

Harmonogram

Czerwiec

- **Konsultacje u promotora**
- **Opracowanie funkcjonalności systemu**
 - Analiza wymagań funkcjonalnych
 - Określenie ról użytkowników (zawodnik, trener, administrator)
 - Specyfikacja procesów biznesowych

Lipiec

- **Projektowanie bazy danych**
 - Schemat bazy danych
 - Relacje między tabelami
 - Indeksy i optymalizacja
- **Przegląd technologii**
 - ASP.NET Core MVC
 - Entity Framework Core
 - PostgreSQL

Sierpień

- **Rozpisanie harmonogramu**
- **Konfiguracja środowiska**
- **Implementacja bazy danych**
 - Migracje Entity Framework
 - Modele danych

Wrzesień

- Projekt UI/UX
 - Makiety
- Praca nad wizualizacją projektu
- Ulepszanie utworzonej podstawowej wersji
 - Do zaimplementowania:
 - CategoryController (CRUD kategorie)
 - TrainingController (CRUD treningi)
 - ScheduleController (harmonogram grup)
 - AttendanceController (obecności)
 - EnrollmentController (zapisy do grup)

Październik

- Rozwijanie systemu
 - Moduł uwierzytelniania i autoryzacji:
 - System logowania/rejestracji
 - Zaawansowane funkcjonalności Group:
 - Automatyczne zarządzanie statusem grup
 - Walidacja liczby zawodników
 - System zapisów:
 - Lista oczekujących
 - Automatyczne potwierdzenia
 - Powiadomienia email?
 - System treningów:
 - Kalendarz treningów
 - Odwoływanie treningów
 - Zarządzanie uczestnikami
- Testowanie
 - Unit testy dla modeli
 - Testy

Listopad

- Realizacja kolejnych funkcjonalności
 - System obecności:
 - Rejestracja obecności

- Statystyki frekwencji
- **Harmonogram:**
 - Cykliczne treningi
 - Zarządzanie salami/miejscami
 - Konflikty terminów
- **Dashboard i raporty:**
 - Statystyki dla trenerów
 - Przegląd aktywności zawodników
- **Testowanie**
 - Testy funkcjonalne
 - Testy UI/UX

Grudzień

- Ostatnie testy systemu
- Poprawa ewentualnych błędów
- Zakończenie po stronie kodu
 - Dokumentacja kodu
- Przygotowanie dokumentacji technicznej
 - Instrukcja instalacji
 - Instrukcja administratora

Styczeń-Luty

- Pisanie pracy dyplomowej
 - Opis architektury systemu
 - Analiza zastosowanych technologii
 - Prezentacja funkcjonalności
 - Wnioski i kierunki rozwoju
- Przygotowanie prezentacji

- Demo systemu
 - Slajdy z kluczowymi punktami
 - Przygotowanie do obrony
-

Kluczowe funkcjonalności do zaimplementowania

Priorytety wysokie:

1. System autoryzacji - logowanie, role, bezpieczeństwo
2. Kompletny CRUD dla wszystkich encji (Category, Training, Schedule, Attendance, Enrollment)
3. Proces zapisów do grup - walidacja, lista oczekujących
4. Kalendarz treningów - tworzenie, odwoływanie, zarządzanie

Priorytety średnie:

1. System powiadomień - email notifications
2. Raporty i statystyki - frekwencja, aktywność grup
3. Import/export danych - CSV, Excel
4. Mobile-responsive design

Priorytety niskie:

1. API REST dla integracji z zewnętrznymi systemami
2. Zaawansowane filtrowanie i wyszukiwanie
3. System płatności (jeśli wymagany)
4. Wielojęzyczność

Funkcjonalności do zrealizowania

Zarządzanie użytkownikami i grupami:

- System grup zajęciowych z możliwością tworzenia, edycji i usuwania grup

- Zarządzanie uczestnikami związanymi z grupami (role: uczestnik, prowadzący, administrator)
- Hierarchia uprawnień i ról w systemie
- Profile użytkowników z danymi personalnymi i historią aktywności

Administracja systemu:

- Panel administratorów z pełnymi uprawnieniami do zarządzania systemem
- Możliwość moderacji grup i użytkowników
- System logowania i monitorowania aktywności

Mechanizm zapisów z priorytetami:

- System kolejek i priorytetów przy zapisywaniu na zajęcia
- Algorytmy przydzielania miejsc w grupach na podstawie kryteriów

System nagród i statystyk (opcjonalnie):

- Zbieranie i prezentacja statystyk aktywności użytkowników
- System punktów lub nagród za uczestnictwo
- Analiza frekwencji i zaangażowania uczestników

Lista rankingowa i cele (opcjonalnie):

- Ranking użytkowników na podstawie aktywności i osiągnięć
- System celów i wyzwań dla uczestników
- Śledzenie postępów i osiągnięć

Funkcjonalności podstawowe:

- Rejestracja i logowanie użytkowników
- Przeglądanie dostępnych grup i zajęć
- System powiadomień i komunikacji
- Kalendarz zajęć i harmonogram
- Możliwość oceniania i komentowania zajęć

Architektura aplikacji:

- Podział na moduły funkcjonalne (użytkownicy, grupy, płatności, statystyki)
- Separacja logiki biznesowej od warstwy prezentacji

Podmienność i konfigurowalność:

- Możliwość wymiany poszczególnych modułów bez wpływu na całość

- Konfiguracja przez pliki konfiguracyjne lub panel administracyjny

System płatności (opcjonalnie):

- Obsługa metod płatności
- Zarządzanie subskrypcjami i opłatami za zajęcia