Dokumentacja Projektu - Sieć Społecznościowa z Neo4J

Informacje podstawowe

Autor: Artur Saganowski

Technologia: Angular 18 (SPA) + Node.js/Express + Neo4J AuraDB

Link do aplikacji live:

`https://social-network-neo4.netlify.app/network`

1 Opis projektu

Aplikacja typu "proof of concept" przedstawiajaca wykorzystanie grafowej bazy danych Neo4J do implementacji sieci społecznościowej. System umożliwia zarzadzanie użytkownikami, tworzenie relacji znajomości oraz wykorzystuje algorytmy grafowe do zaawansowanych funkcji takich jak rekomendacje znajomych czy znajdowanie najkrótszej ścieżki miedzy użytkownikami.

1.1 Cel projektu

Demonstracja możliwości grafowej bazy danych Neo4J w kontekście sieci społecznościowych, ze szczególnym naciskiem na:

- Modelowanie relacji miedzy użytkownikami jako grafu
- Wykorzystanie algorytmów grafowych (shortest path, friends of friends)

2 Szczegóły technologiczne

Frontend

- **Framework:** Angular 18
- **Architektura:** Single Page Application (SPA) z standalone components

Backend

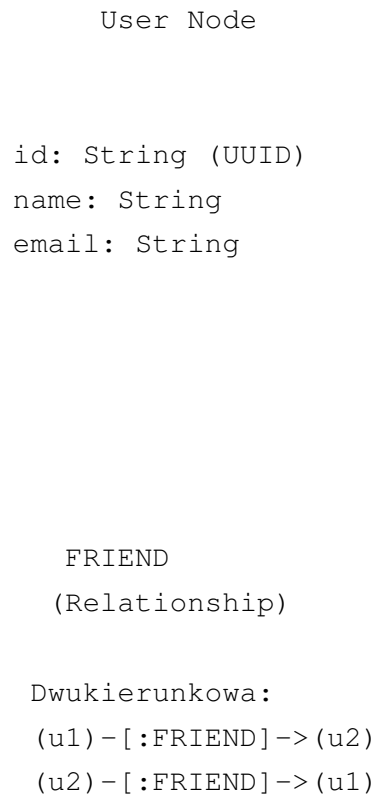
- **Runtime:** Node.js
- **Framework:** Express

Baza danych

- **Typ:** Grafowa baza danych
- **Hosting:** Neo4J AuraDB (DBaaS)

3 Model danych (Graf Neo4J)

3.1 Diagram modelu grafowego



3.2 Przykładowe zapytania Cypher

Tworzenie użytkownika:

```
CREATE (u:User {
  id: $id,
```

```
  name: $name,
  email: $email
})
RETURN u
```

Tworzenie relacji znajomości (dwukierunkowa):

```
MATCH (u1:User {id: $userId1})
MATCH (u2:User {id: $userId2})
CREATE (u1)-[:FRIEND {}]->(u2)
CREATE (u2)-[:FRIEND {}]->(u1)
```

Znajdowanie rekomendacji (Friends of Friends):

```
MATCH (u:User {id: $id})-[:FRIEND]->(friend)-[:FRIEND]->(recommendation
   )
WHERE NOT (u)-[:FRIEND]->(recommendation) AND u <> recommendation
RETURN DISTINCT recommendation, count(*) as mutualFriends
ORDER BY mutualFriends DESC
LIMIT 10
```

Najkrótsza ścieżka miedzy użytkownikami:

```
MATCH (u1:User {id: $from})
MATCH (u2:User {id: $to})
MATCH path = shortestPath((u1)-[:FRIEND*]-(u2))
RETURN nodes(path) as path, length(path) as distance
```

4 Diagram przypadków użycia

Zarzadzanie użytkownikami:

- Dodaj nowego użytkownika
- Zobacz liste użytkowników
- Usuń użytkownika

Zarzadzanie relacjami:

- Utwórz relacje znajomości
- Usuń relacje znajomości

- Zobacz znajomych użytkownika

Analiza sieci:

- Zobacz rekomendacje znajomych (FoF)
- Znajdź najkrótsza ścieżke miedzy osobami
- Zobacz statystyki sieci

5 Wdrożenie (Deployment)

Platforma: Neo4J AuraDB (Cloud DBaaS)
Plan: AuraDB Free

Proces wdrożenia:

1. Utworzono konto na `https://neo4j.com/cloud/aura/`
2. Utworzono instancje bazy danych
3. Skonfigurowano connection string w pliku .env backendu
4. Połaczono repozytorium Git z Netlify (frontend) oraz Render (backend)

6 Instalacja i uruchomienie lokalne

6.1 Backend

```
cd backend
npm install
npm start
```

6.2 Frontend

```
cd frontend
npm install
npm start
```

7 Ciekawe rozwiazania technologiczne

1. SPA z Angular 18 - nowoczesny framework JS
2. Neo4J AuraDB
3. Algorytmy grafowe - shortest path, friends-of-friends
4. REST API
5. Deployment w chmurze - Netlify dla frontend i Render dla backend