|  |  |
| --- | --- |
|  | |
| НАСОКА: СОФТВЕРСКО ИНЖИНЕРСТВО И  ИНФОРМАЦИСКИ СИСТЕМИ | |
| Проектна задача  СЕМИНАРСКА РАБОТА | |
| Предмет: | |
| Неструктурирани бази на податоци | |
|  | |
| ТЕМА: | |
| Складирање на податоци во граф базирани бази на податоци и споредба | |
|  | |
|  | Тим:  Мартин Трајковски 195063 |
|  | Вангел Трајковски 181261 |
|  |  |
| Јуни 2023 | |

[Апстракт 3](#_Toc138583167)

[1. Вовед 5](#_Toc138583168)

[2. Методологија 5](#_Toc138583169)

[2.1. Податоци 5](#_Toc138583170)

[2.2. Агрегација 6](#_Toc138583171)

[2.3. Прашалници 7](#_Toc138583173)

[2.4. 7](#_Toc138583174)

[3. Изработка на апликацијата 7](#_Toc138583175)

[4. Користена литература 8](#_Toc138583176)

Соджина

￼

# Апстракт

.

Слики

[Слика 1 ТypeDef за Hero објект 5](#_Toc64231117)

[Слика 2 Повика на барање за податоци од сервер 6](#_Toc64231118)

[Слика 3 Резолвер од Hero овјект 6](#_Toc64231119)

[Слика 4 Tailwind компонента 6](#_Toc64231120)

[Слика 5 ApolloQuery Компонента 6](#_Toc64231121)

[Слика 6 Структура на Hero шемата 7](#_Toc64231122)

[Слика 7 GraphQL корисничко квери 7](#_Toc64231123)

[Слика 8 Router компонента 8](#_Toc64231124)

[Слика 9 пример за TypeDef 9](#_Toc64231125)

[Слика 10 Имплементирано квери во резолвер 10](#_Toc64231126)

[Слика 11 Vue темплејт слилизиран со tailwind 10](#_Toc64231127)

[Слика 12 Приказ на корискот од темплејтот на слика 11 10](#_Toc64231128)

[Слика 13 Форма за регистрација 12](#_Toc64231129)

[Слика 14 Форма за најава 12](#_Toc64231130)

[Слика 15 Изглед на играта 13](#_Toc64231131)

[Слика 19 Изглед на artefacts компонентата 13](#_Toc64231132)

[Слика 18 Изглед на Companions компонентата 13](#_Toc64231133)

[Слика 17 Изглед на Hero компонентата 13](#_Toc64231134)

[Слика 16 Изглед на Gear компонентата 13](#_Toc64231135)

# Вовед

При изработка на апликации често се случува да ...

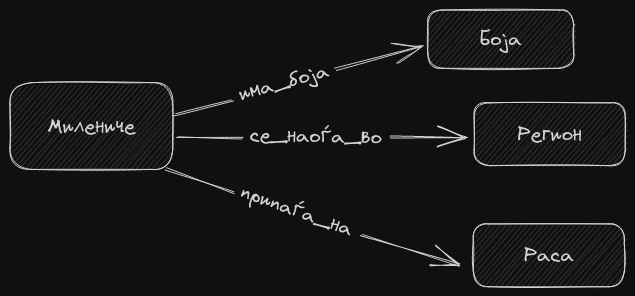
# Методологија

## Податоци

Податоците кои користени од наша страна се множество од миленичиња[[1](#Bookmark1)], ова мноќество е во формат „CSV“, и беше потребна преработа и читање од страна на користените системи за чување на податоци, во прилог е представена слика, која ги отсликува податоците.

Како што можиме да видиме од следната слика, множеството од податоци се состои од еден главен тип на ставка која содржи податоци за миленичињата, и дополнително имаме уште 3 споредни типови кои го опишуваат миленичето, како што се бојата, која според податоците може да се содржи 3 пати, потоа ја имаме расата на миленичето, за која секое милениче имам можност да поседува 2 рефернци доколку е мешано од повеќе раси, во расата се состои и типот на милениче односно, дали миленичето е куче, маче итн, последниот податок е за каде е лоцирано, односно во кој регион од соодветната држава се наоѓа миленичето.

По претставување на овие податоци следува и нивно претставување во вид на граф, како што можи да се види од следната слика, каде се отсликани визуелно форматот на податоците при вметнување во базата на податоци, базата се состои од 3 релации, и во сите случаеви релациите се движат од ентитетот „милениче“ до останатите 3 ентитети.



## Агрегација

Во рамки на граф базите на податоци, податоците се агрегираат така што се почнува од една точка во графот и за неа се пребарува преку врските кои ги има, доколку базата се состои од елементи кои се меѓусебно поврзани со ребра, податоците се агрегираат во една комплетна низа од истите и се испраќаат до корисниците на системот.

Податоците се приспособени за да се вклопат во рамки на системот, при при што беа искористени 2 системи, има 2 различни начини на приспособовање:

**Neo4j:**

За да се приспособат податоците и да се вметнат во овој систем, искористени се прашалници преку кои се читаат податоците од CSV формат, кои подоцна се креираат точките и ребрата на графот.

**OrientDB:**

Во вториот систем кој се одлучивме да го користиме, нема подршка преку прашалници да се прочита директно од CSV датотека, но постои можност за вметнување на податоци со една алатка така наречена ETL (Extractor, Transformer, Loader), оваа алатка работи со конфигурација во JSON формат, преку која е потребно да се кажи на системот како да ги извади податоците (Extract), на кој начин да ги промени и адаптира на системот (Transformer), и како да ги впиши.

Агрегацијата е извршена така што од податоците дадени во повеќе табели се претставени во грат, тоа можи да се види на предходната слика, во која податоците немаат веќе колони, туку имаат атрибути и ребра кои ги поврзуваат ентитетите, тоа можи да се види на следната слика.

## 

## Прашалници

## 

# Изработка на апликацијата

# Користена литература

1. <https://www.kaggle.com/c/petfinder-adoption-prediction>

2. https://tailwindcss.com/

3. https://vuejs.org/