**ДИПЛОМЕН ПРОЕКТ**

**НА ТЕМА**

**ПЛАТФОРМА ЗА ДОБРОВОЛНА ВЗАИМНОПОМОЩ ЗА ДОМАШНИ ЛЮБИМЦИ**

**“Pet Exchange”**

**Ученик: Ръководител консултант:**

Виктор Божидаров Зашев д-р. Янислав Картелов

Професия: Системен програмист

Специалност: Системно програмиране

гр. Пловдив, 2025

Примерно съдържание Цветан

Съдържание

[1. Увод 5](#_Toc133442481)

[1.1 Цел на проекта 6](#_Toc133442482)

[2. Използвани технологии 7](#_Toc133442483)

[2.1 .NET 7 8](#_Toc133442484)

[2.2 Visual Studio 2022 8](#_Toc133442485)

[2.3 C# 9](#_Toc133442486)

[2.4 ASP.NET Core 9](#_Toc133442487)

[2.5 Blazor (Blazor Server) 10](#_Toc133442488)

[2.6 SignalR 11](#_Toc133442489)

[2.7 MSSQL Server 11](#_Toc133442490)

[2.8 MSSQL Server Management Studio 12](#_Toc133442491)

[2.9 Entity Framework Core 12](#_Toc133442492)

[2.10 Razor 13](#_Toc133442493)

[2.11 ASP.NET Core Identity 13](#_Toc133442494)

[2.12 Azure Blob Storage 14](#_Toc133442495)

[2.13 Уеб браузър 14](#_Toc133442496)

[3. Архитектура 16](#_Toc133442497)

[3.1 База данни 16](#_Toc133442498)

[3.1.1 Моделът User 17](#_Toc133442499)

[3.1.2 Моделът Video 18](#_Toc133442500)

[3.1.3 Моделът Training 18](#_Toc133442501)

[3.1.4 Моделът Team 19](#_Toc133442502)

[3.1.5 Моделът IndividualTraining 20](#_Toc133442503)

[3.1.6 Моделът Game 20](#_Toc133442504)

[3.2 Помощни класове 22](#_Toc133442505)

[3.2.1 SpartakAcademyDBContext 22](#_Toc133442506)

[3.2.2 IRepository 23](#_Toc133442507)

[3.2.3 TrainingRepository 24](#_Toc133442508)

[3.2.4 AppAuthenticationStateProvider 25](#_Toc133442509)

[3.2.5 ClaimConverter 26](#_Toc133442510)

[3.3 Services 27](#_Toc133442511)

[3.3.1 Adding services in Program.cs 28](#_Toc133442512)

[3.3.2 IdentityService 28](#_Toc133442513)

[3.3.3 BlobStorageService 29](#_Toc133442514)

[3.3.4 TrainingService 30](#_Toc133442515)

[3.4 Случаи на употреба 31](#_Toc133442516)

[3.4.1 Admin 31](#_Toc133442517)

[3.4.2 Coach 32](#_Toc133442518)

[3.4.3 Player 32](#_Toc133442519)

[4. Ръководство за потребителя 33](#_Toc133442520)

[4.1 Начална страница 33](#_Toc133442521)

[4.2 Форма за вход 33](#_Toc133442522)

[4.3 Форма за регистрация 34](#_Toc133442523)

[4.4 Меню 35](#_Toc133442524)

[4.5 Профил на потребителя 36](#_Toc133442525)

[4.6 Меню с играчите и треньорите 36](#_Toc133442526)

[4.7 Създаване на обект 37](#_Toc133442527)

[4.8 Меню с ресурсите 37](#_Toc133442528)

[5. Инсталиране и стартиране 38](#_Toc133442529)

[6. Заключение 39](#_Toc133442530)

[7. Използвана литература 40](#_Toc133442531)

[7.1 Книги 40](#_Toc133442532)

[7.2 Уеб сайтове 40](#_Toc133442533)

[8. Фигури 41](#_Toc133442534)

Оформяне на дипломния проект:

Обем на дипломния проект – минимум 30 страници

Формат: А4; Брой редове в стр.: 30; Брой на знаците: 60 знака в ред

Общ брой на знаците в 1 стр.: 1800 – 2000 знака

Шрифт: Times New Roman, размер 12

I – 24

1 – 22

* 1. – 20
     1. – 18

1. – 16

I - 14

Текст - 12

Contents

[I. Увод 5](#_Toc192865195)

[II. Основна част 6](#_Toc192865196)

[1. Цел на ДП. 6](#_Toc192865197)

[2. Проучване на съществуващи приложения. 6](#_Toc192865198)

[3. Обхват на проекта 6](#_Toc192865199)

[4. Технологии 6](#_Toc192865200)

[4.1 .NET 8 6](#_Toc192865201)

[5. Изисквания 7](#_Toc192865202)

[● Функционални изисквания 7](#_Toc192865203)

[● Нефункционални изисквания 7](#_Toc192865204)

[● Бизнес изисквания 7](#_Toc192865205)

[● Системни изисквания 7](#_Toc192865206)

[6. Архитектурна документация 7](#_Toc192865207)

[7. Документация на кода: 8](#_Toc192865208)

[● API документация 8](#_Toc192865209)

[8. Потребителска документация: 8](#_Toc192865210)

[9. Авторски права 9](#_Toc192865211)

[III. Заключение 9](#_Toc192865212)

[IV. Списък на използвана литература 10](#_Toc192865213)

[V. Приложения 12](#_Toc192865214)

# Увод

Уводът (въведение) съдържа кратко описание на основните цели и очаквани

резултати.

# Основна част

## 1. Цел на ДП.

Формулира се целта на дипломния проект и задачите, които трябва да бъдат решени,

за да се постигне тази цел.

## 2. Проучване на съществуващи приложения.

Да се започне с проучване, което описва предпоставките за създаване на

приложението. Съдържа описание и анализ на съществуващи подобни програмни

системи и продукти, като се цитират /с номерация, която се изнася накрая на

отделна страница/ съответните литературни източници.

## 3. Обхват на проекта

Какво точно ще бъде разработено и какви са

ограниченията на системата.

## 4. Технологии

Описание на използваните технологии, програмни езици,

платформи и инструменти. Аргументирано се избират подходящи

технологии, като се цитират съответните литературни източници

### 4.1 .NET 8

### 4.2 Visual Studio 2022

Visual Studio 2022 е най-новата версия на средата за разработка (Integrated Development Environment или IDE), разработена от Microsoft. Тя представлява цялостен набор от инструменти и услуги за разработка, които позволяват на разработчиците да създават широк спектър от приложения, включително настолни, уеб, мобилни, игрови и базирани в облака приложения. Средата поддържа над 36 езика за програмиране, сред които: C#, C++, F#, JavaScript, HTML, CSS, XML, Python и много други, както и множество инструменти като Code Editor и Debugger, Nuget Packages, Git, IntelliCode – (интелигентна система за предложения за дописване на код), инструменти за качествен автоматично генериран код, Server Explorer и други. Visual Studio има както платени версии (Professional и Enterprise), така и безплатна пълна версия (Community).

### 4.3 C#

C# е модерен, обектно-ориентиран език за програмиране, разработен от Microsoft. От началото си през 2002 г. бързо се превръща в един от най-популярните езици за програмиране, използвани за разработване на настолни приложения за Windows, уеб приложения, мобилни приложения, игри и др. C# е силно типизиран език, което означава, че всички променливи трябва да бъдат декларирани с определен тип данни. Важна характеристика е така нареченият garbage-collector, чрез който системата за изпълнение автоматично управлява паметта вместо разработчиците, като освобождава обекти, които вече не са необходими. Във връзка с проекта, C# е подходящ избор заради възможността за писане на асинхронен код, който ще позволи по-добра и по-бърза работа на приложението, което ще доведе и до по-добро потребителско изживяване.

### 4.4 ASP.NET CORE

ASP.NET е web application framework, разработен от Microsoft, който позволява на разработчиците да създават динамични, интерактивни, мащабируеми, надеждни и сигурни уеб приложения и уебсайтове. ASP.NET се основава на .NET Framework и поддържа широк набор от езици за програмиране, включително C#. Поддържа и редица модели за програмиране, включително Web Forms, MVC (Model-View-Controller) и Web API. В комбинация с други решения на Microsoft като SignalR и Identity Core чрез ASP.NЕТ може да се създаде стабилно уеб приложение със система за акаунти и роли на различните потребители. ASP.NЕТ може да се комбинира с front-end framework, различен от тези на Microsoft, като Angular, ReactJS и други.

ASP.NET е една от най-популярните технологии за web development и през 2022г. постига около 37% пазарен дял, като едновременно с това приблизително 32% от уеб приложенията днес използват ASP.NET. Заради предимствата си пред подобни технологии като Django и Angular.js, ASP.NET е подходящ за този проект и ще даде възможност за създаване на Blazor Server проект.

### 4.5 MSSQL Server

MSSQL е вид релационна база данни, базирана на SQL (Structured Query Language). При релационните бази данни група от обекти се пази в таблица, като всеки ред съответства на даден обект, а всяка колона ­– на характеристиките му. MSSQL Server осигурява достъпност, мащабируемост, сигурност и производителност. Той поддържа различни типове данни, включително текст, изображения и XML, и осигурява поддръжка на сложни заявки, обработка на транзакции и съхранение на данни. В комбинация с технология като Entity Framework MSSQL Server ни позволява да запазваме информацията за обектите си в релационна база данни без да се налага да пишем ръчно SQL заявки.

### 4.6 MSSQL Server Management Studio

MSSQL Server Management Studio (SSMS) е инструмент с графичен потребителски интерфейс, който се използва за управление и администриране на бази данни на Microsoft SQL Server. Той позволява на администраторите на бази данни и разработчиците да изпълняват широк кръг от задачи като конфигуриране и наблюдение на сървъри, създаване и управление на бази данни, както и писане и изпълнение на SQL заявки. Той предоставя широк набор от функционални възможности за управление на бази данни на SQL Server, включително: Query Editor, Object Explore, Activity Monitor, Database Backup and Restore, Integration, Analysis и Reporting Services

### 4.7 Entity Framework Core

ORM (Object-Relational Mapping) технологиите позволяват на разработчиците да пишат код на познат обектно-ориентиран език (като C# или Java), за да взаимодействат с базата данни, без да се налага да пишат директно SQL заявки и така позволяват на разработчиците да се съсредоточат върху бизнес логиката на приложението. В този проект се използва Entity Framework Core (EF Core), който е ORM технология, разработена от Microsoft. EF Core позволява на разработчиците да работят с базата данни, използвайки специфични за областта обекти и LINQ заявки. Той поддържа различни доставчици на бази данни, включително Microsoft SQL Server, MySQL, PostgreSQL, SQLite и други. С негова помощ може да се създават приложения чрез два подхода– Code First и Database First.

Проектът е създаден с помощта на EF Core и Code First модела, при който се създават първо класовете, отговарящи на моделите в приложението, след което EF Core създава базата данни заедно с необходимите връзки и ключове по модела, изграден от разработчика с помощта на Data Annotations и наследяване и конфигуриране на DbContext класа.

### 4.8 LINQ

Language-Integrated Query (LINQ) е името на набор от технологии, базирани на интегрирането на възможностите за заявки директно в езика C #. Традиционно заявките срещу данни се изразяват като прости низове без проверка на типа при време на компилация или поддръжка на IntelliSense. Освен това трябва да научите различен език за заявки за всеки тип източник на данни: SQL бази данни, XML документи, различни уеб услуги и т.н. С LINQ заявката е първокласна езикова конструкция, подобно на класове, методи, събития. Пишете заявки срещу силно типизирани колекции от обекти, като използвате ключови думи на език и познати оператори. LINQ семейството от технологии осигурява последователно изпитание за обекти (LINQ to Objects), релационни бази данни (LINQ до SQL) и XML (LINQ до XML). LINQ поддържа два вида синтаксис на писане на заявки (LINQ Query Syntax) и (LINQ Method Syntax).

### 4.9 ASP.NET Core Identity

ASP.NET Core Identity е система за управление на потребители, роли и удостоверяване на потребители в ASP.NET уеб приложения. Тя предоставя оторизация за контрол на достъпа до различни ресурси и персонализиране на поведението на аутентикацията и оторизацията, за да отговаря на специфичните нужди на приложението. ASP.NET Core Identity поддържа различни методи за удостоверяване, включително локални акаунти, социални влизания (например Facebook, Google) и двуфакторно удостоверяване. Включва също и вградена поддръжка за хеширане и валидиране на пароли, което предотвратява често срещани уязвимости в сигурността. В проекта е използван ASP.NET Core Identity, защото е необходимо да се поддържат и установяват ролите на различните потребители в системата и той прави изграждането на структурата от роли и профили на потребителите по-лесна и изчистена.

### 4.10 Уеб браузър

Уеб браузърът е приложение, което позволява на потребителите да имат достъп до и да разглеждат уеб страници в интернет. Уеб браузърите предоставят графичен потребителски интерфейс, чрез който потребителите могат да взаимодействат с уеб съдържание, включително текст, изображения, видеоклипове и друга мултимедия. Достъпът до приложенията на Blazor Server, каквото ще е и използваното за проекта, може да се осъществи чрез всеки съвременен уеб браузър, който поддържа JavaScript. Въпреки това, някои по-стари версии на тези браузъри може да не се поддържат напълно или да имат ограничения в производителността при работа с приложенията на Blazor Server. Най-популярните уеб браузъри включват: Google Chrome, Mozilla Firefox, Apple Safari, Brave, Opera и други.

### 4.11 HTML

HTML (Hypertext Markup Language – език за маркиране на хипертекст) се използва за създаване и структуриране на съдържание за уеб. Това е стандартният език, използван за създаване на уеб страници и други документи, които могат да се разглеждат в уеб браузър.

HTML е език за маркиране, което означава, че използва тагове за определяне на структурата и съдържанието на дадена уеб страница. Тези тагове се използват за създаване на заглавия, параграфи, списъци, връзки, изображения, форми, таблици и други елементи на уеб страницата. Съдържанието на уебстраницата се определя чрез комбинация от HTML тагове и текст.

HTML е в основата на Интернет и е от съществено значение за създаването на уеб страници и други уеб базирани приложения. Той се използва в комбинация с други уеб технологии като CSS и JavaScript за създаване на интерактивни и интересни уеб изживявания.

HTML5 е най-новата версия на HTML, която включва нови функции, като поддръжка на мултимедия, нови семантични елементи и подобрена достъпност. HTML5 включва и API(Application Programming Interface) за геолокация, плъзгане и пускане и други усъвършенствани функции, което улеснява разработчиците при създаването на богати и интерактивни уеб приложения.

### 4.12 CSS

CSS (Cascading Style Sheets) е език, който се използва за описание на представянето на документ, написан в HTML или XML. Той определя как елементите на HTML да се показват на екрана, в печата или в други медии. CSS отделя представянето на документа от неговото съдържание и дава възможност на разработчиците да контролират оформлението, типографията, цветовете и други визуални аспекти на дадена уеб страница.

CSS се използва в комбинация с HTML за създаване на визуално привлекателни и адаптивни уебсайтове. Той позволява на разработчиците да прилагат стилизиране към отделни HTML елементи или към групи от елементи, като използват селектори за насочване към конкретни елементи на страницата. CSS също така позволява създаването на адаптивни дизайни, които регулират оформлението и стила на страницата в зависимост от размера на екрана и ориентацията на устройството, което се използва за разглеждане.

CSS се състои от поредица от правила, които определят стилизирането на дадена страница. Всяко правило се състои от селектор, който насочва към HTML елементите, към които ще се прилага стилизирането, и набор от декларации, които определят свойствата на стилизирането и техните стойности. CSS може да се запише в реда на HTML документа, в отделен файл за стилове или да се вгради в един HTML документ.

### 4.13 Javascript

JavaScript е популярен език за програмиране от високо ниво, който се използва предимно за създаване на динамични и интерактивни уеб страници. Създаден е през 1995 г. от Брендън Айх по време на работата му в Netscape Communications Corporation.

JavaScript е „client side“ скриптов език, което означава, че се изпълнява в уеб браузъра на потребителя, а не на уеб сървър. Често се използва в комбинация с HTML и CSS за добавяне на интерактивност и функционалност към уеб страниците. JavaScript се използва и в „server side“ програмирането, като например в Node.js, популярна среда за изпълнение на JavaScript за изграждане на приложения от страна на сървъра.

JavaScript е обектно-ориентиран език, което означава, че позволява на разработчиците да дефинират обекти и техните свойства и методи. Той също така поддържа парадигми на функционалното програмиране, като например функции от по-висок ред и затваряне.

JavaScript разполага с широк набор от вградени функции и обекти, като масиви, низове и регулярни изрази, които го превръщат в мощен и гъвкав език за уеб разработка. Той също така разполага с голяма екосистема от библиотеки и рамки на трети страни, като React, Angular и Vue.js, които осигуряват допълнителна функционалност и инструменти за изграждане на сложни уеб приложения.

### 4.14 Razor

Razor е синтаксис за създаване на динамични уеб страници в ASP.NET Синтаксисът Razor е синтаксис за маркиране, който комбинира HTML и код от страна на сървъра, позволявайки на разработчиците да създават динамични уеб страници, които могат да се изпълняват на сървъра.

Razor бива въведен в ASP.NET MVC 3 като алтернатива на традиционния синтаксис на ASP.NET Web Forms. Той е проектиран така, че да бъде прост и интуитивен, позволявайки на разработчиците да пишат чист и поддържан код. Синтаксисът Razor поддържа различни езици за програмиране, включително C#, Visual Basic и F#.

Синтаксисът на Razor позволява на разработчиците да вграждат код от страна на сървъра в HTML, като използват символа "@". Този код може да включва C# изрази, оператори и структури за управление, като например if/else оператори и цикли. Синтаксисът на Razor също така предоставя функции като оформления, частични изгледи и компоненти на изгледи, които позволяват на разработчиците да създават многократно използваем и модулен код за своите уеб страници.

Едно от основните предимства на Razor е, че позволява на разработчиците да пишат код от страна на сървъра директно в своите HTML маркировки, което улеснява създаването на динамични уеб страници и избягва нуждата от излишни файлове с код. Razor също така осигурява силна проверка на типовете и други функции за безопасност, които помагат за намаляване на грешките и подобряване на цялостното качество на кода.

### 4.15 Bootstrap Framework

Bootstrap е популярна рамка с отворен код, която се използва за създаване на адаптивни уеб проекти, ориентирани към мобилните устройства. Тя е създадена от разработчиците на Twitter Марк Ото и Джейкъб Торнтън и е пусната през 2011 г.

Bootstrap предоставя колекция от предварително създадени стилове CSS и компоненти JavaScript, които разработчиците могат да използват за бързо и лесно създаване на уеб страници и уеб приложения. Рамката е проектирана така, че да бъде гъвкава и приспособима, което позволява на разработчиците да създават свои собствени уникални дизайни, като същевременно използват основната функционалност на рамката.

Bootstrap включва адаптивна решетъчна система, която позволява на разработчиците да създават оформления, които се адаптират към различни размери на екрана и устройствата. Тя включва също така редица компоненти на потребителския интерфейс, като навигационни менюта, форми, бутони, сигнали и модалности, което улеснява създаването на последователни и визуално привлекателни интерфейси.

Bootstrap е съвместим с всички съвременни уеб браузъри и е проектиран да работи добре с други фронт-енд технологии, като например рамки на JavaScript като React и Angular. Той има голяма и активна общност от разработчици, които допринасят за развитието му и осигуряват поддръжка и ресурси за другите, които го използват.

### 4.16 NUnit

## 5. Изисквания

### 5.1 Функционални изисквания

Изчерпателен списък на всички

функции и характеристики, които софтуерът трябва да

притежава.

### 5.2 Нефункционални изисквания

Изисквания, свързани с

производителност, сигурност, мащабируемост, използваемост и

др.

### 5.3 Бизнес изисквания

Целите на проекта от гледна точка на

бизнеса и потребителите.

### 5.4 Системни изисквания

Хардуерни и софтуерни изисквания за

внедряване и работа на системата.

## 6. Архитектурна документация

### 6.1 Архитектурен дизайн:

Описание на софтуерната архитектура,

включително структурни компоненти, като слоеве, модули, бази

данни и интерфейси.

#### 6.1.1 Data Layer

##### Moдели

###### ЕnumExtensions

###### GenderEnum

###### Pet

###### PetAgeEnum

###### PetTypeEnum

###### PublicOffer

###### RoleEum

###### SelectOption

###### Town

###### User

###### UserRequest

##### DbContexts и помощни класове

###### IDbWithNav

###### IDbWithoutNav

###### PetDbContext

###### PublicOfferDbContext

###### TownDbContext

###### UserDbContext

###### UserRequestsDbContext

###### ConnectionString

###### PetExchangeDbContext

#### 6.1.2 Business Layer

###### PetService

###### PublicOfferService

###### TownService

###### UserRequestsService

###### UserService

#### 6.1.3 Test Layer

Приложението се тества с мениджър клас, която се компилира и извършва преди всеки тест. Служи за зареждането на тестовата база данни (в паметта) и други методи за приготовление.

###### Пример за тестов мениджър

###### Пример за тест

#### 6.1.4 ASP.NET MVC (Presentation Layer)

### 6.2 База данни:

Посочва основната софтуерна система, която управлява

взаимодействията между потребителите и базата данни.

Примери са MySQL, PostgreSQL, Oracle, Microsoft SQL Server и

други.

● Дефинира структурата на базата данни, включваща

таблици, атрибути (колони), типове данни и връзки между

таблиците /посочва графично изображение/.

● Посочва езикът, който се използва за извличане и управление

на данни в базата данни.

● Диаграми: архитектурата на базата данни и взаимодействието

между компонентите (UML диаграми).

## 7. Документация на кода

### 7.1 API документация

Описание на интерфейсите, функциите и

методите, които могат да бъдат използвани от други части на

системата или външни приложения.

● Примерни кодови фрагменти: Примери за използването на

ключови функции и модули.

## 8. Потребителска документация

● Ръководство за потребителя: Описание на това как

потребителите могат да използват системата. Това може да

включва инструкции за инсталиране, работа с потребителския

интерфейс и справяне с често срещани проблеми.

● Ръководство за администратора: Описание на това как системата

се администрира и поддържа, включително настройки,

конфигурации и управление на базата данни.

● Тестова документация: - компонентно тестване на поне една

структурна единица. Резултати от тестовете: Доклад за проведените тестове, включително грешки и корекции. Грешки при круда

## 9. Авторски права

Информация за собствеността и правата върху кода и

използваните библиотеки.

# Заключение

Заключението съдържа изводи и предложения за доразвиване на проекта и

възможностите за неговото приложение.

Обобщава постигнатото в дипломния проект, представя възможностите за бъдещо

развитие и усъвършенстване, както и възможностите за приложение на създаденото

приложение.

# Списък на използвана литература

Списъкът с използваната литература включва цитираната и използвана в

записката на дипломния проект литература. Започва на отделна страница от основния

текст. При имената на авторите първо се изписва фамилията. Всички описания в

списъка с използваните източници трябва да са подредени по азбучен ред според

фамилията на първия автор на всяка публикация.

Включва цитираната и използвана в дипломния проект литература. Започва на

отделна страница от основния текст. При имената на авторите първо се изписва

фамилията. Всички описания в списъка с използваните източници трябва да са

подредени по азбучен ред според фамилията на първия автор на всяка публикация.

# Приложения

Да са номерирани (Приложение 1, Приложение 2 ...). Приложенията съдържат

документация, която не е намерила място в текста поради ограниченията в обема й

или за по-добра прегледност и подредба. В текста трябва да има препратка към всяко

приложение.

Вмъкнатите изображения, фигури, диаграми и други, в документацията, да бъдат

подходящо форматиране, като разстоянието на текста преди и след тях да бъде не

повече от 6 пункта. Да няма едно изображение на цяла страница.

Приложенията съдържат документация, която не е намерила място в текста поради

ограниченията в обема й или за по-добра прегледност и подредба. В текста трябва да

има препратка към всички приложения.

Notes

● Коментари в кода: Добра практика е да има смислени коментари

в кода, които обясняват какво правят различните части от

програмата.