

Workouter

Jan Baran

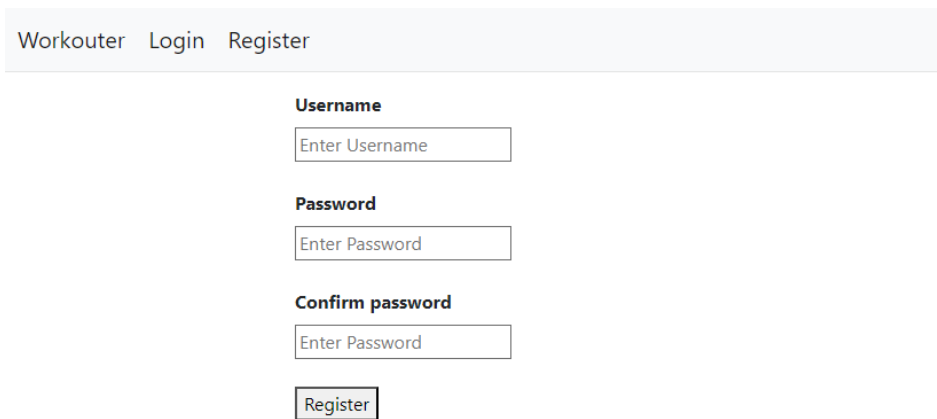
Link: <https://workouter.azurewebsites.net>

1. Opis projektu

Workouter to aplikacja webowa służąca do układania planów treningowych i zapisywania swoich wyników po wizycie na siłowni. Wykorzystuje ona bazę danych Neo4j dostępną w serwisie Neo4j AuraDB. Do jej stworzenia został użyty pythonowy framework Flask, który odpowiada zarówno za backend (zapytania do bazy danych), jak i frontend (renderowanie strony z uprzednio przygotowanych szablonów – plików .html).

2. Instrukcja obsługi

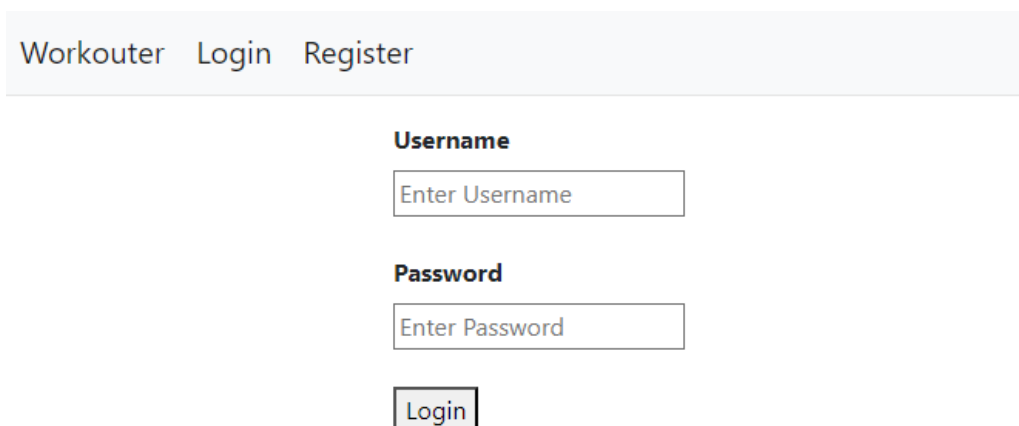
Przy pierwszym wejściu na stronę należy utworzyć konto w aplikacji:



The screenshot shows the registration form of the Workouter application. At the top, there is a navigation bar with links: 'Workouter', 'Login', and 'Register'. Below the navigation bar, the form consists of four input fields: 'Username' with placeholder text 'Enter Username', 'Password' with placeholder text 'Enter Password', and 'Confirm password' with placeholder text 'Enter Password'. Below these fields is a 'Register' button.

Widok 1. Rejestracja

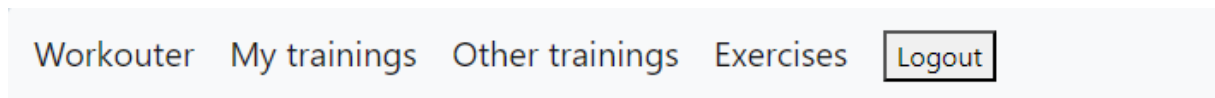
a następnie się na nie zalogować:



The screenshot shows the login form of the Workouter application. At the top, there is a navigation bar with links: 'Workouter', 'Login', and 'Register'. Below the navigation bar, the form consists of two input fields: 'Username' with placeholder text 'Enter Username' and 'Password' with placeholder text 'Enter Password'. Below these fields is a 'Login' button.

Widok 2. Login

Po wykonaniu powyższych kroków wyświetli się następująca strona:



Save your trainings with Workouter!

Widok 3. Strona główna

Zakładka „Exercises” pozwala podejrzeć ćwiczenia, które są dodane na stałe do bazy danych oraz te, które zostały przez użytkownika utworzone. Ćwiczenie dodaje się za pomocą przycisku „Add Exercise”. Użytkownik może usunąć tylko stworzone przez siebie ćwiczenia.

Workouter	My trainings	Other trainings	Exercises	Logout
Add Exercise				
Exercise Name				Action
Incline bench press				Remove
Bench press				
Biceps curl				
Cable flyes				
Calf raises				
Deadlift				
Face pulls				
Hammer curls				
Hip thrust				
Lateral raise				
Pull-up				
Seated rows				
Squat				
Triceps pushdowns				

Widok 4. Ćwiczenia

Po wybraniu ćwiczenia użytkownik zostaje przeniesiony na jego stronę i tam wprowadza dane. Jeśli przykładowo zrobił on 3 serie po 10 powtórzeń ciężarem o wadze 20 kg, to musi 3 razy wcisnąć przycisk „Add set”, wpisując w „Reps” 10, a w „Weight (kg)” 20.

Incline bench press

Reps

Weight (kg)

Add set

Reps	Weight (kg)
10	20
10	20
10	20

Save sets

History

Widok 5. Strona ćwiczenia - ilustracja przykładu

Gdy wprowadzone zostaną wszystkie dane, należy wcisnąć „Save sets” – sprawi to, że wynik od teraz będzie zapisany w bazie danych i wyświetlany na stronie w dziale „History”:

[Workouter](#) [My trainings](#) [Other trainings](#) [Exercises](#) [Logout](#)

Incline bench press

Reps

Weight (kg)

[Add set](#)

Reps	Weight (kg)
------	-------------

[Save sets](#)

History

11/09/2023

Reps	Weight (kg)
10	20
10	20
10	20

Widok 6. Strona ćwiczenia - zapisane serie z 9 listopada

Wracając do zakładki „My trainings” pozwala stworzyć użytkownikowy własny trening, czyli kontener, w którym może trzymać interesujące go ćwiczenia:

Workouter	My trainings	Other trainings	Exercises	Logout
Add training				
Training Name				Action
Arms				Remove
Legs				Remove
Lower body				Remove
Upper body				Remove

Widok 7. Moje treningi

Po wybraniu treningu należy dodać do niego ćwiczenia – robi się to poprzez wciśnięcie przycisku „Add Exercises”, wybraniu z listy interesujących użytkownika ćwiczeń (tych samych, które są w zakładce „Exercises”) i zatwierdzeniu operacji przyciskiem „Add”:

Workouter

My trainings

Other trainings

Exercises

Logout

Add Exercises

Exercise Name

Bench press

Cable flyes

Face pulls

Hip thrust

Exercise name

Exercise Name

☐ Incline bench press

☐ Biceps curl

☐ Calf raises

☐ Deadlift

☐ Hammer curls

☐ Lateral raise

☐ Pull-up

☐ Seated rows

☐ Squat

☐ Triceps pushdowns

Close

Add

Widok 8. Strona treningu - wybór ćwiczeń do dodania

Ostatnią zakładką jest „Other trainings” – to w niej można zobaczyć wszystkie treningi, które stworzyli inni użytkownicy. Jeśli użytkownikowi spodoba się trening, może wcisnąć widniejący przy nim przycisk „follow” – sprawi to, że trening przestanie się tu wyświetlać, a zacznie w zakładce „My trainings”. Tak pozyskanych treningów nie da się edytować, ale można zapisywać w nich swoje wyniki siłowe.

Workouter	My trainings	Other trainings	Exercises	Logout
-----------	--------------	-----------------	-----------	--------

Training Name	Action
Góra	Follow

3. Diagramy

W Workouterze zastosowano następujący model danych:

1. Węzły:

- User – reprezentujący użytkownika
- Exercise – reprezentujący ćwiczenie
- Training – reprezentujący trening
- Completed – reprezentujący zapisane serie wykonanego ćwiczenia

2. Relacje:

- ADDED_TRAINING – między użytkownikiem a treningiem (gdy użytkownik doda trening)
- FOLLOWS - między użytkownikiem a treningiem (gdy użytkownik „zafollowuje” trening innego użytkownika)
- ADDED_EXERCISE – między użytkownikiem a ćwiczeniem (gdy użytkownik doda ćwiczenie)
- INCLUDES – między treningiem a ćwiczeniem (gdy trening zawiera ćwiczenie)
- INCLUDES_SETS – między ćwiczeniem a zapisanymi seriami ćwiczenia

Diagram UML:

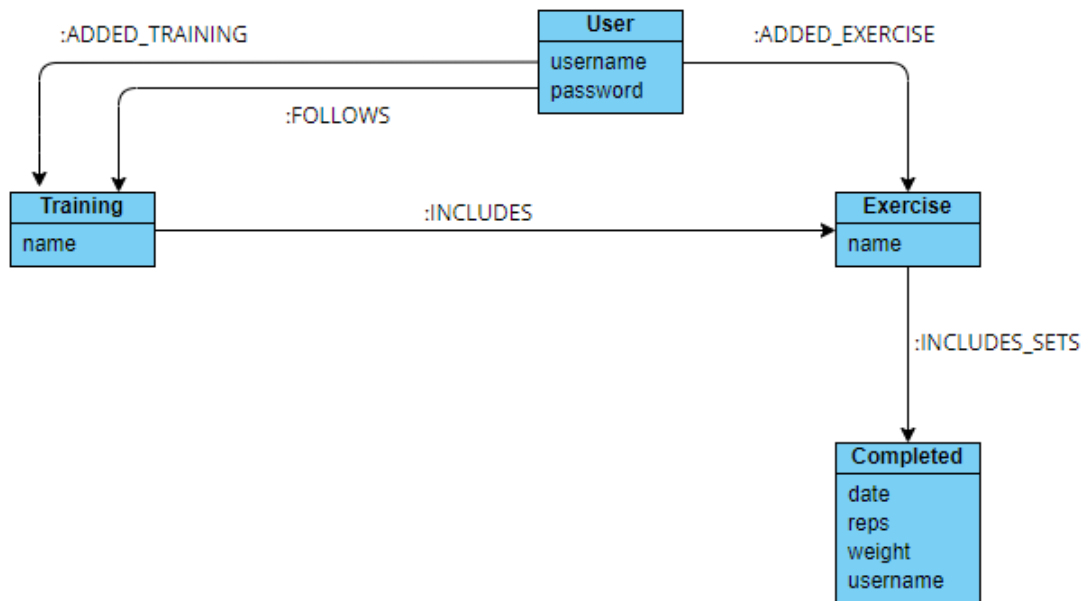


Diagram 1. Diagram UML

Przykładowy fragment grafu reprezentującego bazę danych:

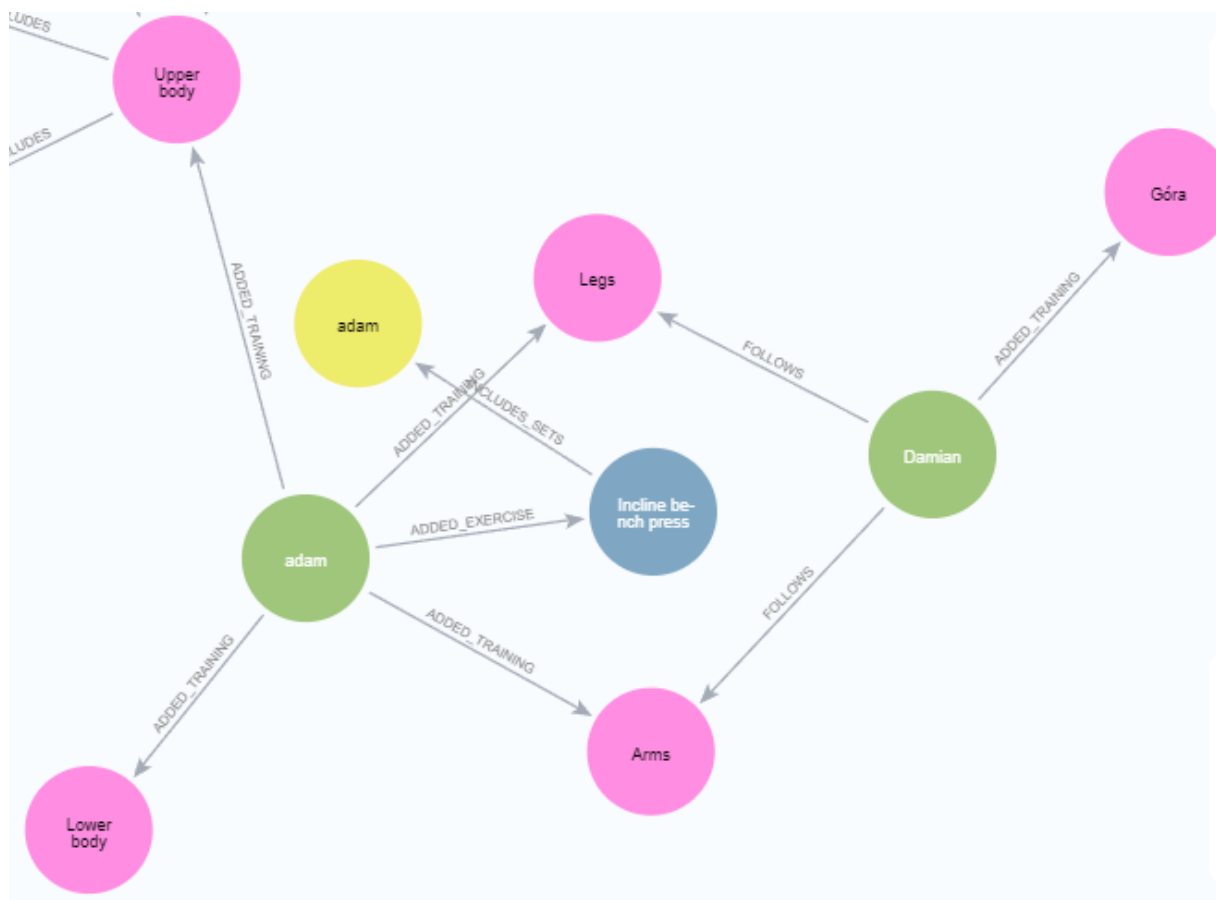


Diagram 2. Wizualizacja fragmentu bazy danych (Neo4j Workspace)

Kolory:

- Zielony – użytkownicy
- Różowy – treningi
- Niebieski – ćwiczenia
- Żółty – zapisane serie ćwiczeń

Zapisane serie ćwiczeń mają tę samą nazwę co użytkownicy, gdyż zapisują użytkownika, który wykonał ćwiczenie.

4. API

Endpoints (z pominięciem tych, których jedynym celem jest zwrócenie szablonu ze stroną):

Endpoint	Metoda HTTP	Opis
/login	POST	Uwierzytelnia użytkownika (zwracany JWT token)
/register	POST	Tworzy konto nowemu użytkownikowi
/mytrainings_protected	GET	Zwraca listę utworzonych i śledzonych treningów
/mytrainings	POST	Dodaje nowy trening przypisany do użytkownika
/mytrainings/delete/<int:id>	DELETE	Usuwa trening
/othertrainings_protected	GET	Zwraca listę treningów, które są stworzone przez innych użytkowników i nieśledzone
/exercises	POST	Dodaje nowe ćwiczenie przypisane do użytkownika
/exercises_protected	GET	Zwraca listę ćwiczeń, które są domyślnie w bazie lub użytkownik je dodał
/exercises/delete/<int:id>	DELETE	Usuwa ćwiczenie
/training/exercises/<int:id>	GET	Zwraca ćwiczenia zawierające się w treningu
/training/exercisesUnadded/<int:id>	GET	Zwraca ćwiczenia, które są dostępne w endpointzie „/exercises”, ale nie zawierają się w treningu
/training	POST	Dodaje ćwiczenie do treningu
/training/update	PUT	Usuwa ćwiczenie z treningu
/follow	POST	Dodaje trening jako śledzony dla konkretnego użytkownika
/unfollow	PUT	Usuwa trening ze śledzonych
/checkowner	POST	Zwraca informację czy użytkownik jest właścicielem treningu
/exercise/<int:id>	GET	Zwraca nazwę ćwiczenia, w którym można zapisać dane
/exercise	POST	Dodaje dane do ćwiczenia
/exercise/<int:id>/history	GET	Zwraca historię ćwiczenia

5. Wdrożenie

Aby uruchomić projekt należy posiadać zainstalowany język Python w wersji 3.10 (na inne wersje twórca nie daje gwarancji) oraz zainstalować do niego biblioteki komendą:

```
pip install -r requirements.txt
```

Serwer uruchamia komenda:

```
flask run
```

Niestety, tak uruchomiona aplikacja nie będzie działać poprawnie, gdyż dane potrzebne do uruchomienia serwera (uri i hasło do AuraDB) są ukryte w zmiennych środowiskowych. Działająca aplikacja jest dostępna pod linkiem podanym na początku dokumentacji. Na platformę Azure jest wdrażana automatycznie dzięki Github Actions. Po założeniu konta użytkownik sam musi stworzyć swoje treningi, jedyne do czego ma dostęp to kilka stworzonych uprzednio ćwiczeń i pozostałe po testach aplikacji treningi innych użytkowników.