

Collegium Witelona Uczelnia Państwowa w Legnicy  
Informatyka, 3 rok  
Semestr zimowy 2024/2025

# Projektowanie i programowanie systemów internetowych I I projekt

## Sprawozdanie



Dawid Skowroński  
Jakub Trznadel  
Jakub Kozubek  
Mateusz Tęcza

Prowadzący: mgr inż. Krzysztof Rewak

# Spis treści

<b>1</b>	<b>Opis projektu</b>	<b>2</b>
<b>2</b>	<b>Opis poszczególnych ról w grupie</b>	<b>2</b>
<b>3</b>	<b>Opis wymagań i założeń projektowych</b>	<b>2</b>
3.1	Wymagania funkcjonalne . . . . .	2
3.2	Wymagania нефункционалне . . . . .	3
3.3	Założenia projektowe . . . . .	4
<b>4</b>	<b>Opis działania aplikacji / systemu</b>	<b>5</b>
4.1	Komunikacja i architektura . . . . .	5
<b>5</b>	<b>Opis wykorzystanych technologii, narzędzi i rozwiązań technicznych</b>	<b>5</b>
<b>6</b>	<b>Opis event stormingu + zrzuty z modelowania</b>	<b>6</b>
<b>7</b>	<b>Backlog</b>	<b>7</b>
<b>8</b>	<b>Estymata</b>	<b>8</b>
<b>9</b>	<b>Opis sprintów + zrzuty</b>	<b>9</b>
9.1	Sprint 1: Podstawy aplikacji i logowanie użytkownika . . . . .	9
9.2	Sprint 2: Profil użytkownika . . . . .	9
9.3	Sprint 3: Tworzenie i zarządzanie wyjazdami . . . . .	9
9.4	Sprint 4: System wydatków . . . . .	10
9.5	Sprint 5: Rozliczanie wyjazdów i powiadomienia . . . . .	10
9.6	Sprint 6: Panel administratora . . . . .	10
9.7	Sprint 7: Testy i wdrożenie . . . . .	10
<b>10</b>	<b>Interfejs aplikacji (zrzuty) + krótki opis</b>	<b>11</b>
<b>11</b>	<b>Informacje uruchomieniowe w środowisku developerskim</b>	<b>15</b>
<b>12</b>	<b>Opis wdrożenia aplikacji i sposobu hostowania</b>	<b>16</b>
<b>13</b>	<b>Podsumowanie, wnioski</b>	<b>16</b>

# 1 Opis projektu

Projekt zakłada stworzenie systemu do zarządzania wyjazdami i wspólnymi wydatkami, gdzie użytkownicy mogą rejestrować się za pomocą e-maila lub Facebooka, tworzyć wyjazdy, zapraszać innych, dodawać wydatki oraz zarządzać płatnościami. Wydatki, z przypisaną kwotą, walutą, opisem, datą, kategorią i opcjonalnym miejscem, są domyślnie dzielone po równo między uczestników, ale można to zmieniać. Użytkownicy mogą oznaczać swoje części jako zapłacone, co musi zostać zatwierdzone przez autora wydatku, a płatności są tagowane różnymi metodami.

## 2 Opis poszczególnych ról w grupie

### **Skład zespołu**

- Dawid Skowroński - Team leader/Backend
- Jakub Trznadel - Frontend
- Jakub Kozubek - Tester
- Mateusz Tęcza - Backend

## 3 Opis wymagań i założeń projektowych

### 3.1 Wymagania funkcjonalne

#### **Rejestracja i logowanie:**

- Możliwość rejestracji użytkownika za pomocą e-maila oraz konta na Facebooku.
- Funkcjonalność logowania oraz odzyskiwania zapomnianego hasła.

#### **Tworzenie i zarządzanie wyjazdami:**

- Użytkownicy mogą tworzyć wyjazdy i zapraszać innych uczestników.
- Możliwość dodania uczestników jako „offline”.

#### **Dodawanie wydatków:**

- Użytkownicy mogą dodawać wydatki z dokładnym określeniem kwoty, waluty, opisu, daty oraz kategorii.
- Istnieje możliwość dodania opcjonalnego miejsca (np. restauracji lub sklepu).

#### **Zarządzanie podziałem kosztów:**

- Wydatki mogą być przypisywane do konkretnych uczestników wyjazdu.
- Możliwość równomiernego podziału kosztów lub dostosowanego podziału, w zależności od potrzeb.

#### **Zarządzanie płatnościami:**

- Użytkownicy mogą oznaczać część wydatku jako zapłaconą.
- Akceptacja płatności przez autora wydatku.
- Możliwość przypisania formy płatności (np. gotówka, Blik, Revolut).

#### **Podsumowanie kosztów:**

- Użytkownicy mogą zobaczyć łączną kwotę do spłaty oraz szczegóły rozliczeń między uczestnikami.

#### **Edycja profilu:**

- Użytkownicy mają możliwość edycji swojego profilu i zmiany hasła.

#### **Panel administratora:**

- Administrator posiada panel do zarządzania użytkownikami, w tym możliwość ich usuwania czy edytowania uprawnień.

#### **Backend i frontend:**

- Oddzielony backend i frontend zapewniający większą skalowalność i łatwość rozwoju systemu.
- Komunikacja między frontendem a backendem przez REST API.

#### **Dokumentacja OpenAPI:**

- Pełna dokumentacja API backendu w standardzie OpenAPI, umożliwiająca łatwą integrację i rozszerzanie funkcjonalności.

## **3.2 Wymagania niefunkcjonalne**

### **Wydajność:**

- Optymalizacja czasu odpowiedzi systemu, szczególnie przy dodawaniu wydatków czy przeglądaniu podsumowań.

### **Skalowalność:**

- System musi być w stanie obsługiwać rosnącą liczbę użytkowników i wyjazdów bez pogarszania wydajności.
- Możliwość rozbudowy o nowe funkcjonalności w przyszłości.

### **Niezawodność:**

- System musi być odporny na błędy, z minimalnym czasem przestoju i stabilnością działania.
- Przechowywanie danych użytkowników i wydatków w bezpieczny sposób, zapewniając integralność danych.

### **Bezpieczeństwo:**

- Uwierzytelnianie i autoryzacja z użyciem standardów, takich jak OAuth dla logowania przez social media.

### 3.3 Założenia projektowe

#### Technologie:

- Backend aplikacji oparty będzie na ASP.NET, ponieważ jest to framework, z którym zespół ma doświadczenie, a także zapewnia on doskonałą integrację z ekosystemem .NET, co umożliwia łatwą rozbudowę i optymalizację aplikacji.
- C# zostanie wybrany jako główny język programowania w projekcie, co zapewni wysoką wydajność, łatwość rozbudowy kodu i spójność z ASP.NET.
- Frontend aplikacji zostanie zbudowany przy użyciu JavaScript jako głównego języka skryptowego, który jest szeroko znany i dobrze wspierany.
- HTML i CSS będą wykorzystywane do tworzenia struktury oraz stylizacji stron aplikacji, co pozwoli na uzyskanie responsywnego i estetycznego interfejsu użytkownika.
- Bootstrap będzie używany do stylizacji interfejsu użytkownika, zapewniając szybki i responsywny design, który był już wykorzystywany w poprzednich projektach zespołu.

#### Środowiska deweloperskie:

- Visual Studio będzie głównym środowiskiem IDE dla programistów Backend, ze względu na pełną integrację z ASP.NET oraz bazą danych MsSQL, a także wsparcie dla rozwoju aplikacji w C#.
- Visual Studio Code będzie używane przez programistów Frontend, ponieważ jest to lekkie i elastyczne środowisko z dużą liczbą rozszerzeń, ułatwiających pracę nad kodem JavaScript/TypeScript.

#### Baza danych:

- Do przechowywania danych aplikacji zostanie wybrana MsSQL. Jest to baza danych, która współpracuje z .NET, a także jest dobrze znana zespołowi, co zapewni szybkie i efektywne zarządzanie danymi użytkowników oraz transakcjami.

#### Komunikacja i zarządzanie projektem:

- Zespół będzie korzystać z Discord jako narzędzia komunikacyjnego, co pozwoli na sprawny przepływ informacji oraz bieżące rozwiązywanie problemów w projekcie.
- Kod źródłowy aplikacji będzie zarządzany przy użyciu Github, co umożliwi wersjonowanie, współpracę, kontrolowanie zmian i zapewni przejrzystość w procesie developmentu.

#### Komunikacja między backendem a frontendem:

- REST API będzie wykorzystywane do komunikacji pomiędzy frontendem a backendem aplikacji, co umożliwi łatwą integrację z różnymi platformami i zapewni oddzielenie warstwy prezentacji od logiki aplikacji.

#### Zgodność z urządzeniami mobilnymi:

- Aplikacja będzie responsywna, co oznacza, że będzie działać poprawnie na urządzeniach mobilnych (smartfony, tablety) oraz komputerach stacjonarnych.

#### **Testowanie:**

- Przed wdrożeniem aplikacja zostanie dokładnie przetestowana.
- Testowanie obejmować będzie również testy integracyjne oraz UI/UX, aby upewnić się, że interfejs użytkownika jest intuicyjny, a aplikacja działa płynnie na różnych urządzeniach.

## **4 Opis działania aplikacji / systemu**

Aplikacja opiera się na architekturze klient-serwer z oddzielnym frontendem i backendem. Backend został zaimplementowany w ASP.NET Core i udostępnia REST API do komunikacji z frontendem. Baza danych to Microsoft SQL Server, w której przechowywane są informacje o użytkownikach, wyjazdach, wydatkach i transakcjach. Frontend został zbudowany w i komunikuje się z backendem za pomocą żądań HTTP (GET, POST, PUT, DELETE).

### **4.1 Komunikacja i architektura**

Frontend komunikuje się z backendem poprzez REST API, wysyłając zapytania w formacie JSON. Backend obsługuje autoryzację i walidację danych oraz wykonuje operacje na bazie danych. Baza danych przechowuje encje takie jak Users, Trips, Expenses, Payments, korzystając z Entity Framework Core. Serwer jest hostowany na webio.pl (backend) oraz seohost.pl (frontend).

## **5 Opis wykorzystanych technologii, narzędzi i rozwiązań technicznych**

- ASP.NET - wybrany jako główny framework do tworzenia aplikacji z uwagi na doświadczenie zespołu oraz jego doskonałą integrację z ekosystemem .NET.
- C# - preferowany język programowania w projekcie, będący najczęściej używanym narzędziem w pracy z .NET, zapewniający wydajność i łatwość rozbudowy kodu.
- HTML i CSS - podstawowe technologie do tworzenia struktury i stylizacji stron internetowych.
- JavaScript - główny język Frontend-u, dobrze znany i wygodny dla programistów odpowiedzialnych za interfejs użytkownika.
- TypeScript - rozszerzenie JavaScriptu, zapewniające lepszą kontrolę typów i większą czytelność kodu, wykorzystywane tam, gdzie potrzebna jest skalowalność i wydajność.
- Visual Studio - preferowane środowisko IDE dla programistów Backend, z uwagi na rozbudowaną integrację z ASP.NET i MsSQL.

- Visual Studio Code - środowisko znane i używane przez Frontend do efektywnej pracy nad kodem.
- Bootstrap - stosowany już w poprzednich projektach przez programistę Frontend.
- Discord - narzędzie komunikacyjne używane przez zespół, zapewniające sprawny przepływ informacji.
- MsSQL - baza danych wybrana ze względu na efektywną współpracę z .NET oraz znajomość przez zespół.
- Github - platforma do zarządzania kodem źródłowym, wspierająca wersjonowanie, współpracę oraz kontrolę zmian.
- Trello - narzędzie ułatwiające tworzenie i zarządzanie sprintami podczas projektu.
- Mural - narzędzie do przeprowadzenia sesji event stormingu.
- xUnit - do testów jednostkowych.
- Cypress - do testów E2E.

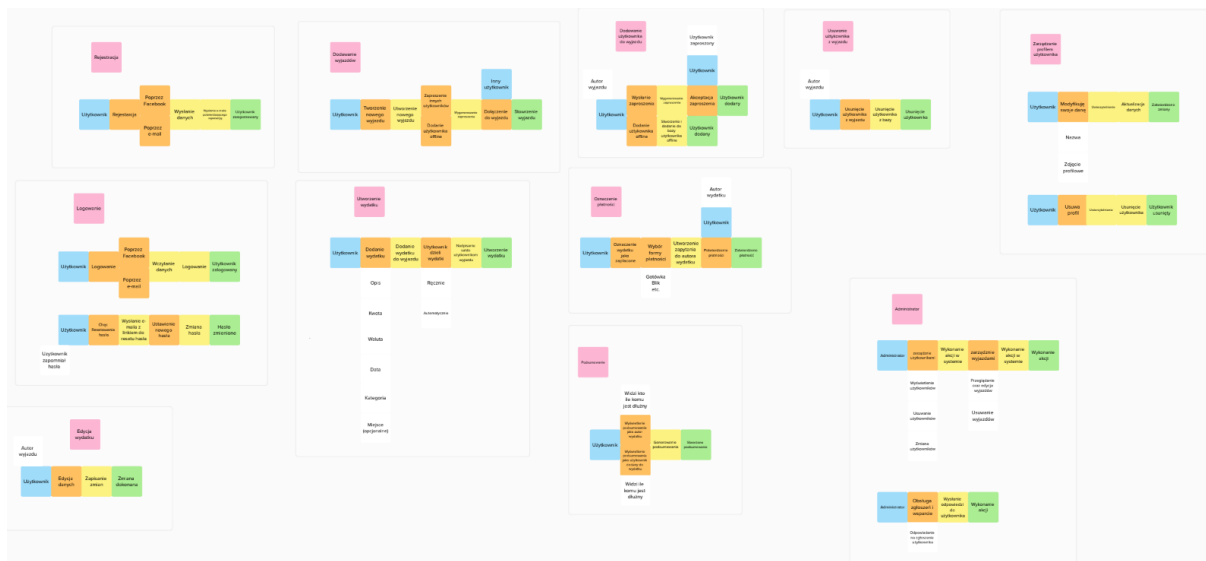
## 6 Opis event stormingu + zrzuty z modelowania

Do przeprowadzenia sesji wykorzystaliśmy Mural oraz Discord. Całość sesji trwała 2 godziny. W tym czasie powstało wiele grup zdarzeń zakładające różne scenariusze korzystania z aplikacji jako użytkownik bądź administrator. Każdy z członków grupy przedstawił swój pomysł na działanie poszczególnych funkcji aplikacji, a następnie wspólnie zebraliśmy je w całość. Na podstawie wytycznych projektowych, które posłużyły nam jako punkt odniesienia, ustaliliśmy kluczowe zdarzenia i mechanizmy działania aplikacji. Proces ten pozwolił nam na wypracowanie kompletnego zarysu funkcji, który spełnia założenia projektowe oraz uwzględnia potrzeby przyszłych użytkowników.



Rysunek 1: Legenda i zrzut z modelowania

## Efekt końcowy:



Rysunek 2: Efekt końcowy z sesji event storming

## 7 Backlog

- Utworzenie szkieletu aplikacji
- Utworzenie bazy danych
- Utworzenie podstawowego interfejsu testowego
- Rejestracja użytkownika
- Logowanie użytkownika
- Logowanie przez Facebook'a
- Wprowadzenie profilu użytkownika
- Implementacja edycji danych użytkownika
- Resetowanie hasła na profilu
- Tworzenie wyjazdu
- Implementacja wprowadzania danych wyjazdu (miejsce/data/itp)
- Implementacja zapraszania użytkownika do wyjazdu
- Wprowadzenie kodów zaproszeń
- Zaimplementowanie systemu wydatków:
  - Utworzenie opcji dodania wydatku do wyjazdu



- Utworzenie opcji edycji wydatku
- Implementacja systemu dzielenia wydatków
- Implementacja systemu ręcznego dzielenia wydatków
- Implementacja historii wydatków
- Wprowadzenie edycji wyjazdów
- Implementacja systemu rozliczania całości wyjazdu
- Powiadomienia
- Rozliczanie wyjazdów
  - Zaznaczenie uregulowania długu
  - Zaznaczenie metody płatności
- Panel administratora
  - Wprowadzenie zarządzania użytkownikami
  - Wprowadzenie zarządzania wyjazdami
  - Wprowadzenie zarządzania zgłoszeniami błędów
- Implementacja i przeprowadzenie testów
- Utworzenie ostatecznej wersji interfejsu użytkownika
- Deployment

## 8 Estymata

Poszczególne zadania	Dawid Skowroński	Jakub Trznadel	Mateusz Tęcza	Jakub Kozubek	Średnia
Utworzenie szkieletu aplikacji	4.00.00	6.00.00	3.00.00	4.30.00	4.22.30
Utworzenie bazy danych	1.00.00	1.00.00	1.00.00	1.00.00	1.00.00
Utworzenie podstawowego interfejsu testowego	1.30.00	1.00.00	1.00.00	0.30.00	1.00.00
Rejestracja użytkownika	2.00.00	1.30.00	2.00.00	2.00.00	1.52.30
Logowanie użytkownika	2.00.00	3.00.00	2.00.00	2.00.00	2.15.00
Logowanie przez Facebook'a	1.30.00	2.00.00	1.30.00	1.00.00	1.30.00
Wprowadzenie profilu użytkownika	4.00.00	4.00.00	4.00.00	4.00.00	4.00.00
Implementacja edycji danych użytkownika	2.00.00	1.30.00	1.00.00	1.30.00	1.30.00
Implementacja edycji zdjęcia użytkownika	1.00.00	1.30.00	1.30.00	1.30.00	1.22.30
Resetowanie hasła na profilu	1.00.00	1.00.00	1.30.00	1.00.00	1.07.30
Tworzenie wyjazdu	15.00.00	15.00.00	15.00.00	16.30.00	15.22.30
Implementacja wprowadzania danych wyjazdu miejsce/data/tp	7.00.00	6.00.00	6.00.00	8.00.00	6.45.00
Implementacja zapraszania użytkownika do wyjazdu	4.00.00	3.00.00	4.00.00	4.00.00	3.45.00
Wprowadzenie kodów zaproszeń	1.00.00	1.00.00	2.00.00	2.00.00	1.30.00
Implementacja historii wydatków(listy)	2.00.00	4.00.00	2.30.00	2.00.00	2.37.30
Wprowadzenie edycji wyjazdów	1.00.00	1.00.00	0.30.00	0.30.00	0.45.00
Zaimplementowanie systemu wydatków	16.00.00	17.30.00	16.30.00	17.30.00	16.52.30
Utworzenie opcji dodania wydatku do wyjazdu	5.00.00	5.30.00	5.00.00	5.00.00	5.07.30
Utworzenie opcji edycji wydatku	1.00.00	1.00.00	0.30.00	1.00.00	0.52.30
Implementacja systemu dzielenia wydatków	4.00.00	4.00.00	4.30.00	4.00.00	4.07.30
Implementacja systemu ręcznego dzielenia wydatków	2.00.00	2.00.00	1.30.00	1.30.00	1.45.00
Implementacja systemu rozliczania całości wyjazdu	4.00.00	5.00.00	5.00.00	6.00.00	5.00.00
Powiadomienia	2.00.00	3.00.00	2.30.00	3.00.00	2.37.30
Panel Administratora	4.00.00	5.00.00	3.30.00	3.30.00	4.00.00
Wprowadzenie zarządzania użytkownikami	2.00.00	2.00.00	1.30.00	1.00.00	1.37.30
Wprowadzenie zarządzania wyjazdami	1.00.00	2.00.00	1.00.00	1.30.00	1.22.30
Wprowadzenie zarządzania zgłoszeniami błędów	1.00.00	1.00.00	1.00.00	1.00.00	1.00.00
Implementacja i przeprowadzenie testów	6.00.00	7.00.00	7.00.00	8.30.00	7.07.30
Utworzenie ostatecznej wersji interfejsu użytkownika	20.00.00	20.00.00	18.00.00	25.00.00	20.45.00
Deployment	25.00.00	20.00.00	25.00.00	15.00.00	21.15.00
Suma	104.00.00	106.00.00	102.00.00	104.00.00	104.00.00

Rysunek 3: Estymata

## 9 Opis sprintów + zrzuty

Czas trwania sprintów: 1 tydzień

### 9.1 Sprint 1: Podstawy aplikacji i logowanie użytkownika

Cel: Stworzenie szkieletu aplikacji oraz funkcjonalności związanych z rejestracją i logowaniem użytkownika.

Zakres zadań:

- Utworzenie szkieletu aplikacji.
- Utworzenie bazy danych.
- Utworzenie podstawowego interfejsu testowego.
- Rejestracja użytkownika.
- Logowanie użytkownika.
- Logowanie przez Facebook'a.

### 9.2 Sprint 2: Profil użytkownika

Cel: Wprowadzenie profilu użytkownika z możliwością edycji danych.

Zakres zadań:

- Wprowadzenie profilu użytkownika.
- Implementacja edycji danych użytkownika.
- Resetowanie hasła na profilu.

### 9.3 Sprint 3: Tworzenie i zarządzanie wyjazdami

Cel: Dodanie funkcjonalności tworzenia i edycji wyjazdów.

Zakres zadań:

- Tworzenie wyjazdu.
- Implementacja wprowadzania danych wyjazdu (miejsce/data/itp).
- Implementacja zapraszania użytkownika do wyjazdu.
- Wprowadzenie kodów zaproszeń.
- Wprowadzenie edycji wyjazdów.

## 9.4 Sprint 4: System wydatków

Cel: Implementacja systemu wydatków dla wyjazdów.

Zakres zadań:

- Utworzenie opcji dodania wydatku do wyjazdu.
- Utworzenie opcji edycji wydatku.
- Implementacja systemu dzielenia wydatków.
- Implementacja systemu ręcznego dzielenia wydatków.
- Implementacja historii wydatków (listy).

## 9.5 Sprint 5: Rozliczanie wyjazdów i powiadomienia

Cel: Dodanie funkcji rozliczania wyjazdów i systemu powiadomień.

Zakres zadań:

- Implementacja systemu rozliczania całości wyjazdu.
- Rozliczanie wyjazdów:
  - Zaznaczenie uregulowania długu.
  - Zaznaczenie metody płatności.
- Powiadomienia.

## 9.6 Sprint 6: Panel administratora

Cel: Wprowadzenie funkcji dla administratora.

Zakres zadań:

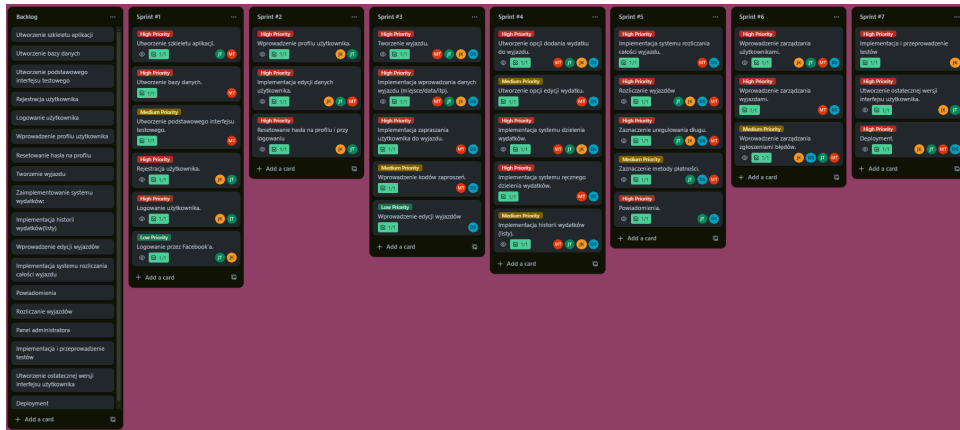
- Wprowadzenie zarządzania użytkownikami.
- Wprowadzenie zarządzania wyjazdami.
- Wprowadzenie zarządzania zgłoszeniami błędów.

## 9.7 Sprint 7: Testy i wdrożenie

Cel: Zakończenie projektu i przygotowanie aplikacji do wdrożenia.

Zakres zadań:

- Implementacja i przeprowadzenie testów.
- Utworzenie ostatecznej wersji interfejsu użytkownika.
- Deployment.

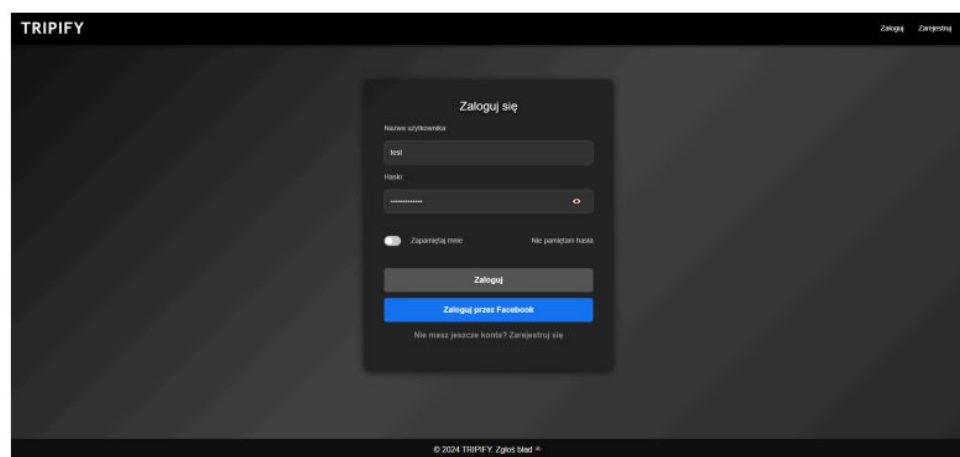


Rysunek 4: zrzut ekranu z Trello

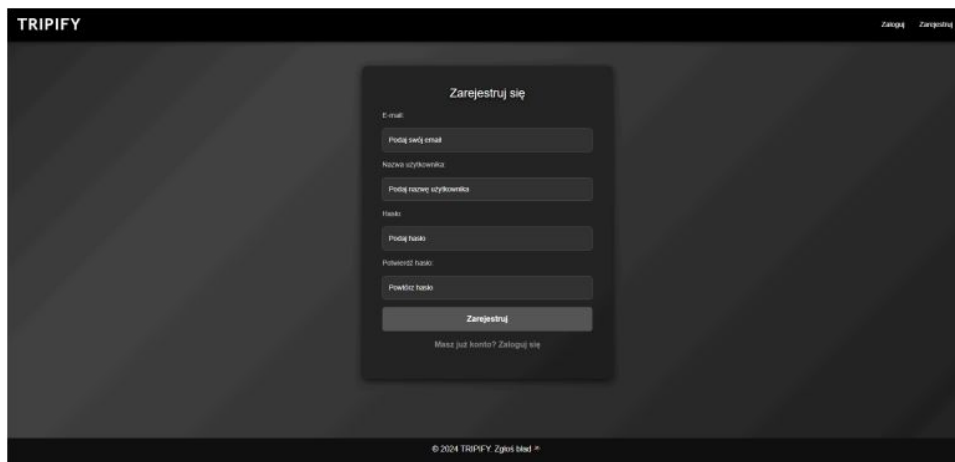
## 10 Interfejs aplikacji (zrzuty) + krótki opis



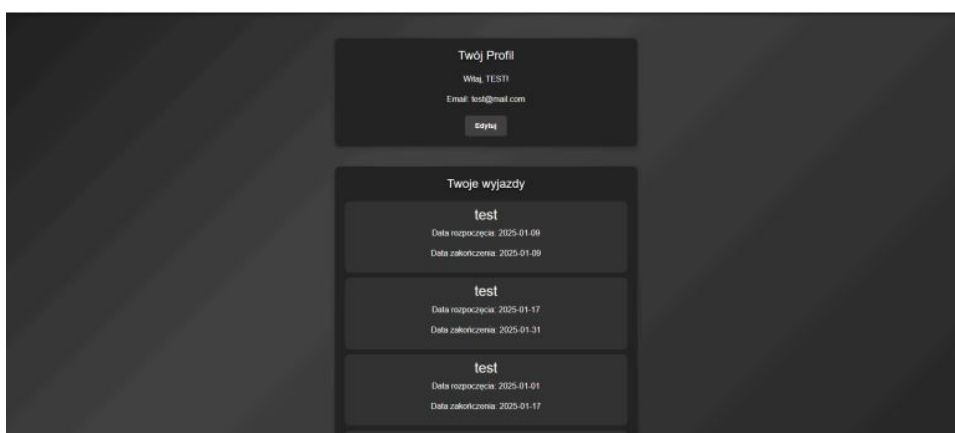
Rysunek 5: Strona powitalna na którą można stworzyć lub dołączyć do wyjazdu.



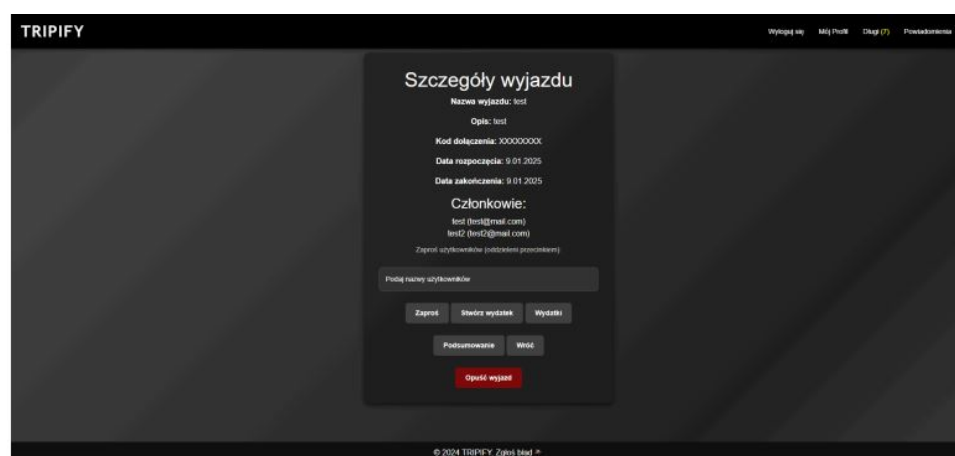
Rysunek 6: Logowanie.



Rysunek 7: Rejestracja.



Rysunek 8: Profil na którym widoczne są nasze dane jak i wyjazdy.



Rysunek 9: Szczegóły wyjazdu, na których możemy zobaczyć członków wyjazdu oraz zarządzać nim.

**Podsumowanie**

Wybierz walutę: PLN

KATEGORIA	ŁĄCZNA KWOTA
Transport	0.00 PLN
Jedzenie	40.00 PLN
Zakwaterowanie	0.00 PLN
Rozrywka	50.00 PLN
Inne	0.00 PLN
<b>ŁĄCZNIE</b>	<b>90.00 PLN</b>

© 2024 TRIPIFY. Złoty błęd \*

Rysunek 10: Tworzenie wyjazdu.

**Stwórz nowy wyjazd**

Proszę wpisać wyjazd

Proszę wpisać wyjazd

Typ wyjazdu

Proszę wybrać

Data rozpoczęcia

01.01.2024

Data zakończenia

01.01.2024

Zapisać wydatki (nieobowiązkowo)

Proszę wpisać wydatki

**Stwórz wyjazd**

Rysunek 11: Podstrona do tworzenia wydatku.

**Stwórz Wydatek**

Proszę wpisać wydatek

Kategoria

Wybierz kategorię

Kwota

Proszę wpisać

Währung

PLN

Data

01.01.2024

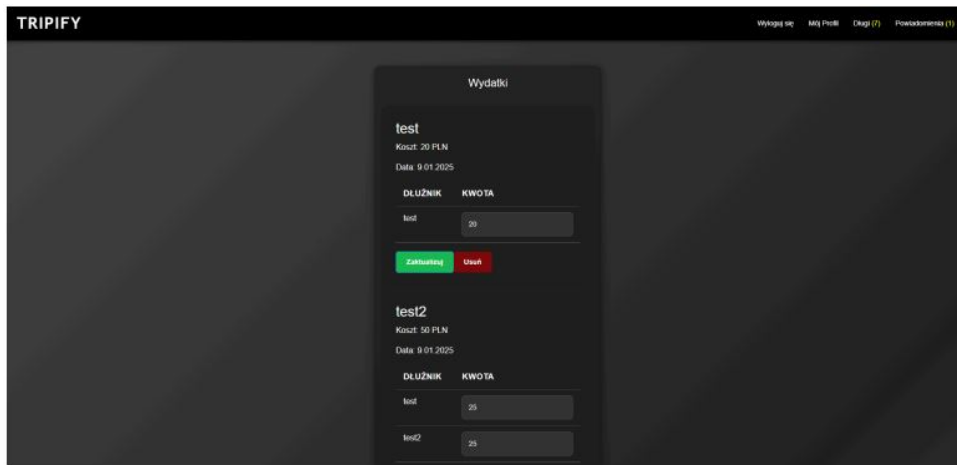
Lokalizacja

Proszę wpisać

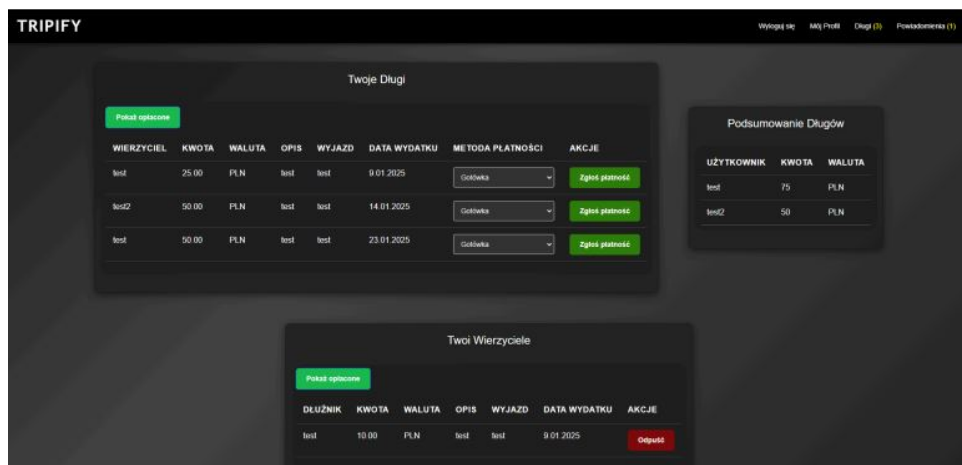
**Zapisz Wydatek** **Anuluj**

© 2024 TRIPIFY. Złoty błęd \*

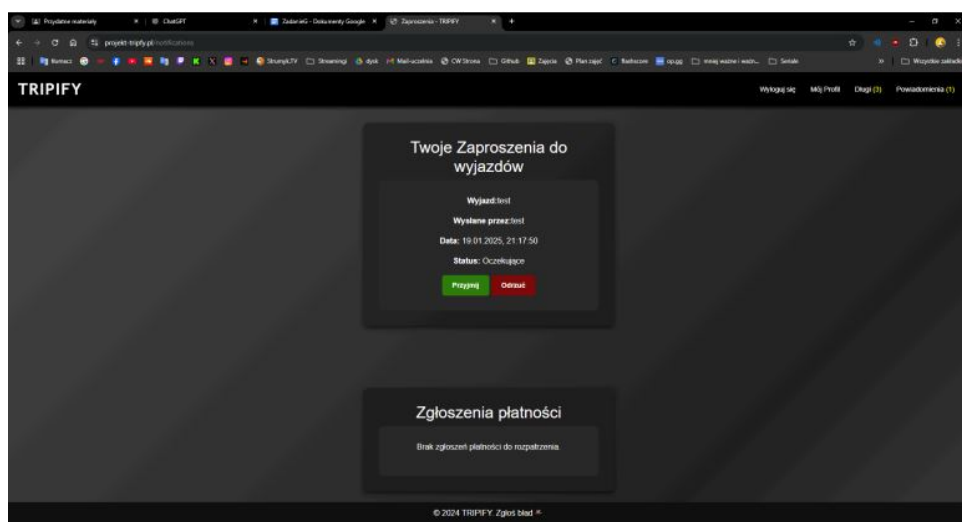
Rysunek 12: Modyfikacja wydatku.



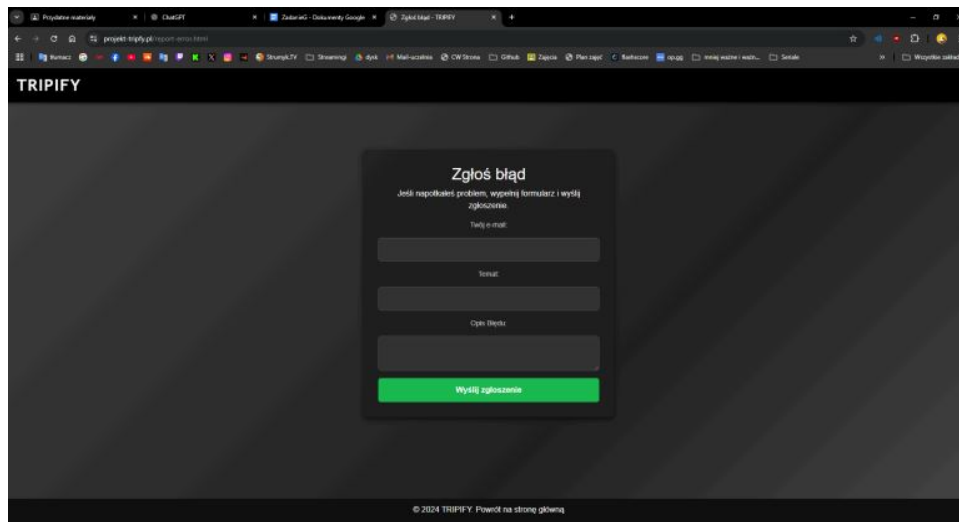
Rysunek 13: Strona długów, gdzie widoczne są moje długi jak i moi wierzyciele.



Rysunek 14: Akceptacja zaproszeń oraz zgłoszeń płatności.



Rysunek 15: Zgłaszanie błędów.



Rysunek 16: Panel administracyjny.

## 11 Informacje uruchomieniowe w środowisku developerskim

### 1. Pobierz repozytorium:

Wejdź na repozytorium GitHub i sklonuj projekt na swoje lokalne środowisko.

### 2. Konfiguracja backendu:

Otwórz projekt backendu w Visual Studio. Następnie przejdź do pliku 'Program.cs' i zmień ustawienia CORS, aby adresy backendu i frontendu były lokalne (zaktualizuj odpowiednio URL do serwera frontendu i backendu).

### 3. Konfiguracja pliku appsettings.json:

Otwórz plik 'appsettings.json' w projekcie backendu. Upewnij się, że wszystkie ustawienia, takie jak połączenia z bazą danych, są poprawnie skonfigurowane.

### 4. Uruchomienie backendu:

Po wprowadzeniu zmian uruchom aplikację. Jeśli konfiguracja przebiegła pomyślnie, aplikacja powinna uruchomić się bez problemu, a przeglądarka automatycznie otworzy interfejs Swagger, gdzie będziesz mógł testować dostępne endpointy API.

### 5. Konfiguracja frontendu:

Otwórz folder frontend w Visual Studio Code. Następnie zaktualizuj wszystkie adresy w funkcjach 'fetch', zmieniając je na lokalne adresy backendu.

### 6. Uruchomienie frontendu:

Uruchom Live Server na pliku 'index.html' w folderze frontendu. Po uruchomieniu, frontend powinien poprawnie połączyć się z lokalnym backendem, umożliwiając korzystanie z aplikacji.

### 7. Dodatkowe kroki:

Upewnij się, że masz zainstalowane wszystkie wymagane zależności zarówno w backendzie, jak i frontendzie.



## 12 Opis wdrożenia aplikacji i sposobu hostowania

Aplikacja została wdrożona na dwóch osobnych serwerach: webio.pl dla backendu oraz seohost.pl dla frontendu. Backend został wdrożony jako aplikacja ASP.NET hostowana na serwerze obsługującym .NET, natomiast frontend został umieszczony na serwerze wspierającym statyczne strony internetowe. W celu monitorowania działania aplikacji zastosowano mechanizmy logowania oraz systemy powiadomień o błędach.

## 13 Podsumowanie, wnioski

### Podsumowanie:

Realizacja projektu dostarczyła kompleksowego rozwiązania, które spełnia wszystkie założone wymagania funkcjonalne i нефункционалне. Aplikacja pozwala na łatwe zarządzanie wyjazdami i podziałem kosztów pomiędzy uczestnikami, oferując intuicyjny interfejs użytkownika oraz solidny backend. Dwuwarstwowa architektura oparta na rozdzieleniu backendu i frontendu umożliwia nie tylko elastyczność rozwoju, ale także łatwe wdrażanie nowych funkcjonalności w przyszłości.

Projekt został wdrożony zgodnie z planem na dwóch różnych serwerach: webio.pl dla backendu oraz seohost.pl dla frontendu, co zapewnia optymalne działanie aplikacji w środowisku produkcyjnym.

### Wnioski każdego z członków zespołu:

- **Dawid Skowroński:** Projekt "Tripify", aplikacja do rozliczania wyjazdów był ciekawym pomysłem na aplikację, zarazem stanowiąc dla nas pewne wyzwanie. Jednak na koniec wszyscy jesteśmy zadowoleni z rezultatu jaki udało nam się uzyskać. Rola którą pełniłem w zespole (Team leader, backend) odpowiadała mi, ponieważ praca z backendem stanowczo bardziej pasuje mi aniżeli zabawa frontendem. Opracowanie backendu z wykorzystaniem simpleapibackend uważam, że wyszło nam z Mateuszem bardzo dobrze, a a koordynacja prac oraz pilnowanie założeń projektowych jako teamleader przebiegała sprawnie. Podsumowując, praca nad naszą aplikacją była doskonałą okazją do nauki współpracy w zespole i zarządzania projektem IT. Uzyskane doświadczenie i nabyte umiejętności na pewno przydadzą się podczas pracy w zawodzie.
- **Mateusz Tęcza:** Praca w projekcie "Tripify" było ciekawym doświadczeniem, zadaniem aplikacji jest rozliczanie wyjazdów, co końcowo dopięło swój cel. Głównym moim zadaniem był backend – to na nim skupiałem się najbardziej, dbając o logikę działania aplikacji i integrację z bazą danych. Samą pracę w backendzie oceniam pozytywnie, bardzo dobrze współpracowało mi się z całą grupą przy projekcie. Zajmowałem się też deploymentem, więc trochę zabawy z wdrażaniem i konfiguracją było, ale finalnie wszystko działa jak potrzeba. To doświadczenie pozwoliło mi jeszcze bardziej rozwijać umiejętności w projektowaniu API, pracy z bazą danych oraz procesie wdrażania aplikacji, a także nauczyło efektywnej współpracy w zespole.
- **Jakub Kozubek:** "Tripify" jest ciekawym projektem aplikacji, która ułatwia życie na wyjazdach. Na pewno samo rozpoczęcie i odnalezienie się w oddzielnym backendem i frontendem było ciężkie z początku potem nieźle się w tym odnaleźliśmy. Na

początku miałem być testerem ale wyszło, że robiłem po trochu w każdym obszarze. Razem z Kuba zaprojektowaliśmy frontend oraz w większości zajmowałem się rejestracją, logowaniem oraz mailingiem. Samo testowanie przyszło mi z trudnością średnio podobała mi się przydzielona rola w projekcie. Nie wiem czy testy, które wykonałem są wystarczające. Sam deployment na samym początku też sprawił nam bardzo dużo kłopotów ponieważ robiliśmy to pierwszy raz, ale koniec końców strona została wdrożona. Podsumowując jak aplikacja spełnia wszystkie swoje założenia z czego jestem zadowolony, interfejs użytkownika wygląda ładnie ale przydałby się osobny frontend dla urządzeń mobilnych, a pracując jako tester poraz pierwszy czułem zakłopotanie ale koniec końców chyba wszystko się udało.

- **Jakub Trznadel:** Praca nad projektem była udana, choć mogła być bardziej efektywna. To moje pierwsze doświadczenie z odseparowanym backendem i frontendem, co wymagało trochę czasu na adaptację. Jako osoba odpowiedzialna za frontend, musiałem zmierzyć się z wieloma wyzwaniami, zwłaszcza w zakresie integracji z backendem. Początkowo miałem trudności w dostosowaniu komunikacji między frontem a backendem za pomocą REST API. Ponadto, integracja z zewnętrznymi bibliotekami i stylowanie aplikacji w sposób responsywny stanowiło kolejne wyzwanie, które wymagało sporego nakładu pracy, szczególnie na urządzeniach mobilnych. Mimo to, stopniowo przezwyciężałem te trudności, ucząc się nowych narzędzi i technik, a finalnie udało się stworzyć spójną i funkcjonalną aplikację, która spełniała wszystkie założenia projektu.