**Документация на проекта "The Book of Recipes"**

**1. Въведение**

Проектът "The Book of Recipes" е уеб приложение, което позволява на потребителите да разглеждат, добавят и управляват рецепти. Проектът следва принципите на MVC (Model-View-Controller) архитектурата.

**2. Участници в проекта**

**Даниел Бойчев и Николай Табальов**

**3. Технологии и инструменти**

* **Език за програмиране**: C#
* **Framework**: ASP.NET Core
* **База данни**: Entity Framework Core с In-Memory Database
* **Тестова рамка**: xUnit
* **IDE**: Visual Studio

**4. Структура на проекта**

**Проектът "The Book of Recipes" е организиран в следната структура, която следва стандартните практики за ASP.NET Core приложения. Всяка папка и файл имат специфична роля в архитектурата на приложението.**

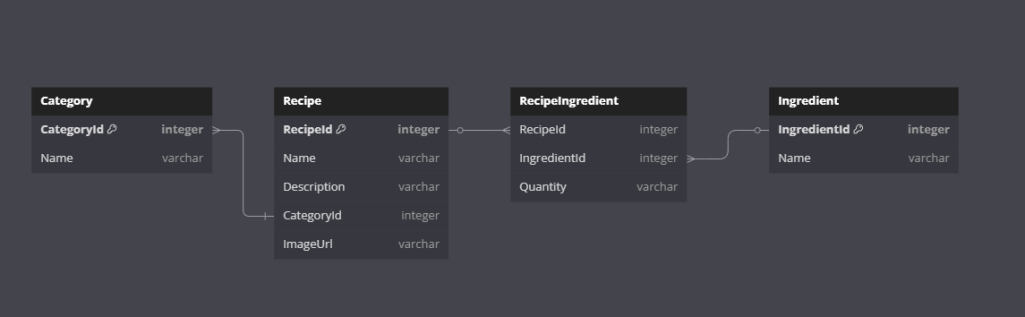
**4.1. Папки и файлове**

* **Models:**
  + **Съдържа класовете, които представляват данните в приложението. Всеки клас е свързан с таблица в базата данни.**
    - **Recipe.cs: Клас, представляващ рецепта. Съдържа свойства като RecipeId, Name, Description, Instructions, CategoryId, ImageUrl и RecipeIngredients.**
    - **Category.cs: Клас, представляващ категория на рецепти. Съдържа свойства като CategoryId и Name.**
    - **RecipeIngredient.cs: Клас, представляващ връзката между рецепти и съставки. Съдържа свойства за идентификатор на рецептата и идентификатор на съставката.**
    - **ErrorViewModel.cs: Клас, представляващ модела за грешка, който съдържа RequestId за идентифициране на конкретна заявка.**
* **Controllers:**
  + **Съдържа контролерите, които обработват HTTP заявките и взаимодействат с моделите.**
    - **HomeController.cs: Контролер, отговарящ за началната страница на приложението. Включва методи за показване на рецепти и обработка на грешки.**
    - **CategoriesController.cs: Контролер, отговарящ за управлението на категориите. Включва методи за показване на категории и рецепти по категории.**
    - **RecipesController.cs: Контролер, отговарящ за CRUD операциите на рецептите. Включва методи за добавяне, редактиране, изтриване и показване на рецепти.**
    - **AccountController.cs: Контролер, отговарящ за управлението на акаунтите. Включва методи за регистрация на акаунт, влизане в акаунта и управаление на акаунта.**
* **Views:**
  + **Съдържа представленията (views), които се рендират на клиентската страна. Всяко представление е свързано с контролер.**
    - **Home: Папка, съдържаща представления за HomeController, включително Index.cshtml и Error.cshtml.**
    - **Categories: Папка, съдържаща представления за CategoriesController, включително Index.cshtml и Recipes.cshtml.**
    - **Recipes: Папка, съдържаща представления за RecipesController, включително Index.cshtml, Create.cshtml, Edit.cshtml, Delete.cshtml и Details.cshtml.**
    - **Account: Папка, съдържаща представления за AccountController, включително Edit.cs, Login.cs, Register.cs и**

**Profile.cs.**

* **Data:**
  + **Съдържа контекста на базата данни и конфигурацията на Entity Framework.**
    - **ApplicationDbContext.cs: Клас, наследяващ DbContext, който представлява контекста на базата данни. Съдържа DbSet свойства за Recipes, Categories и RecipeIngredients.**
* **wwwroot:**
  + **Съдържа статични файлове, като CSS, JavaScript и изображения.**
    - **css: Папка, съдържаща стиловите файлове на приложението.**
    - **js: Папка, съдържаща JavaScript файлове.**
    - **images: Папка, съдържаща изображения, използвани в приложението.**

**4.2. Връзките между таблиците в базата данни.**

****

**5. Основни функции**

* **Преглед на рецепти**: Потребителите могат да разглеждат списък с рецепти, включително описание, инструкции и съставки.
* **Филтриране по съставки**: Потребителите могат да търсят рецепти по определени съставки.
* **Добавяне на нови рецепти**: Потребителите могат да добавят нови рецепти с всички необходими полета.
* **Филтриране на категории**: Потребителите могат да търсят рецепти по определени категории.

**6. Тестове**

Проектът включва unit тестове за основните контролери, които проверяват функционалността на методите. Тестовете са написани с помощта на xUnit и включват:

* **Тестове за HomeController**:
  + Проверка на метода **Index**, който връща случайни рецепти.
  + Проверка на метода **Index**, който връща рецепти по съставки.
  + Проверка на метода **Error**, който връща правилния модел.
* **Тестове за CategoriesController**:
  + Проверка на метода **Index**, който връща списък с категории.
  + Проверка на метода **Recipes**, който връща рецепти за определена категория.
* **Тестове за RecipesController**:
  + Проверка на CRUD операциите за рецепти.

**7. Заключение**

Проектът "The Book of Recipes" демонстрира основните принципи на разработка на уеб приложения с ASP.NET Core и включва функционалности за управление на рецепти и категории. Тестовете осигуряват надеждност и стабилност на приложението.