

Generator Sudoku

Aplikacja przeglądarkowa napisana przez Emila Kobyłeckiego i Agnieszkę Kowalczyk w ramach zaliczenia przedmiotu Inżynieria Internetu.

1. Ogólny opis aplikacji

Aplikacja służy do generowania sudoku i obsługi powiązanych zapytań. Na odpowiednie żądanie użytkownika, wyświetlana jest tablica sudoku z brakującymi polami do uzupełnienia, przedstawiona w czytelny sposób w przeglądarce:

Sudoku to solve

	_	1	5		_	3	2		7	_	_	
	6	_	3		7	9	_		_	5	2	
	9	_	2		_	_	_		_	6	_	

	_	8	9		_	7	5		_	1	6	
	_	2	1		_	6	9		_	_	_	
	3	_	_		1	_	4		9	7	8	

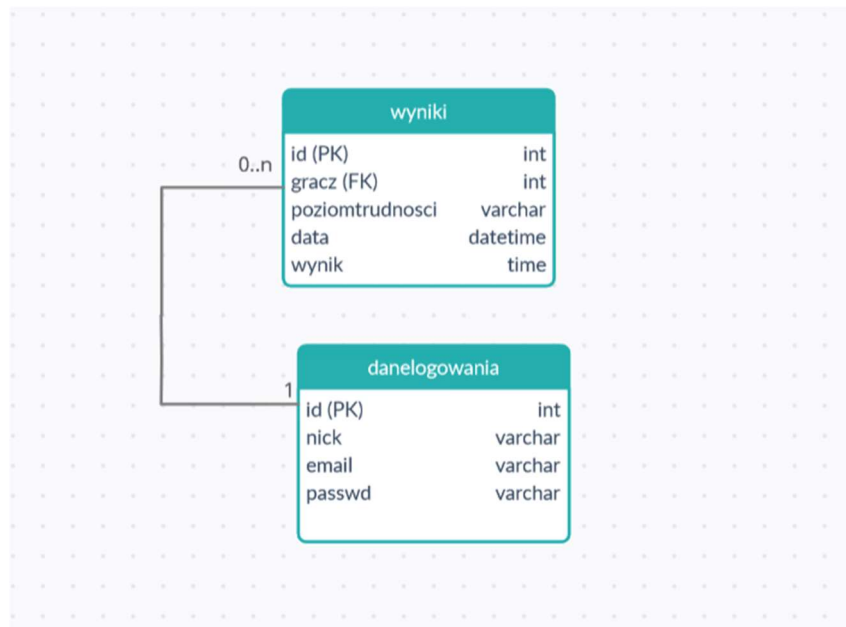
	2	3	_		_	1	_		_	8	_	
	_	_	_		_	_	7		_	_	9	
	_	_	7		2	8	_		6	4	1	

Jest możliwość wyboru jednego spośród trzech poziomów trudności łamigłówki - łatwego, średniego lub trudnego. Ponadto aplikacja oferuje możliwość założenia konta użytkownika, który po rejestracji uzyskuje możliwość zapisywania wyników czy przeglądania archiwalnych. Konto użytkownika jest w pełni obsługiwane- można się na nie powtórnie zalogować, wyświetlać oraz edytować wprowadzone dane czy trwale usunąć założone konto.

2. Wykorzystane technologie i narzędzia

Aplikacja została napisana zgodnie z założeniami API Rest oraz zaimplementowana w języku JavaScript. Zarządzaniem wszelkimi zależnościami w projekcie zajął się menadżer pakietów- Yarn, natomiast narzędziem zapewniającym przenośność kodu pomiędzy różnymi platformami i

odpowiednio przetwarzającym zmienne środowiskowe był cross-env. Wykorzystano bazę danych MySQL umieszczoną na lokalnym serwerze uruchamianą przy pomocy pakietu XAMPP. Baza składa się z dwóch prostych tabel- jednej przechowującej dane dotyczące użytkowników oraz drugiej przechowującej wyniki. Tabele połączone są relacją jeden do wielu (jeden użytkownik może mieć zapisane wiele wyników).



Dla łatwiejszej obsługi bazy danych w kodzie, wykorzystano narzędzie wspierające budowanie zapytań oraz migracji- Knex. Wsparciem przy rzutowaniu obiektów obsługiwanych przez program na relacyjną bazę danych był Objectection.js- odpowiadał za mapowanie obiektowo - relacyjne. Ważnym elementem umożliwiającym logowanie użytkownika oraz przetwarzanie różnych żądań pochodzących od użytkownika był JWT - Json Web Token. Bezpieczne haszowanie haseł wprowadzanych do bazy danych zrealizowano dzięki algorytmowi Argon2.

3. Funkcjonalności

Strona powitalna nawigująca po reszcie funkcjonalności dostępna jest pod adresem:
<http://localhost:3198/>.

a. Sudoku

Do wygenerowania sudoku służy żądanie GET pod adresem: `/api/sudoku/'nr'`, gdzie pod 'nr' należy podstawić jedną z liczb w zależności od wybranego poziomu trudności (0- łatwy, 1- średni, 2- trudny).

Aby wystać rozwiązaną łamigłówkę do sprawdzenia, należy jako zalogowany użytkownik wystać żądanie POST pod adres: `/api/sudoku`. Liczby należy wpisywać kolejnymi wierszami w tablicach w formacie JSON:

```
{  
  "row1": [5, 3, 4, 6, 7, 8, 9, 1, 2],  
  "row2": [6, 7, 2, 1, 9, 5, 3, 4, 8],  
  "row3": [1, 9, 8, 3, 4, 2, 5, 6, 7],  
  "row4": [8, 5, 9, 7, 6, 1, 4, 2, 3],  
  "row5": [4, 2, 6, 8, 5, 3, 7, 9, 1],  
  "row6": [7, 1, 3, 9, 2, 4, 8, 5, 6],  
  "row7": [9, 6, 1, 5, 3, 7, 2, 8, 4],  
  "row8": [2, 8, 7, 4, 1, 9, 6, 3, 5],  
  "row9": [3, 4, 5, 2, 8, 6, 1, 7, 9]  
}
```

Jeżeli odpowiedź została przesłana w prawidłowy sposób, użytkownik otrzyma stosowny komunikat "Prawidłowo rozwiązałeś sudoku!" lub "Nieprawidłowo rozwiązane sudoku :(" wraz z statusem 200 oznaczający że żądanie zostało przetworzone bez zakłóceń.

b. Konto użytkownika

W celu rejestracji konta użytkownika należy wysłać żądanie POST pod adres */api/danelogowania*, wysyłając dane użytkownika w formacie JSON:

```
{  
  "nick": "my_nick",  
  "passwd": "my_passwd",  
  "email": "my@email.com"  
}
```

Jeżeli podane dane są prawidłowe, i konto zostanie założone, użytkownikowi zostanie zwrócony komunikat zawierający zapisane dane oraz status 201 oznaczający utworzenie nowego zasobu.

Aby zalogować się na istniejące konto, należy pod adres */api/danelogowania/login* wysłać żądanie GET z danymi do zalogowania w postaci JSON-a:

```
{  
  "nick": "my_nick",  
  "passwd": "my_passwd"  
}
```

Po poprawnym zalogowaniu, zostanie zwrócony komunikat o powodzeniu operacji wraz ze statusem 200, natomiast w sytuacji podania nieprawidłowych danych (nicku bądź hasła) również zostanie wyświetlona odpowiednia informacja ze statusem 400.

Aby uzyskać informacje na temat zapisanych danych użytkownika (nick, email), należy po zalogowaniu się na swoje konto wysłać żądanie GET pod adres */api/danelogowania/mydata*.

Istnieje możliwość edycji danych zalogowanego użytkownika. Aby skorzystać z tej funkcjonalności należy wysłać żądanie PUT pod adres */api/danelogowania* wraz z poprawionymi danymi w formacie JSON:

```
{  
  "nick": "corrected_nick",  
  "passwd": "corrected_passwd",  
  "email": "corrected@emial.com"  
}
```

Jeżeli operacja zakończy się sukcesem, użytkownik otrzyma stosowny komunikat wraz ze statusem 201.

Użytkownik może wylogować się z konta za pomocą żądania GET pod adresem */api/danelogowania/logout*.

W przypadku, gdy użytkownik chce usunąć swoje konto, może to wykonać poprzez wysłanie żądania DELETE pod adres */api/danelogowania*. Operację należy potwierdzić hasłem. W przypadku pomyślnego zakończenia akcji, użytkownik otrzyma komunikat "Usunięto użytkownika" wraz ze statusem 200.

c. Wyniki

Wyniki są automatycznie dodawane do konta użytkownika, w momencie przesyłania przez zalogowanego użytkownika rozwiązania łamigłówki. Aby wyświetlić archiwalne wyniki, zalogowany gracz może wysłać żądanie GET pod adresem */api/wyniki*. Przechowywanych jest 10 najlepszych rezultatów.