base64_encode($data_third['Driver picture']).'" alt="...">'?></td>
133                         </tr>
134                         <tr>
135                             <td><?php echo $data_second["Driver name"];?></td>
136                             <td><?php echo $data_first["Driver name"];?></td>
137                             <td><?php echo $data_third["Driver name"];?></td>
138                         </tr>
139
140                         </tbody>
141                     </table>
```

Sl. 4.9. Program za prikaz prva tri plasirana vozača u utrci u datoteci „mainPage.php“

U *accordion* elementima prikazani su cijeli rezultati utrke te poredak nakon kvalifikacija. (Sl. 4.10.)

```
<div class="accordion" id="accordionExample">
    <div class="accordion-item">
        <h2 class="accordion-header" id="headingOne">
            <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-
                Race results
            </button>
        </h2>
        <div id="collapseOne" class="accordion-collapse collapse" aria-labelledby="headingOne" data-bs-parent="#accordionExample">
            <div class="accordion-body">
                <table class="table table-striped table-hover">
                    <thead>
                        <tr>
                            <th scope="col">Position</th>
                            <th scope="col">Driver name</th>
                            <th scope="col">Team</th>
                            <th scope="col">Time</th>
                            <th scope="col">Points</th>
                        </tr>
                    </thead>
                    <tbody>

                    <?php···
                    ?>
                        <tr>
                            <th scope="row"><?php echo $race_result["Position"];?></th>
                            <td><?php echo $race_result["Driver name"];?></td>
                            <td><?php echo $race_result["Team"];?></td>
                            <td><?php echo $race_result["Time"];?></td>
                            <td><?php echo $race_result["Points"];?></td>
                        </tr>

                    <?php···
                    ?>
```

Sl. 4.10. Prikaz *accordion* elementa u *modalu* u datoteci „mainPage.php“

Na kraju *modal* elementa prikazane su informacije o stazi. Prvo je prikazana slika staze, a ispod nje nalazi se tablica sa dodatnim informacijama o stazi za velike obožavatelje koji žele saznati svaki detalj. U slučaju da se utrka još nije održala, u *modalu* je tablično prikazan raspored svih sjednica tijekom vikenda te iste informacije o stazi.

Na stranici „Standings“, kojoj možemo pristupiti preko navigacijske trake klikom na gumb „Standings“, tablično je prikazan poredak vozača i timova. Slično kao i na početnoj stranici ispod navigacijske trake smještene su informacije o stranici koje sadrže naslov stranice i dva gumba koja aktiviraju *collapse* element Bootstrapa. *Collapse* je jako sličan *accordionu* jer ima padajuću zavjesu na kojoj prikazuje informacije ali se aktivira gumbom. Na slici 4.11. prikazana je implementacija dva gumba. Svaki gumb aktivira svoj *collapse* element. Elementi su postavljeni jedan iznad drugoga. Prvi element je pri dolasku na stranicu automatski otvoren i prikazuje poredak vozača. Ako korisnik želi pogledati i poredak timova kliknemo na gumb “Constructor standings” i otvoriti će se poredak timova ispod poretka vozača.

```
        <div class="col-md-5 standings_buttons">
            <div class="btn-group" role="group" aria-label="Basic example">
                <a class="btn btn-primary btn-lg" data-bs-toggle="collapse" href="#driverStandings" role="button" aria-expanded="true" aria-cont
                    Driver standings
                </a>
                <a class="btn btn-primary btn-lg" data-bs-toggle="collapse" href="#constructorStandings" role="button" aria-expanded="false" ari
                    Constructor standings
                </a>
            </div>
        </div>
    </div>
```

Sl. 4.11. Program za gumbe za prebacivanje poredaka u *accordion* elementu u datoteci „stadings.php“

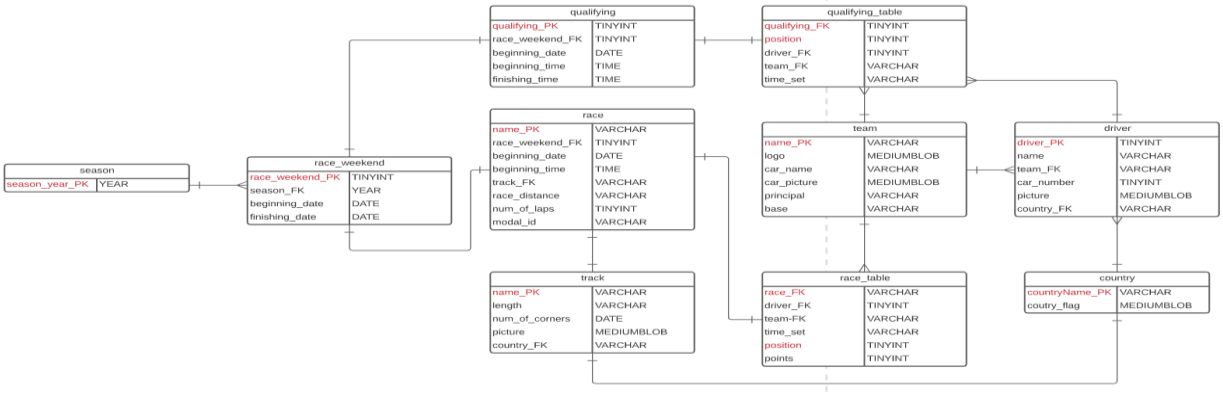
Stranici „Teams/Drivers“ također se pristupa preko navigacijske trake. Ta stranica prikazuje najbitnije informacije o timovima i njihovim vozačima kao što su ime tima, sjedište tima, slika auta, ime vozača, slika vozača i slično. Za prikaz timova također su upotrijebljene *row* i *col* klase. U jednom retku prikazana su dva tima koja zauzimaju istu količinu mjesta na ekranu. U mobilnom prikazu timovi su prikazani jedan ispod drugoga. Informacije o timovima prikazane su pomoću *tab* načina prikaza, također dostupnog u Bootstrapu. To je dinamičan način prikaza informacija jer na vrhu postoji mala navigacijska traka koja sadrži *tabove* tj. linkove na različite informacije koje se mogu prikazati u tijelu ovoga elementa. Tijelo elementa nalazi se ispod navigacijske trake u elementu čiji atribut *class* sadrži klasu „*tab-content*“. (Sl. 4.12.)

```
<nav>
    <div class="nav nav-tabs" id="nav-tab" role="tablist">
        <button class="nav-link active" id="nav-home-tab" data-bs-toggle="tab" data-bs-target="#nav-<?php echo $team_data["Car name"];?>" type="button" role="tab" aria-controls="nav
        <button class="nav-link" id="nav-profile-tab" data-bs-toggle="tab" data-bs-target="#nav-driver-<?php echo $team_data["Car name"];?>" type="button" role="tab" aria-controls="
        <?php echo '<img src="'.base64_encode($team_data['Team logo']).'"height="50px" alt="..." id="teamLogo">'?>
    </div>
</nav>

<div class="tab-content" id="nav-tabContent">
    <div class="tab-pane fade show active" id="nav-<?php echo $team_data["Car name"];?>" role="tabpanel" aria-labelledby="nav-home-tab">
        <p><?php echo $team_data["Team name"];?></p>
        <div class="row">
            <div class="col-sm-7 team-car"><?php echo '<img src="'.base64_encode($team_data['Car picture']).'"width="100%" alt="...">'?></div>
            <div class="col-sm-5 team-info">
                <table class="table">
                    <tbody>
                    <tr>
                        <th scope="row">Principal:</th>
                        <td><?php echo $team_data["Team principal"];?></td>
                    </tr>
                    <tr>
                        <th scope="row">Base:</th>
                        <td><?php echo $team_data["Team base"];?></td>
                    </tr>
                    <tr>
                        <th scope="row">Car:</th>
                        <td><?php echo $team_data["Car name"];?></td>
                    </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
```

Sl. 4.12. Program za *tab* prikaz timova i vozača u datoteci „standings.php“

## 4.3 Izrada serverskog sučelja

Prvi korak u izradi serverskog sučelja je kreiranje skripte koji se spaja na bazu podataka. Za spajanje na bazu podataka koristiti će se PHP programski jezik. Na slici 4.12. prikazana je funkcija *connectToDB()*, koja se nalazi u datoteci „connenction.php“, u kojoj se stvara varijabla *$conn* i koristi funkcija za spajanje na bazu podataka. Funkcija prima 4 parametra: *host*, *username*, zaporku i ime baze podataka. U ovom slučaju *host* je „*localhost“*, *username* je „*root“*, zaporke nema, a ime baze podataka je „formula1“. U uvjetnoj naredbi *if* se ispituje postoji li varijabla *$conn*. Ako ne postoji, funkcija *die()* ispisuje poruku da je spajanje neuspješno i izbacuje pogrešku. Ako varijabla *$conn* postoji funkcija *connectToDB()* vraća vezu kao varijablu. Ovaj način spajanja na bazu podataka naziva se proceduralan pristup.




Sl. 4.12. Funkcija spajanja na bazu podataka u datoteci „connection.php“

Drugi korak je uključivanje funkcije *connectToDB()* u sve skripte koje trebaju imati konekciju na bazu podataka. To se radi pomoću naredbe *include_once* i u navodnicima navodimo datoteku u kojoj se nalazi funkcija za spajanje na bazu. Korišteno je *include_once* jer će bez toga funkcija *connectToDB()* biti definirana svaki put kada se u datoteci pozove, što u PHP-u nije moguće.

Datoteka zaslužna za dohvaćanje podataka za početnu stranicu je „mainPageData.php“. Datoteka sadrži četiri funkcije: *getFinishedRaces()*, *getUpcommingRaces()*, *getFinishedRacesResults()*, *getFinishedQualyResults()*. Sve funkcije imaju istu strukturu i razlikuju se samo u SQL upitima koji se šalju bazi podataka. Na slici 4.13. prikazana je prva funkcija koja dohvaća podatke o završenim utrkama. Ti podaci će se upotrijebiti u karticama koje prikazuju glavne informacije o utrkama.

```php
function getFinishedRaces(){
    $conn = connectToDB();

    $sql = "SELECT CONCAT(DAY(rw.beginning_date),'.',MONTH(rw.beginning_date),'.') AS 'Beggining Date',
    CONCAT(DAY(finishing_date),'.',MONTH(finishing_date),'.') AS 'Finishing Date', r.name_PK AS 'Race name',
    c.country_flag AS 'Country flag', r.modal_id AS 'Race name PK', t.picture AS 'Track picture', t.name_PK AS 'Track name',
    t.country_FK AS 'Country', t.length AS 'Track length', t.num_of_corners AS 'Number of corners', r.race_distance AS 'Race distance',
    r.num_of_laps AS 'Number of laps'
    FROM `race_weekend` AS rw, `race` AS r, `track` AS t, `country` AS c
    WHERE rw.race_weekend_PK=r.race_weekend_FK AND r.track_FK=t.name_PK AND t.country_FK=c.countryName_PK AND r.beginning_date <= NOW()
    ORDER BY rw.beginning_date";

    $finished_races_query_result = mysqli_query($conn, $sql);

    $finished_races = array();

    if (mysqli_num_rows($finished_races_query_result) > 0) {
        while($row = $finished_races_query_result-> fetch_assoc()) {
            $finished_races[]=$row;
        }

    }
    else{
        echo "No result found";
    }
    mysqli_close($conn);

    return $finished_races;
}
```

Sl.4.13. Funkcija *getFinishedRaces()* u „mainPageData.php“ datoteci

U varijabli *$sql* sprema se SQL upit koji se šalje bazi podataka. U upitu se odabire dan i mjesec početka i kraja svake 'runde' koja je već završena, ime utrke, zastava države, slika staze, broj zavoja na stazi, broj krugova u utrci, duljina staze, duljina cijele utrke i šifra *modal_id* koja služi za povezivanje modala i gumba koji ga otvara preko *id* atributa u gumbu i modalu. Kako bi upit funkcionirao primarni i strani ključevi između tablica koje se koriste trebaju biti isti i kako bi se odabrale utrke koje su već završene koristi se funkcija *NOW()*. Rezultat funkcije NOW() se uspoređuje sa početnim datumom 'runde'. Taj upit se šalje bazi pomoću funkcije *mysqli_query()*, a rezultat upita sprema se u varijablu *$finished_races_query_result*. Stvorena je varijabla *$finished_races* koja je niz. U uvjetnoj naredbi *if* ispituje se ima li redaka u varijabli *$finished_races_query_result* tako da rezultat funkcije *mysqli_num_rows()* bude veći od nula. Ako u varijabli *$finished_races_query_result* ima redaka, u *while* petlji se pojedini red upita stavlja u varijablu *$row* koja se dalje sprema u varijablu *$finished_races* kako bi bilo jasnije što koja funkcija vraća. Veza se zatvara i formatirani rezultat upita vraća se tamo gdje je funkcija

```php
<?php
    $upcomming_races_data = array();
    $upcomming_races_data = getUpcommingRaces();
    $upcomming_race = $upcomming_races_data[0];
    echo $upcomming_race['Race name'];
?>
```

Sl.4.14 Pozivanje funkcije *getUpcommingRaces()* u „mainPage.php“

pozvana. Postupak dobivanja nadolazećih utrka je isti osim što u uspoređivanju početnog datuma 'runde' i funkcije *NOW()* tražimo da je početak 'runde' veće od rezultata funkcije *NOW()* tj. tražimo datume poslije datuma koji daje funkcija *NOW()*.

Na slici 4.14 prikazan je poziv funkcije *getUpcommingRaces()* u „mainPage.php“ datoteci. Prvo je stvoren novi niz koji će primiti rezultat iz funkcije. Iz rezultata je uzet samo prvi redak, što odgovara prvoj slijedećoj utrci koja nadolazi. Funkcija *echo* na ekran ispisuje ime 'runde' koja je slijedeća na rasporedu. Ova funkcija koristi se svaki put kada nam je potreban ispis podataka iz

```
275      <?php
276          $upcomming_race_details_data = array();
277          $upcomming_race_details_data = getUpcommingRaceDetails();
278          $upcomming_races_counter = 0;
279
280          foreach($upcomming_races_data as $datas){
281      ?>
```

Sl. 4.13. Pozivanje funkcije *getUpcommingRaceDetails()* u „mainPage.php“

baze na ekran. Za prikaz rasporeda sjednica u utrkama koje će se tek dogoditi, dodatno se koristi funkcija *getUpcommingRaceDetails()* koja radi na isti princip kao već objašnjena funkcija, a služi za dohvaćanje vremenskog i datumskog početka kvalifikacija i utrke.

```
317  <tbody>
318  <tr>
319      <th scope="row"><?php echo $race_details["Qualy date"];?></th>
320      <td>Qualifying</td>
321      <td><?php echo $race_details["Qualy time"];?></td>
322  </tr>
323  <tr>
324      <th scope="row"><?php echo $race_details["Race date"];?></th>
325      <td>Race</td>
326      <td><?php echo $race_details["Race time"];?></td>
327  </tr>
328  </tbody>
```

Sl. 4.16. Primjer ispisa podataka u *modal* element u datoteci „mainPage.php“

Na slici 4.15. prikazan je poziv funkcije *getUpcommingRaceDetails()*. U *foreach* petlji prolazi se kroz svaki redak u tablici i ispisuje ga se na ekran u *modal* element. (Sl. 4.16.)

Prikaz informacija o utrkama koje su se već dogodile malo je kompliciraniji. Funkcija *getFinishedRaces()* je ista kao *getUpcommingRaces()* tj. daje istu vrstu informacija ovisno o utrci. Postaje kompliciranije ako se želi saznati još informacija o utrci. U datoteci „finishedRacedPodiumDetails.php“ nalaze se tri funkcije: *getFirstPlaces()*, *getSecondPlaces()*, *getThirdPlaces()*. (Sl. 4.17.)

```
6   function getFirstPlaces(){
7       $conn = connectToDB();
8
9       $sql="SELECT driver.name AS 'Driver name', race.name_PK, driver.picture AS 'Driver picture'
10      FROM driver, race_table, race, race_weekend
11      WHERE driver.driver_PK=race_table.driver_FK AND race_table.position=1
12      AND race_weekend.race_weekend_PK=race.race_weekend_FK AND race.name_PK=race_table.race_FK
13      ORDER BY race_weekend.race_weekend_PK";
14
15      $first_place_query_result = mysqli_query($conn, $sql);
16      $first_place = array();
17
18      if (mysqli_num_rows($first_place_query_result) > 0) {
19          while($row = $first_place_query_result-> fetch_assoc()) {
20          $first_place[]=$row;
21      }
22
23      }
24      else{
25          echo "No result found";
26      }
27      mysqli_close($conn);
28
29      return $first_place;
30  }
```

Sl. 4.17. Funkcija *getFirstPlaces()* u „mainPageData.php“ datoteci

Sve tri funkcije funkcioniraju na isti princip tj. dohvaćaju podatke o vozačima koji su završili prvi, drugi, odnosno treći u utrci. Za potpuni poredak kvalifikacija i utrke koriste se funkcije *getFinishedRacesResult()* i *getFinishedQualyResult()*. (Sl. 4.18.) Iz razloga što u jednoj *foreach* petlji ne možemo iterirati kroz više tablica, za svaki prolazak kroz *foreach* petlju ručno stvoreni iterator prolazi kroz rezultate funkcija koje prikazuju podij utrke (funkcije *getFirstPlaces()*, *getSecondPlaces()* i *getThirdPlaces()*) i potpune rezultate (funkcije *getFinishedRacesResult()*, *getFinishedQualyResult()*). Iteratori su postavljeni na nula i na kraju svakog prolaska kroz *foreach* petlju povećavaju se za jedan. Na slici 4.19. prikazano je odabiranje reda tablice onog indeksa kojeg je i brojač te ispis tih podataka u aplikaciju. Iterator broji do 19 jer indeks prvog reda tablice je 0, a u sezoni vozi 20 vozača. Slike su direktno spremljene u bazu podataka te je potrebno njihovo kodiranje kako bi se mogle prikazati na stranici pomoću funkcije *base64_encode()*. Podaci na ostalim stranicama prikazane su na isti način kao i što je objašnjeno. Jedina razlika u funkcijama su SQL upiti koji se šalju bazi podataka.

```php
function getFinishedRacesResults(){
    $conn = connectToDB();

    $sql = "SELECT rt.position AS 'Position', d.name AS 'Driver name', t.name_PK AS 'Team',
    rt.time_set AS 'Time', rt.points AS 'Points'
    FROM race_table AS rt, race AS r, driver AS d, team AS t
    WHERE rt.race_FK=r.name_PK AND rt.driver_FK=d.driver_PK AND rt.team_FK=t.name_PK
    ORDER BY r.race_weekend_FK";

    $finished_races_results_query_result = mysqli_query($conn, $sql);

    $finished_races_results = array();

    if (mysqli_num_rows($finished_races_results_query_result) > 0) {
        while($row = $finished_races_results_query_result-> fetch_assoc()) {
        $finished_races_results[]=$row;
    }

    }
    else{
        echo "No result found";
    }
    mysqli_close($conn);

    return $finished_races_results;
}
```

Sl. 4.148. Funkcija *getFinishedRacesResults()* u „mainPageData.php“ datoteci

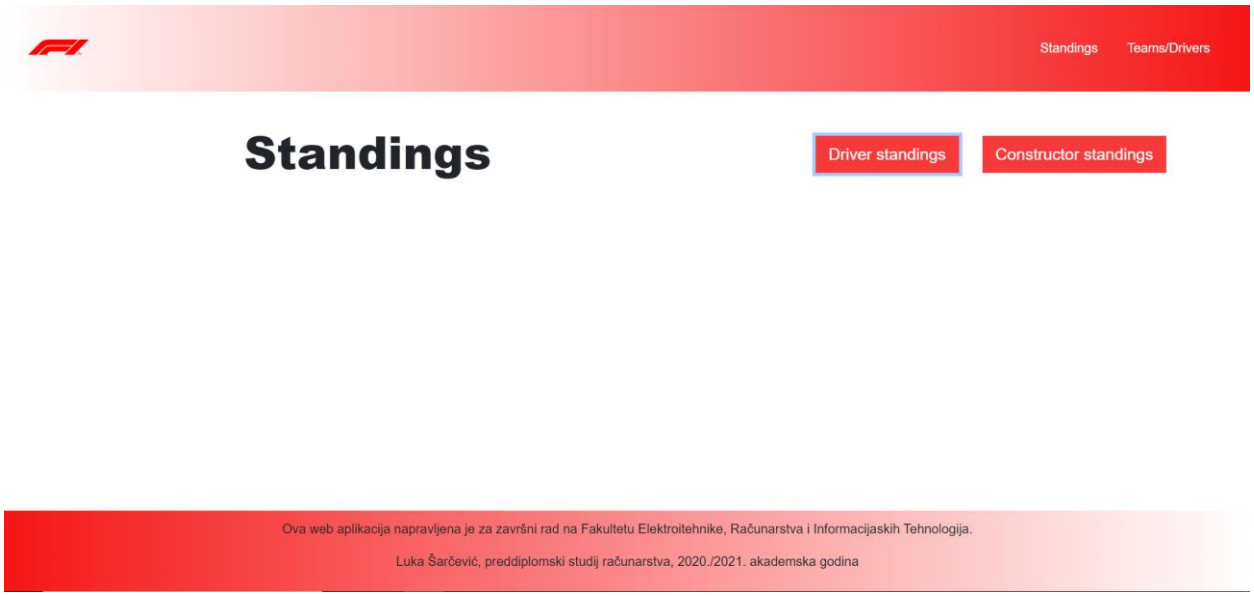
```php
<?php
    $finished_races_results_counter = 0;
    while($finished_races_results_counter <= 19){
        $race_result = $finished_races_results[$race_row_counter+$finished_races_results_counter];

?>
    <tr>
        <th scope="row"><?php echo $race_result["Position"];?></th>
        <td><?php echo $race_result["Driver name"];?></td>
        <td><?php echo $race_result["Team"];?></td>
        <td><?php echo $race_result["Time"];?></td>
        <td><?php echo $race_result["Points"];?></td>
    </tr>
<?php
    ++$finished_races_results_counter;
    }
?>
```

Sl. 4.19. Prikaz iteratora kroz tablicu dobivenu funkcijom *getFinishedRacesResults()*

Informacije se na ostalim stranicama prikazuju na isti način kao na početnoj stranici, pomoću *foreach* petlje te iteratora ako se koristi više tablica u jednoj *foreach* petlji.

Podnožje stranica izrađeno je da ostane na dnu ako na stranici nema informacija. Ako se upotrijebi običan <footer> element, u slučaju prazne stranice, podnožje je izravno ispod

navigacijske trake što ne izgleda privlačno. Na slici 4.20. se vidi kako podnožje ostaje na dnu stranice iako nema sadržaja koji pokriva cijelu stranicu.

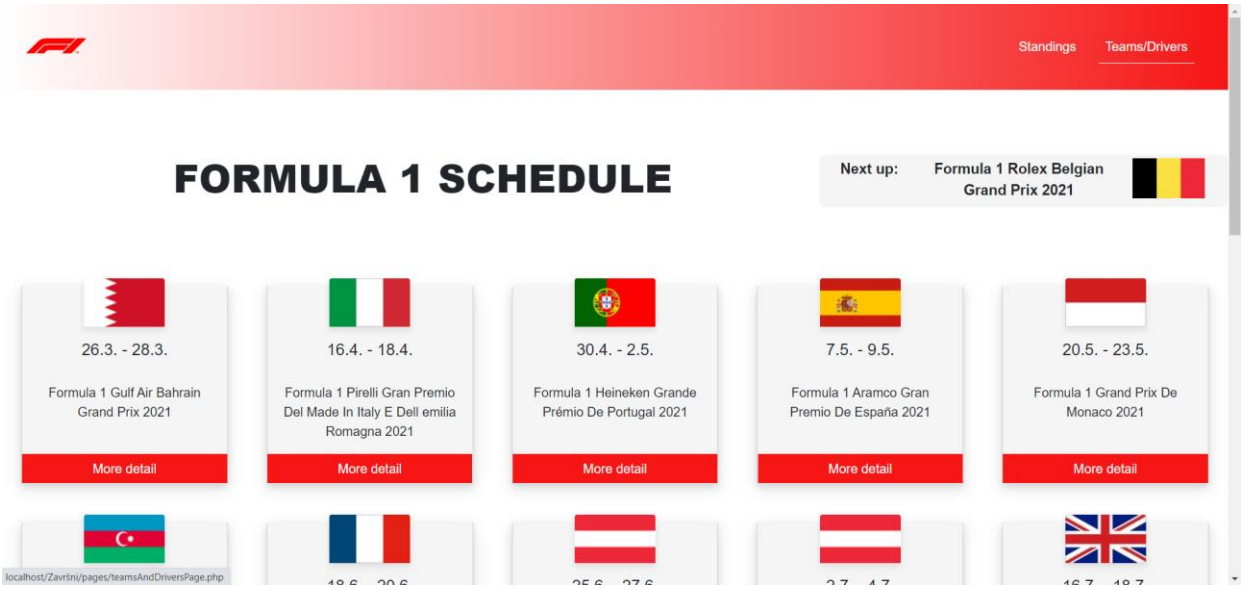



Sl. 4.20. Prikaz podnožja na dnu stranice

# 5. PREGLED ZAVRŠENE APLIKACIJE

U zadnjem poglavlju prikazane su slike završene aplikacije i sve njezine funkcionalnosti te neka rješenja za probleme koji nisu bili nabrojani. Uz prikaz na računalu također će biti prikazan i mobilni prikaz aplikacije.

Slikama 5.1. i 5.2. prikazana je početna stranica aplikacije u računalnom i mobilnom prikazu. Gumbi u navigacijskoj traci imaju *hover* animaciju. Kada se mišem prođe iznad gumba pojavi se linija ispod gumba vidljivo na gumbu „Teams/Drivers“.

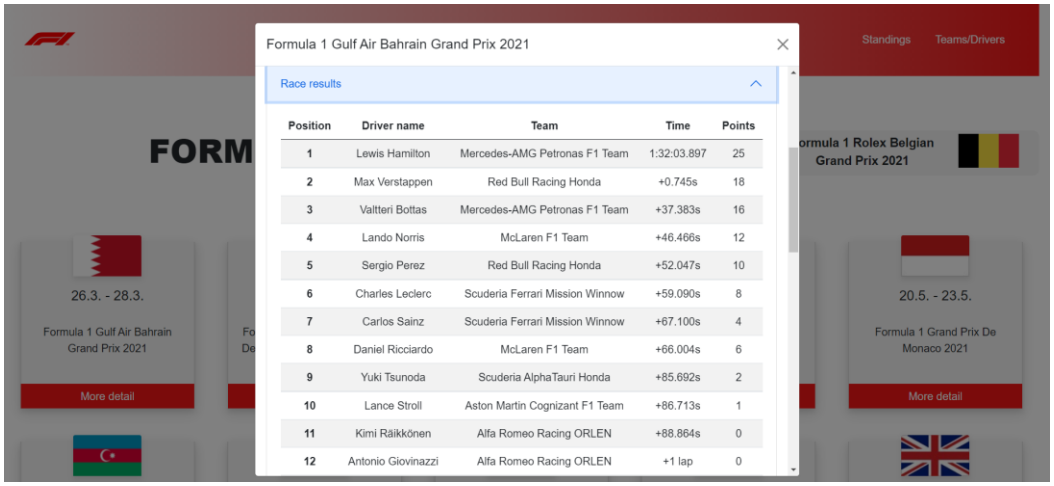



Sl. 5.1. Početna stranica u računalnom prikazu

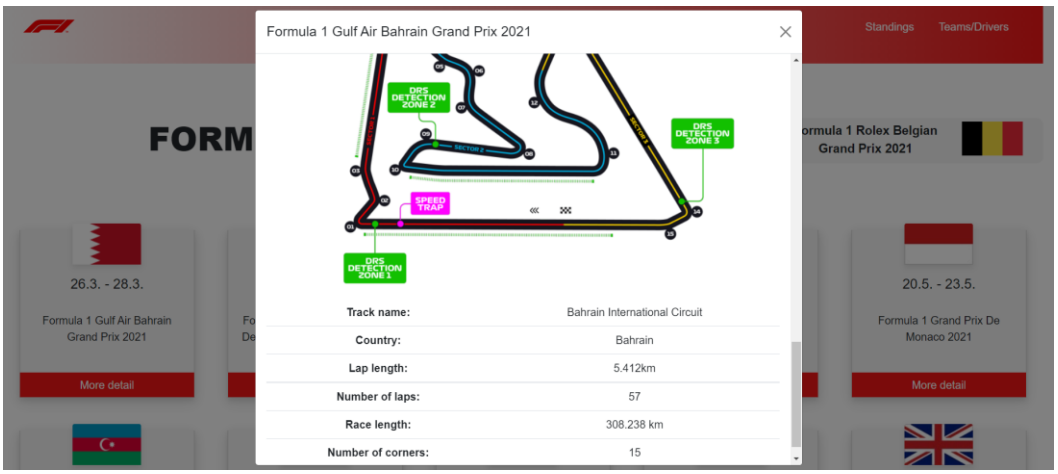
Sl. 5.2. Početna stranice u mobilnom prikazu

Gumb „More information“ vodi do *modula* sa više informacija. Na slikama 5.3., 5.4. i 5.5. prikazane su informacije o završenim utrkama. Sve informacije ne stanu na jednu sliku pa su podijeljene na više slika tako da je na prvoj slici početak *modula*, zatim druga slika prikazuje tablicu rezultata utrke koja se nalazi ispod prva tri plasirana vozača u utrci, a informacije o stazi se nalaze na kraju *modula* ispod tablice rezultata utrke.

Sl. 5.3. Prikaz prva tri plasirana vozača u *modul* elementu

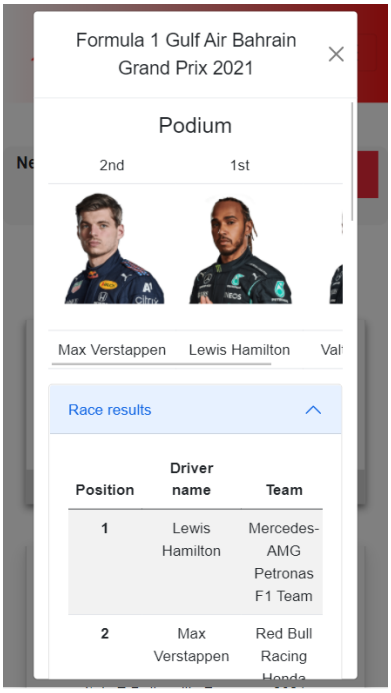



Sl. 5.4. Prikaz rezultata utrke u *modul* elementu

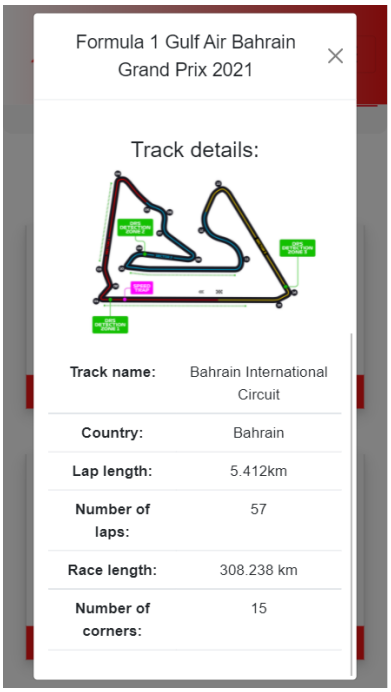



Sl. 5.5. Prikaz informacija o stazi u *modul* elementu

Rezultati kvalifikacija se prikazuju tablično kao rezultati utrke na slici 5.6. pa nema potrebe za njihov prikaz. Sve tri slike dio su istog *modula*. Na slikama 5.6. i 5.7. prikazan je isti *modul* u mobilnom prikazu.




Sl. 5.6. Prikaz prva tri plasirana vozača i rezultata utrke u *modul* elementu u mobilnom prikazu



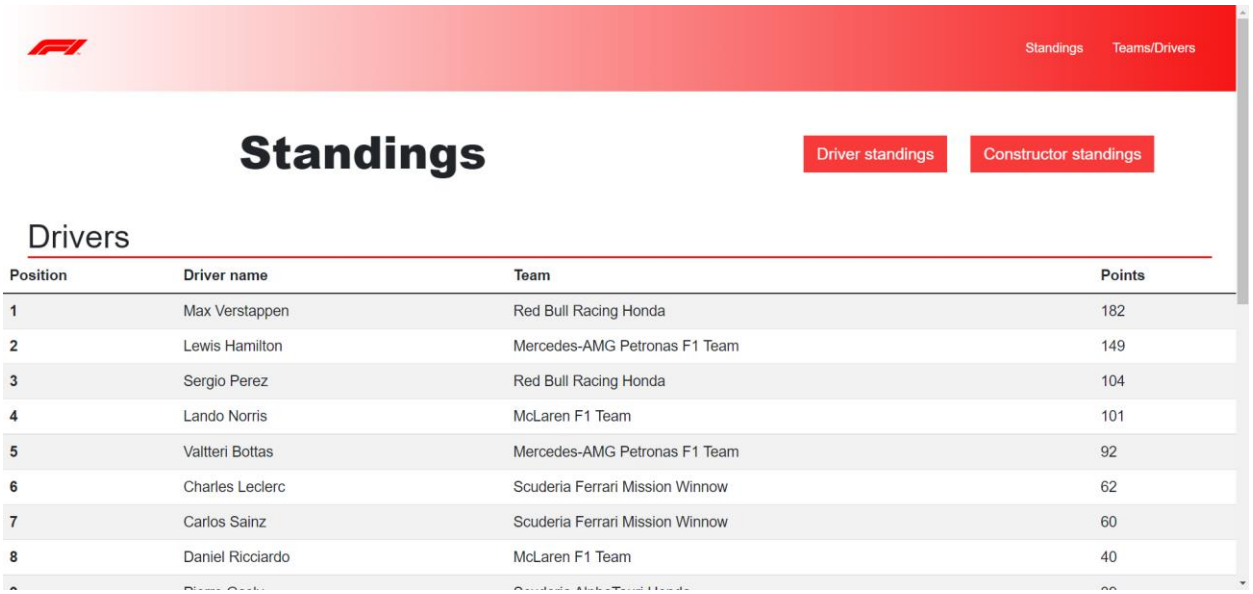

Sl. 5.7. Prikaz informacija o stazi u *modul* elementu u mobilnom prikazu

U mobilnom prikazu prva tri plasirana vozača ne stanu jedan pokraj drugoga i zbog toga je na tablicu u kojoj se oni nalaze primijenjen *overflow* u CSS-u. *Overflow* je postavljen na „auto“, slike su smanjene sa izvorne veličine od 206 x 206 piksela na veličinu od 120 x 120 piksela. Isti *overflow* primijenjen je na tablice rezultata utrke i kvalifikacije. (Sl. 5.8.) Ove promjene se primjenjuju kada širina ekrana postane manja od 500 piksela. U slučaju da je element širi od roditeljskog elementa on se neće proširiti preko njega nego će se automatski stvoriti traka za pomicanje sadržaja lijevo-desno.

```
110  @media only screen and (max-width:500px){
111      .modal-body .table img{
112          height: 120px;
113      }
114
115      .modal_finished .table{
116          display: block;
117          overflow: auto;
118          width: 100%;
119      }
120
121      .accordion .table{
122          display: block;
123          overflow: auto;
124      }
125  }
```

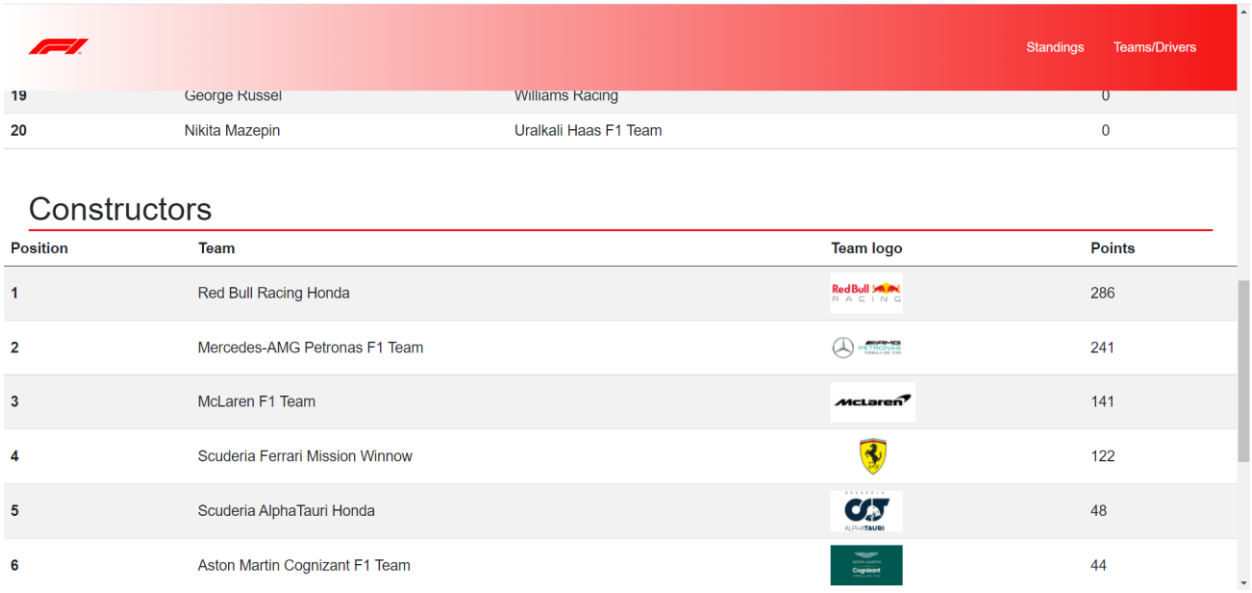
Sl. 5.8. Program u CSS-u za prilagođavanje sadržaja šireg od roditeljskog elementa u datoteci „mainPage.css“

Na slikama 5.9. i 5.10. prikazana je stranica „Standings“. Tablica za poredak vozača je automatski otvorena, a klikom na gumb „Constructor standings“ otvara se tablica poretka timova ispod tablice poretka vozača.



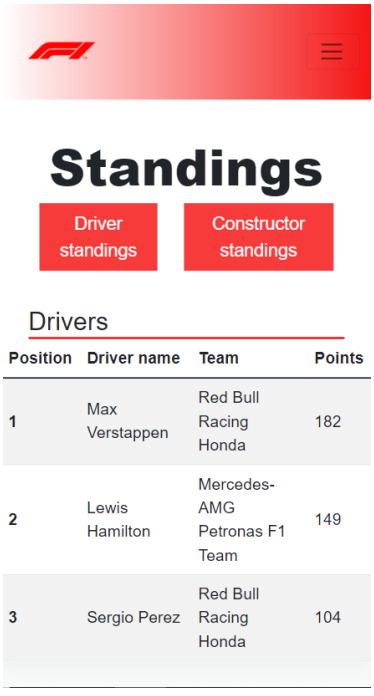

| Position | Driver name | Team | Points |
|---|---|---|---|
| 1 | Max Verstappen | Red Bull Racing Honda | 182 |
| 2 | Lewis Hamilton | Mercedes-AMG Petronas F1 Team | 149 |
| 3 | Sergio Perez | Red Bull Racing Honda | 104 |
| 4 | Lando Norris | McLaren F1 Team | 101 |
| 5 | Valtteri Bottas | Mercedes-AMG Petronas F1 Team | 92 |
| 6 | Charles Leclerc | Scuderia Ferrari Mission Winnow | 62 |
| 7 | Carlos Sainz | Scuderia Ferrari Mission Winnow | 60 |
| 8 | Daniel Ricciardo | McLaren F1 Team | 40 |

Sl. 5.9. Poredak vozača na stranici „Standings“



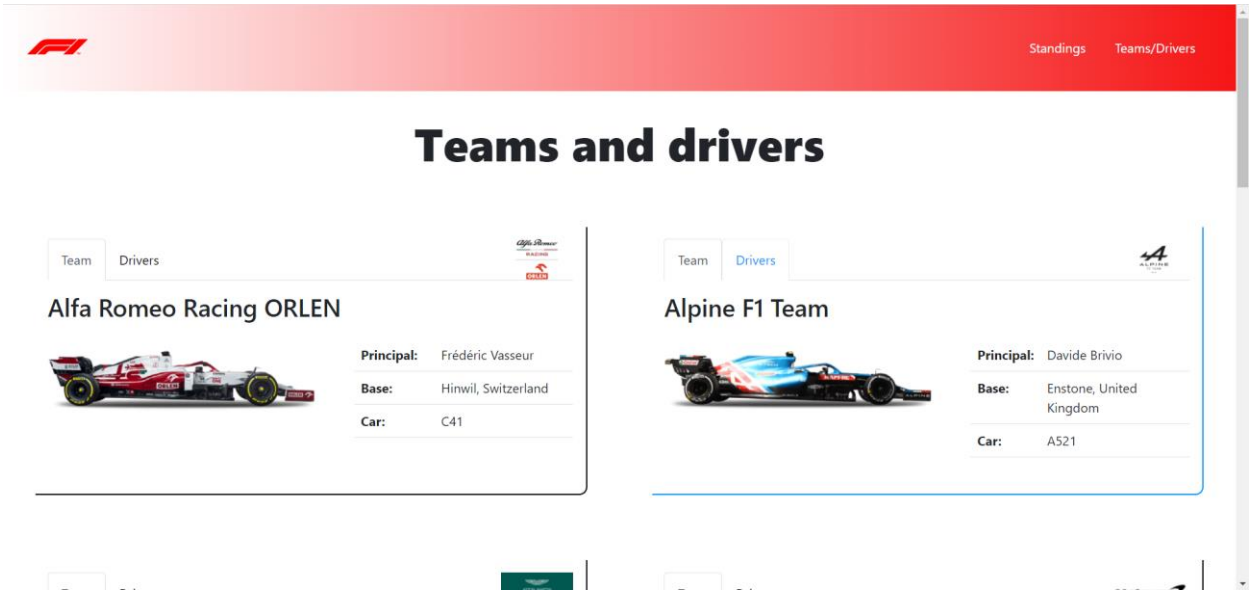

Sl. 5.10. Poredak timova na stranici „Standings“

Slika 5.11. prikazuje mobilni prikaz poretka vozača na stranici „Standings“. Tablica poretka timova prikazuje iste podatke kao na slici 5.10. ali u mobilnom obliku kao tablica poretka vozača na slici 5.11. pa ju je nepotrebno prikazivati.

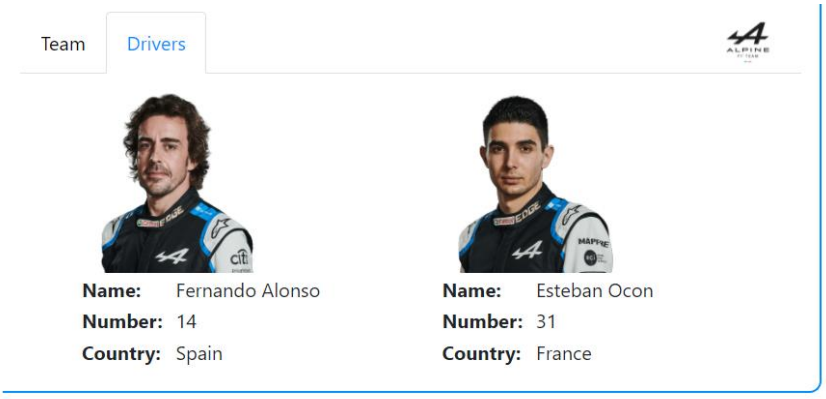



Sl. 5.11. Mobilni prikaz pretka vozača na stranici „Standings“

Stranica „Teams/Drivers“ prikazana je na slici 5.12. Podaci o timu automatski su prikazani ali klikom na gumb „Drivers” pokazuju se podaci o vozačima koji voze za tim. (Sl. 5.13.) U CSS-u dodane su animacije za prelazak miša preko tima.

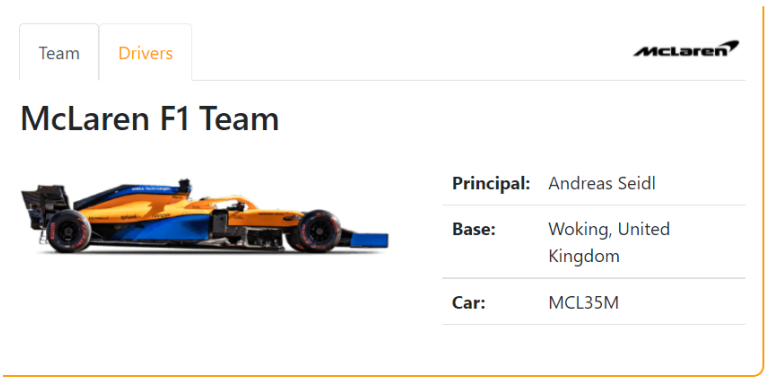



Sl. 5.122. Prikaz informacija o timovima na stranici „Teams/Drivers“




Sl. 5.13. Prikaz informacija o vozačima na stranici „Teams/Drivers“

Rub elementa je crn ali prelaskom miša preko elementa tima, rub mijenja boju u boju tima. U ovom slučaju to je plava boja. U slučaju tima „Mclaren F1 Team” ta boja je narančasta, kao što je prikazano na slici 5.14. Prelazak miša preko gumba također utječe na promjenu boje u boju tima. Svaki tim ima svoju boju što je prikazano na slici 5.15.

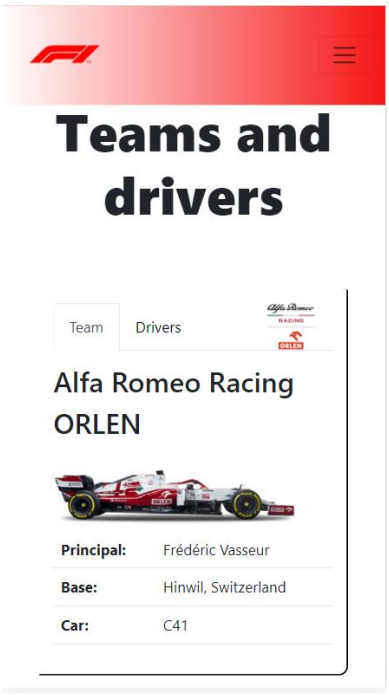



Sl. 5.14. Prikaz promjene boje ruba elementa pri prelaskom miša iznad tima „Mclaren F1 Team“

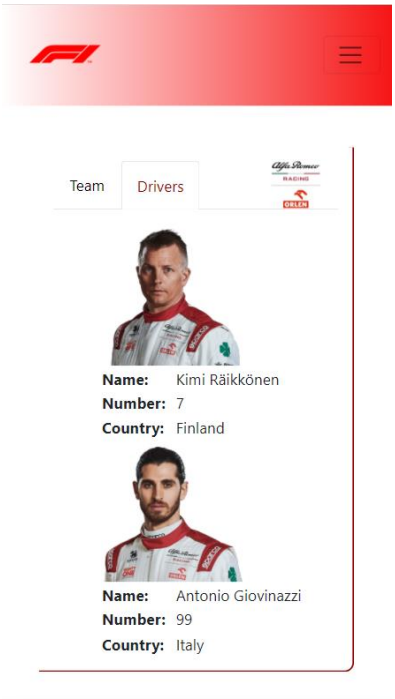
```
#C41:hover{
    border-color: #900000;
}

#A521:hover{
    border-color: #0090ff;
}

#AMR21:hover{
    border-color: #006f62;
}

#MCL35M:hover{
    border-color: #ff9800;
}

#RB16B:hover{
    border-color: #0600ef;
}

#AT02:hover{
    border-color: #2b4562;
}

#SF21:hover{
    border-color: #dc0000;
}

#VF-21:hover{
    border-color: #000000;
}

#FW43B:hover{
    border-color: #005aff;
}

#W12:hover{
    border-color: #00d2be;
}
```

Sl. 5.15. CSS kôd za mijenjanje boja ruba tima pri prelaskom miša iznad tima u datoteci „teamsAndDrivers.css“

Na slikama 5.16 i 5.17.  prikazan je mobilni prikaz stranice „Teams/Drivers“.




Sl. 5.16. Mobilni prikaz informacija o timovima na stranici „Teams/Drivers“




Sl. 5.17. Mobilni prikaz informacija o vozačima na stranici „Teams/Drivers“

# 6. ZAKLJUČAK

Sa rastom popularnosti Formule 1 raste i potreba za glavnim informacijama o sportu kao što je raspored utrka, informacije o stazama na kojima se utrke održavaju te informacije o timovima i vozačima.

Cilj ovog završnog rada bio je dizajn i izrada web aplikacije za evidenciju utrka Formule 1. Aplikacija intuitivno i jednostavno prikazuje gotove i nadolazeće utrke, detalje o utrkama ovisno jesu li se dogodile ili će se tek dogoditi. Dodatno prikazuje potpuni poredak timova i vozača te o njima nudi informacije za najveće ljubitelje sporta. Aplikacija je u potpunosti responzivna i funkcionalna tj. sama se ažurira i postavlja informacije koje su u skladu sa ažuriranim podacima u bazi podataka. Korisničko sučelje aplikacije napravljeno je pomoću HTML-a, CSS-a te Bootstrap-a. Serversko sučelje napravljeno je pomoću PHP-a, a baza podataka pomoću okruženja XAMPP i phpMyAdmin administrativnog alata za lakši dizajn baze podataka.

Ova aplikacije je spoj prednosti oba navedena postojeća rješenja jer ne zamara posjetitelja sa nepotrebnim medijskim sadržajem i dovoljno je lagan za korištenje i za starije osobe jer je Formula 1 poznata po obožavateljima starije dobi.

# LITERATURA


[1] Microsoft, »HTML For Beginners The Easy Way: Start Learning HTML & CSS Today,« 2021.. [Mrežno]. Dostupno na: https://html.com/#Welcome_You8217ve_Found_the_Easiest_Way_to_Learn_HTML_and_CSS. [Datum pristupa 12. 7. 2021.].

[2] W3Schools, »HTML introduction, HTML tutorial,« Refsnes Dana, 2021. [Mrežno]. Dostupno na: https://www.w3schools.com/html/html_intro.asp. [Datum pristupa 12. 7. 2021.].

[3] HTML.com, »HTML For Beginners The Easy Way: Start Learning HTML & CSS Today,« 2021.. [Mrežno]. Dostupno na: https://html.com/#Welcome_You8217ve_Found_the_Easiest_Way_to_Learn_HTML_and_CSS. [Datum pristupa 12. 7. 2021.].

[4] F. e. r. i. t. tehnologija, »LV1 - Uvod u HTML,« 2020.. [Mrežno]. dostupno na: https://moodle.srce.hr/2020-2021/mod/folder/view.php?id=1535646. [Datum pristupa 12. 7. 2021.].

[5] W3Schools, »CSS Tutorial,« 2021.. [Mrežno]. dostupno na: https://www.w3schools.com/css/. [Datum pristupa 12. 7. 2021.].

[6] M. W. Docs, »CSS: Cascading Style Sheets,« 2021.. [Mrežno]. dostupno na: https://developer.mozilla.org/en-US/docs/Web/CSS. [Datum pristupa 12. 7. 2021.].

[7] Bootstrap, »Introduction,« 2021.. [Mrežno]. dostupno na: https://getbootstrap.com/docs/5.0/getting-started/introduction/. [Datum pristupa 12. 7. 2021.].

[8] A. Friends, »XAMPP,« 2021.. [Mrežno]. dostupno na: https://www.apachefriends.org/index.html. [Datum pristupa 12. 7. 2021.].

[9] JavaTpoint, »XAMPP Tutorial,« 2021.. [Mrežno]. dostupno na: https://www.javatpoint.com/xampp. [Datum pristupa 12. 7. 2021.].

[10] TutorialsPoint, »PHP Introduction,« 2021.. [Mrežno]. dostupno na: https://www.tutorialspoint.com/php/php_introduction.htm. [Datum pristupa 12. 7. 2021.].

[11] W3Schools, »PHP MySQL Database,« 2021.. [Mrežno]. dostupno na: https://www.w3schools.com/php/php_mysql_intro.asp. [Datum pristupa 12. 7. 2021.].

# SAŽETAK

Tema završnog rada je izraditi web aplikaciju za evidenciju Formula 1 utrka. Za izradu korisničkog sučelja koristili su se HTML, CSS i Bootstrap, a za serversko sučelje PHP i XAMPP. U radu su objašnjene tehnologije korištene za izradu te je postupak izrade aplikacije također detaljno objašnjen uz mnoštvo slika koje prikazuju dijelove programa. Aplikacija se sastoji od 3 zasebne stranice. Početna stranica prikazuje kronološki raspored utrka i prikazuje različite informacije ovisno o tome dali je utrka završena ili se još nije dogodila. Na stranici „Standings“ prikazane su tablice timova i vozača poredani po bodovima te na stranici „Teams/Drivers“ prikazane su najbitnije informacije o timovima i vozačima.

Ključne riječi: Evidencija, Formula 1, HTML, web aplikacija, XAMPP

# ABSTRACT

**Web application for tracking Formula 1 races**

Theme of this undergraduate thesis is a web application for tracking Formula 1. HTMl, CSS and Bootstrap were used for creating the front-end, and XAMPP and PHP for the back-end. Paper offers information on the technologies used to create the application and a detailed explanation of the process of making the application with pictures that show pieces of code. Application is made out of three pages. First is the main page that shows the race schedule and shows information dependent on whether the race happened or it is yet to happen. Page "Standings" shows driver and team standings and on the last page "Teams/Drivers" most important information about the teams and drivers is shown.



Key words: Formula 1, HTML, schedule, web application, XAMPP

# ŽIVOTOPIS

Luka Šarčević rođen je 25. kolovoza 1999. godine u Pakracu. Završio je Osnovnu školu braće Radića u Pakracu 2014. godine. Za vrijeme osnovnog školovanja, 2013. godine završava Osnovnu glazbenu školu u Pakracu, instrument gitara. Iste godine upisuje Srednju školu Pakrac, smjer opća gimnazija koju završava 2018. godine sa vrlo dobrim uspjesima. Nakon položene mature upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer preddiplomski studij računarstva. Posjeduje znanje u govoru, čitanju i pisanju engleskog i njemačkog jezika te vozačku dozvolu B kategorije. Također posjeduje znanje programskih jezika HTML, CSS, PHP, MySQL, Bootstrap, C, C++, C#, JAVA, Matlab, tehnologija Visual Studio, Android Studio i upravljanje Microsoft Office alatima.

Luka Šarčević

_____

# PRILOZI

- Uz printanu verziju završnoga rada nalazi se i optički disk
    - Optički disk sadrži: .docx i .pdf formate završnoga rada, mapu izvornog kôda i sve slike koje se nalaze u bazi podataka
- Kôd se također može pronaći na Git repozitoriju na linku: https://gitlab.com/luka_sarcevic/zavrsni-rad-web-aplikacija-za-evidenciju-formule-1