

DEVELOPMENT OF AN ELECTRONIC PLATFORM FOR COOKING RECIPES

*Assoc. Prof. PhD Dimitar Minchev, Burgas Free University, mitko@bfu.bg
Simona Dimitrova, graduate student, szhdimitrova@gmail.com*

Abstract: This publication examines the development of a web-based application for a personal culinary diary, which facilitates the storage, management, and sharing of recipes in a structured, flexible, and visually accessible environment.

Keywords: Microsoft, Visual Studio, ASP.NET Core, Razor Pages, Entity Framework Core, Microsoft SQL Server, MSTest.

РАЗРАБОТВАНЕ НА ЕЛЕКТРОННА ПЛАТФОРМА ЗА ГОТВАРСКИ РЕЦЕПТИ

*доц. д-р Димитър Минчев, Бургаски свободен университет, mitko@bfu.bg
Симона Димитрова, дипломант, szhdimitrova@gmail.com*

Абстракт: Настоящата публикация разглежда разработката на уеб базирано приложение за личен кулинарен дневник, което да улеснява съхраняването, управлението и споделянето на рецепти в структурирана, гъвкава и визуално достъпна среда.

Ключови думи: Microsoft, Visual Studio, ASP.NET Core, Razor Pages, Entity Framework Core, Microsoft SQL Server, MSTest.

Въведение

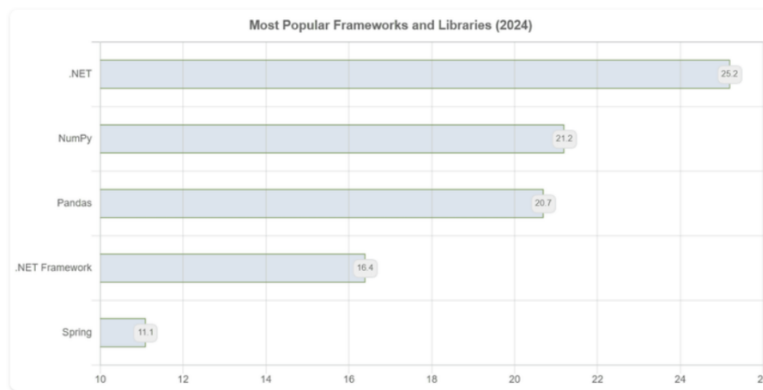
В съвременния дигитален свят все повече хора търсят удобни начини да организират личната си информация, включително и своите кулинарни идеи. Домашните готвачи и любителите на готвенето често събират рецепти от различни източници, експериментират със съставки и създават собствени вариации на ястия. За да се улесни съхранението и управлението на тази информация, е разработено приложение за личен кулинарен дневник, който позволява на потребителя да записва, редактира и преглежда своите рецепти в удобна и структурирана форма, както и при желание да споделя уникалните си рецепти с други потребители на платформата.

Разработва се електронна платформа, в която потребителите разполагат с възможност да създават, управляват и споделят свои собствени рецепти. Още при създаването на рецепта може да се определи дали тя да бъде публична – достъпна за всички потребители в системата – или частна, видима само за самия автор. Тази гъвкавост предоставя контрол върху съдържанието и нивото на споделяне. В допълнение, платформата позволява по всяко време да се променя видимостта на вече създадена рецепта, което улеснява адаптирането към нуждите и предпочитанията на потребителя. Тази функционалност е от съществено значение за изграждането на персонализирано потребителско изживяване и подкрепя принципите на поверителност и свобода при споделяне на съдържание.

Избор на технологии

В разработката на проекта са използвани технологии от екосистемата на Microsoft: Visual Studio [1], ASP.NET Core Razor Pages [2], Entity Framework Core [3], Microsoft SQL Server и MSTest [4]. Microsoft предлага пълноценна екосистема от технологии, които са тясно интегрирани и осигуряват висока степен на надеждност, сигурност и поддръжка. Използването на ASP.NET Core Razor Pages в комбинация с Entity Framework Core и MSSQL дава възможност за създаване на мощни уеб приложения с висока производителност и мащабируемост. В допълнение, Visual Studio – водещата интегрирана среда за разработка на Microsoft – предлага богат набор от инструменти за дебъг, автоматизирано тестване, което значително ускорява процеса на разработка и поддръжка.

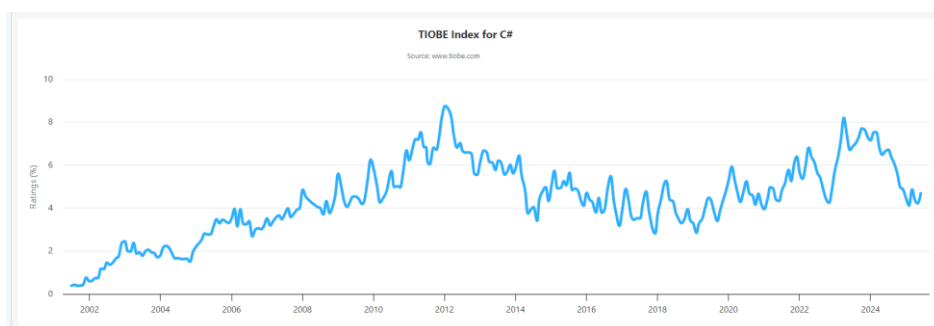
NET платформата се използва от приблизително 25% от всички софтуерни разработчици по света, което я прави една от най-популярните.



Фиг. 1. Статистически данни за най-популярните frameworks и библиотеки [5]

ASP.NET Core е уеб framework, който е част от .NET платформата, а C# е основният език за програмиране, използван за създаване на приложения с него. Всички основни компоненти — контролери, модели, Razor страници, Blazor компоненти, middleware и API логика — се пишат на C#.

През 2025 година C# продължава да бъде един от най-популярните езици за програмиране в света. Според TIOBE Index [6] за май 2025 г., C# заема 5-то място с дял от 4.22%, което го поставя сред водещите езици в индустрията. Това е доказателство за неговата стабилна позиция и широко приложение в корпоративни и уеб среди.



Фиг. 2 Графика от Tiobe.com за май 2025 г., C# заема 5-то място с дял от 4.22%, което го поставя сред водещите езици в индустрията.[6]

В годишното проучване на Stack Overflow Developer Survey, C# е класиран на 8-мо място по използваемост, като 27.1% от всички анкетирани разработчици го използват, а сред

професионалните разработчици този процент достига 28.8% [5, 6, 7, 8]. Това показва, че езикът е предпочитан не само от любители, но и от професионалисти, които го използват в реални проекти.

Според доклада на JetBrains Developer Ecosystem, броят на активните C# програмисти в света е приблизително 3.05 милиона, което го прави един от най-широко използваните езици в корпоративна среда. [5, 6, 7, 8]

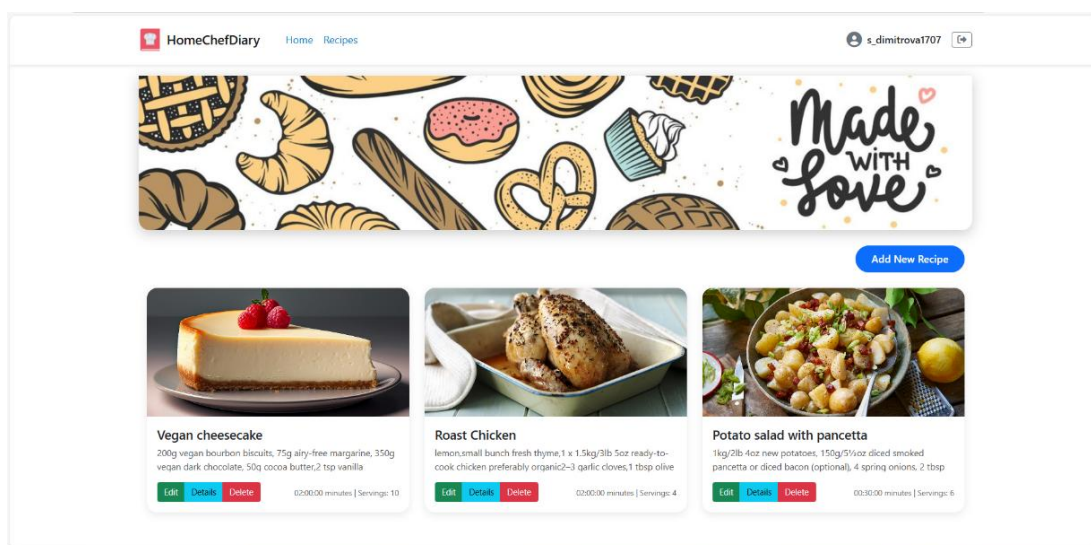
Причините за тази популярност включват модерния синтаксис, високата производителност, богатата стандартна библиотека и кросплатформената поддръжка чрез .NET.

Разработка на платформата

Платформата представлява съвременно уеб приложение, разработено с помощта на ASP.NET Core Razor Pages, което предоставя дигитално пространство за управление на лични и споделени рецепти. Основната идея на проекта е да улесни потребителите – както любители, така и опитни готвачи – при създаването, съхраняването и споделянето на кулинарни рецепти в структурирана, достъпна и персонализируема форма.

Платформата реализира пълния CRUD цикъл (Create, Read, Update, Delete), като осигурява интуитивен и адаптивен потребителски интерфейс за добавяне на нови рецепти, редактиране на съществуващи, преглед на детайли и изтриване на нежелани записи.

Ключова особеност е възможността потребителите да избират нивото на видимост на всяка рецепта още при създаването ѝ — тя може да бъде лична (частна) или публична, достъпна за всички потребители в системата. Тази настройка може да се променя по всяко време, като това дава свобода и контрол върху споделянето на съдържанието. Платформата цели да съчетае персонален кулинарен дневник с обществено достъпна библиотека от рецепти, подобрявайки потребителското изживяване чрез баланс между индивидуалност и общност.



Фиг. 3. Изглед от страница с личните рецепти на потребител

В уеб приложението асинхронното програмиране играе ключова роля за осигуряване на бърза реакция и висока производителност при работа с база данни и потребителски

заявки. То е реализирано чрез използването на ключовите думи `async` и `await` в методите, които извършват операции, изискващи време – основно четене и запис на данни в базата данни чрез Entity Framework Core.

Асинхронното програмиране позволява изпълнението на дълготрайни операции, като заявки към база данни, без да се блокира основната нишка на приложението. Това означава, че уеб сървърът може да продължи да обслужва други потребителски заявки, докато текущата операция се изпълнява във фонов режим. По този начин се постига по-висока производителност и мащабируемост на приложението, особено когато то се използва от множество потребители едновременно.

Логиката за достъп до данни се реализира с помощта на Entity Framework Core (EF Core), който действа като ORM (Object-Relational Mapper). С него се елиминира нуждата от писане на „суров“ SQL код – вместо това се използват .NET класове за взаимодействие с базата данни.

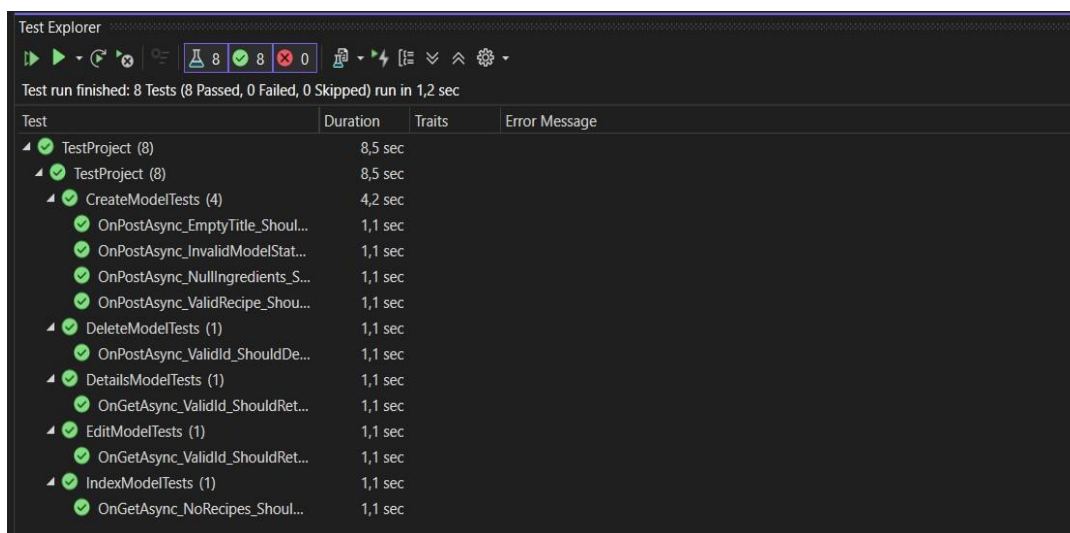
В този проект е използван Entity Framework Core съвместно с ASP.NET Core Identity, с цел осигуряване на възможност за автентикация и управление на потребители. Това е особено полезно в контекста на лична платформа за рецепти, където достъпът може да бъде ограничен до конкретен потребител.

Системата предоставя възможност за регистрация и вход в потребителски акаунт, което е ключова функционалност за осигуряване на персонализиран достъп до ресурсите на платформата. Всеки потребител може да създаде свой собствен профил, чрез който да създава, редактира и съхранява рецепти, както и да използва останалите функционалности, достъпни само за регистрирани потребители.

Използваната архитектура осигурява мащабируемост и стабилност, като същевременно улеснява поддръжката и бъдещото разширяване на функционалностите – например чрез добавяне на филтри по време за приготвяне, брой порции или ниво на трудност.

Изследване и анализ на разработената платформа

В рамките на разработката на уеб платформата бяха реализирани компонентни тестове, които имат за цел да валидират коректната работа на основните функционалности в приложението. Тестването е осъществено с помощта на MSTest – официалната тестова рамка на Microsoft – и Entity Framework Core InMemory, което позволява провеждането на изолирани тестове без използването на реална база данни.



Test	Duration	Traits	Error Message
TestProject (8)	8,5 sec		
TestProject (8)	8,5 sec		
CreateModelTests (4)	4,2 sec		
OnPostAsync_EmptyTitle_Shoul...	1,1 sec		
OnPostAsync_InvalidModelStat...	1,1 sec		
OnPostAsync_NullIngredients_S...	1,1 sec		
OnPostAsync_ValidRecipe_Shou...	1,1 sec		
DeleteModelTests (1)	1,1 sec		
OnPostAsync_ValidId_ShouldDe...	1,1 sec		
DetailsModelTests (1)	1,1 sec		
OnGetAsync_ValidId_ShouldRet...	1,1 sec		
EditModelTests (1)	1,1 sec		
OnGetAsync_ValidId_ShouldRet...	1,1 sec		
IndexModelTests (1)	1,1 sec		
OnGetAsync_NoRecipes_Shoul...	1,1 sec		

Фиг. 4. Изглед от Test Explorer

Тези тестове повишават **надеждността, устойчивостта и поддръжаемостта** на приложението, като позволяват бъдещи модификации без риск от неволно нарушаване на основната бизнес логика.

Тестовите обхващат ключовите Razor Pages модели, свързани с операциите по създаване, редактиране, изтриване и преглед на рецепти в приложението. В обобщение на проведените тестове могат да се направят следните изводи:

1. **CreateModelTests**

Тестван е моделът за създаване на нова рецепта. Извършени са проверки както при валидни, така и при невалидни входни данни. Сценариите включват: Успешно създаване на валидна рецепта; Отказ от запис при липсващо заглавие или съставки; Обработка на невалидна ModelState. Тестовите гарантират, че при грешка потребителят остава на текущата страница и рецептата не се записва в базата.

2. **DeleteModelTests**

Проверен е процесът на изтриване на рецепта по зададен Id. Установено е, че при валиден идентификатор рецептата се премахва от базата и потребителят се пренасочва към основната страница. Това осигурява надеждно поведение при изтриване на данни.

3. **DetailsModelTests**

Извършена е проверка на зареждането на детайлна информация за рецепта. При валиден Id се удостоверява, че моделът успешно зарежда съответната рецепта и визуализира необходимата информация. Тестовите потвърждават точността на данните, представяни пред потребителя.

4. **EditModelTests**

Тестван е моделът за зареждане на данни при редакция на рецепта. Проверено е, че при зададен валиден Id, съществуващата информация се зарежда коректно за редактиране. Това гарантира безпроблемното стартиране на процеса по актуализация на данни.

5. **IndexModelTests**

Извършен е тест на главната страница, която показва списък с рецепти. Проверява се поведението при празна база данни – очаква се връщане на празен списък без грешки. Това удостоверява стабилността на началната страница в различни състояния на системата.

Тези тестове обхващат критичните модули за работа с рецепти и осигуряват по-висока надеждност и устойчивост на системата. Използването на In-Memory база улеснява провеждането на тестовете и позволява по-бърз цикъл на разработка и валидиране на логиката, без да се разчита на реална база от данни.

Заклучение

Настоящата публикация представя разработено уеб приложение, което изпълнява ролята на дигитален кулинарен дневник, предоставящ възможност за създаване, управление и споделяне на рецепти. Проектът успешно отговаря на нарастващата нужда от персонализирани платформи за съхраняване на кулинарна информация, като съчетава функционалност, гъвкавост и лесна употреба в единна система.

Реализацията на приложението премина през всички ключови етапи на софтуерното инженерство – от анализ на потребителските нужди и проектиране на архитектурата, до имплементация, тестване и финално представяне. При разработката бяха използвани съвременни технологии от .NET екосистемата – ASP.NET Core Razor Pages, Entity Framework Core, Microsoft SQL Server, както и MSTest за проверка на функционалността. Това допринесе за изграждане на стабилна архитектура с ясно разделение на слоеве – презентационен, логически и слой за достъп до данни.

Приложението предоставя на потребителя не само възможност да съхранява и редактира рецепти, но и да контролира тяхната видимост – функция, която повишава нивото на лична ангажираност и сигурност. Благодарение на това, приложението може да се използва както за създаване на лична база от рецепти, така и като средство за споделяне на вдъхновение и кулинарни идеи с общността.

Проектът демонстрира ефективното приложение на принципите на уеб разработката и добрите практики при изграждане на потребителски ориентирани системи. Разработката показва как чрез ясна архитектура, модерен технологичен стек и добро познаване на нуждите на крайните потребители може да се създаде практична, полезна и адаптивна уеб платформа.

Използвана литература

- [1] Microsoft, Visual Studio Documentation, <https://learn.microsoft.com/en-us/visualstudio/>
- [2] Microsoft, Introduction to Razor Pages in ASP.NET Core, <https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-9.0>
- [3] Vincent Maverick Durano, Getting Started with Entity Framework Core, <https://www.codeproject.com/Articles/1218427/Getting-Started-with-Entity-Framework-Core-Buildin>
- [4] Microsoft, MSTest Unit Testing Guide, <https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-mstest-getting-started>
- [5] Chandra, Yuvraj, C# Popularity, Usage, and Developer Momentum in 2025, <https://www.zenrows.com/blog/c-sharp-popularity>
- [6] TIOBE Index, The C# Programming Language, <https://www.tiobe.com/tiobe-index/csharp/>
- [7] W3techs, Usage statistics of ASP.NET, <https://w3techs.com/technologies/details/pl-aspnet>
- [8] Reddit, What database do you use as a .NET developer team?, https://www.reddit.com/r/dotnet/comments/1i0w1a6/what_database_do_you_use_as_a_net_developer_team/