

# Sudoku se vším všudy

Ročníkový projekt

Vladimír Vávra 4.E

Gymnázium, Praha 6, Arabská 14

7.března 2022

# O projektu

- Cíl práce: Vytvořit škálovatelný základ pro stejnojmennou maturitní práci
- Aktuální funkce projektu:
  - Vytvoření uživatelského rozhraní pro hraní klasického sudoku a různých variant
  - Registrační systém uživatelů s možností více druhů přihlášení
  - Schopnost programu řešit sudoku včetně přidružených variant
  - Generátor sudoku různé typy sudoku různých velikostí dle obtížnosti

# Použité technologie

## ■ MERN Stack

- React, Material-UI, SASS – frontend
- Express + NodeJS – backend server
- MongoDB – databáze



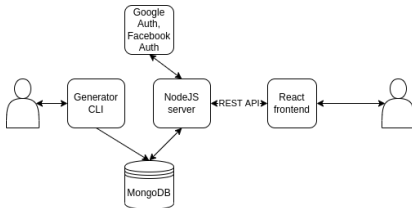
## ■ Vývoj:

- VS Code
- Git + dropbox – zálohování kódu



# Architektura

- Uživatel interaguje s frontendem
- Frontend se na sudoku dotazuje NodeJS serveru přes REST API (Client side rendering)
- NodeJS server vybírá zadání (s řešením) z MongoDB databáze + provede variační práce
- Pro vygenerování sudoku – interakce s CLI na serveru (pro admina)



Architektura aplikace

# Architektura

## ■ Frontend

- Atomic design (atoms, molecules, organisms, templates, pages)
- Redux state (návrhový vzor kontext)
- Další návrhové vzory: při vývoji proxy na server, observer (event listenery)

## ■ Backend

- Architektura Boba Martina
- Využití návrhových vzorů: Dependency injection (IoC), továrních metod, adaptérů, fasád
- Test-driven development – test před kódem – 90% kódu má testy – unit, integrační

# Autentifikace

- Typy:
  - Local (heslem)
  - Google Auth
  - Facebook Auth
- Při stejném emailu možnost přihlášení více providery
- Endpoint pro změnu hesla

# Ukázka autentifikace





# Ukázka her

# Algoritmus řešení sudoku

- Backtrackingový algoritmus – prohledávání do hloubky
- Při pokládání čísel kontrola, zda mohu položit – rekurze / jiné číslo / backtrack
- $O(M^N)$ , kde  $M$  je počet možných dosazovaných čísel a  $N$  počet políček, za které dosazujeme.

# Algoritmus generování sudoku

- 1 Vytvoř validní mřížku pro daný typ sudoku
  - Upravený algoritmus řešení
  - Při přidání políčka je číslo vyškrtáno z míst, kde nemůže být
  - Další číslo vybíráno pouze z čísel, které mohou na políčku být
- 2 Uber z mřížky N políček (přímo úměrné obtížnosti)
- 3 Vyřeš sudoku a zkontroluj, zda má 1 řešení. Má? Ulož ho do databáze. Nemá? Vrať odstraněná políčka a jdi na krok 2.

## Algoritmus na výrobu variant

- Možnost vytvoření více zadání z jedné vygenerované mřížky s kvadratickou časovou složitostí (v závislosti na velikosti mřížky).
- Uživatel nepozná rozdíl, bude-li mít rozestup mezi variantami téhož sudoku
- Operace, které můžeme provést jsou:
  - Permutace čísel – celkem  $9!$  možností
  - Rotace matice – 4 možnosti
  - Transpozice matice – dle hlavní diagonály, vedlejší diagonály, osy  $x$ , osy  $y$  – 4 možnosti
- $N! * 4 * 4$
- pro  $9 \times 9$  zadání – 5 806 080 variant.

# Generátor

- CLI rozhraní pro administrátora
- Velké mřížky těžké obtížnosti trvají dlouho vygenerovat → nutnost manuální kontroly
- `npm run generate`
- Nutné zadat:
  - Typ sudoku
  - Velikost mřížky
  - Obtížnost
  - Počet zadání tohoto typu

# Ukázka generátoru

# Závěr

- Podařilo se splnit většinu zadání a vytvořit škálovatelný základ pro maturitní práci
- Do maturitní práce:
  - Dokončení rozpracované interní ekonomiky (využití coinů)
  - Implementace dalších typů her
  - Vylepšení UI pro podporu dalších funkcí (vlastní zadání a jeho vyřešení)
  - Bonus: vyřešení z fotografie

# Zdroje

- vscode icon:  
`https://commons.wikimedia.org/wiki/File:Visual\_Studio\_Code\_1.35\_icon.svg`
- dropbox icon:  
`https://danielgamrot.cz/skvely-tip-pro-dropbox/`
- git icon: `https://commons.wikimedia.org/wiki/File:Git\_icon.svg`