Obraz zawierający tekst, Czcionka, biały, czarne i białe

Opis wygenerowany automatycznie

Dokumentacja wstępna

Bazy danych 2

Projekt

*„Książka Kucharska Online”*

Mateusz Kołacz  
Mikołaj Bańkowski

Karol Bogumił

Jakub Poćwiardowski

prowadzący

Wojciech Sitek

Warszawa 2024

Spis treści

[1. Skład zespołu: 3](#_Toc164525980)

[2. Opis projektu: 3](#_Toc164525981)

[3. Realizacja projektu: 3](#_Toc164525982)

[4. Technologie programistyczne: 7](#_Toc164525983)

[5. Wymagania funkcjonalne: 7](#_Toc164525984)

# 1. Skład zespołu:

1. Jakub Poćwiardowski
2. Mikołaj Bańkowski
3. Karol Bogumił
4. Mateusz Kołacz

# 2. Opis projektu:

Celem projektu jest stworzenie bloga kulinarnego, który będzie udostępniał swoim użytkownikom możliwość:

* założenia konta,
* zamieszczania postów z przepisami kulinarnymi,
* komentowania postów innych użytkowników,
* dodawania postów do ulubionych

Każdy z użytkowników będzie miał również możliwość przeglądanie ich na swoim profilu.

# 3. Realizacja projektu:

Projekt zostanie zrealizowany zapewne w architekturze trójwarstwowej. Aplikacja zostanie podzielona na następujące warstwy:

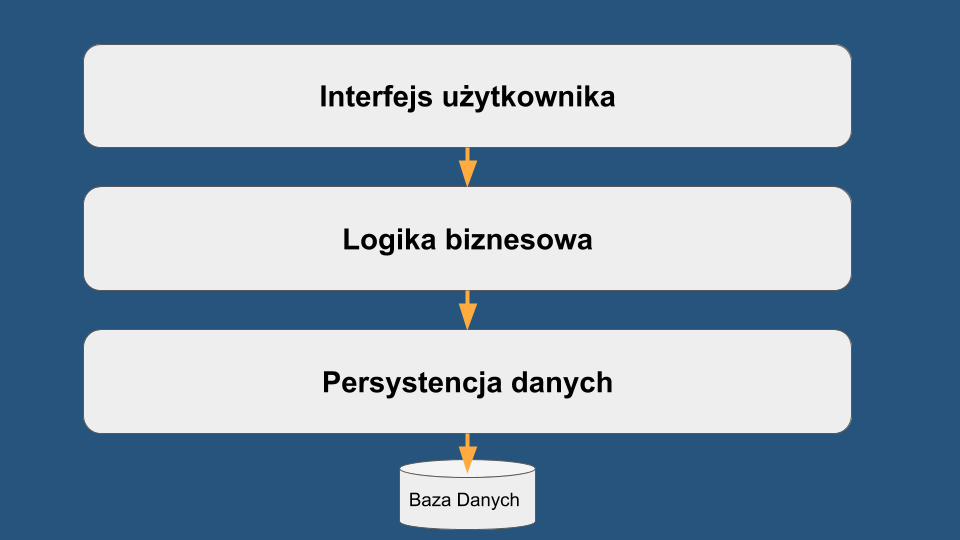
* Warstwę prezentacji danych (Frontend)
* Warstwę biznesową( Backend)
* Warstwę bazodanową

Frontend będzie składać się z trzech widoków:

* Ekran logowania
* Strona bloga
* Profil użytkownika

Każdy widok będzie w osobnym module. Do tego niezbędne będą również serwisy:

* Serwis strzelający do odpowiednich endpointów po dane
* Serwis handlujący logowanie



Podstawowymi modułami backendu będą Encje, Kontrolery oraz Repozytoria dla każdej z tabel. Konieczne będzie również stworzenie osobnego modułu dla Security oraz logiki logowania się do aplikacji.

Baza danych reprezentowana będzie przez następujące tabele:

* Media,
* User,
* Post,
* Favourite Post
* Comment,
* Preparation Step
* Ingredient

**Model E-R**

Obraz zawierający tekst, diagram, zrzut ekranu, linia

Opis wygenerowany automatycznie

**Model Logiczny**

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

# 4. Technologie programistyczne:

Baza danych: MySQL

Backend: Java 17, Spring Boot (Hibernate, JPA, REST Api), Security – JWT, Logging - SLF4J

Frontend: Angular (Typescript, Html, CSS & SASS)

# 5. Wymagania funkcjonalne:

* Wstępnie planowana jest możliwość
  + Przeglądania wpisów na blogu
  + Udostępniania przepisów, postów
  + Dodawania postów do ulubionych
  + Przeglądania ulubionych postów
  + Zalogowania się jako czytelnik
  + Komentowania wpisów jako czytelnik
  + Przeglądania profilu użytkownika
  + Edycji profilu użytkownika

# 6. Podsumowanie

Rozdział ma na celu podsumować prace projektowe i zaprezentować efekty naszej pracy.

## 6.1. Zrealizowane funkcjonalności

Funkcjonalności które udało się zrealizować:

* + **Przeglądania wpisów na blogu**
  + **Udostępniania przepisów, postów**
  + **Dodawania postów do ulubionych**
  + **Dodawania nowych postów**
  + **Przeglądania ulubionych postów**
  + **Zalogowania/rejestracji się jako czytelnik**
  + **Komentowania wpisów jako czytelnik**
  + **Przeglądania profilu użytkownika**
  + **Edycji profilu użytkownika**

## 6.2 Warstwa wizualna

Warstwa wizualna została stworzona za pomocą Angulara w oparciu o architekturę MVVM. Jest to aplikacja internetowa typu SPA(Single-Page-Aplication), składająca się z trzech widoków: ekran logowania/rejestracji, strona bloga, profil użytkownika



*Architektura MVVM*

**Model** reprezentuje dane i logikę biznesową aplikacji. W Angularze modele często są zwykłymi TypeScript-owymi klasami lub interfejsami.

**Widok** (View) Odpowiada za prezentację danych i interakcję użytkownika. W Angularze widoki są reprezentowane przez komponenty.

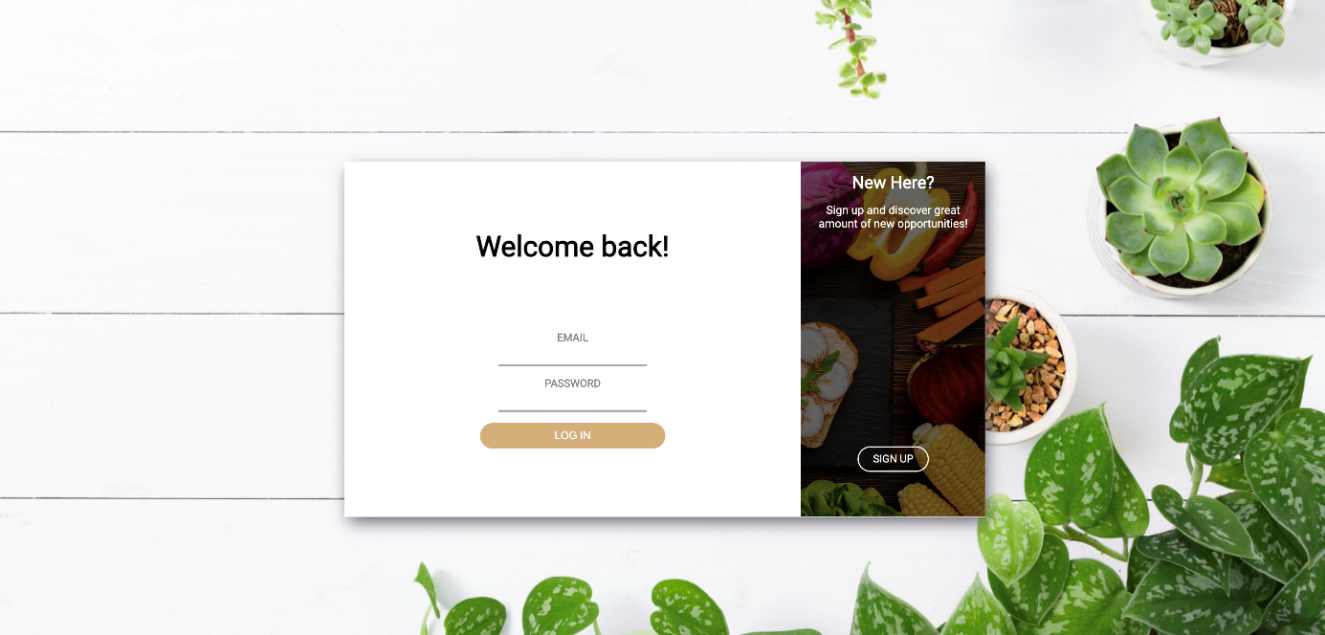
W Angularze obiekt **ViewModel** nie istnieje jako oddzielna instancja. Zamiast tego, logika i dane obiektu ViewModel są często zarządzane przez komponenty Angulara.

**Dwukierunkowe wiązanie danych** (Two-Way Data Binding) to jedna z kluczowych cech Angu-lara, która ułatwia synchronizację danych pomiędzy modelem a widokiem oraz między widokiem a modelem. Zmiany w modelu automatycznie odzwierciedlają się w widoku, a zmiany w widoku są odzwierciedlane w modelu.

***Widok – ekran logowania/rejestracji***

Funkcjonalności:

* + Zalogowanie się/ zarejestrowanie się jako czytelnik

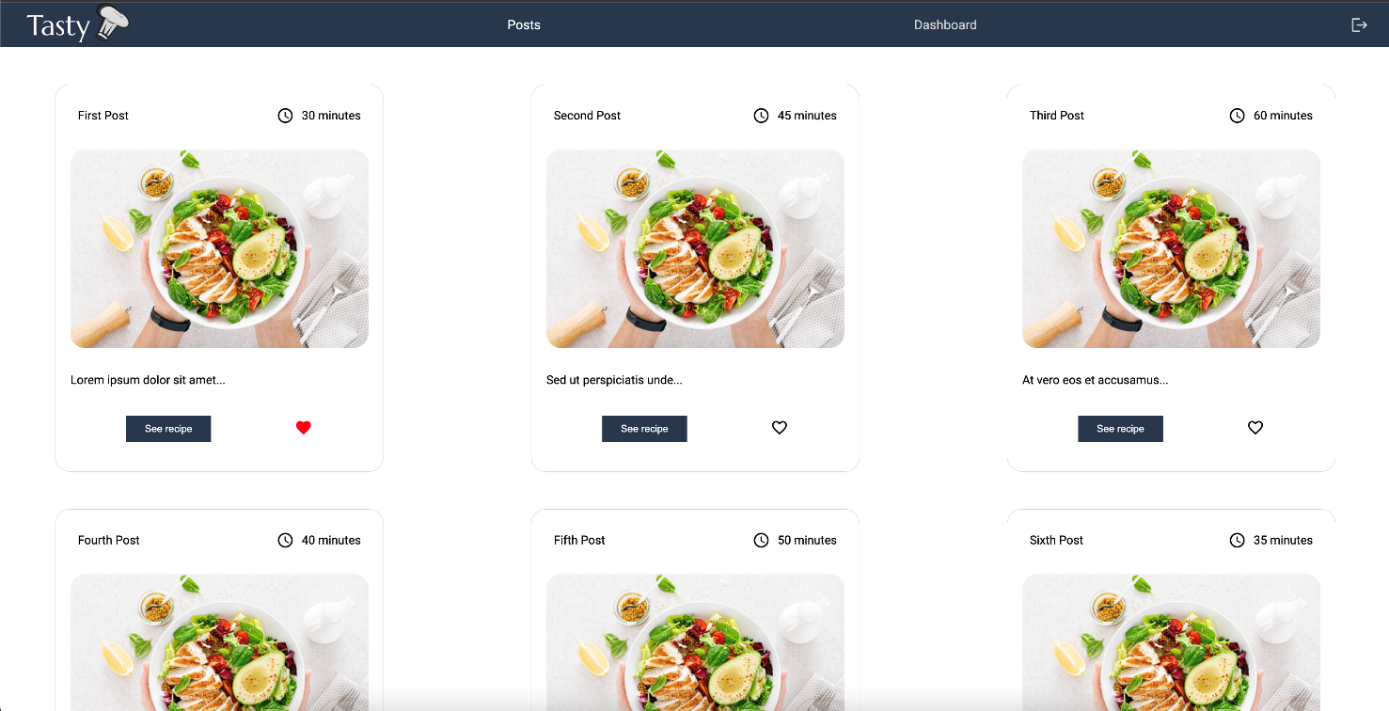
****

*ekran logowania*

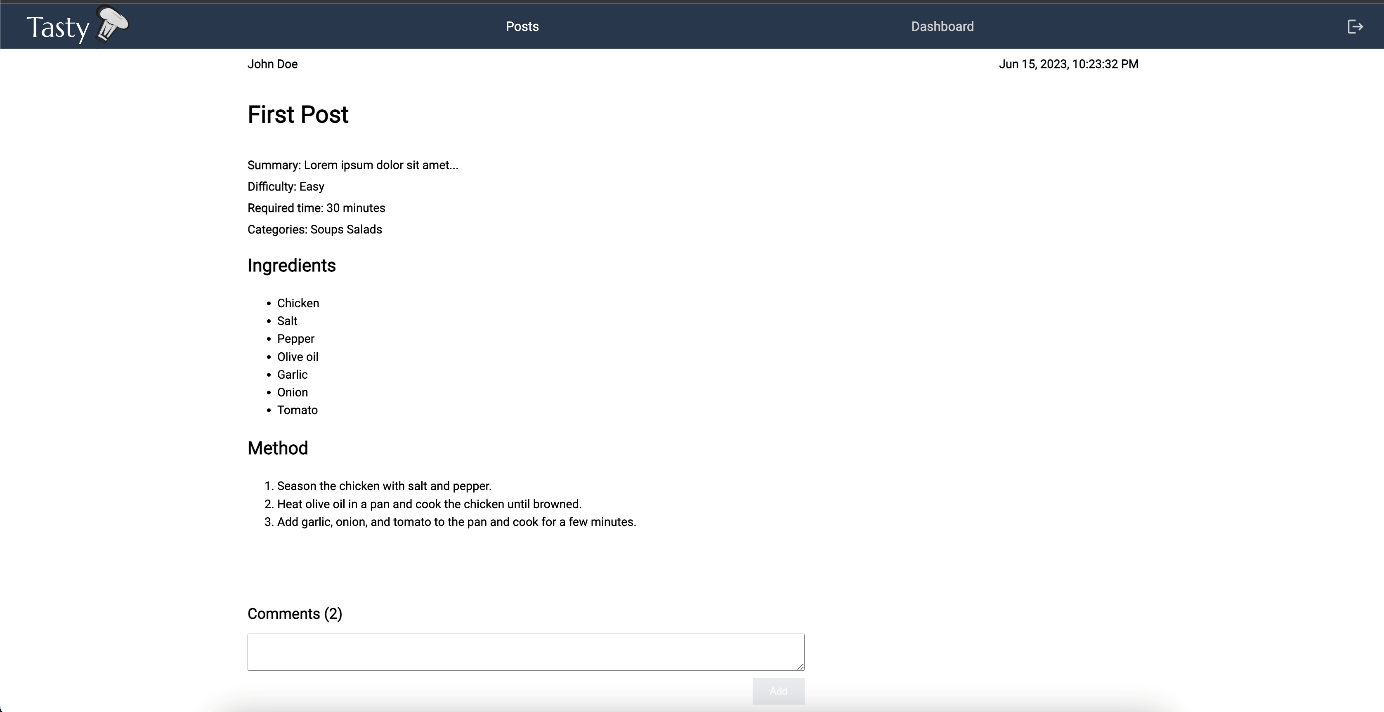
**Widok – *strona bloga***

Funkcjonalności:

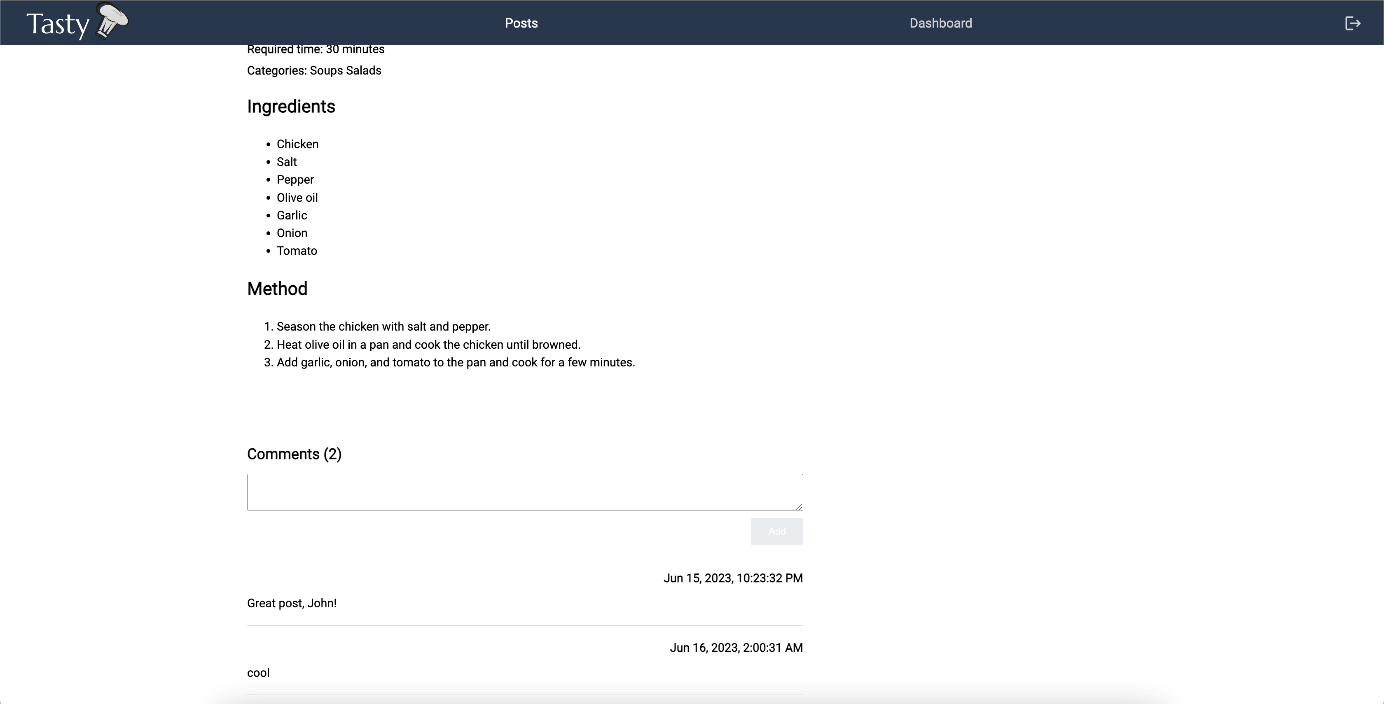
* + Przeglądania wpisów na blogu
  + Dodawania postów do ulubionych
  + Komentowania wpisów jako czytelnik

**

*Strona główna bloga*

**

*po wejściu w przepis*

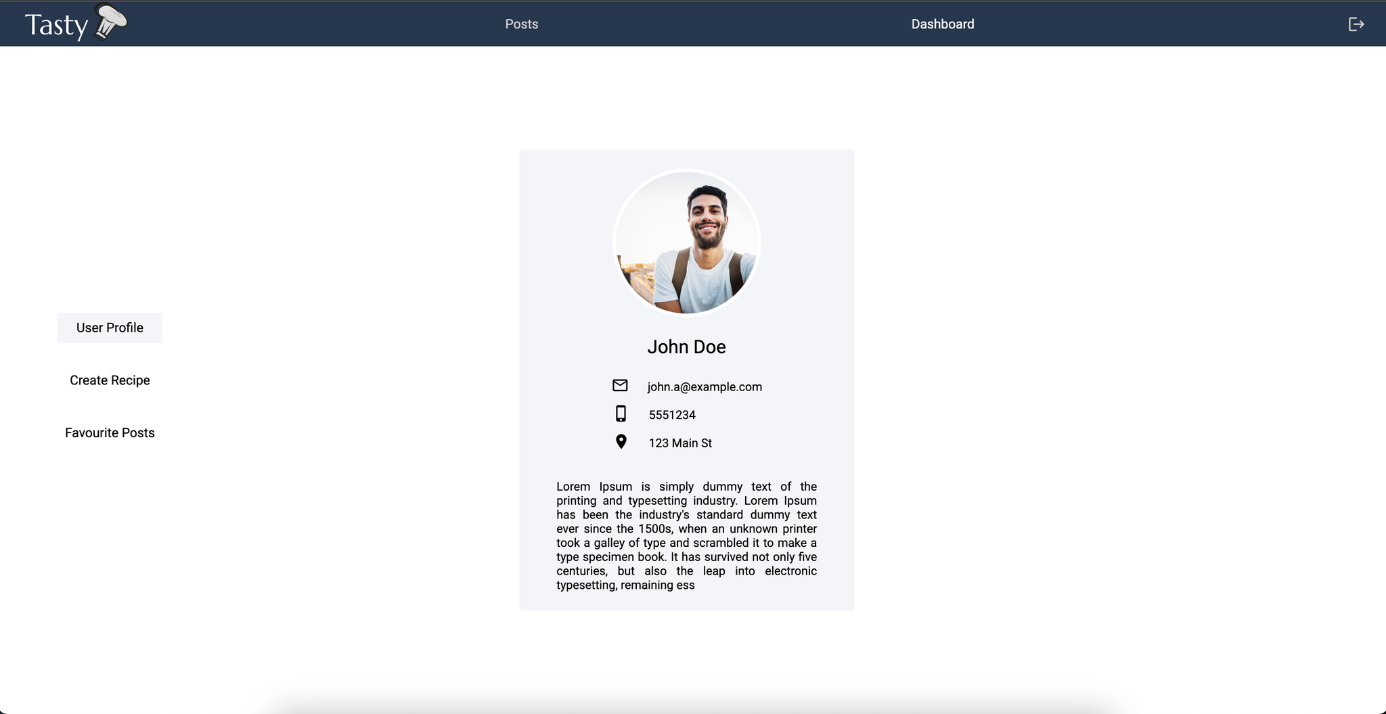
**

*po wejściu w przepis cd.*

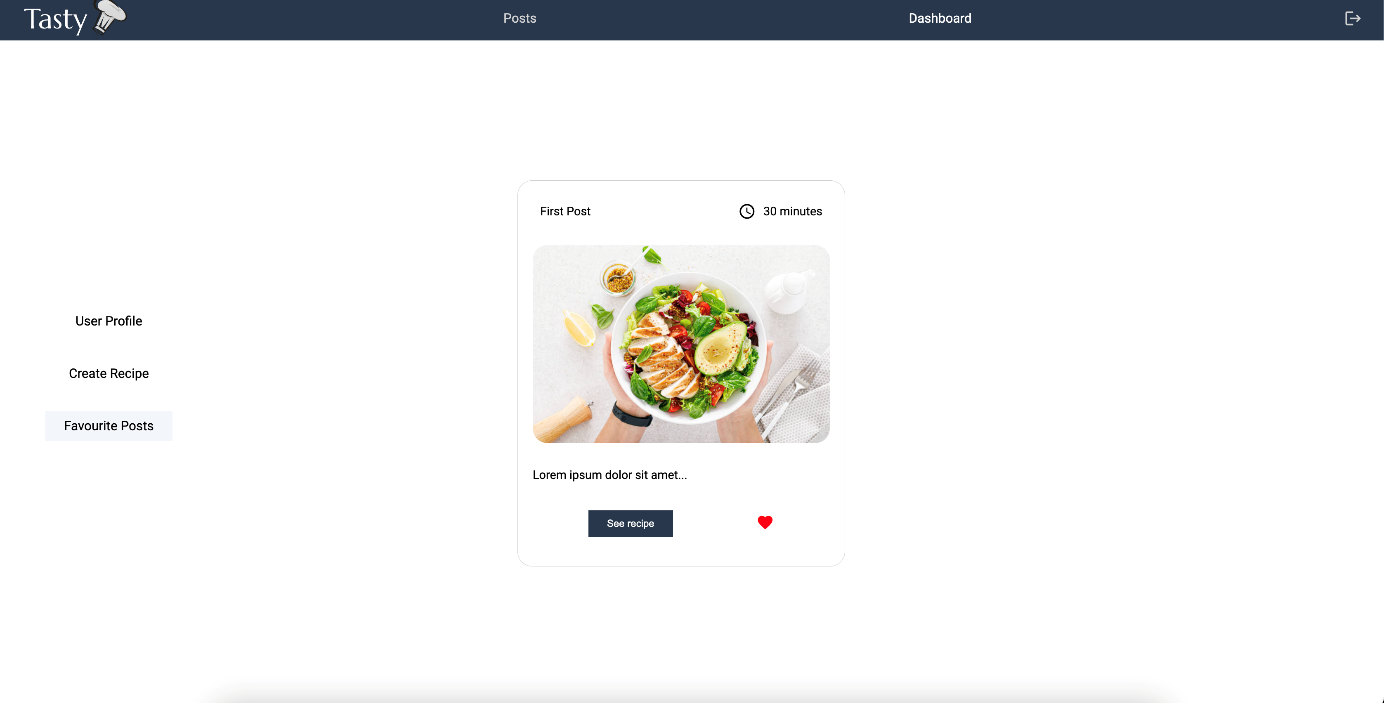
**Widok – *profil użytkownika***

Funkcjonalności:

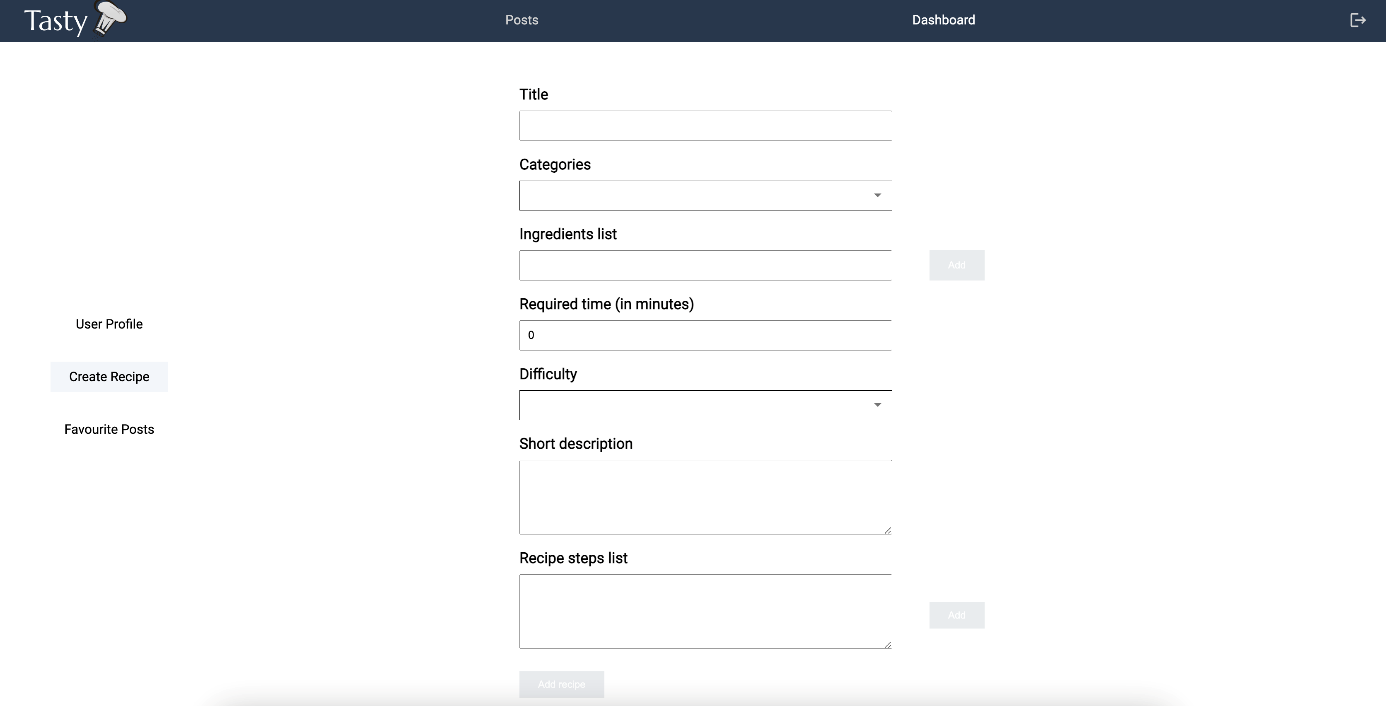
* + Przeglądania profilu użytkownika
  + Przeglądania ulubionych postów
  + Dodawanie nowych postów



*Profil użytkownika*



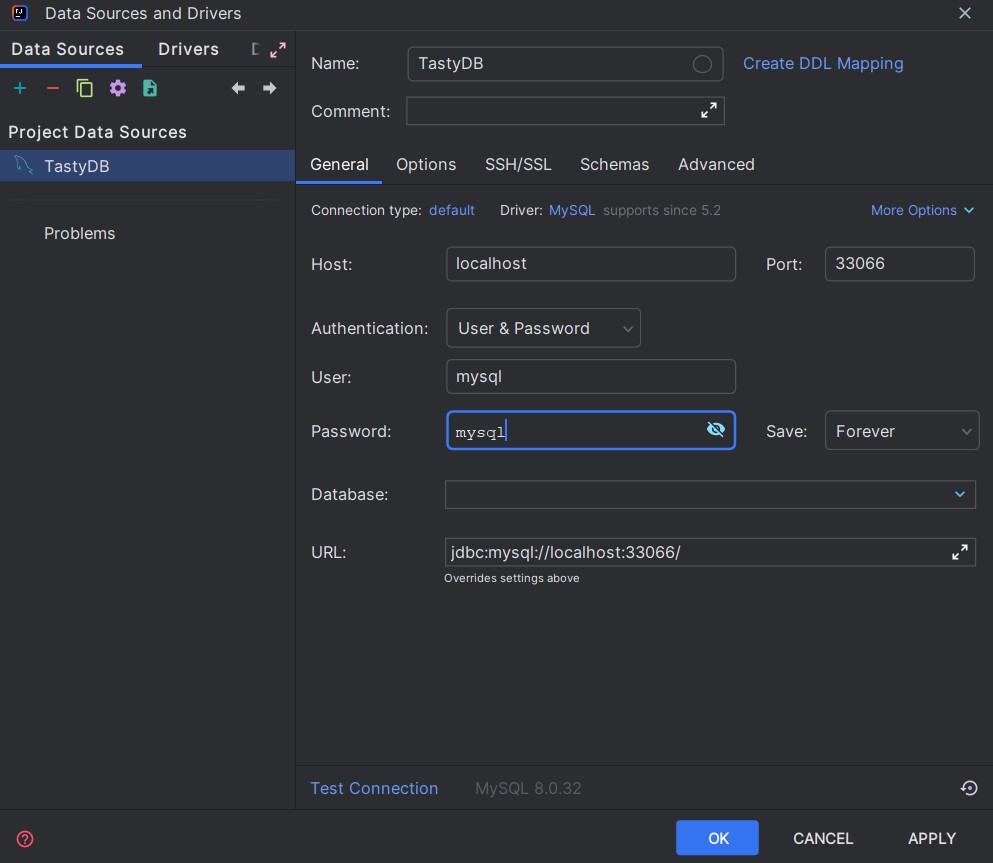
*przeglądanie ulubionych*

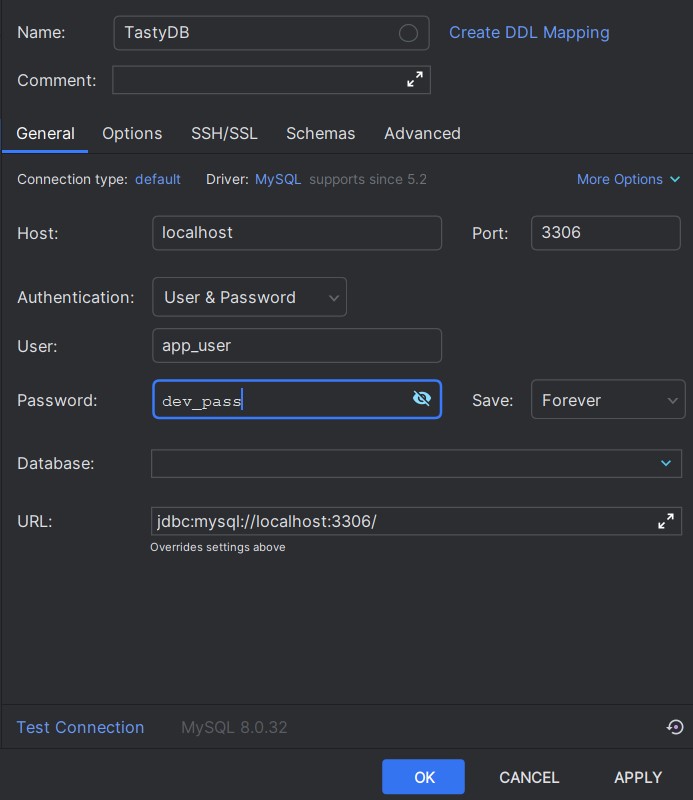


*tworzenia nowego postu*

## 6.2 Baza danych

Konfiguracja bazy danych MySQL

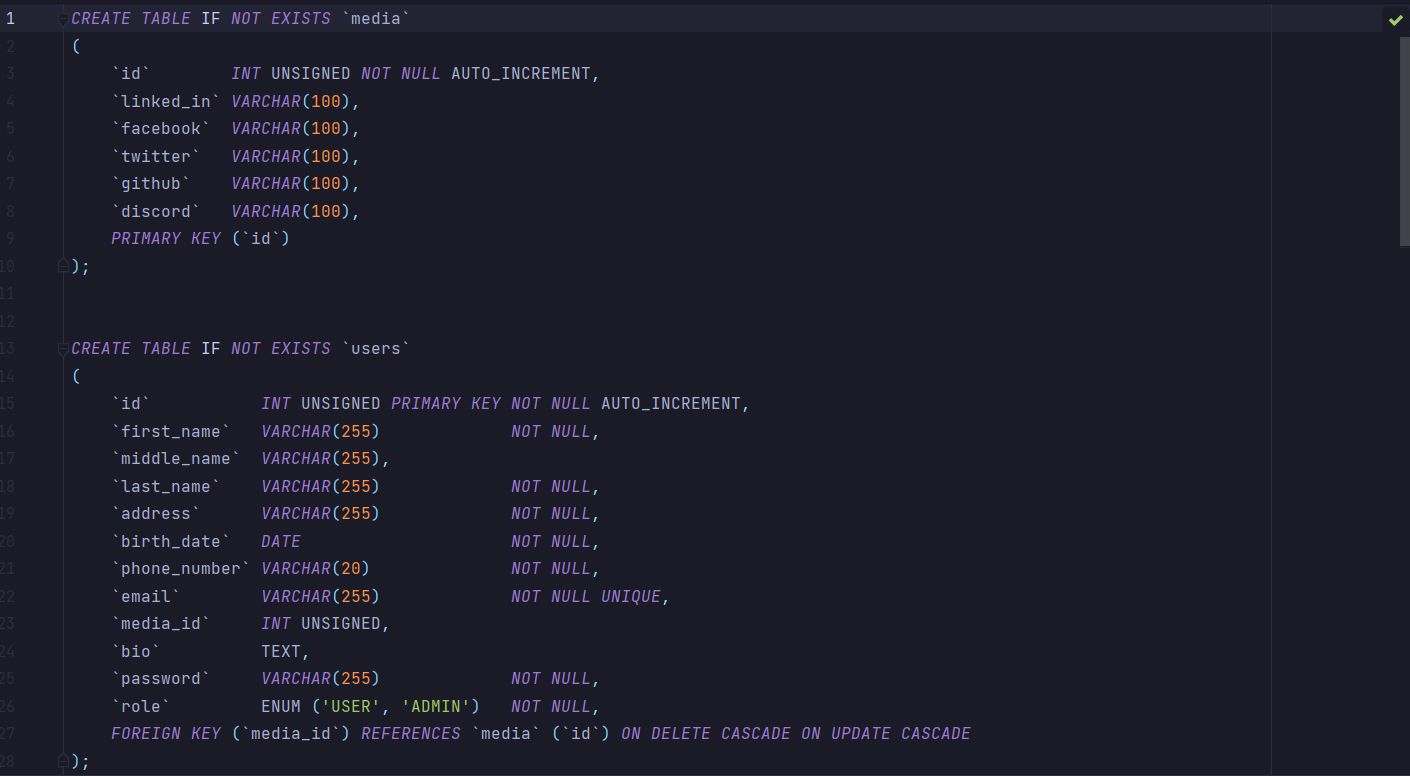




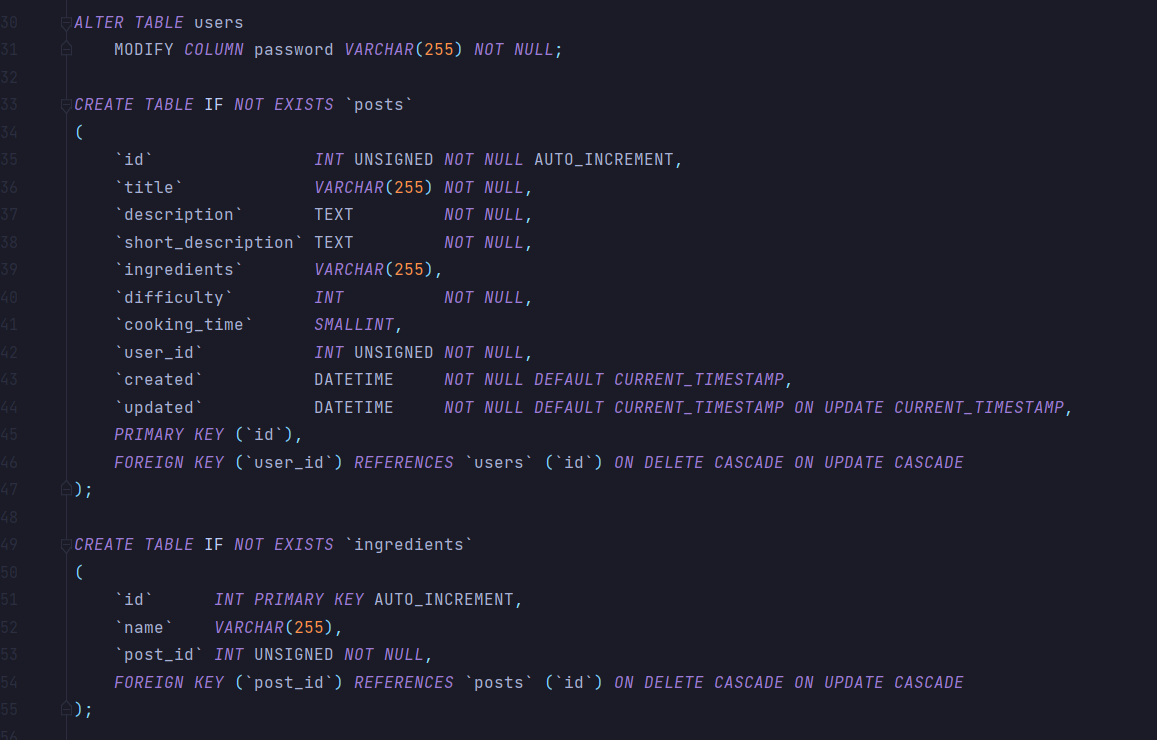
Zostały przygotowane dwa skrypty:

1. Skrypt tworzący bazę danych *schema.sql*
2. Skrypt z danymi testowymi *data.sql*

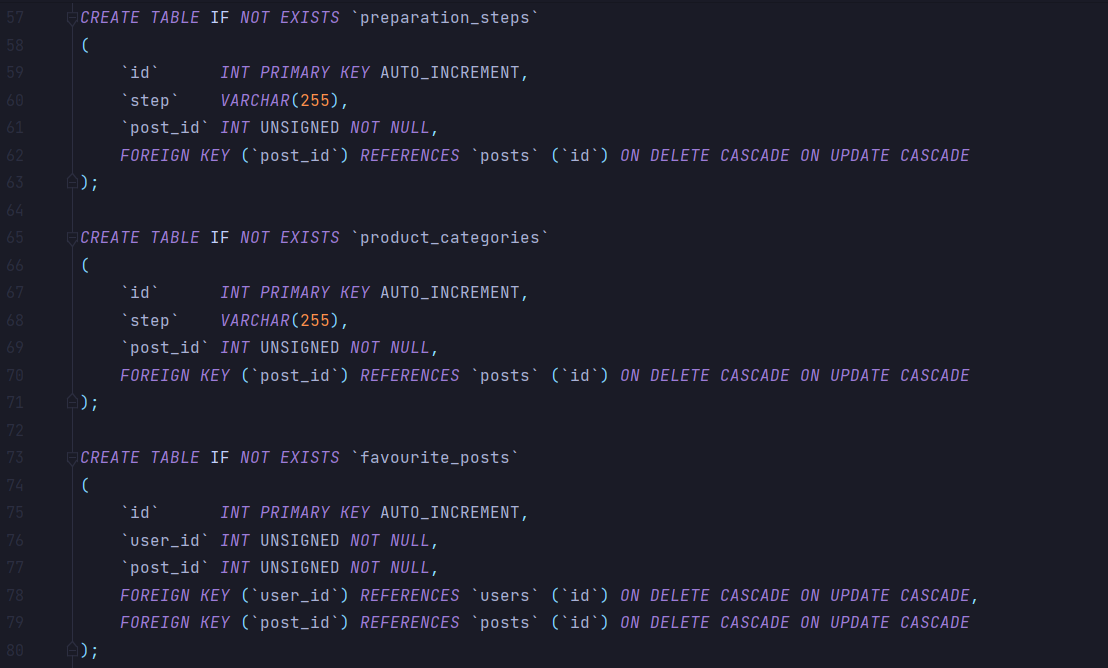
Prezentacja skryptu tworzącego bazę danych:



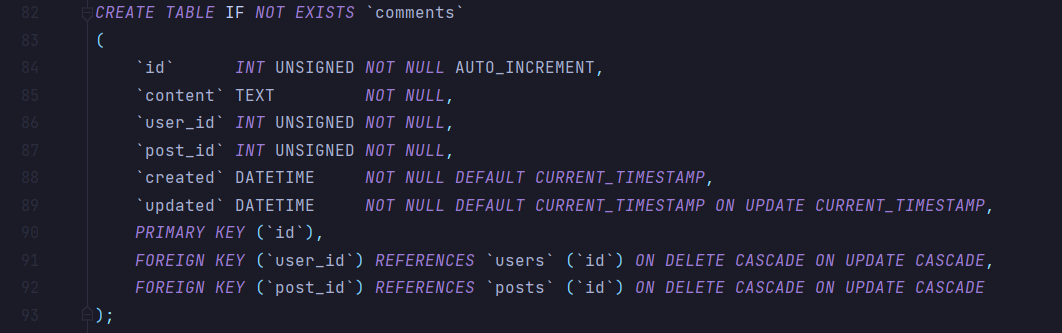
*schema.sql*

**

*schema.sql cd.*

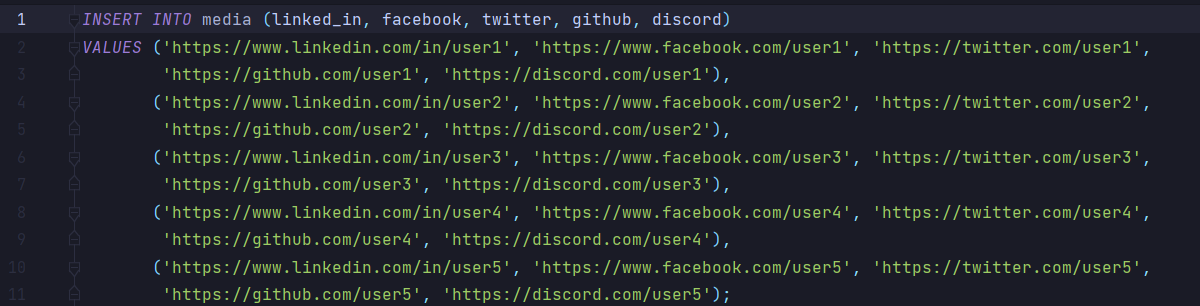


*schema.sql cd.*

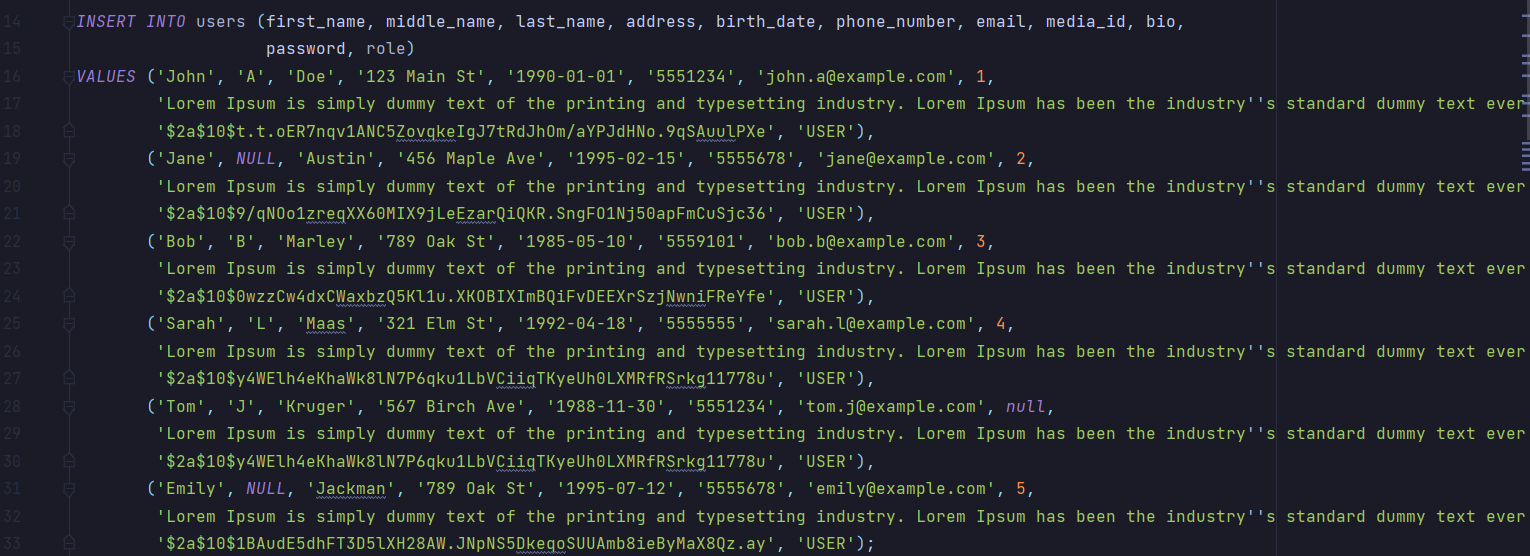
**

*schema.sql cd.*

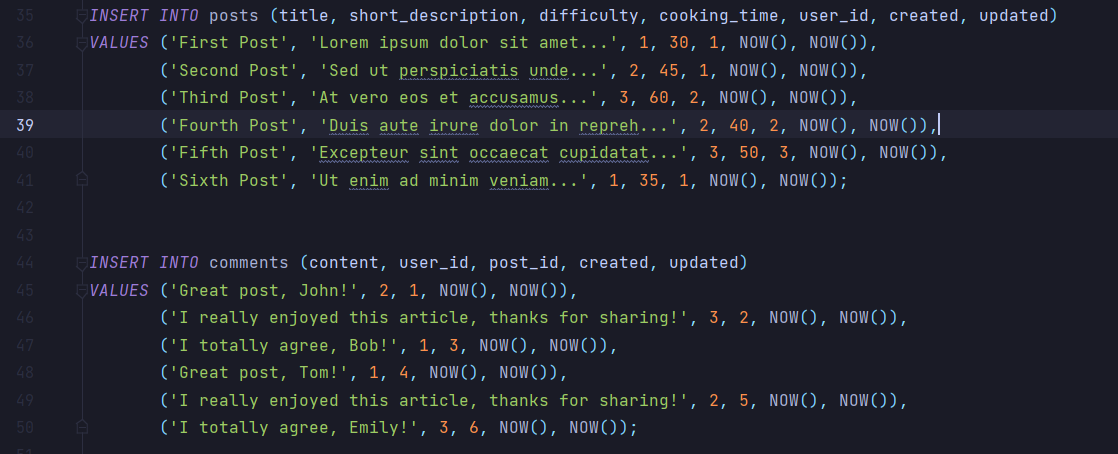
Prezentacja skryptu zapewnianiającego dane testowe:



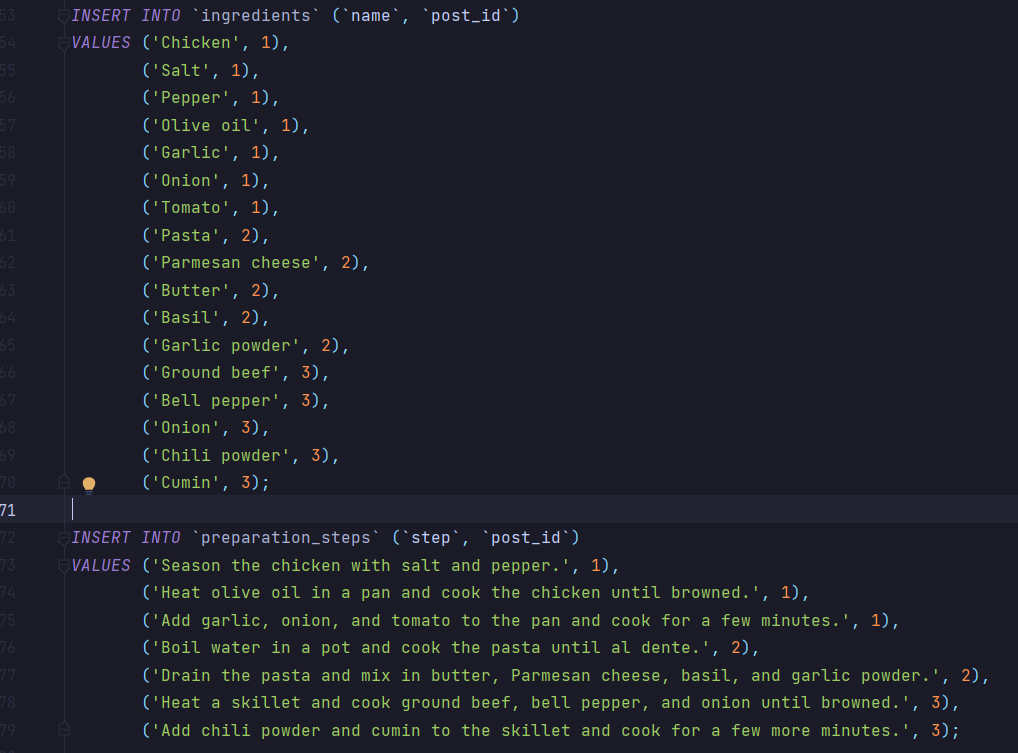
*data.sql*

**

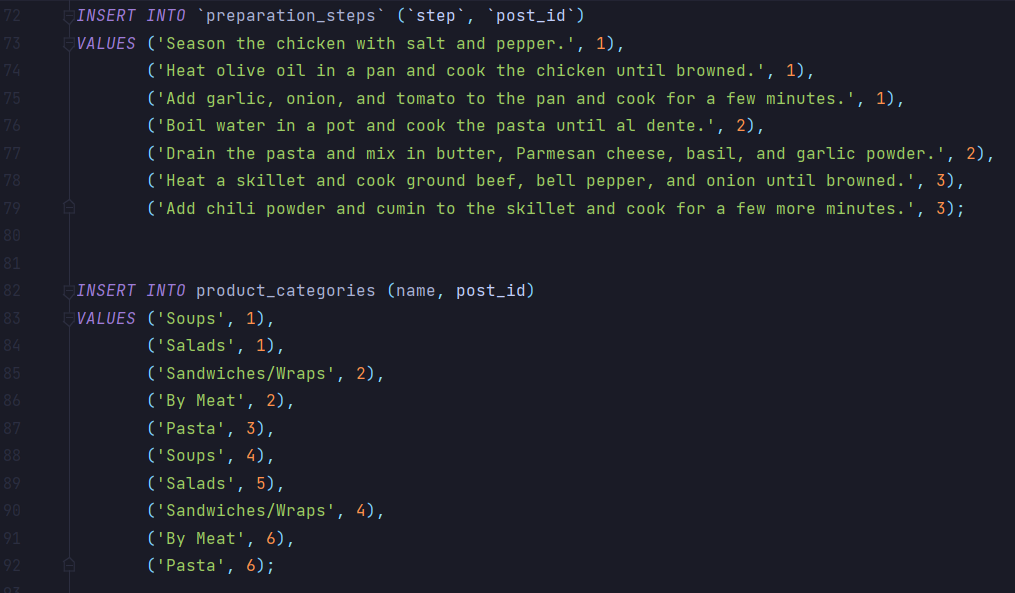
*data.sql cd.*

**

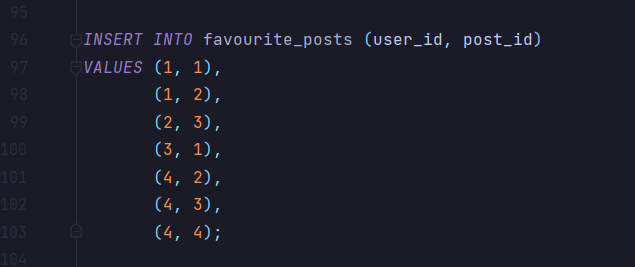
*data.sql cd.*

**

*data.sql cd.*

**

*data.sql cd.*

**

*data.sql cd.*