WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH

Projektowanie aplikacji internetowych (PAINT)

Meal Planner with AI

Raport końcowy projektu

Autorzy:
Jan Kaliwoda,
Kacper Rogowski,
Szymon Pietraszewski,
Maciej Paliszewski

8 czerwca 2025

Spis treści

1 Temat projektu

Celem projektu **Meal Planner with AI** jest stworzenie nowoczesnej aplikacji webowej wspierającej użytkownika w planowaniu posiłków, zarządzaniu składnikami oraz **tworzeniu przepisów z wykorzystaniem sztucznej inteligencji**. Wykorzystuje nowoczesny stos technologiczny (frontend, backend, AI, DevOps), by zapewnić efektywne i intuicyjne doświadczenie użytkownika.

2 Zespół projektowy i role

- **Kacper Rogowski** - lider projektu, DevOps / Fullstack Developer
- **Szymon Pietraszewski** - frontend developer (React, HTML, CSS, interfejs użytkownika)
- **Jan Kaliwoda** - backend developer (Django, REST API, zarządzanie danymi i autoryzacją)
- **Maciej Paliszewski** - AI developer (stworzenie własnego modelu AI)

3 Szczegółowe założenia funkcjonalne aplikacji

- Rejestracja i logowanie użytkownika (w tym przez Google)
- Definiowanie preferencji dietetycznych i alergii
- Planowanie posiłków z użyciem kalendarza
- Zapisywanie posiadanych przez użytkownika składników
- Ręczne wprowadzanie składników do wyszukiwania
- Generowanie przepisów z użyciem sztucznej inteligencji
- Planowanie posiłków na podstawie przepisów i rekomendacji
- Generowanie listy zakupów na podstawie zaplanowanych posiłków

4 Szczegółowe założenia architektoniczne aplikacji

Architektura aplikacji została zaprojektowana w oparciu o **model trójwarstwowy (three-tier architecture)**, co zapewnia modularność, przejrzystość oraz łatwość skalowania. Każda warstwa systemu posiada jasno określoną odpowiedzialność:

- **Warstwa prezentacji (frontend)** – oparta o React oraz TypeScript i JavaScript. Odpowiada za interakcję z użytkownikiem i komunikację z backendem poprzez REST API.
- **Warstwa logiki biznesowej (backend)** – zaimplementowana w Django z użyciem Django REST Framework. Przetwarza dane przychodzące z frontend'u, zarządza sesjami, autoryzacją, regułami biznesowymi oraz integracją z modułem AI.

- **Warstwa danych (baza danych)** – wykorzystuje PostgreSQL do trwałego przechowywania informacji: kont użytkowników, składników, przepisów, planów posiłków, preferencji i historii aktywności.
- **Moduł AI** – stanowi osobny komponent odpowiedzialny za analizę składników i generowanie przepisów. Wykorzystuje modele językowe oraz uczenie maszynowe do interpretacji danych wejściowych i tworzenia trafnych rekomendacji.
- **Środowisko uruchomieniowe** – cała aplikacja wdrażana jest za pomocą Docker i Docker Compose, co umożliwia łatwe testowanie i replikację środowiska na różnych maszynach. Każda warstwa jest osobnym kontenerem.

Zaproponowana architektura umożliwia niezależne rozwijanie poszczególnych komponentów, co jest istotne dla dalszej rozbudowy aplikacji (np. tryb offline, udostępnianie przepisów, personalizacja modeli AI).

5 Wykorzystanie narzędzi programowych w kontekście założeń funkcjonalnych i architektonicznych

W poniższej tabeli przedstawiono zestawienie wykorzystanych technologii oraz ich powiązanie z funkcjonalnościami i warstwami aplikacji:

Technologia / narzędzie	**Warstwa / moduł**	**Funkcjonalność**
React + TypeScript	Frontend	Formularze, kalendarz, dodawanie składników, wyświetlanie przepisów, UX/UI
Tailwind CSS	Frontend	Stylowanie interfejsu, responsywność aplikacji
Django + Django REST Framework	Backend	Autoryzacja, przetwarzanie danych, logika biznesowa, API
PostgreSQL	Baza danych	Przechowywanie danych użytkowników, składników, przepisów, planów
TensorFlow, scikit-learn pandas, NumPy NER, GloVe embeddings	Moduł AI	Generowanie przepisów z użyciem składników wejściowych Przetwarzanie danych wejściowych i wyjściowych dla modelu Przetwarzanie języka naturalnego, wykrywanie nazw składników
Docker	Cały system	Izolacja komponentów aplikacji, konteneryzacja środowiska
Docker Compose	DevOps	Orkiestracja serwisów: frontend, backend, baza danych, AI
OAuth2 (Google)	Backend / Frontend	Logowanie użytkowników przez konto Google
Vite	Frontend	Budowanie aplikacji, szybki hot reload

6 Wykorzystanie technologii

Aplikacja korzysta z trójwarstwowej architektury:

- **Frontend:** React, HTML, CSS, JavaScript, TypeScript, Tailwind CSS
- **Backend:** Django, Django REST Framework
- **Baza danych:** PostgreSQL
- **AI:** TensorFlow, scikit-learn, pandas, NumPy, NER, GloVe embeddings
- **Wdrożenie:** Docker, Docker Compose

7 Opis realizacji

Pracę nad projektem *Meal Planner with AI* rozpoczęliśmy od dokładnej analizy potrzeb użytkownika oraz określenia zakresu funkcjonalnego aplikacji. Naszym celem było stworzenie systemu, który nie tylko wspiera planowanie posiłków, ale także wykorzystuje sztuczną inteligencję do generowania przepisów na podstawie dostępnych składników.

Etap planowania

Na początku podzieliliśmy się rolami zgodnie z kompetencjami każdego członka zespołu: frontend, backend, AI oraz DevOps. Wspólnie ustaliliśmy listę funkcjonalności oraz architekturę systemu, zakładając model trójwarstwowy i konteneryzację z użyciem Dockera. Już na tym etapie zaplanowaliśmy integrację modułu AI jako osobnego komponentu.

Tworzenie komponentów frontend i backend

Zaczęliśmy od implementacji podstawowych funkcji logowania i rejestracji, w tym także przez Google. Frontend oparty został na React i Tailwind CSS, a backend na Django REST Framework. Na tym etapie napotkaliśmy problemy m.in. z komunikacją między frontendem a backendem (błąd CORS), które rozwiązaliśmy przez odpowiednią konfigurację serwera i środowisk `.env`.

Praca nad modułem AI

Moduł AI był jednym z najbardziej wymagających elementów. Początkowo planowaliśmy wykorzystać gotowe zestawy przepisów, jednak okazały się one niespójne i niekompletne. W związku z tym opracowaliśmy własny proces przetwarzania danych z użyciem bibliotek `pandas` i `NumPy`. Do rozpoznawania składników w treści przepisów wykorzystaliśmy Named Entity Recognition (NER) oraz embeddingi GloVe.

Integracja planowania posiłków i kalendarza

Aplikacja łączy planowanie posiłków z widokiem kalendarza tygodniowego. Posiłki użytkownika (śniadanie, lunch, obiad) są pobierane z API i wyświetlane w odpowiednich dniach i porach tygodnia. Użytkownik może przeglądać swoje zaplanowane posiłki, przełączać tygodnie, wybrać konkretny dzień, a także kliknąć posiłek, aby zobaczyć szczegóły przepisu – składniki i kroki przygotowania. Dzięki temu interfejs jest czytelny i funkcjonalny, wspierając codzienne planowanie żywienia.

8 Moduły systemu

- **Moduł użytkownika:** rejestracja, logowanie
- **Moduł składników:** zarządzanie składnikami w lodówce
- **Moduł przepisów:** generowanie i zapisywanie przepisów z wykorzystaniem AI
- **Moduł planowania posiłków:** przypisywanie przepisów do kalendarza
- **Moduł listy zakupów:** automatyczne generowanie listy na podstawie przepisów

Dalsze wyzwania dla projektu Meal Planner with AI:

1. **Wykrywanie podobnych przepisów**
   Implementacja algorytmu, który sugeruje użytkownikowi podobne przepisy na podstawie wybranego dania (np. przez embeddingi składników lub tytułów).
2. **Współdzielenie listy zakupów i jadłospisu**
   Możliwość udostępniania listy zakupów lub planu posiłków innym użytkownikom (np. rodzinie, współlokatorom).
3. **Analiza wartości odżywczych**
   Automatyczne wyliczanie kalorii, białka, tłuszczu i węglowodanów na podstawie składników przepisu.
4. **Tryb offline**
   Umożliwienie korzystania z aplikacji bez dostępu do internetu (np. przez cache'owanie przepisów i składników).
5. **Zaawansowane filtrowanie przepisów**
   Filtrowanie po czasie przygotowania, liczbie składników, poziomie trudności, kuchni świata itp.
6. **Własne notatki i modyfikacje przepisów**
   Możliwość dodawania własnych notatek do przepisów oraz zapisywania ich zmodyfikowanych wersji.

9 Napotkane problemy i niezrealizowane założenia

- **Generowanie listy zakupów** na podstawie wybranych posiłków.
- **Zbiór danych**, na którym trenowano model, charakteryzował się niską jakością ze względu na brak spójności, niekompletność.
- Nie stworzono generowania przepisów na podstawie preferencji użytkownika.
- Problemy z nawigacją w sekcji storage w działaniu navbaru.

10 Instrukcja instalacji

10.1 Wymagania

- Docker

10.2 Kroki

1. Należy pobrać i rozpakować archiwum repozytorium:
   https://github.com/JanKaliwoda/Meal-Planner-with-AI.git
2. Należy w repozytorium stworzyć plik o nazwie `.env` i wypełnić go następującym kodem:

   Listing 1: Plik .env

   ```
   DJANGO_SECRET_KEY='django-insecure-73&d0jzj561v1kq7o+t(6j!
       dz_aje843f+i#blsct@$&qv^9-d'
   ```

3. Należy w folderze `/frontend/` stworzyć plik o nazwie `.env` i wypełnić go następującym kodem:

   Listing 2: Plik .env w folderze /frontend/

   ```
   VITE_API_URL="http://localhost:8000/"
   VITE_GOOGLE_CLIENT_ID=693228826757-
       dmcp2cnc9hdccg8dkhkn7k2fsmu9d6cp.apps.googleusercontent.com
   ```

4. W folderze repozytorium należy otworzyć terminal i wykonać komendę:
   docker compose up –watch –build
   Uwaga: ze względu sporą ilość bibliotek ten proces może potrwać do 15 minut w zależności od sprzętu
5. Po pomyślnym włączeniu się kontenerów należy otworzyć drugi terminal w folderze repozytorium i zaimportować bazy danych komendami:
   docker compose run –rm app sh -c "python manage.py import_ingredients_alldata"
   docker compose run –rm app sh -c "python manage.py import_recipes"
   Uwaga - te procesy mogą zająć do 20 minut w zależności od sprzętu
6. Aplikacja jest gotowa do użytku

11 Instrukcja użytkownika

1. **Zarejestruj się lub zaloguj (również przez Google i email)**
2. **Uzupełnij profil** (preferencje dietetyczne, alergie)
3. **Dodaj składniki**
4. **Wygeneruj przepis** na podstawie składników
5. **Zaplanuj posiłki** w kalendarzu
6. **Aktualizuj lodówkę** po zakupach

12 Podsumowanie

Meal Planner with AI to nowoczesna aplikacja webowa wspierająca użytkownika w codziennym planowaniu posiłków, zarządzaniu składnikami oraz tworzeniu list zakupów. Projekt został zrealizowany z wykorzystaniem trójwarstwowej architektury, nowoczesnych frameworków frontendowych (React, Tailwind CSS), backendowych (Django REST Framework) oraz sztucznej inteligencji (TensorFlow, scikit-learn, GloVe embeddings, Named Entity Recognition).

Meal Planner with AI stanowi solidną bazę do dalszego rozwoju – zarówno pod kątem funkcjonalności, jak i skalowalności aplikacji. Projekt pokazuje praktyczne zastosowanie sztucznej inteligencji w życiu codziennym i stanowi przykład kompleksowej aplikacji webowej opartej na nowoczesnym stosie technologicznym.