



**Politechnika  
Śląska**

## Współczesne aplikacje webowe

Kierunek: Informatyka

Autor:

*Kamil Grabowski*

*Bartosz Bugla*

*Bartosz Maliniecki*

Gliwice, 2023/2024

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Cel projektu . . . . .	2
1.2	Role w projekcie . . . . .	2
1.3	Etapy pracy . . . . .	2
<b>2</b>	<b>Realizacja zadania</b>	<b>3</b>
2.1	Założenia projektu . . . . .	3
2.2	Narzędzia i technologie . . . . .	4
<b>3</b>	<b>Baza danych</b>	<b>5</b>
3.1	Model relacyjny . . . . .	5
3.2	Encje . . . . .	6
3.3	Związki . . . . .	6
<b>4</b>	<b>Wnioski</b>	<b>7</b>
4.1	Podsumowanie . . . . .	7

# 1 Wprowadzenie

## 1.1 Cel projektu

Projekt ma na celu rozwinięcie platformy wspierającej procesy dydaktyczne poprzez stworzenie systemu umożliwiającego wygodne przekazywanie, gromadzenie i zarządzanie różnorodnymi zasobami edukacyjnymi. Jest to narzędzie, które ma ułatwić nauczycielom oraz studentom organizację oraz dostęp do materiałów dydaktycznych oraz zadań związanych z danymi kursami.

## 1.2 Role w projekcie

- **Kamil Grabowski** - backend, stworzenie bazy danych, tworzenie dokumentacji
- **Bartosz Bugla** - frontend, stworzenie bazy danych, tworzenie dokumentacji
- **Bartosz Maliniecki** - frontend, backend, tworzenie dokumentacji

## 1.3 Etapy pracy

- **Etap 1.** - W ramach etapu pierwszego należy przygotować prezentację projektu systemu, w której zostanie przedstawiona jego architektura oraz opis stosu technologicznego. Dodatkowo, konieczne jest przygotowanie dokumentacji, która będzie zawierała szczegółowe informacje na temat projektowanej architektury, technologii użytych w projekcie oraz kluczowych założeń i celów systemu.
- **Etap 2.** - W ramach etapu drugiego należy przygotować prezentację części serwerowej systemu, która będzie zawierała kilka działających endpointów. Te endpointy będą odpowiadały za obsługę różnych żądań HTTP, umożliwiających interakcję z aplikacją przez klienta.
- **Etap 3.** - W ramach etapu trzeciego należy przygotować prezentację działającego w pełni systemu, który będzie zawierał wszystkie komponenty (zarówno frontend, jak i backend) oraz funkcjonalności zaimplementowane w ramach projektu. Ten etap obejmuje prezentację działającej aplikacji, w której użytkownicy będą mogli korzystać z wszystkich zaprojektowanych i zaimplementowanych funkcji.

## 2 Realizacja zadania

### 2.1 Założenia projektu

- **System Logowania** - Wprowadzenie mechanizmu logowania zapewnia, że dostęp do systemu mają tylko uprawnione osoby. Umożliwia to również personalizację doświadczenia użytkownika i zabezpiecza dane.
- **Role Użytkowników** - System przewiduje istnienie dwóch głównych ról: nauczyciela i studenta. Każda z tych ról posiada różne uprawnienia i możliwości w ramach systemu, co pozwala na lepszą organizację pracy i komunikację.
  - **Nauczyciel** - Może tworzyć nowe repozytoria (kursy), ogłaszać zadania, zarządzać uczestnikami kursów oraz oceniać nadesłane prace.
  - **Student** - Może dołączać do repozytoriów (kursów), składać wykonane zadania i otrzymywać informacje zwrotne.
- **Repozytoria Kursów** - Nauczyciele mogą tworzyć dedykowane repozytoria dla swoich kursów, które służą jako centralne miejsca do przekazywania materiałów, zadań i komunikacji z uczniami. Każde repozytorium może być indywidualnie skonfigurowane pod kątem dostępu i zasad uczestnictwa.
- **Zarządzanie Zadaniem** - Nauczyciele mogą ogłaszać zadania, wyznaczać terminy ich realizacji oraz oceniać prace nadesłane przez studentów. System pozwala na łatwe przeglądanie statusu realizacji zadań przez uczestników kursu.
- **Akceptacja Uczestników** - Dołączenie studenta do repozytorium kursu wymaga akceptacji nauczyciela, co daje kontrolę nad tym, kto może uczestniczyć w danym kursie.

## 2.2 Narzędzia i technologie

### 1. Frontend

**Framework:** React

**Język Programowania:** JavaScript/TypeScript

**Środowisko:** Visual Studio Code

### 2. Backend

**Framework:** .NET 8.0

**Język Programowania:** C#

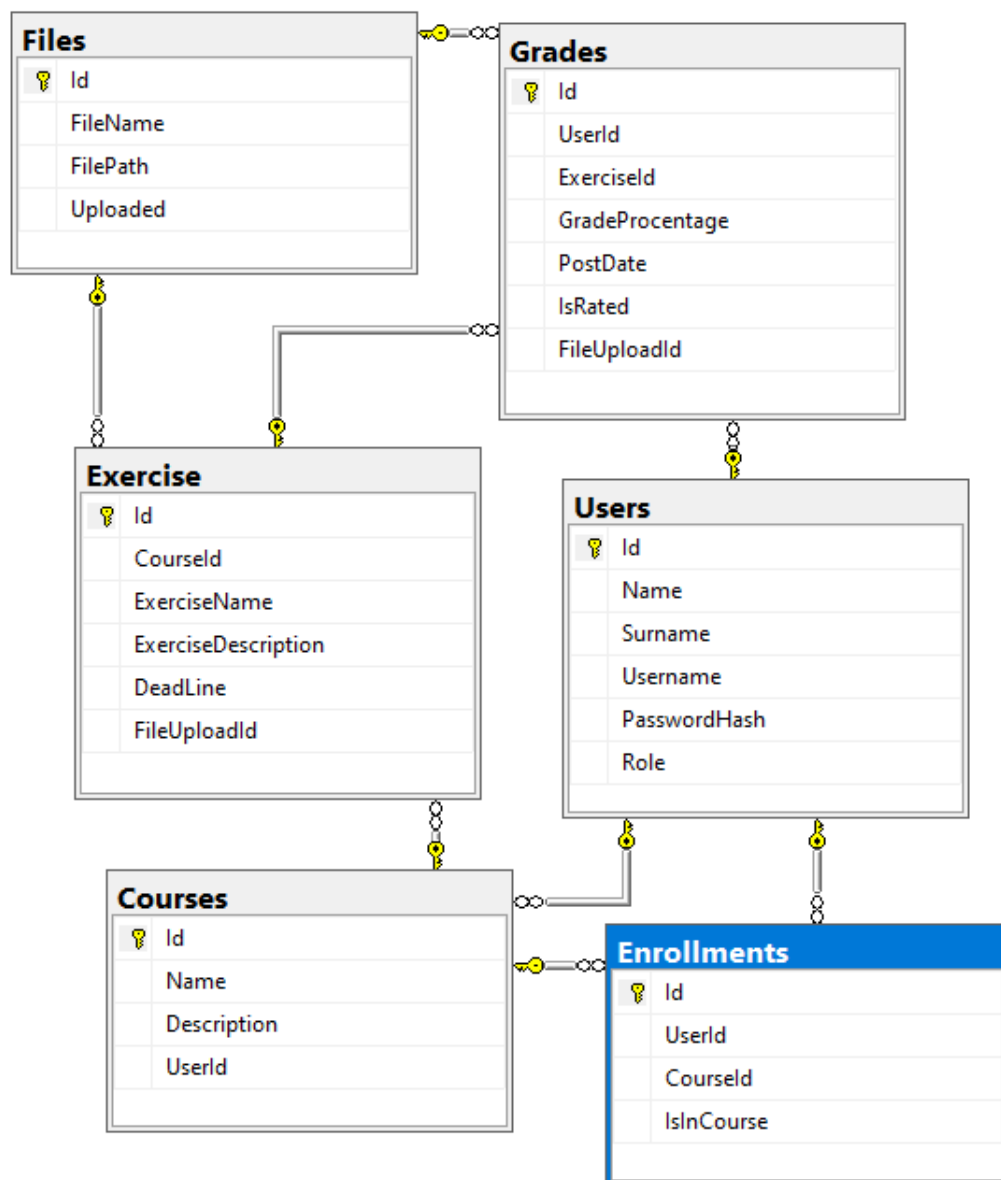
**Środowisko:** Visual Studio

### 3. Baza danych - Do zarządzania bazą danych wykorzystaliśmy *SQLite*. Została ona stworzona przy pomocy *Entity Framework Core*.

### 4. System kontroli wersji: Git

## 3 Baza danych

### 3.1 Model relacyjny



## 3.2 Encje

1. Users - ID User[PK], Name, Surname, Username, Password, Role(User or Teacher)
2. Courses - ID Course[PK], Name, Description
3. Enrollments - ID Enrollment[PK], Is in course? (bool)
4. Exercise - ID Exercise[PK], Name, Description, Deadline
5. Grades - ID Grades[PK], Grade, Post date, Is Rated (bool)
6. Files - ID File[PK], File name, File path, Uploaded (date)

## 3.3 Związki

1. Course
  - Jaki nauczyciel? - Course - User 1:n, nauczyciel może tworzyć wiele kursów
2. Enrollments
  - Jaki course? - Enrollment - Course 1:n, jakiego kursu dotyczy zapis na przedmiot
  - Jaki user? - Enrollment - User 1:n, każdy student może się zapisać do kursu
3. Exercise
  - Jaki course? - Exercise - Course 1:n, jakiego kursu dotyczy zadanie
  - Jaki file? - Exercise - File 1:n, jaki plik dotyczy zadania
4. Grades
  - Jaki user? - Grades - User 1:n, student może mieć wiele ocen
  - Jakie exercise? - Grades - Exercise 1:n, jakie zadanie dotyczy oceny
  - Jaki file? - Grades - File 1:n, jaki plik dotyczy zadania

## 4 Wnioski

Podczas całego procesu tworzenia projektu należy zwrócić szczególną uwagę na:

- Przeprowadzenie oceny stabilności, wydajności i bezpieczeństwa działania systemu, w celu identyfikacji ewentualnych problemów i możliwości ich poprawy.
- Analiza zgodności z założeniami projektowymi i oczekiwaniami użytkowników, aby ocenić, czy system spełnia założone cele i dostarcza wartość dodaną.
- Przeprowadzenie oceny użyteczności interfejsu użytkownika, aby zidentyfikować obszary wymagające ulepszeń i poprawić doświadczenie użytkownika.
- Wyciągnięcie wniosków i nauk z przeprowadzonego projektu, aby wykorzystać je w przyszłych działaniach i projektach.

### 4.1 Podsumowanie

Zaprezentowany plan projektu stanowi wstępną koncepcję i roadmapę działań, jednakże należy mieć na uwadze, że w trakcie pracy nad projektem mogą pojawić się zmiany i dostosowania. Elastyczność i zdolność do adaptacji do nowych warunków oraz potrzeb projektowych są kluczowe dla sukcesu realizacji projektu. Dlatego też plan może ulec ewolucji w miarę postępu prac i pojawienia się nowych informacji lub wymagań. Ważne jest, aby zespół projektowy był świadomy tej elastyczności i gotowy na ewentualne dostosowania w trakcie procesu realizacji projektu.