

KittchenBook

APLIKACJA DO ZARZĄDZANIA PRZEPISAMI

PRZEMYSŁAW DRYGAS, JAKUB GÓRNAŚ, MACIEJ IDZIKOWSKI

1. Opis aplikacji

Kitchen Book to aplikacja webowa służąca do zarządzania domowymi przepisami kulinarnymi. Umożliwia użytkownikom dodawanie, edytowanie, organizowanie oraz wyszukiwanie przepisów według kategorii i składników. Dodatkowo pozwala tworzyć plan posiłków i generować listy zakupów na podstawie wybranych receptur. Aplikacja została zaprojektowana z myślą o użytkownikach indywidualnych oraz rodzinach, które chcą w prosty sposób organizować swoje gotowanie i zakupy.

2. Technologia

2.1. Stack technologiczny

- Frontend: HTML5, CSS3, czysty JavaScript
- Backend: Node.js + Express.js
- Baza danych: Microsoft SQL Server, TypeORM
- Języki: JavaScript, TypeScript
- Inne narzędzia: GitHub (kontrola wersji), LocalStorage (przechowywanie danych w przeglądarce)

3. Struktura projektu

3.1. Frontend

Struktura katalogów:

- /frontend
 - /assets – zasoby graficzne (np. logo)
 - /css – arkusze stylów (reset.css, styles.css)
 - /js
 - /api – komunikacja z backendem (np. recipesAPI.js)
 - /ui – komponenty UI odpowiedzialne za interakcje i rendering (layout.js, recipesUI.js)
 - /views – logika widoków odpowiadająca za poszczególne strony (recipesView.js, mealplannerView.js)
 - main.js – punkt wejściowy aplikacji
 - /pages – dokumenty HTML (recipes.html, mealplanner.html)
 - index.html, favicon.ico, package.json

3.2. Backend

Struktura backendu oparta na Express.js:

- /routes – definicje tras REST API
- /services – logika biznesowa
- /entities – modele danych (TypeORM)
- /controllers – logika obsługi zapytań
- app.ts – punkt startowy aplikacji Express
- ormconfig.js – konfiguracja połączenia z bazą danych

4. Funkcjonalności aplikacji

1. **Dodawanie nowego przepisu** – umożliwia użytkownikowi dodanie nowego przepisu do swojej kolekcji.
2. **Edycja przepisu** – pozwala edytować istniejące przepisy, modyfikując ich składniki i instrukcje.
3. **Organizacja przepisów według kategorii** – przypisywanie przepisów do kategorii takich jak śniadanie, obiad, deser.
4. **Wyszukiwanie po składnikach** – przeszukiwanie przepisów na podstawie posiadanych składników.
5. **Edycja planu posiłków** – użytkownik może zaplanować posiłki na wybrane dni tygodnia.
6. **Lista zakupów** – na podstawie przepisów lub planu posiłków tworzona jest lista zakupów.

5. Grupy docelowe

5.1. Osoby gotujące w domu

Osoby indywidualne, które chcą uporządkować i zorganizować swoje przepisy kulinarne w jednym miejscu.

5.2. Członkowie gospodarstwa domowego

Rodziny i współlokatorzy, którzy wspólnie planują posiłki i zakupy spożywcze.

6. Korzyści dla użytkownika

- **Łatwe przechowywanie i organizacja przepisów** – użytkownik ma dostęp do wszystkich przepisów w jednym miejscu.
- **Efektywne planowanie posiłków** – dynamiczny interfejs umożliwia szybkie uzupełnianie planu.
- **Optymalizacja zakupów** – lista zakupów tworzona automatycznie na podstawie przepisów ułatwia codzienne zakupy.

7. Backend – opis API

W backendzie zaimplementowano następujące elementy:

- Uruchomienie serwera Express.js z podstawowymi zabezpieczeniami (np. middleware)
- Endpoint do dodawania przepisów: POST /api/recipes
- Endpoint do aktualizacji przepisów: PUT /api/recipes/:id
- Endpoint do pobierania przepisów: GET /api/recipes/:id
- Endpoint do usuwania przepisów: DELETE /api/recipes/:id
- Obsługa połączenia z relacyjną bazą danych SQL Server z wykorzystaniem TypeORM

8. Frontend – funkcjonalności

W aplikacji frontendowej zaimplementowano:

- Interfejs użytkownika dla wszystkich funkcjonalności aplikacji
- Komunikację z backendem poprzez API
- Użycie LocalStorage do przechowywania stanu użytkownika (np. lista zakupów)
- Responsywność ekranów (dla urządzeń mobilnych i desktopów)

9. Dalsze etapy rozwoju

- Dodanie systemu routingu pomiędzy podstronami (np. hash routing lub prosty SPA)
- Implementacja testów jednostkowych i integracyjnych
- Refaktoryzacja kodu frontendowego z rozdzieleniem na moduły

- Przygotowanie prezentacji projektu (np. demo lub screencast)
- Dodanie interfejsu do zarządzania listą zakupów

10. Uruchamianie aplikacji

Backend (Express.js)

1. Przejdź do katalogu /backend
2. Zainstaluj zależności: npm install
3. Uruchom serwer: npm start

Frontend

1. Frontend jest serwowany przez backendowy serwer, wystarczy uruchomić przeglądarkę pod odpowiednim adresem. Np. <http://localhost:3000/index.html>