SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij

VIZUALIZACIJA PODATAKA

Vizualizacija podataka o statistici rangiranih igrača unutar igre Apex Legends

Seminarski rad

Andreja Nađ

Osijek, 2021.

##### SADRŽAJ

[1. UVOD 1](#_Toc82250974)

[1.1. Zadatak završnog rada 2](#_Toc82250975)

[2. UPRAVLJANJE PAMETNOM KUĆOM PRILAGOĐENOM OSOBAMA S INVALIDITETOM I PREGLED STANJA U PODRUČJU 3](#_Toc82250976)

[2.1. Problemi osoba s invaliditetom 3](#_Toc82250977)

[2.2. Pametna kuća 3](#_Toc82250978)

[2.2.1. Sustavi pametne kuće 4](#_Toc82250979)

[2.2.2. Prednosti i nedostaci pametne kuće 5](#_Toc82250980)

[2.3. Izazovi upravljanja pametnom kućom za osobe s invaliditetom 7](#_Toc82250981)

[2.4. Trenutno stanje u području upravljanja pametnom kućom za osobe s invaliditetom 7](#_Toc82250982)

[2.4.1. Izvedba pametne kuće pomoću internet stvari 7](#_Toc82250983)

[2.4.2. Izvedba pametne kuće Kinect senzorom 9](#_Toc82250984)

[2.4.3. Aplikacije i digitalni asistenti 10](#_Toc82250985)

[3. PRIJEDLOG MODELA SUSTAVA PAMETNE KUĆE PRILAGOĐENE OSOBAMA S INVALIDITETOM 12](#_Toc82250986)

[3.1. Zahtjevi na sustav 12](#_Toc82250987)

[3.1.1. Zahtjevi na upravljački dio 12](#_Toc82250988)

[3.1.2. Zahtjevi na mobilnoj aplikaciji 12](#_Toc82250989)

[3.2. Sklopovska arhitektura sustava pametne kuće 13](#_Toc82250990)

[3.3. Programska arhitektura sustava pametne kuće 14](#_Toc82250991)

[3.4. Načini multimodalnog upravljanja pametnom kućom 14](#_Toc82250992)

[3.5. Postupci obrade prirodnog jezika 15](#_Toc82250993)

[4. RJEŠENJE ZA UPRAVLJANJE PAMETNOM KUĆOM 17](#_Toc82250994)

[4.1. Korištene tehnologije, programski jezici i razvojna okolina 17](#_Toc82250995)

[4.1.1. Sklop Croduino NOVA2 17](#_Toc82250996)

[4.1.1. Računalna platforma Arduino 17](#_Toc82250997)

[4.1.2. Biblioteke ESPAsyncWebServer i ESPAsyncTCP 17](#_Toc82250998)

[4.2. Opis korištenog sklopovlja 18](#_Toc82250999)

[4.3. Programsko rješenje upravljanja pametnom kućom 19](#_Toc82251006)

[4.3.1. WebServer i WebSocket 19](#_Toc82251007)

[4.3.2. Pristupna točka bežične mreže 19](#_Toc82251008)

[4.3.3. Upravljanje sustavom 20](#_Toc82251009)

[5. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE 21](#_Toc82251010)

[5.1. Korištene tehnologije, programski jezici i razvojna okolina 21](#_Toc82251011)

[5.1.1. Program Android Studio 21](#_Toc82251012)

[5.1.2. Fragmenti 21](#_Toc82251013)

[5.1.3. Klasa WebSocket 22](#_Toc82251014)

[5.1.4. Klasa AsyncTasks 22](#_Toc82251015)

[5.1.5. Lista RecyclerView 22](#_Toc82251016)

[5.1.6. Klasa SpeechRecognizer 23](#_Toc82251017)

[5.1.7. Klasa TextToSpeech 23](#_Toc82251018)

[5.2. Programsko rješenje na strani klijenta 23](#_Toc82251019)

[5.3. Programsko rješenje na strani poslužitelja 26](#_Toc82251020)

[5.3.1. Fragment Home 26](#_Toc82251021)

[5.3.2. Fragment Settings 30](#_Toc82251022)

[5.3.3. Fragment Device 32](#_Toc82251023)

[5.3.4. Pametne preporuke 33](#_Toc82251024)

[5.4. Komuniciranje u sustavu 35](#_Toc82251025)

[6. OPIS NAČINA RADA I ISPITIVANJE SUSTAVA 37](#_Toc82251030)

[6.1. Opis načina rada sustava 37](#_Toc82251031)

[6.2. Ispitivanja i rezultati sustava 39](#_Toc82251032)

[6.2.1. Ispitni slučaj 1 39](#_Toc82251033)

[6.2.2. Ispitni slučaj 2 39](#_Toc82251034)

[6.2.3. Ispitni slučaj 3 40](#_Toc82251035)

[6.3. Analiza rezultata ispitivanja sustava 41](#_Toc82251036)

[ZAKLJUČAK 42](#_Toc82251037)

[LITERATURA 43](#_Toc82251038)

[SAŽETAK 46](#_Toc82251039)

[ABSTRACT 47](#_Toc82251040)

[ŽIVOTOPIS 48](#_Toc82251041)

[PRILOZI (na DVD-u) 49](#_Toc82251042)

# UVOD

Izrada ovog projekta svodi se na prikaz podataka koji su sakupljeni sa web stranice „Apex Legends Status“ koja ima bazu od više od trinaest milijuna igrača. Podatci se skupljaju i izmijenjuju svake minute te su podatci korišteni u ovom radu uzeti sa iste stranice u malom skupu. Statistika se odnosi na prikaz rangiranih igrača i najčešće korištenih likova unutar rangiranih igara.

# KORIŠTENE TEHNOLOGIJE

## Biblioteka D3.js

D3.js je JavaScript biblioteka za manipulaciju dokumentima na temelju podataka. D3 omogućuje povezivanje proizvoljnih podataka s objektnim modelom dokumenta (DOM), te primijenjivanje transformacije vođene podacima na dokument.

## SVG

SVG je skraćenica od Scalable Vector Graphics. SVG definira vektorsku grafiku za web u XML formatu. Svaki element i svaki atribut u SVG datotekama može se animirati. SVG se integrira s W3C standardima kao što su DOM i XSL

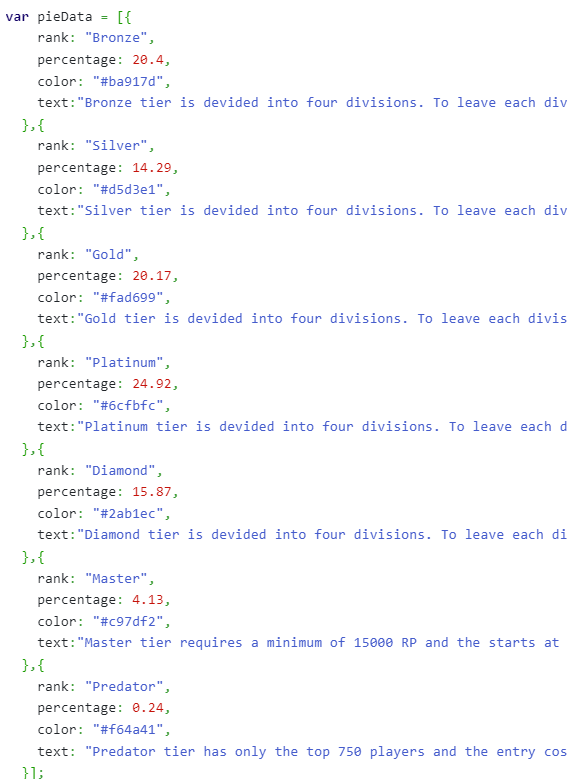
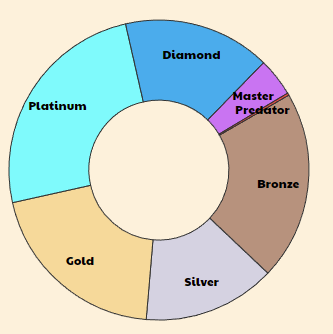
## JavaScript

JavaScript je najpopularniji svjetski programski jezik weba. JavaScript je skriptni programski jezik, koji se izvršava u web pregledniku na strani korisnika. Napravljen je da bude sličan Javi, zbog lakšega korištenja, ali nije objektno orijentiran kao Java, već se temelji na prototipu i tu prestaje svaka povezanost s programskim jezikom Java. JavaScript je lako naučiti, ne morata se preuzimati i besplatan je.

# PROGRAMSKO RJEŠENJE

Prilikom izrade prvo je potrebno odrediti bazu podataka. Pošto ovaj rad ima tri različita prikaza, potrebno nam je minimalno toliko i podataka.

Prvi prikaz je kružni graf koji je prikazan na slici 1 koristi podatke čiji je dio prikazan na istoj slici.



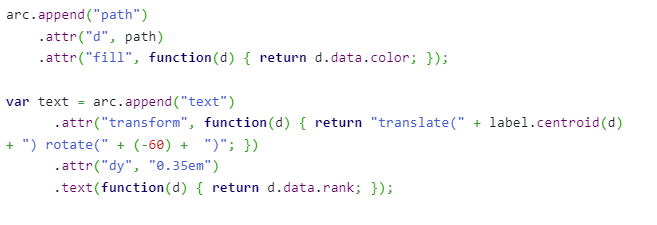
*Slika 1. Kružni prikaz postotka rangiranih igrača sa rupom i korišteni podaci*

Varijabla *pieSvg* je vezana na svg element određen unutar *html*-a. Taj element već ima određene dimenzije te se one dobivaju koristeći funkciju *attr()* sa odgovarajućim nazivom atributa. Osim toga potreban je radius i *g* element koji predstavlja cijeli kružni graf sa početnom pozicijom rotacije. Unutar tog *g* elementa se nalaze drugi *g* elementi klase *arc* koji predstavljaju odvojene dijelove kružnog grafa. Njihove dimenzije ovise o danim podatcima te su automatski dodjeljene. Varijabla *percentageText* je varijabla na koju će se zapisivati trenutačni postupak dijela kružnog grafa kada se pređe preko njega putem pokazivača.



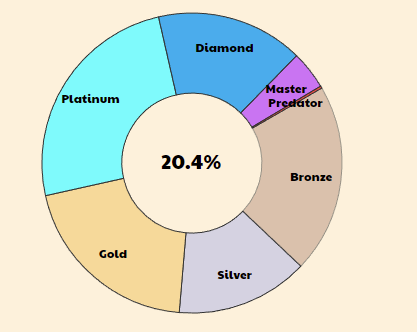
*Slika 2. Kreiranje kružnog grafa 1/2*

Prije spomenuti *g* elementi klase *arc* koji predstavljaju odvojene dijelove kružnog grafa su roditelji *path* elementa koji zapravo ima koordinate i dimenzije svakog dijela kružnog grafa. Gledajući da su *path* elementi isto što su *bar* elementi kod stupičastih grafova, onda na njih primijenjujemo elemente koje želimo da se prikazuju na određenom dijelu ili atribute za određeni dio kao što su boja ili slika na elementu. Varijabla text predstavlja naslov svakog ranga koji se pozicionira na dijelove kružnog prikaza.



*Slika 3. Kreiranje kružnog grafa 2/2*

Kao prije spomenuto, na *path* element postavljamo atribute i funkcije vezane za te elemente. Neke od njih su funkcije *on()* koje se bave događajima kao na primjer kada pokazivač pređe preko elementa, klikne na element ili makne se sa elementa. Prijelaz preko elementa je prikazan na donjoj slici gdje se traži trenutačni naziv elementa i putem toga se prikazuje postotak ranga elementa unutar prije spomenute varijable *percentageText*.

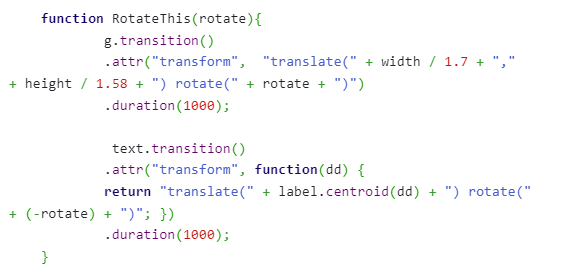
*Slika 4. Mouseover funkcionalnost na kružnom grafu*

Funkcija *on()* koja se bavi događajem na klik je prikazana dolje. Putem nje se rotira kružni graf tako da se kliknuti dio kruga prikaže prema desno gdje se nalazi div na kojem se prikazuje opisni tekst ovisan o dijelu koji je pritisnut, te se ispod teksta nalazi još jedan kružni graf koji se mijenja ovisno o kliknutom elementu.



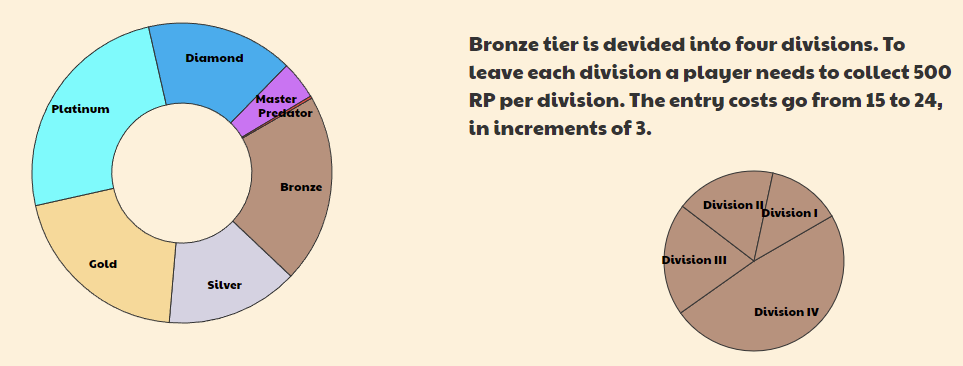
*Slika 5. Click funkcionalnost na kružnom grafu*

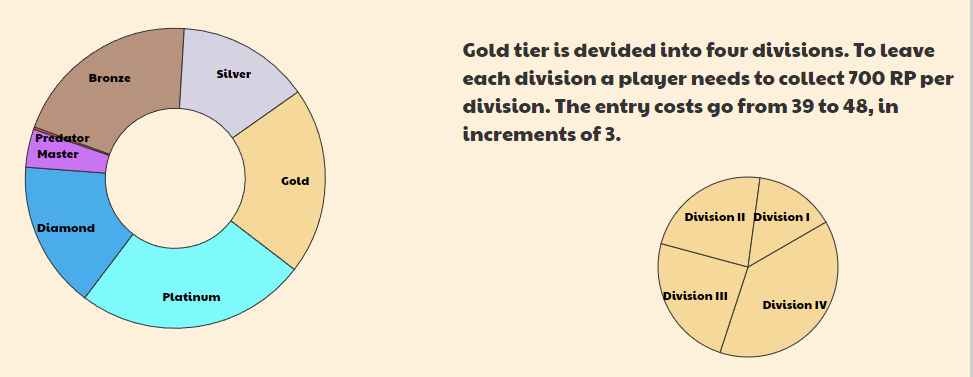
Funkcija *RotateThis()* rotira kružni graf te sve labele dijelova u zadanom broju.



*Slika 6. Funkcija RotateThis()*

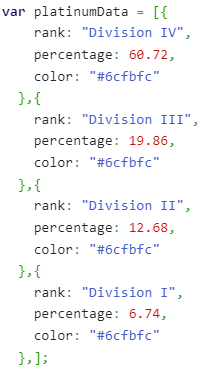
Slika 7 prikazuje početno stanje stranice i stanje kada se klikne na *„Gold“* dio u kružnom grafu.





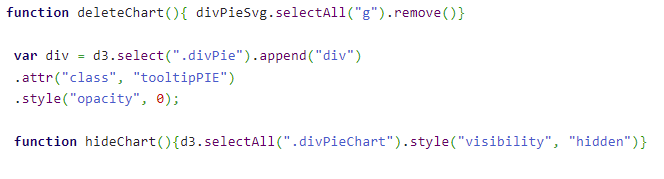
*Slika 7. Prikaz rada funkcije RotateThis() i click funkcionalnosti*

Drugi kružni graf ima više podataka koji se koriste te se podaci mijenjaju prema kliknutom elementu u prvom kružnom grafu.



*Slika 8. Prikaz jednog dijela podataka za drugi kružni graf*

Važne funkcije za rad ovog grafa su funkcija *deleteChart()* koja je odgovorna za brisanje prethodnog grafa prije prikaza novog. Ako se prethodni ne obriše, grafovi će se gomilati jedan na drugoga. Druga važna funkcija je *ChangeChart()* koja kreira kružni graf kao i prethodni. Ovaj graf ima dodatni alat za opis koji se aktivira kada pokazivač pređe preko kružnog grafa.



*Slika 9. Funkcija deleteChart()*

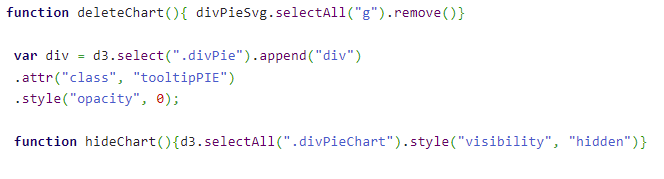


*Slika 10. Prvi dio funkcije ChangeChart() za kreiranje kružnog prikaza prema predanim podacima*

Alat za opis koji se aktivira prikazuje postotak elementa te se pomiče za pokazivačem tako da ga prati. Pozicija pokazivača se dobiva putem *event* varijable koja predstavlja objekt klase *MouseOver* te sadrži atribute *clientX* i *clientY* koji predstavljaju koordinate pokazivača.

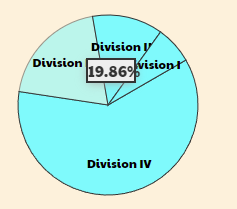
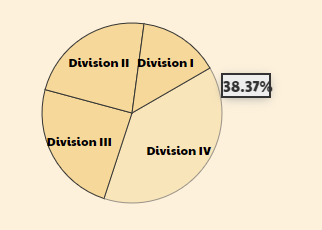


*Slika 11. Drugi dio funkcije ChangeChart() - funkcionalnost prelaska pokazivčala*



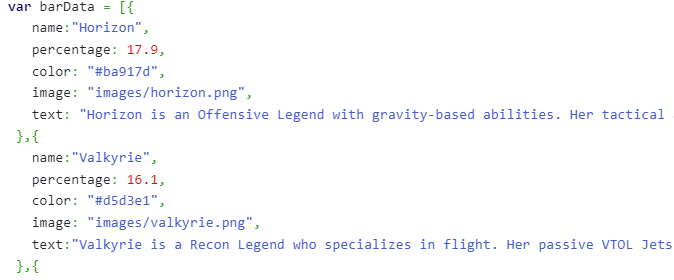
*Slika 12. Osnovne postavke za kreiranje alata za opis*

Slika 13 prikazuju funkcionalnost alata za opis koji prati pokazivač.



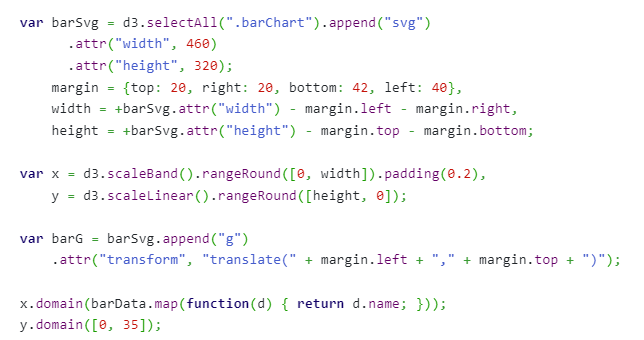
*Slika 13. Prikaz rada alata za opis i ChangeChart() funkcije*

Treći i zadnji graf je stupičasti graf čiji je dio podataka prikazan na slici ispod.



Slika 14. Prikaz dijela podataka za stupičasti graf

Slika 15 prikazuje kreiranje stupičastog grafa zadavanjem njegovih granica.



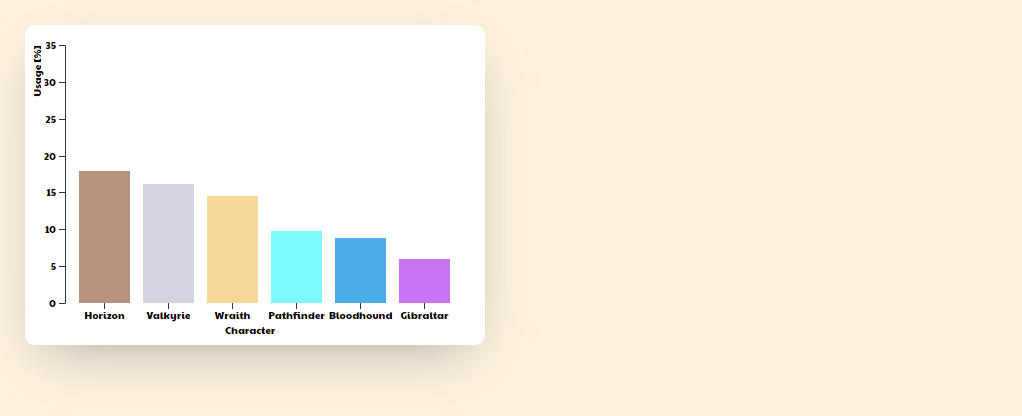
*Slika 15. Kreiranje stupičastog grafa 1/2*

Slika 16 prikazuje dodjeljivanje osi grafu i elementa na osima kao i iscrtavanje stupaca bazirano na danim podacima.



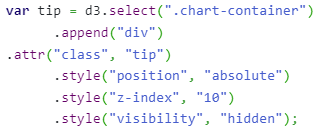
*Slika 16. Kreiranje stupičastog grafa 2/2*

Slika 17 prikazuje prikaz stupičastog grafa bez ikakvih dodatnih mogućnosti.



Slika 17. Osnovni prikaz grafa

Iznad stupa na kojem je pokazivač, pokazuje se alat za opis sličan prethodnom. Aktiviran je metodom *onMouseOver()* koja isto tako aktivira funkciju *imageAdd()* kojanadodaje sliku dimenzija 100x100 iznad alata za opis.

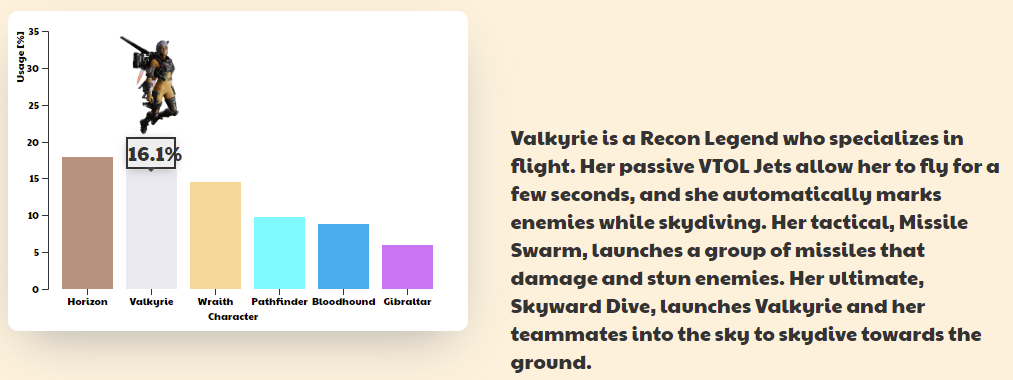
 

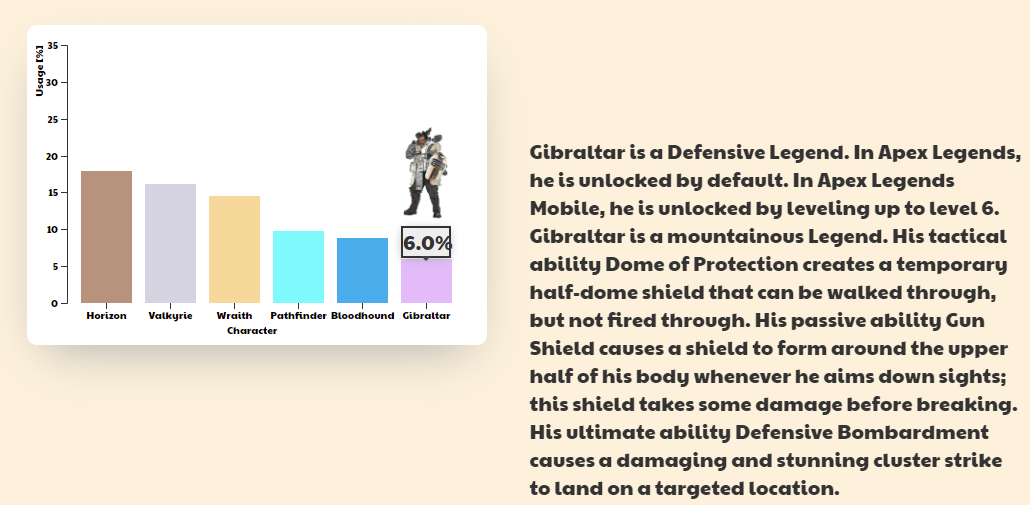
*Slika 18. Alat za opis i funkcija imageAdd()*



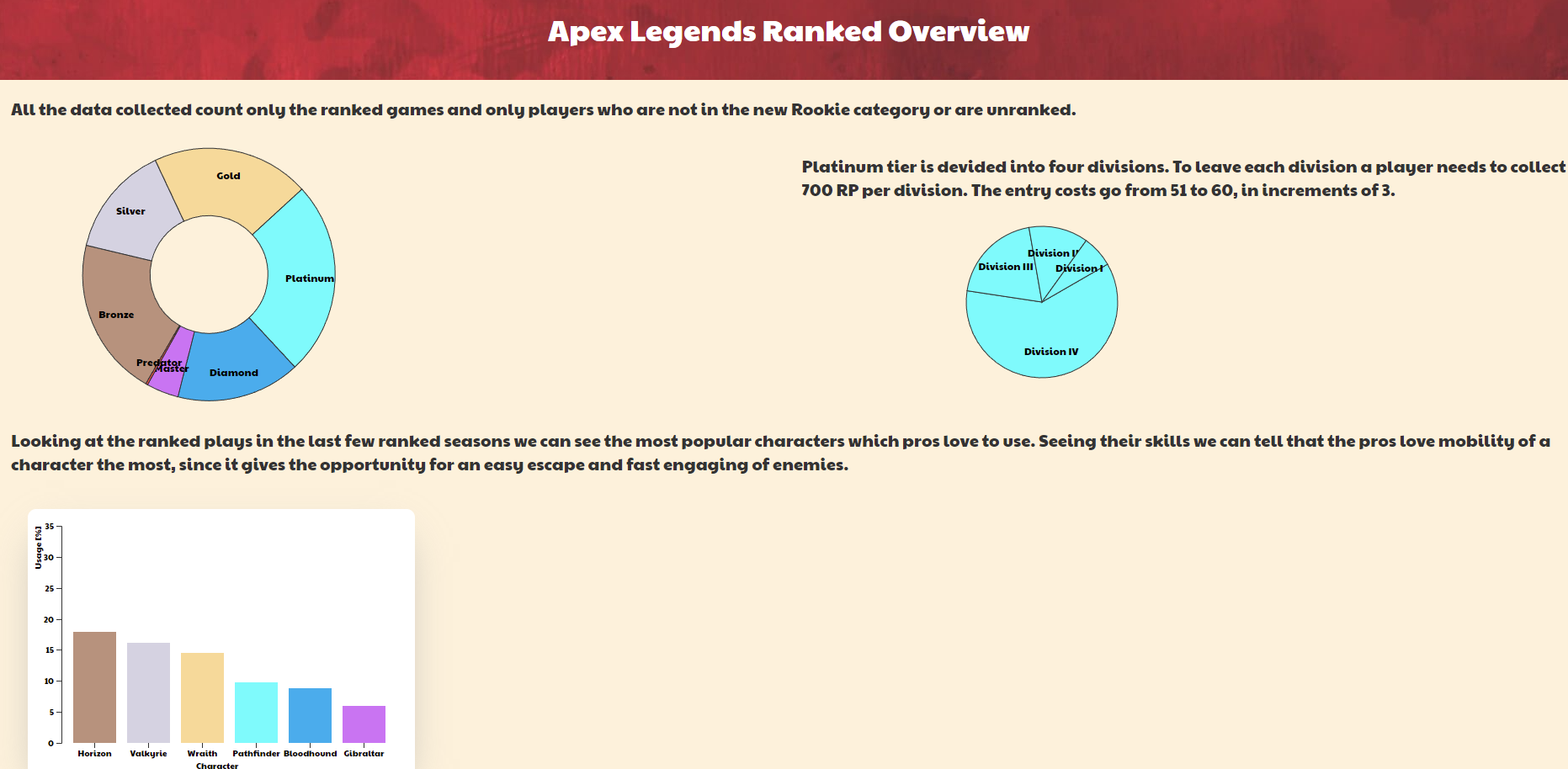
*Slika 19. Funkcija onMouseOver()*

Slika 20 prikazuje dva primjera rada funkcije *onMouseOver().*





*Slika 20. Primjer rada funkcije onMouseOver()*



Slika 21. Prikaz gotove stranice

#### ZAKLJUČAK

Najveće poteškoje izašle su iz manjka primjera programskog koda koji su rađeni na biblioteci D3.js verziji sedam. Dosta pronađenih primjera su rađeni u verziji četiri i tri koje koriste zastarjele funkcije. Teško je pronaći kako da se neke zastarjele stvari koriste u verziji sedam. To se najviše osjetilo u početnom kutu kružnog grafa, varijabla koju smo u prijašnjim verzijama mogli automatski dohvatiti na lagan način, no u ovoj verziji nisam uspješno našla lagan način pa sam morala improvizirati sa zadavanje za koliko se rotira umjesto računanjem.

Vizualizacija ovih podataka je bila zanimljiva za izraditi, iako ponekad sam morala odustat od nekih ideja ili zbog premalo vremena ili zbog ne pronalaska funkcija u novoj verziji biblioteke D3.js. Sve u svemu, D3 je jako moćan alat zbog kojeg je uopće bilo moguće napraviti ovakv projekt u kratkom vremenu bez previše programiranja.

Stranica je izrađena putem biblioteke *bootstrap* kako bi omogućili responzivnost dizajnu.