

Vizualizacija podataka

**Statistika Leicester City**

Projektni zadatak

Kristman Denis

Osijek, 2016.

**Sadržaj**

[**1.** **Uvod** 3](#_Toc455500296)

[**2.** **Opis projektnog zadatka** 4](#_Toc455500297)

[**2.1** **Projektni zadatak** 4](#_Toc455500298)

[**3.** **Programsko rješenje** 5](#_Toc455500299)

[**4.** **Korištene tehnologije** 11](#_Toc455500300)

[**5.** **Zaključak** 12](#_Toc455500301)

[**Literatura** 13](#_Toc455500302)

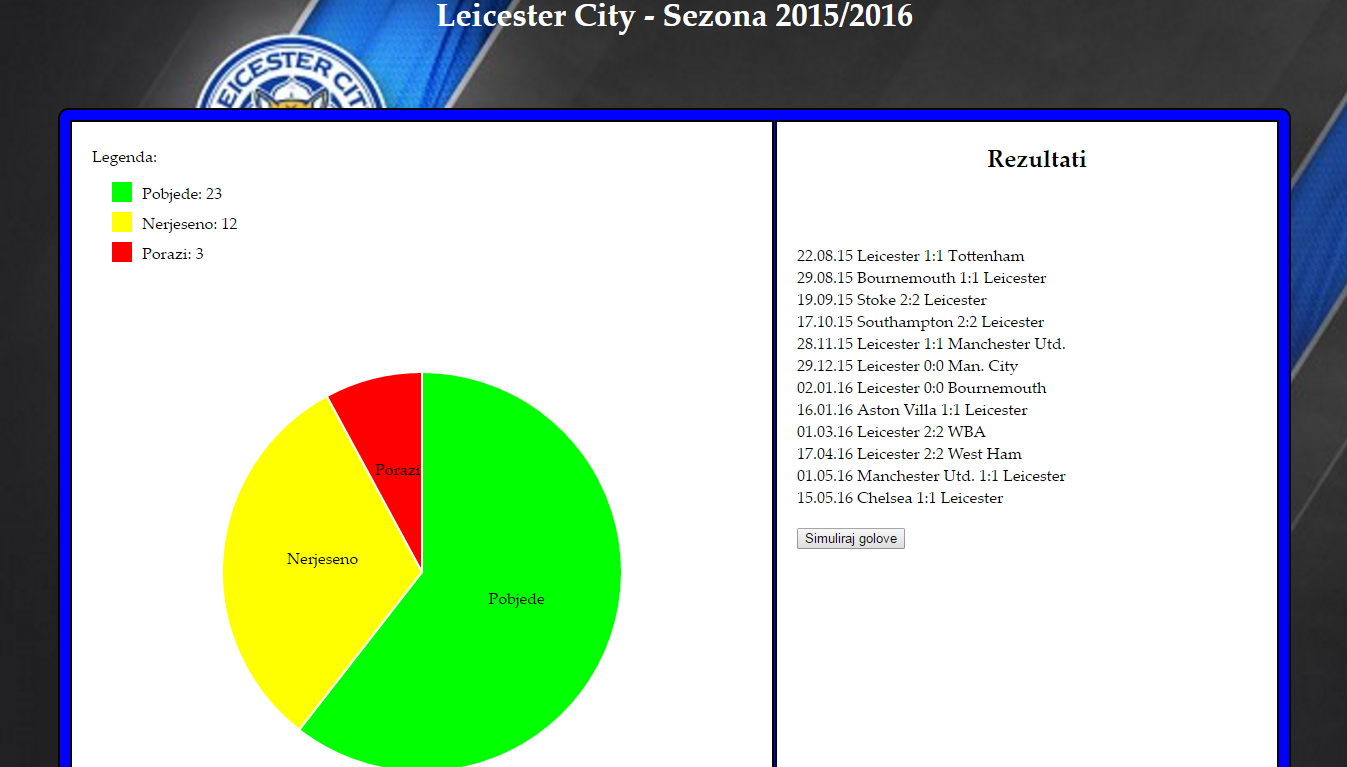
1. **Uvod**

Vizualizacija podataka može se opisati kao proučavanje vizualne reprezentacije podataka koji su definirani kao informacije zamišljene u nekom shematskom obliku. To je oblik grafičke interpretacije podataka sa ciljem lakšeg razmatranja odnosa (relacija) između podataka, uključujući atribute ili varijable za jedinicu informacija. Jednostavno rečeno cilj vizualizacije podataka je olakšavanje analize prikazanih podataka.

U ovome radu opisan je način izrade vizualizacije podataka za nogometni klub Leicester City, odnosno prikaz rezultata utakmice tijekom nogometne sezone 2015/2016. pomoću „pie Chart“ prikaza, kao i simulacija golova jedne utakmice iz navedenog skupa podataka.

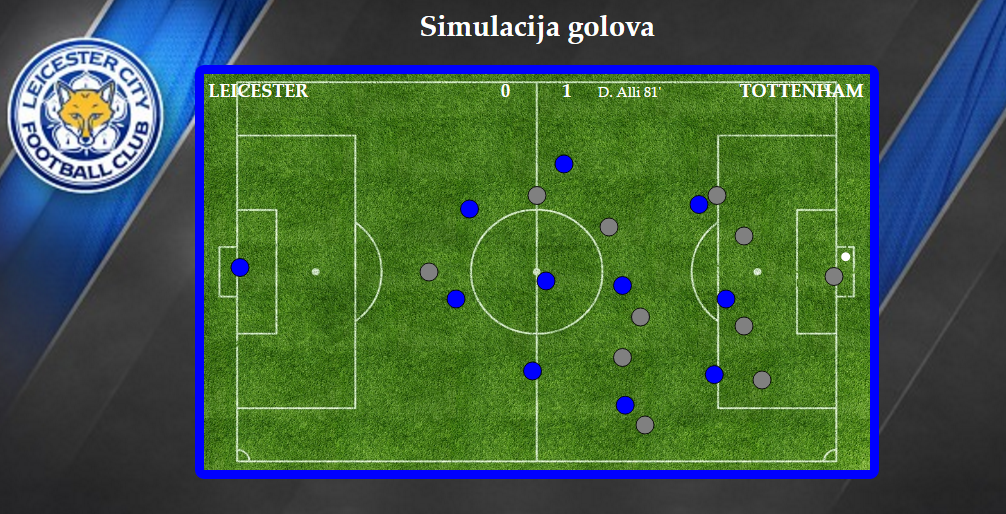
1. **Opis projektnog zadatka**
   1. **Projektni zadatak**

Tema projekta je prikaz korištenja vizualizacije podataka, izrađena je vizualizacija podataka za nogometni klub „Leicester City“. Oblikovanje elemenata web stranice odrađeno je u CSS datoteci „leicester.css“ dodijeljivanjem vrijednosti selektorima koji su prethodno određeni u html datoteci za svaki element. Prikaz vizualizacije odrađen je pomoću kružnog „Pie Chart“ grafa. Klikom na pojedini segment grafa omogućen je prikaz skupa podataka koji se nalazi u tom segmentu kružnog grafa, također za navedeni graf postoji legenda koja bolje opisuje pojedine segmente. Prikazano na slici 1.



Slika 1. Prikaz grafa i ispis rezultata

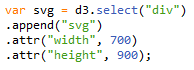
Uz sami prikaz rezultata svih utakmica nogometnog kluba tijekom sezone postoji mogućnost simulacije golova sa jedne nogometne utakmice što je omogućeno pomoću više JavaScript tranzicija. Prikazano na slici 2.



Slika 2. Prikaz simulacije golova

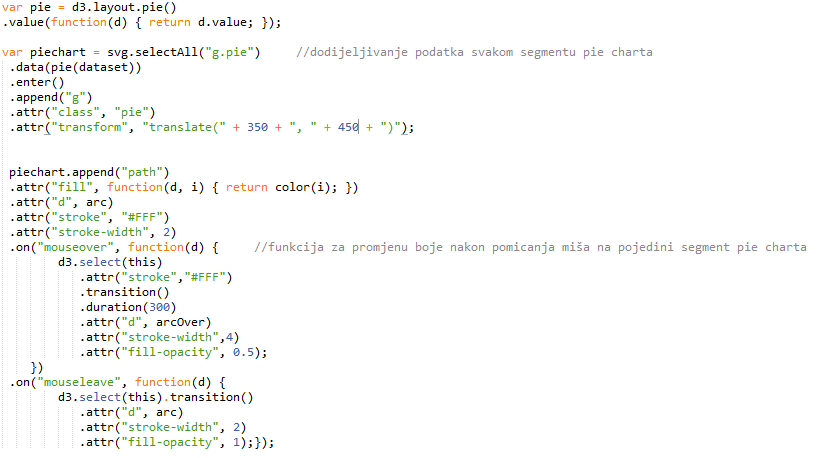
1. **Programsko rješenje**

Skup podataka korišten za izradu ove vizualizacije imeplementiran je u html dokument, odnosno u tri različite varijable: dobiveneUtakmice, nerjeseneUtakmice, izgubljeneUtkamice. Svaki od tih podatkovnih skupova implementiran je u jedan od segmenata kružnog grafa. Vizualizacija podataka odnosno „Pie Chart“ prikaz nalazi se unutar jednog svg elementa koji je dodan na web stranicu pomoću D3 JavaScript biblioteke. Prikazano na slici 3.



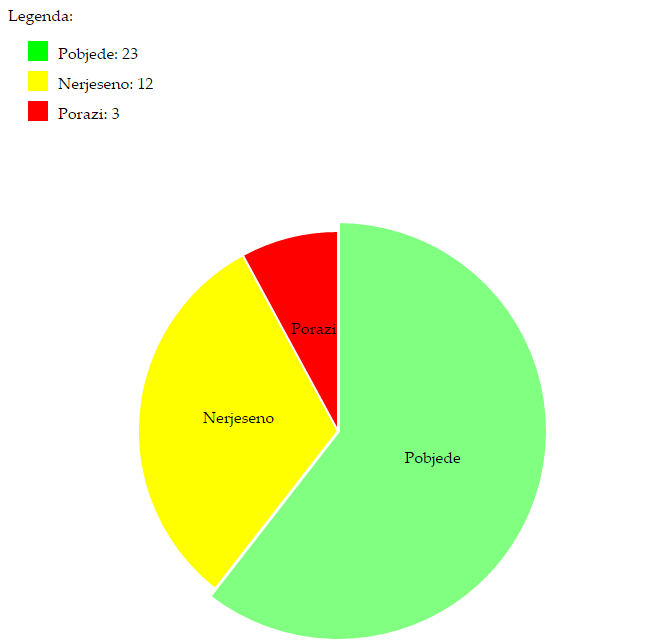
Slika 3. Dodavanje SVG elementa

U samome SVG elementu prikazan je „Pie Chart“ u kojemu se nalaze podaci o odigranim utakmicama, te legenda koja malo bolje prikazuje pojedini segment grafa odnosno konkretne vrijednosti koje se u njemu nalaze. Na slici 4. je prikazan programski kod za prikazivanje navedenog kružnog grafa.



Slika 4. Programski kod za „Pie Chart“

U prikazanom programskom kodu također se može vidjeti da je poboljšana sama vizualizacija grafa tako što je omogućeno da nakon postavljanja samog miša na pojedini segment dolazi do povećanja tog segmenta što je određeno samom varijablom „arcOver“ te se smanjuje vidljivost segmenta pomoću atributa „fill-opacity“. Sve je odrađeno pomogu tranzicija iz D3 JavaScript biblioteke. Nakon što maknemo miš sa segmenta taj će se dio grafa vratiti u prvobitni oblik, što je također omogućeno pomoću tranzicije, odnosno smanjivanjem veličine tog segmenta na prvobitnu te postavljanjem „fill-opacity“ na 1 kako bi se vratila boja koju smo imali na početku prikaza. Prikaz grafa nakon postavljanja miša na njega prikazan je na slici 5.



Slika 5. Vizualizacija segmenta „Pobjede“

Na prethodnoj slici vidljivo je da na svakom dijelu grafa postoji naziv, ali nisu poznate konkretne vrijednosti koje opisuju pojedine segmente. U tu svrhu prikazana je legenda za ovaj graf kako bi bilo poznato koliko je pobjeda ostvario Leceister City, isto tako nerješenih i poraženih utakmica. Prikazano na slici 5.

Legenda je izrađena pomoću „g“ elemenata iz D3 JavaScript biblioteke. Svaki od tih elemenata ima svoju kockicu sa bojom koja je jednaka onom segmentu iz grafa kojeg detaljnije opisuje. Boje su implementirane pomoću ručno napisane ordinalne palete boja, koja se koristi za graf i legendu (prikazano na slici 7.). Pokraj svake kockice prikazan je tekst sa brojem utakmica za pojedini dio grafa, tekst je dodan u svaki g element te su mu određene koordinate po xy osi. Programski kod za izradu legende za graf prikazan je na slici 8.



Slika 7. Ordinalna skala boja za graf i legendu



Slika 8. Programski kod za izradu legende

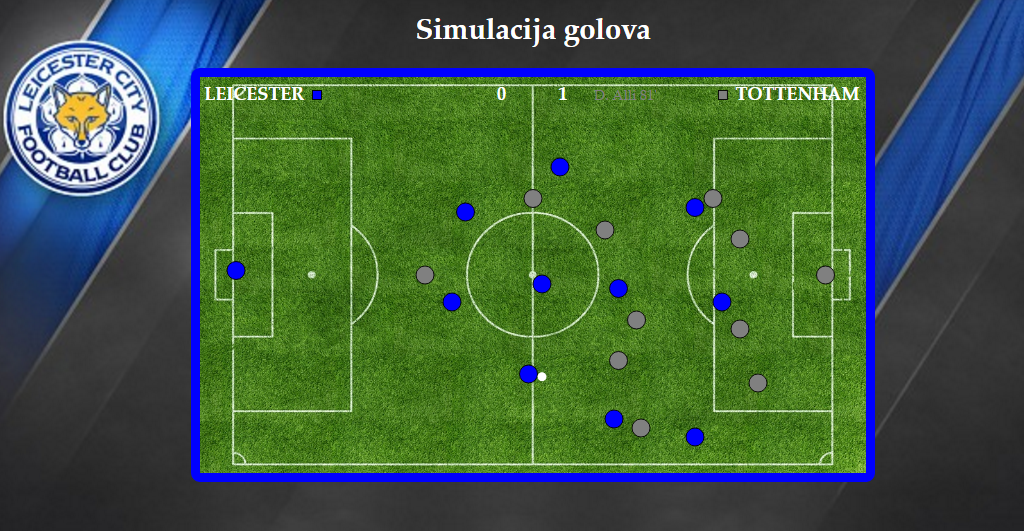
Nakon što je odrađena konkretna vizualizacija podataka potrebno je prikazati skupove podataka koji se nalaze unutar samog grafa. Klikom na pojedini dio grafa prikazuju se rezultati utakmica u zasebnom HTML blok elementu „div“. Kako bi se rezultati kvalitetno prikazali potrebno je dodati na graf jednu funkciju koja se pokreće na klik miša odnosno „on click“ funkciju. Problem je kako će funkcija prepoznati koji je dio grafa kliknut te je to odrađeno pomoću tri uvjeta, klikom na određeni dio grafa provjerava se koja je vrijednost tog dijela odnosno da li je to broj 3 za izgubljene utakmice, 12 za nerješene ili 25 za pobjede. Na taj način funkcija prepoznaje na koji smo dio kliknuli. Ukoliko smo kliknuli na dio grafa „Pobjede“ uspoređuje se vrijednost tog podatkovnog skupa koja je 25 sa „if grananjima“ te se može primjetiti da se zadovoljava prvo „if grananje“ gdje je „value == 25“. Nakon toga potrebno je osigurati da u „div“ bloku nema nepotrebnih „text“ elemenata te se zbog toga prvo brišu svi „text“ elementi iz „div“ bloka prije upisivanja novih. Kada je osigurano da nema tekstualnih elemenata u bloku potrebno je u „.data“ proslijediti podatakovni skup koji ćemo ispisati“, te se nakon toga pomoću funkcije upisuje jedan po jedan element skupa u „text“ element koji će se prikazati.

Kako bi se osiguralo da se podaci ne brišu i ne ispisuju ponovno za neki dio grafa ukoliko su već prikazani dodane su bool varijable u if uvjet zbog prepoznavanja da li je određeni dio već prikazan. Programski kod za ispis rezultata prikazan je na slici 9.



Slika 9. Programski kod za ispis rezultata

Na prethodnoj slici može se primjetiti da u drugom if grananju postoji dio koda za prikaz tipke koja nas prebacuje na simulaciju golova. Simulacija golova odrađena je na jednoj utakmici iz skupa nerješenih utakmica. Vidljivo je da klikom na „button“ otvara se link na datoteku „simulacija\_gola.html“ gdje je prikazana simulacija. Simulacija će se otvoriti u novoj kartici što je određeno atributom „target“ i njegovom vrijednosti „blank“. Izgled prozora simulacije može se pogledati na slici 10.



Slika 10. Izgled simuacije nakon klika na tipku

Pogledom na samu fotografiju može se zaključiti da je sam izgled nogometnog terena odrađen pomoću SVG elementa iz D3 JavaScript biblioteke. Plavi obrub terena određen je u samoj CSS datoteci pomoću atributa border i njegovim pripadajućim vrijednostima.

Igrači su napravljeni pomoću SVG elemenata „circle“ koji imaju pripadajući atribut radijus.Svaki igrač ima svoje koordinate prije početka akcije te se pomoću više tranzicija igrači pomiču na druge koordinate. Kod svake tranzicije potrebno je odrediti njen „delay“ odnosno kada će se ona pokrenuti kao i njeno trajanje „duration“ koje je označeno u milisekundama. Tranzicije se neće pokretati ovisno o tome kojim su redosljedom napisane u kodu nego ovisno o „delay-u“ pa je potrebno tranzicijama koje bi se trebale dogoditi kasnije, potrebno odrediti ispravan „delay“. Prikaz programskog koda za neke od igrača sa njihovim pripadajućim tranzicijama nalazi se na slici 11.



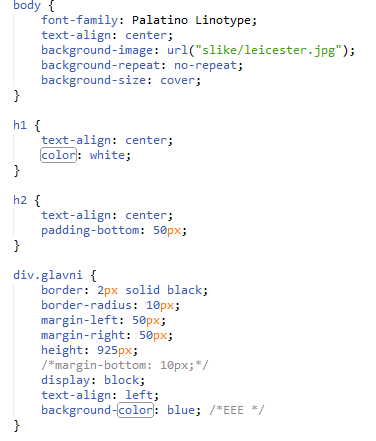
Slika 11. Programski kod za igrače

Rezultat i nazivi klubova prikazani su dodavanjem tekstualnih elemenata u SVG. Svaki element ima svoje xy koordinate. Pored imena klubova nalaze se dvije kockice (rect elementi) koje označavaju koja boja igrača njima pripada. Nakon postignutog pogotka ponovno se izvode tranzicije nad tekstualnim elementima rezultata te im se mijenja atribut „text“.

1. **Korištene tehnologije**

Kao što se već moglo zaključiti jedna od osnovnih komponenti ovog projektnog zadatka je korištenje D3 JavaScript biblioteke. D3.Js biblioteka nam omogućuje manipulaciju HTML elementima prvenstveno u smislu jednostavne i brze izrade vizualizacija za web. Temelji se na jednostavnoj manipulaciji HTML, CSS i SVG elemenata/objekata putem JavaScripta.

Kompletno oblikovanje web stranice ovog projekta odrađeno je u sklopu CSS „leicester.css“ datoteke u kojoj se nalazi oblikovanje svih elemenata vizualizacije grafa, prikaza rezultata kao i simulacije golova. Korišteni su jednostavni, klasni i id selektori. Dio koda iz css datoteke za oblikovanje nekih od elemenata ovog projekta vidljiv je na slici 12.



Slika 12. Dio oblikovanja iz „leicester.css“

Na prikazanoj slici može se vidjeti već spomenuto korištenje jednostavnog selektora za „body“ gdje je određena pozadina projekta i njena svojstva kao i poravnanje i font teksta. Id selektor korišten je pri oblikovanju glavnog blok elementa projektnog zadatka gdje se može vidjeti plava pozadina koja se mogla prethodno uočiti na slikama projekta.

1. **Zaključak**

Projektni zadatak je uspješno odrađen zahvaljujući prethodno stečenim znanjima iz kolegija „Vizualizacija podataka“ kao i kvalitetnim laboratorijskim vježbama iz navedenog kolegija. Vidljivo je da je u sklopu vizualizacije podataka na webu izuzetno bitno korištenje D3.Js JavaScript biblioteke te je ona jedan od neizostavnih elemenata kvalitetne vizualizacije podataka. Vizualizacijom podataka omogućeno je jednostavan uvid u podatku i njihovu međusobnu povezanost kao i lakšu analizu tih podataka.

Na odrađenom projektnom zadatku pomoću kružnog grafa lagano se može zaključiti kakve je rezultate ostvarila navedena momčad i bez prethodnog uvida u konkretne brojeve, te koji je odnos između pobjeda i izgubljenih utakmica. Nije potrebno gledati skupove podataka i što se u njima nalazi nego je omogućen jednostavan uvid u njih pomoću kružnog grafa. Može se zaključiti da je Leicester City ostvario dobre rezultate u prošloj sezoni u svega nekoliko sekundi pogleda na ovaj „Pie Chart“.

Kod simulacije golova izuzetno je bitno paziti na tranzicije koje se izvode kod svakog igrača kako bi to izgleda dobro u realnom vremenu, te je potrebna izuzetna preciznost u usklađivanju kretanja pojedinih igrača. Sigurno je da u tome segmentu projekta stoji velika mogućnost napretka i unaprijeđivanja.

# **Literatura**

[1] Predlošci iz laboratorijskih vježbi iz kolegija „Vizualizacija podataka“.

[2] <https://d3js.org/>

[3] <http://www.w3schools.com/>