

Architektura i technologie

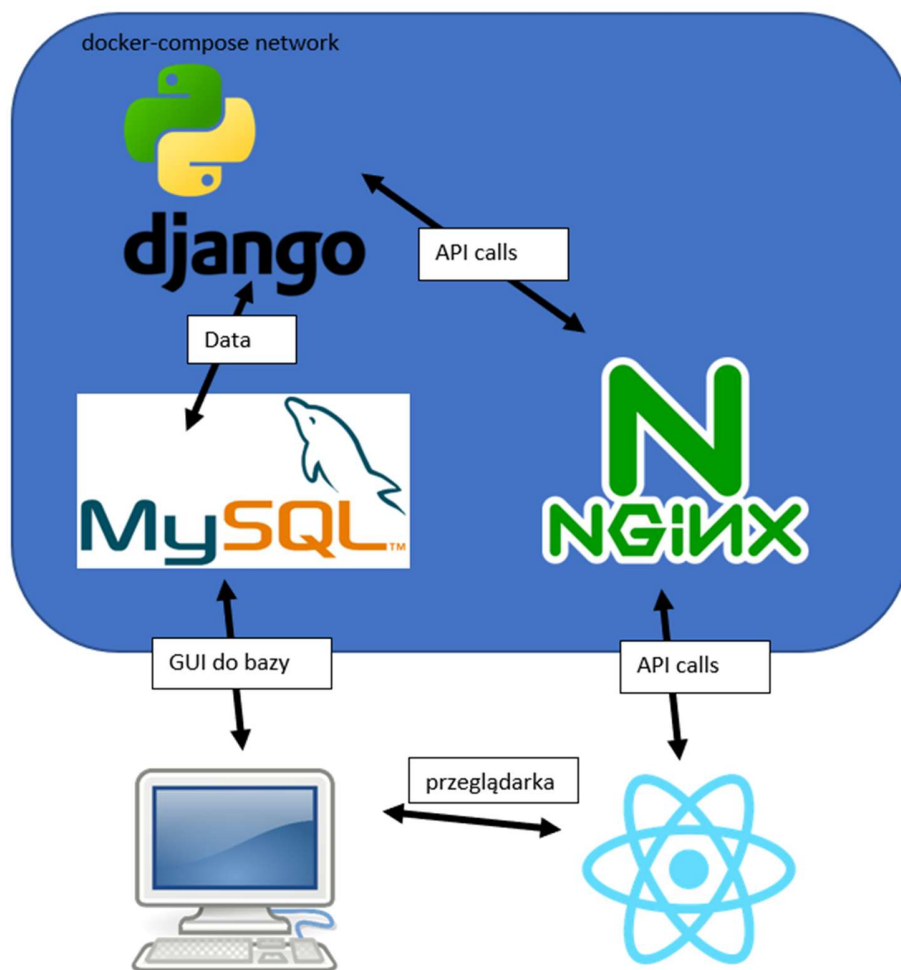
1. O projekcie

Aplikacja internetowa która pozwala nauczycielom tworzyć kursy, do których mogą zapisać się uczniowie. Aplikacja wykorzystuje elementy gamifikacji w celu uatrakcyjnienia przekazywania wiedzy. Uczniowie tworzą postać w ramach kursu, która jest ulepszana poprzez wykonywanie zadań umieszczonych na kursie. Uczniowie mogą wykorzystać tę postać do wykonania misji co daje im bonusy w ramach kursu. W ramach projektu stworzony będzie kurs mający kształtować nawyki proekologiczne.

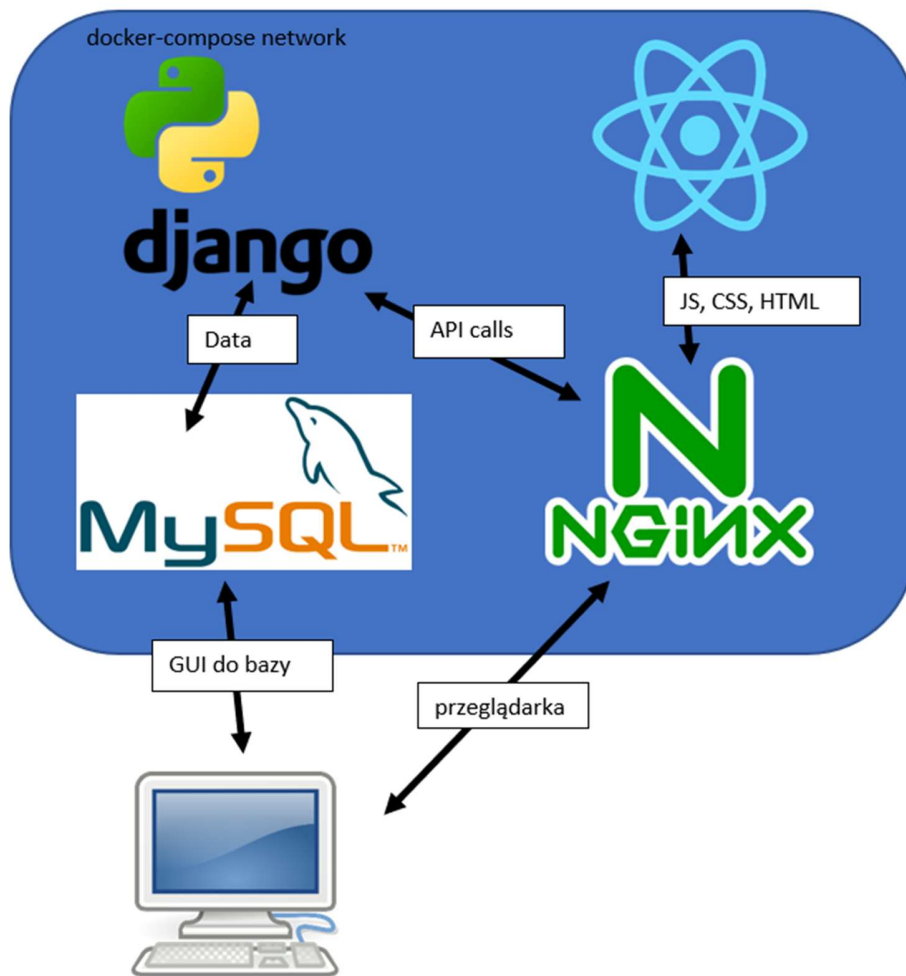
2. Architektura

Do zbudowania naszej aplikacji postanowiliśmy wykorzystać kontenery dockerowe. Zdecydowaliśmy się na to rozwiązanie ponieważ dzięki niemu unikniemy błędów wynikających z różnych wersji zainstalowanych programów na naszych komputerach oraz unikniemy problemów związanych z brakiem wymaganych modułów. Będziemy wykorzystywać 4 kontenery, na których będą postawione: baza danych, backend, frontend oraz proxy do naszej strony internetowej.

Podczas pracy nad naszą aplikacją, by w prosty sposób modyfikować wygląd naszej strony, będziemy mieli następujący model architektury:



Gdy nasza aplikacja będzie już gotowa i będziemy chcieli udostępnić ją klientom to zastosujemy następujący model:



3. Technologie i gotowe komponenty

Do stworzenia naszego systemu wykorzystaliśmy jedno z najbardziej popularnych frameworków używanych do tworzenia stron internetowych:

- Docker - wirtualne środowisko zbudowane z kontenerów z niezbędnymi komponentami naszej aplikacji. Wszyscy developerzy mogą łatwo pobrać daną wersję systemu. W kontenerze są potrzebne rzeczy już zainstalowane więc nie będzie problemów związanych z niezgodnością wersji. Łatwo coś takiego na zewnętrzny serwer przenieść
- Django - backend naszego systemu obsługujący zapytania API. Zaawansowane narzędzie, które dobrze wykorzystane świetnie poradzi sobie z bazą danych oraz z przekierowaniem pomiędzy stronami
- React - frontend, który jest połączenie html i javascript. Tworzymy nasze własne obiekty które później możemy wielokrotnie wykorzystać co się idealnie nada dla naszego pomysłu
- MySQL - jedna z bardziej popularnych relacyjnych baz danych, będziemy w niej trzymać dane
- Nginx - służy do obsługi certyfikatów oraz służy jako proxy dla naszej aplikacji

4. Zakres prototypu architektonicznego

4.1. Wymagania jakościowe

- wydajność - nasza aplikacja ma być szybka i użytkownicy nie powinni na nic czekać podczas korzystania z naszego systemu
- niezawodność - w przypadku gdy nastąpi jakiś błąd, np. padnie baza danych, powinna zostać załadowana kolejna baza danych, tak żeby użytkownicy nawet nie zauważyli tego błędu
- ochrona - użytkownicy nie mają bezpośredniego dostępu do bazy danych, ani backendu, korzystają z naszej strony wykorzystującej certyfikat SSL (Ngnix)
- skalowalność - wraz ze wzrostem użytkowników naszej aplikacji, nie powinniśmy mieć problemu z zwiększeniem wymagań ilościowych
- łatwość użycia - uczniowie oraz nauczyciele nie powinni mieć problemów z korzystaniem aplikacji

4.2. Funkcje produktu weryfikujące architekturę systemu

- Rejestracja i logowanie - każdy użytkownik musi mieć swoje własne konto, do którego nikt inny nie ma dostępu
- Stworzenie postaci - postacie mają odzwierciedlać użytkowników i wspomagać elementy gamifikacji w naszym systemie
- Tworzenie kursów - prowadzący powinni móc tworzyć kursy, do których będą zapisywać się uczniowie
- Tworzenie zadań - prowadzący powinni móc tworzyć zadania w danych kursach
- Zapisywanie się do kursów - klienci powinni w prosty sposób móc zapisać się do kursu
- Rozwiązywanie zadań - rozwiązywanie zadań powinno być intuicyjne, żeby osoby egzaminowane mogły się skupić na zadaniu a nie na tym jak przesłać odpowiedź
- System nagród - prowadzący powinien móc ustalać własne nagrody i kary dla poszczególnych czynności w kursie
- System nawyków - możliwość dodania nawyków, które mają za zadanie przypominać o powtarzających się czynnościach
- Tworzenie materiałów naukowych - prowadzący powinni móc przedstawiać wiedzę w swoich kursach w ciekawy sposób
- Rankingi - możliwość wyświetlania rankingu uczestników kursów oraz poszczególnych zadań

5. Podejście do testowania

Do przeprowadzania testów wykorzystamy program Selenium. Wraz z dodawaniem nowych funkcjonalności będziemy również rozwijać nasz tester. Testy będą automatyczne. Po pracy nad kodem, uruchamiamy testera i on da nam znać czy wszystko jest w porządku. Błędy zgłaszamy na GitHub w zakładce Issues z labels "bug".