

1. **Tytuł projektu:** System Optymalizacji Ewolucyjnej Harmonogramów

2. **Akronim:** OptiSlots

3. **Lista członków zespołu:**

- Piotr Bonar
- Jakub Borsuk
- Aleksander Stepaniuk
- Kacper Zakrzewski

4. **Opiekun zespołu:** Michał Przewoźniczek

5. **Logo projektu:**



## **6. Syntetyczny opis**

Projekt *Systemu Optymalizacji Ewolucyjnej Harmonogramów* ma na celu optymalizację procesu tworzenia harmonogramów, znacząco redukując przy tym nakład pracy administracyjnej. Zamiast sztywnych tur rejestracji i jednostronnego narzucania spotkań, proces polega na asynchronicznym zbieraniu preferencji użytkowników w pewnym oknie czasowym. Następnie system iteracyjnie ulepsza plan z wykorzystaniem algorytmu genetycznego. Jednostka odpowiedzialna za tworzenie harmonogramu ma możliwość konfiguracji całego procesu, ustalając kluczowe parametry, takie jak czas trwania dnia, godziny wyłączone z planowania oraz niezbędne zasoby (np. wymagane sale z określonymi parametrami) dla danych grup uczestników.

### **Główne cele biznesowe**

Zwiększenie efektywności operacyjnej i zmniejszenie liczby błędów w procesie poprzez automatyzację tworzenia harmonogramów dla dużych grup użytkowników. Głównym celem jest maksymalizacja satysfakcji użytkowników poprzez lepsze dopasowanie planów do ich indywidualnych potrzeb.

### **Założenia**

System generuje harmonogramy w oparciu o preferencje użytkowników, dążąc przy tym do poprawy ich ogólnej jakości i średniej satysfakcji, może jednak nie spełnić wszystkich indywidualnych oczekiwań.

### **Kluczowe ryzyka**

Algorytm genetyczny jest z natury niedeterministyczny, co stanowi kluczowe ryzyko. Musi on zostać precyzyjnie skalibrowany, aby zwracał akceptowalne rozwiązania w rozsądnym czasie obliczeniowym. System musi bezwzględnie egzekwować twarde ograniczenia, aby nie generować niepoprawnych planów.

## Technologie:

Python, Django, Django-REST, Django-Admin, C++, CMake, Redis, PostgreSQL, Docker, Docker-Compose, Git, GitHub, Next.js, React, Tailwind-CSS, HTML5, CSS3, Postman, Vercel, Node.js

### 7.1 Roadmapa projektu

**Wrzesień 2025:** Analiza wymagań, przygotowanie makiet UI/UX i opracowanie architektury systemu.

**Październik 2025:** Implementacja backendu i frontendu, integracja przez REST API oraz rozwój algorytmów optymalizacyjnych.

**Listopad 2025:** Dodanie logiki użytkownika, testy systemu, zabezpieczenia i przygotowanie wersji beta.

**Grudzień 2025:** Finalizacja dokumentacji, prezentacja wersji demonstracyjnej i zakończenie projektu.

### 7.2 Wersja jeszcze krótsza?

**Wrzesień:** plan i projekt

**Październik:** implementacja

**Listopad:** testy i dopracowanie

**Grudzień:** prezentacja i zakończenie

