



Univerzitet u Banjoj Luci
Prirodno-matematički fakultet

SEMINARSKI RAD

Predmet: Funkcionalno programiranje

Tema: Prikaz karata za igru

Mentor:

doc. dr Milana Grbić

Student:

Anel Popaja

Sadržaj

1	Uvod	1
2	Tehničke karakteristike	2
2.1	Pokretanje aplikacije	2
3	PostgreSQL baza	4
4	Izgled i korištenje aplikacije	5
5	Zaključak	7
6	Literatura	8

1 Uvod

U ovom seminarskom radu predstaviti ćemo način programiranja, metode izrade i finalni izgled web-aplikacije implementirane pomoću programskog jezika Haskell. Prvobitni cilj ovog rada jeste upoznavanje sa programiranjem i dizajnom web-aplikacije u potpuno funkcionalnom okruženju. Zbog određenih ograničenja jezika Haskell, te nedostatka materijala i pomoćnih biblioteka, odlučili smo da značajan dio ove aplikacije bude implementiran kroz programski jezik Elm, koji je također baziran na funkcionalnoj paradigmi.

Kao rezultat dizajna i programiranja želimo dobiti aplikaciju koja se može koristiti kao informacioni resurs vezan za prikaz karata iz video-igre “Gwent: The Witcher Card Game”.

Aplikacija i ostali resursi vezani za aplikaciju se mogu pronaći u git repozitoriju: <https://github.com/anelpopaja/HaskellWebApp>.

2 Tehničke karakteristike

Aplikacija se primarno sastoji iz dva dijela, klijentskog i serverskog, a komunikacija između ova dva dijela se realizuje putem http zahtjeva i odgovora u JSON formatu.

Klijentska strana je realizovana u funkcionalnom jeziku Elm, dok je server pisan u programskom jeziku Haskell. Dodatno, za bazu smo odlučili da koristimo PostgreSQL sistem.

2.1 Pokretanje aplikacije

Kako bismo pokrenuli aplikaciju na sistemu prvo je potrebno instalirati nekoliko komponenti. Ovo uputstvo za pokretanje je namijenjeno za Windows 10/11 operativni sistem.

Na računar je potrebno instalirati paket resursa za rad sa haskell-om i paket za elm, čija se instalacijska uputstva mogu pronaći na sljedećim linkovima:

<https://www.haskell.org/downloads/>,

<https://guide.elm-lang.org/install/elm.html>.

Sljedeći korak je instalacija PostgreSQL DBMS-a. Ovaj alat će nam omogućiti da integrišemo bazu podataka u našu aplikaciju. Link za preuzimanje navednog paketa je: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>.

Zbog činjenice da se jezik Elm kompajlira u JavaScript, potrebno je instalirati Node.js sa sljedećeg linka: <https://nodejs.org/en/download/>.

Nakon što smo instalirati potrebno komponente, pokrećemo Command Prompt sa administratorskim privilegijama u folderu „server“, te pozivamo sljedeće komande:

stack buid,

stack exec migrate-db,

stack exec run.

Nakon uspješnog poziva komandi, terminal će da prikaže poruku „Runing Basic Server“, te tako znamo da je server pokrenut.

Idući korak je pokretanje klijentske strane aplikacije. Potrebno je da pokrenemo novi terminal, kao i u prošlom slučaju, ali ovog puta u folderu „klijent“, te da pozovemo sljedeću komandu:

npx elm-live src/Main.elm.

Pri uspješnom izvršenju komande, terminal će da prikaže poruku „Server has been started!“, te tada možemo poduzeti posljednji korak ka pokretanju aplikacije.

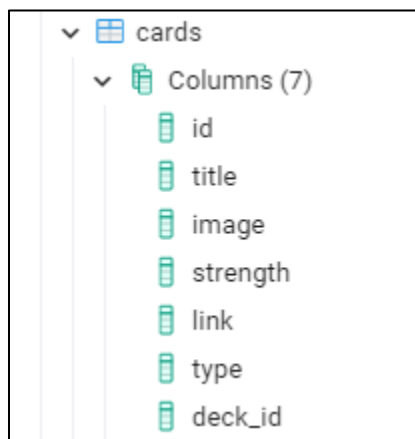
Preostalo je samo da otvorimo web-stranicu na kojoj se aplikacija nalazi, što je u našem slučaju: <http://localhost:8000>. Ukoliko web-pretraživač izbací „CORS“ grešku, to možemo riješiti tako što u „Target“ polje prečice za Google Chrome pretraživač dodamo:

--disable-web-security --user-data dir="C:/ChromeDevSession".

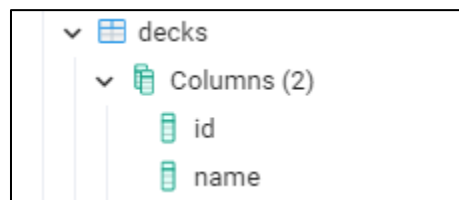
Nakon ovih koraka i otvaranja navedene web-stranice u pretraživaču će se prikazati ciljanja aplikacija.

3 PostgreSQL baza

Prilikom procesa kreiranja servera u poglavlju 2.1, pozvali smo naredbu *stack exec migrate-db*. Tom naredbom se kreiraju tabele unutar „gwentDB“ baze, koje su potrebne za rad aplikacije. Tačnije, kreirali smo tabelu „cards“ i tabelu „decks“. Izgled ovih tabela sa njihovim kolonama je prikazan na Slika 1. i Slika 2.



Slika 1. Prikaz tabele "cards"

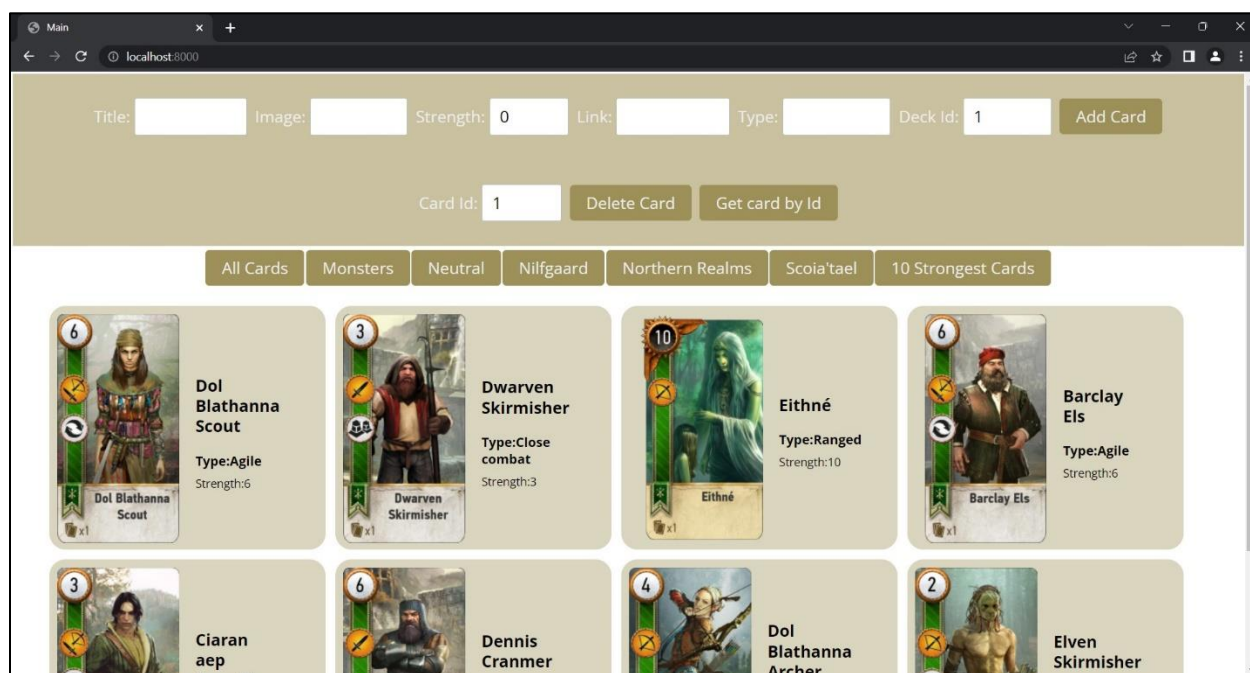


Slika 2. Prikaz tabele "decks"

Bazu popunjavamo sa relevantnim podacima tako što ih kroz pgAdmin 4 aplikaciju importujemo iz .csv file-ova koji se nalaze u git repozitoriju.

4 Izgled i korištenje aplikacije

Otvaranjem web-stranice <http://localhost:8000> prezentuje nam se web-aplikacija za prikaz karata za igru. Izgled aplikacije je prikazan na Slika 3.



Slika 3. Izgled web-aplikacije

Aplikacija se sastoji iz dva dijela, od kojih je prvi dio zaglavlje. U ovom dijelu se nalaze polja za unos podataka nove karte. Nakon što korisnik unese sve relevantne informacije, klikom na dugme „Add Card“, karta se dodaje u bazu.

Svaka karta, koja se dodaje u bazu, mora da pripada nekom od pet mogućih špilova. Svaki od tih špilova ima svoj unikatan id koji je naveden u nastavku:

1. Monsters,
2. Neutral,
3. Nilfgaard,

4. Northern Realms,
5. Scoia'tael.

Ispod polja za dodavanje nove karte se nalazi i polje pomoću kojeg se, na osnovu id-a karte, iz baze prikazuje tražena karta sa mogućnosti da se ista i obriše, zavisno na koje dugme se klikne.

Ispod zaglavlja aplikacije se nalazi prostor za prikaz karti koje se nalaze u bazi. Korisniku je ponuđena mogućnost prikaza svih karti ili pak prikaza samo onih karti koje pripadaju određenom špilu. Također, osim izbora špila, korisniku se nudi i opcija da prikaže deset karti koje imaju najveću snagu (strength vrijednost).

Pojedinačna karta je prikazana u svom zasebnom bloku. Blok se sastoji iz lijevog i desnog dijela, gdje se u lijevom dijelu prikazuje izgled te karte, dok se u desnom dijelu prikazuju osnovne informacije vezane za tu kartu. Klikom bilo gdje unutar bloka određene karte, u novom prozoru pretraživača, se otvara web-stranica sa detaljnijim informacijama o toj karti.

5 Zaključak

U ovom radu smo prezentovali funkcionalnu web-aplikaciju implementiranu pomoću programskih jezika Haskell i Elm. Cilj nam je bio da se поближе upoznemo sa funkcionalnom paradigmom i programiranjem u navedenim jezicima, te da napravimo koristan web aplikaciju.

Nakon ostvarenja prvobitnog cilja, te analize rezultata i procesa kroz koje smo prošli do njegovog ostvarenja, možemo zaključiti da programiranje web-aplikacija u funkcionalnim jezicima, iako vrlo moguće, nije nimalo jednostavno i pruža vrlo ograničene rezultate.

6 Literatura

Babić, M. (25. April 2022). *FUNKCIONALNO-PROJEKAT*. Preuzeto od GitHub:

<https://github.com/suncanjenamesecini/FUNKCIONALNO-PROJEKAT>

Elm - The Complete Guide (a web development video tutorial). (29. September

2022). Preuzeto od Udemy: [https://www.udemy.com/course/elm-the-](https://www.udemy.com/course/elm-the-complete-guide/?referralCode=C7120849E19E4866E9C3)

[complete-guide/?referralCode=C7120849E19E4866E9C3](https://www.udemy.com/course/elm-the-complete-guide/?referralCode=C7120849E19E4866E9C3)

freeCodeCamp.org. (3. July 2018). *Elm crash course - Building unbreakable*

webapps fast. Preuzeto od YouTube:

<https://www.youtube.com/watch?v=kEitFAY7Gc8>

Persistent :: Yesod Web Framework Book- Version 1.6. (29. September 2022).

Preuzeto od Yesod Web: <https://www.yesodweb.com/book/persistent>

Real World Haskell. (29. September 2022). Preuzeto od GitHub:

<https://github.com/MondayMorningHaskell/RealWorldHaskell>

Yank, K. (25. May 2017). *Developer Happiness on the Front End with Elm*.

Preuzeto od YouTube: <https://www.youtube.com/watch?v=kuOCx0QeQ5c>