МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

**ПРОФЕСИОНАЛНА ГИМНАЗИЯ “ГЕН. ВЛАДИМИР ЗАИМОВ” гр. СОПОТ**

4330 гр. Сопот, ул. ”Иван Вазов” №1, тел./факс: /03134/ 83-31, 83-32, e-mail: [pgzaimov@yahoo.com](mailto:pgzaimov@yahoo.com)

**ДИПЛОМЕН**

**ПРОЕКТ**

**Тема: Разработка на уеб платформа за изготвяне и следене на хранителен режим**

*Ученик: Надежда Любенова Антонова*

***Професия:*** *код 481020 „Системен програмист“*

***Специалност:*** *код 4810201 „Системно програмиране“*

***Консултант:*** *Живко Маджев*

Сопот, 2023 г.

**ЗАДАНИЕ ЗА ДИПЛОМЕН ПРОЕКТ**

**Тема: РАЗРАБОТКА НА УЕБ ПЛАТФОРМА ЗА ИЗГОТВЯНЕ И СЛЕДЕНЕ НА**

**ХРАНИТЕЛЕН РЕЖИМ**

**Изисквания за разработката на дипломния проект:**

Да се проектира и реализира уеб платформа с трислойна MVC архитектура, която да организира и управлява изготвянето на хранителен режим за потребители.

Уеб платформата да се състои от публична част, която дава информация за дейността и целта на системата, както и за треньорите, регистрирани в нея.

Потребителите могат да се свързват с даден треньор от платформата, с когото желаят да работят. След регистрация и одобрение на заявката от треньора, потребителите могат да преглеждат изготвения за тях индивидуален режим, както и да следят своя прогрес.

Треньорите могат да изготвят хранителен режим за всеки потребител в зависимост от техните цели и възможности.

Системният администратор се грижи за управлението на треньорите в уеб платформата, като може да добавя, премахва или редактира информация за треньор.

Дизайнът на потребителския интерфейс, както и използваните технологии трябва да бъдат съобразени с функционалността на приложението.

**Съдържание**

**Увод (Въведение) 5**   
 **ПЪРВА ГЛАВА.** Проучване…………………………………………………………..

1.1. Предпоставка за създаване 6

1.2. Анализ на съществуващи решения 6

1.3. Цел на дипломната работа 7

**ВТОРА ГЛАВА.** Проектиране……………………………..

2.1. Използвани технологии и подготовка на работната среда

2.1.1. Microsoft Visual Studio

2.1.2. C#(C Sharp).....................................................................................................

2.1.3. Трислойна архитектура на приложението MVC

2.1.4. ASP.NET Core

2.1.5. HTML…………………………………………………………………….

2.1.6. CSS……………………………………………………….……….

2.1.7. JavaScript

2.1.9. База от данни

2.2. Изисквания към софтуерния продукт

2.2.1. Софтуерни

2.2.2. Интерфейсни

2.2.3. Хардуерни

**ТРЕТА ГЛАВА.** Описание на реализирания проект

3.1. Структура на базата данни………………………………………….......................

3.2. Структура на MVC приложението – Controllers Class Diagram………………

3.2.1. Описание и конфигуриране на моделите за работа и обмен…..………..

3.2.2. Описание на CRUD операциите - особености на отделни ACTION…...

3.2.3. Работа с View моделите........................................

3.2.4. Описание на Razor Pages и Razor Views (TagHelpers и HtmlHelpers)......

3.3. Как работи схемата Route Engine………………………………………...

3.3. Как работи шаблона на Contoller с GET и POST заявки към Actions….

**Приложение**

Индекс на фигурите

**Увод**

Плановете за хранителен режим са предназначени да осигурят енергиино ефективен здравословен комфорт и спестяване на време за човека.

Здравето на човека като състояние на пълно физическо, психично и социално благополучие, е резултат от комплексното въздействие на екологични, социалноикономически, медицински и психо-емоционални фактори. Върху човека въздействат много фактори, като някои от тях укрепват здравето, а други го увреждат (замърсена жизнена среда, шум, нервно-психично напрежение, неправилно хранене, намалена двигателна активност, злоупотреба с алкохол и никотин). Човекът трябва не само да се предпазва от неблагоприятните фактори на природната и социалната среда, но да познава добре и да се възползва оптимално от т.нар. здравословни фактори на околната среда. Здравето и болестта са две форми на съществуване на човека, но между тях няма резки граници

Диета е контролирано приемане на храни и течности с цел постигане на определени резултати, измервани в килограми, мускулна маса, мазнини и др. С други думи това е хранителен режим. Управлението на пропорцията между белтъчините, мазнините и въглехидратите, които се приемат с храната е от основно значение за постигане на максимално добри резултати при управление на даден хранителен режим или при определено коригиране на килограми, поради заболяване, практикуване на спорт или някаква друга необходимост.

Постигането и поддържането на здраве е продължаващ, непрекъснат процес, който се повлиява от еволюцията на знанията за здравословен начин на живот и практиките за грижата за здравето, личните усилия и медицинската помощ.

**ГЛАВА 1.**

**ПРОУЧВАНЕ**

**1.1. Предпоставка за създаване на продукта**

Основната дейност на уеб платформата е свързан с изготвянето и следене на хранителен режим. Всеки режим е индивидуален, направен по строго определени стъпки. Професионалните служители на платформата са коректни към всеки клиент.

Чрез комбинация от психология, технология и човешки коучинг, платформата помога на хиляди потребители да постигнат личните си цели за здраве и уелнес. Потребителите са съчетани с един от многото обучени треньори, за да им помогне да ги напътстват по пътя им към здравето и да осигурят необходимата подкрепа за постигане на целите им.

**1.2. Анализ на съществуващи решения и реализации**

Реших да създам своя уеб платформа, където да помогна на хората да станат по-добри версии на себе си, променяйки начина си на живот. Професионалните ни треньори осигуряват на клиентите удобство и доверие.

**1.3. Цел на дипломната работа**

Целта на този проект е да се разработи уеб платформа за изготвяне и следене на хранителен режим, където потребителите могат да се свързват с даден треньор от платформата, с когото желаят да работят**.** След регистрация и одобрение на заявката от треньора, потребителите могат да преглеждат изготвения за тях индивидуален режим, както и да следят своя прогрес.

След избора на треньор и одобрението му, ще се направи разговор с клиента, на който ще снема пълен антропометричен, здравен и хранителен статус. Също така хранителният режим трябва да е съобразен с неговата възраст, тегло, физическо и здравословно състояние, както и с начина му на живот. На базата на информацията, която получи, ще създаде специално за клиента персонализиран хранителен режим, който ще отговаря на нуждите на организма му и ще е спрямо неговите вкусови предпочитания. Персонално изработеният хранителен режим включва най-необходими покупки, разнообразни варианти на всяко хранене (най-често между 4 и 6) и междинни хранения (над 5), базирани на предварително попълнен въпросник, в който потребителите уточняват предпочитани храни, както и такива които искат да бъдат избегнати на всяка цена.  
Суплементарен план (хранителни добавки), ако имат желание да използват такива.

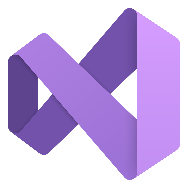
Хранителният режим може да бъде допълнително съобразен с предпочитани брой хранения дневно, работотно време и други индивидуални фактори. Потребителите ще имат възможността да отбелязват всяко хранене, изразходена енергия.

Химичен състав на храните – кратко описание на всяка една храна калории съдържа. Консултации за всяка седмица от режимите, за да се дискутира всичко необходимо и да се направят нужните промени по режимите, ако такива са необходими – на базата на данните, които се предоставят и това как се чувстват.

**ГЛАВА 2.**

**ПРОЕКТИРАНЕ НА СТРУКТУРАТА НА**

**2.1. Използвани технологии и подготовка на работната среда /**технологии: Visual Studio, MVC, ASP.Net Core; средства: C#, HTML, CSS, JavaScrip**/**

 **2.1.1. Microsoft Visual Studio** е мощна [интегрирана среда за разработка](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%B0%D0%BD%D0%B0_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D0%B7%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0" \o "Интегрирана среда за разработка) на софтуерни приложения за [Windows](https://bg.wikipedia.org/wiki/Windows) и за платформата [.NET Framework](https://bg.wikipedia.org/wiki/.NET_Framework). Използва се за разработка на конзолни и [графични потребителски интерфейс](https://bg.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D0%BD_%D0%BF%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%B8%D1%82%D0%B5%D0%BB%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81" \o "Графичен потребителски интерфейс) приложения, както и Windows Forms или WPF приложения, [уеб сайтове](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1%D1%81%D0%B0%D0%B9%D1%82" \o "Уебсайт), [уеб приложения](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5" \o "Уеб приложение) и [уеб услуги](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0" \o "Уеб услуга) на всички поддържани платформи от Microsoft Windows, [Windows Mobile](https://bg.wikipedia.org/wiki/Windows_Mobile), Windows CE, .NET Framework, .NET Compact Framework и Microsoft Silverlight.

Visual Studio предоставя мощна интегрирана среда за писане на код, компилиране, изпълнение, дебъгване (както за високо така и за машинно ниво), тестване на приложения, дизайн на потребителски интерфейс (форми, диалози, уеб страници, визуални контроли и други), моделиране на данни, моделиране на класове, изпълнение на тестове, пакетиране на приложения и стотици други функции.

Visual Studio поддържа различни [езици за програмиране](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D1%86%D0%B8_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5" \o "Езици за програмиране) (например: [C#](https://bg.wikipedia.org/wiki/C_Sharp), VB.NET, [C](https://bg.wikipedia.org/wiki/C_(%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5))/[C++](https://bg.wikipedia.org/wiki/C%2B%2B), [F#](https://bg.wikipedia.org/w/index.php?title=F_Sharp&action=edit&redlink=1), [XML](https://bg.wikipedia.org/wiki/XML)/[XSLT](https://bg.wikipedia.org/wiki/XSLT), [HTML](https://bg.wikipedia.org/wiki/HTML)/[XHTML](https://bg.wikipedia.org/wiki/XHTML), [JavaScript](https://bg.wikipedia.org/wiki/JavaScript), [CSS](https://bg.wikipedia.org/wiki/CSS) и други) и различни технологии за разработка на софтуер (Win32, COM, [ASP.NET](https://bg.wikipedia.org/wiki/ASP.NET), ADO.NET Entity Framework, Windows Forms, WPF, Silverlight и още десетки други Windows и .NET технологии).

**2.1.2. C#** е обектно ориентиран език за програмиране, разработен от Microsoft като част от софтуерната платформа .NET. Стремежът още при създаването на C# езика е бил да се създаде прост, модерен, обектно ориентиран език с общо предназначение. Основа за C# са C++, Java и донякъде езици като Delphi, VB.NET и C. Той е проектиран да балансира мощност (C++) с възможност за бързо разработване (Visual Basic и Java). Те представляват съвкупност от дефиниции на класове, които съдържат в себе си методи, а в методите е разположена програмната логика – инструкциите, които компютърът изпълнява. Програмите на C# представляват един или няколко файла с разширение .cs, в които се съдържат дефиниции на класове и други типове. Тези файлове се компилират от компилатора на C# до изпълним код и в резултат се получават асемблита – файлове със същото име, но с различно разширение (.exe или .dll).

**2.1.3. Трислойна архитектура MVC - „Model-view-controller“.** В [софтуерното инженерство](https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80%D0%BD%D0%BE%D1%82%D0%BE_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D1%81%D1%82%D0%B2%D0%BE" \o "Софтуерното инженерство) **многослойната архитектура** (наричана още N-слойна архитектура) е [архитектура](https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0" \o "Софтуерна архитектура) от тип [клиент-сървър](https://bg.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D1%8A%D1%80%D0%B2%D1%8A%D1%80" \o "Клиент-сървър), в която [интерфейсът](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)" \o "Интерфейс (програмиране)), обработката на приложения и съхранението и обработката на [данни](https://bg.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D0%B8" \o "Данни) са логически разделени на отделни модули. Най-разпространената форма на многослойна архитектура е трислойната архитектура.

Многослойната архитектура осигурява модел, по който разработчиците могат да създават гъвкави приложения, които могат да бъдат използвани многократно. При разделянето на едно приложение на слоеве, разработчиците имат възможността да добавят или променят отделен слой, вместо да преработват цялото приложение.

Всички съвременни бизнес приложения имат достъп до различен вид данни като част от тяхната основна функционалност. След като [релационните бази данни](https://bg.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B8_%D0%B1%D0%B0%D0%B7%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8" \o "Релационни бази данни) добиха широка популярност, IT индустрията се преориентира от еднослоен модел (mainframe) към модела **[клиент-сървър](https://bg.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D1%8A%D1%80%D0%B2%D1%8A%D1%80" \o "Клиент-сървър)**. При него са налице **клиент**, който включва презентационна логика и голямата част от бизнес логиката на приложението, както и **сървър** включващ в себе си хранилище за данни (data storage) и част от бизнес логиката под формата на готови и съхранени заявки към базата данни.

Трислойната архитектура обикновено се състои от презентационен слой ([потребителски интерфейс](https://bg.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%B8%D1%82%D0%B5%D0%BB%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81" \o "Потребителски интерфейс)) на най-високо ниво в приложението. Той служи за прякото взаимодействие с потребителя и изпращането на заявки към бизнес слоя. Не трябва да има директна връзка между този слой и слоят за [бази данни](https://bg.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8" \o "Бази данни). Бизнес слоят служи за обработка на данните и работните процеси. Той комуникира както с презентационния слой, така и с базите данни. Слоят за данни комуникира само с бизнес логиката и служи за съхранение данни и тяхното използване. Често отделните модули се разработват и на различни платформи. Разработена е от Джон Дж. Донован, в неговата компания OEC (Open Environment Corporation), основана в [Кеймбридж](https://bg.wikipedia.org/wiki/%D0%9A%D0%B5%D0%B9%D0%BC%D0%B1%D1%80%D0%B8%D0%B4%D0%B6_(%D0%9C%D0%B0%D1%81%D0%B0%D1%87%D1%83%D0%B7%D0%B5%D1%82%D1%81)" \o "Кеймбридж (Масачузетс)), Масачузетс.

Освен стандартните предимства на модулния [софтуер](https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80" \o "Софтуер), трислойната архитектура е проектирана да позволява да бъде променен или заменен независимо, който и да е от трите слоя, за да отговаря на промени в изискванията или технологиите. Например, промяна на [операционната система](https://bg.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0" \o "Операционна система) на презентационния слой би засегнала само кода за [потребителският интерфейс](https://bg.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%B8%D1%82%D0%B5%D0%BB%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81" \o "Потребителски интерфейс).

В повечето случай, [потребителският интерфейс](https://bg.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%B8%D1%82%D0%B5%D0%BB%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81" \o "Потребителски интерфейс) работи на десктоп система и използва стандартен графичен интерфейс, функционалната логика за обработка на данните, която може да се състои от един или повече отделни модула, работещи на десктоп система или сървърно приложение, и РСУБД (релационните системи за управление на бази данни) работещи на сървър [база данни](https://bg.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8" \o "База данни) или [мейнфрейм](https://bg.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B9%D0%BD%D1%84%D1%80%D0%B5%D0%B9%D0%BC" \o "Мейнфрейм), който съдържа логиката за съхранението и обработката на базите данни. Междинният слой може самият да бъде многослоен. В такъв случай цялостната архитектура се нарича N-слойна архитектура.

Трислойната архитектура се състои от следните три слоя:

В областта на уеб разработката трислойната архитектура често се използва в уеб сайтове, най-често т.нар. „електронни магазини“, които се състоят от 3 слоя:

1. Front-end слой (Клиентски слой) – този слой, е съдържанието, което се възпроизвежда от [уеб браузърът](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D1%8A%D1%80" \o "Уеб браузър) и достига до крайния потребител. То може да бъде генерирано статично или динамично.
2. Среден слой – той реализира логиката на самото уеб приложение. Това се осъществява със [сървър](https://bg.wikipedia.org/wiki/%D0%A1%D1%8A%D1%80%D0%B2%D1%8A%D1%80" \o "Сървър), който обработва и генерира динамично съдържание. Този сървър може да е с платформа един от следните езици – [Ruby on Rails](https://bg.wikipedia.org/wiki/Ruby_on_Rails), Java EE, [ASP.NET](https://bg.wikipedia.org/wiki/ASP.NET), [PHP](https://bg.wikipedia.org/wiki/PHP), ColdFusion, [Perl](https://bg.wikipedia.org/wiki/Perl), Node.js.
3. Back-end слой – той представлява [база-данни](https://bg.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8" \o "База данни), която се състои, както от самата информация в таблици, така и от система за управлението и (RDBMS).

**2.1.4. ASP.NET Core** е безплатна софтуерна рамка за уеб разработка, с отворен код. Също така тя се явява и следващата стъпка в еволюцията ASP.NET. Тя е разработена съвместно от Microsoft и общността, която е събрала през годините на своето развитие. ASP.NET Core е модуларна софтуерна рамка, която може да върви както на пълната .NET рамка, така и на крос-платформената .NET Core. Въпреки, че е нова софтуерна рамка, изградена върху нов web stack, тя има висока степен на съвместимост с ASP.NET MVC.

Спрямо своите предшественици ASP.NET Core поддържа нова функция – т.нар. „side by side versioning”. При нея различни приложения, които използват една и съща машина, могат да таргетират различни версии на ASP.NET Core, в зависимост от версиите (и нуждите) си. Това не е възможно с по-стари издания на ASP.NET.

**2.1.5. HTML** е основният маркиращ език за описание и дизайн на уеб страници. HTML е стандарт в интернет, а неговите стандарти се определят от международния консорциум W3C. Текущата версия на стандарта е HTML 5.0 (от 28 октомври 2014 г.), а предходната стабилна версия е HTML 4.1.

Описанието на документа става чрез специални елементи, наречени HTML елементи или техните маркери, които се състоят от тагове и съответстващите етикети (HTML tags) и ъглови скоби (като например елемента <html>). HTML елементите са основната градивна единица на кода, който изграждат уеб страниците. Чрез тях се форматира, графично оформя текста и неговите отделните части в рамките на една уеб страница, като например заглавия, цитати, текстови раздели, хипертекстови препратки и т.н. Най-често HTML елементите са групирани по двойки <h1> и </h1>.

В повечето случаи HTML кодът е написан в текстови редактори, с файлов формат .html, .htm, dhtml и се качва и хоства на [сървъри](https://bg.wikipedia.org/wiki/%D0%A1%D1%8A%D1%80%D0%B2%D1%8A%D1%80" \o "Сървър), които са онлайн в интернет или са част от [www](https://bg.wikipedia.org/w/index.php?title=Www&action=edit&redlink=1) мрежата. Тези .html файлове съдържат програмно на таговете на HTML и текстово съдържание със маркери и коментари – също инструкции за [браузъра](https://bg.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D1%8A%D1%80" \o "Браузър), за това какъв точно тип е .html страницата, а също за това как да се показва текстът, особено що се отнася до езиковите характеристики. За да се илюстрира как се включва текст в HTML код: <маркер> Някакъв текст. </край на маркера>. [уеб браузърите](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D1%8A%D1%80" \o "Уеб браузър) са програмирани от своя страна така, в повечето случаи, макар че някои браузъри могат да имат съответно проблеми на версията, за да могат да прочетат HTML документите и да ги покажат на екрана като уеб страници. Браузърите не показват самите HTML тагове, освен ако не се отиде в менюто за да се направи това, така че те „интерпретират“ (тоест [парсват](https://bg.wikipedia.org/w/index.php?title=Parsing&action=edit&redlink=1" \o "Parsing (страницата не съществува))) съдържанието на страницата като код и текст за да могат след работа на процесора да покажат желаното уеб-съдържание.

Основното предимство на HTML е, че уеб-страниците, които са го включват в кода си, могат да се разглеждат чрез показването им от браузъра на екрана на повечето устройства. Уебстраницата може да има дизайн, който дори изглежда с добър дизайн с помощта на [CSS](https://bg.wikipedia.org/wiki/CSS) или „правилно оформен“ (например с помощта на [C#](https://bg.wikipedia.org/wiki/C)), както върху монитора на [персоналния компютър](https://bg.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D0%B5%D0%BD_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8E%D1%82%D1%8A%D1%80" \o "Персонален компютър), но също и върху миниатюрния дисплей на [пейджър](https://bg.wikipedia.org/w/index.php?title=%D0%9F%D0%B5%D0%B9%D0%B4%D0%B6%D1%8A%D1%80&action=edit&redlink=1" \o "Пейджър (страницата не съществува)) или дисплея на [мобилен телефон](https://bg.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D0%B8%D0%BB%D0%B5%D0%BD_%D1%82%D0%B5%D0%BB%D0%B5%D1%84%D0%BE%D0%BD" \o "Мобилен телефон).

HTML може да прикрепя скриптове писани на езици като [JavaScript](https://bg.wikipedia.org/wiki/JavaScript), който е помощен за HTML, и това променя поведението на дадена уеб страница. [Cascading Style Sheets](https://bg.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) се използват, като това се прави за да се определя изгледа и оформлението на текста и други включени в страницата изображения и илюстриращи материали. [World Wide Web Consortium](https://bg.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C) поддържа както HTML, така и CSS, и насърчава използването на CSS в HTML страниците още от 1997. Това допринася за разделяне съдържанието и структурата на уеб страниците от тяхното визуално представяне.

**2.1.5. CSS** е език за програмиране и също описание на уеб дизайн програмни стилове (език за стилови файлове, style sheet language) – използва се основно за описание на онлайн представянето на уеббазиран документ, който написан на език за маркиране. Най-често се използва допълнително към чистия HTML, но се прилага и върху XML уебстраници и документи. Спецификацията на CSS официално се поддържа от W3C.

CSS още в началото на развитието на www започва да се добавя към стандартния HTML с цел да бъдат разделени съдържанието и структурата на уеб страниците отделно от тяхното визуално представяне. Преди стандартите за CSS, установени от W3C през 1995 г., съдържанието на сайтовете и стила на техния дизайн са писани в една и съща HTML страницата. В резултат на това HTML кодът се превръща в сложен и нечетлив, а всяка промяна в проекта на даден сайт изисквала корекцията да бъде нанасяна в целия сайт страница по страница. Използвайки CSS, настройките за форматиране могат да бъдат поставени в един-единствен файл и тогава промяната ще бъде отразена едновременно на всички страници, които използват този CSS файл.  
  
 **2.1.6. JavaScrip** е интерпретируем език за програмиране, разпространяван с повечето уеб браузъри. Поддържа обектно ориентиран и функционален стил на програмиране. Създаден е в Netscape през 1995 г. Най-често се прилага към HTML-а на интернет страница с цел добавяне на функционалност и зареждане на данни. Може да се ползва също за писане на сървърни скриптове JSON, както и за много други приложения. JavaScript не трябва да се бърка с Java, съвпадението на имената е резултат от маркетингово решение на Netscape. Javascript е стандартизиран под името EcmaScript.

JavaScript е програмен език, който позволява динамична промяна на поведението на браузъра в рамките на дадена HTML страницата. JavaScript се зарежда, интерпретира и изпълнява от уеб браузъра, който му осигурява достъп до Обектния модел на браузъра. JavaScript функции могат да се свържат със събития на страницата (например: движение/натискане на мишката, клавиатурата или елемент от страницата, и други потребителски действия). JavaScript е най-широко разпространеният език за програмиране в интернет. Прието е JavaScript програмите да се наричат скриптове

**2.1.7. База от данни** (БД, още база от данни) представлява колекция от логически свързани данни в конкретна предметна област, които са структурирани по определен начин. В първоначалния смисъл на понятието, използван в компютърната индустрия, базата от данни се състои от записи, подредени систематично, така че компютърна програма да може да извлича информация по зададени критерии.

Поддръжката на база от данни се осъществява от т.нар. [система за управление на бази от данни](https://bg.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B7%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B8_%D0%BE%D1%82_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8" \o "Система за управление на бази от данни) (СУБД).

Система за управление на бази данни е компютърно приложение ([софтуер](https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80" \o "Софтуер)) създадено за комуникация между потребителя, други приложения, както и други БД, с цел да се сравнят и анализират данни. Общото специфично предназначение на СУБД е да позволи определянето, създаването, заявки, актуализацията и администрирането на бази данни. Добре известни СУБД включват MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, SAP HANA, и IBM DB2. Бази данни не са съвместими с различните СУБД, за това различните СУБД работят със стандартни като SQL и ODBC или JDBC, за да позволи на всяко приложение да работи с различни СУБД, а така и с различни БД. Управлението на БД често се избира от модела им, които те подкрепят. Най-използвани системи от бази данни от 1980 г. насам са всички поддържани релационния модели на езика SQL. Често срещано е СУБД да се нарича само „база данни“.

2.2. Изисквания към софтуерният продукт

**Windows Server 2019** е деветата версия на операционната система Windows Server от Microsoft, като част от семейството операционни системи Windows NT. Това е втората версия на сървърната операционна система, базирана на платформата Windows 10, след Windows Server 2016. Windows Server е операционната система, която свързва локалните среди с услуги на Azure, което разрешава хибридни сценарии и максимизират съществуващите инвестиции, включително:

* Уникални хибридни възможности с Azure за разширяване на вашия център за данни и максимизиране на инвестициите
* Усъвършенствана многослойна защита за подобряване на положението със защитата
* По-бързи иновации за приложения, което позволява на разработчиците и ИТ специалистите да създават нови приложения и да ги модернизират, както и
* Несравнима хиперконвергентна инфраструктура за развиване на вашата инфраструктура на център за данни

**Internet Information Services(IIS)** е пакет от сървърни услуги, включващ уеб сървър, пощенски сървър и сървър за споделяне на файлове. Създаден от Microsoft, той е конкурентът на Apache и sendmail при Unix системите. Най-важния от горепосочените компоненти е уеб сървърът. Той поддържа всички стандарти за уеб като HTTP 1.0/1.1. За разлика от Apache, той може да работи само под Windows. IIS Express, олекотена (4,5–6,6 MB) версия на IIS, се предлага като самостоятелен безплатен сървър и може да се инсталира на Windows XP със Service Pack 3 и следващите версии на Microsoft Windows. IIS 7.5 Express поддържа само HTTP и HTTPS протоколи. Той е преносим, съхранява конфигурацията си за всеки потребител, не изисква административни привилегии и се опитва да избегне конфликт със съществуващи уеб сървъри на същата машина. IIS Express може да бъде изтеглен отделно или като част от WebMatrix или Visual Studio 2012 и по-нови версии. По подразбиране IIS Express обслужва само локален трафик.

**Microsoft SQL Server** е система за управление на релационни бази данни, разработена от Microsoft. Като сървър на база данни, това е софтуерен продукт с основна функция за съхраняване и извличане на данни, както се изисква от други софтуерни приложения - които могат да работят или на същия компютър, или на друг компютър в мрежа (включително Интернет). Microsoft SQL Server 2019 е предназначено за фирми и организации от всякакъв мащаб – малките, средни и корпоративни предприятия могат да избират измежду различни издания на SQL Server, подходящи за техните нужди.

**SQL Server Management Studio**(**SSMS**) – (Студио за управление на [SQL Server](https://bg.wikipedia.org/wiki/Microsoft_SQL_Server)) е софтуерно приложение, стартирано за първи път с [Microsoft](https://bg.wikipedia.org/wiki/Microsoft) [SQL Server 2005](https://bg.wikipedia.org/w/index.php?title=SQL_Server_2005&action=edit&redlink=1), което се използва за конфигуриране, управление и администриране на всички компоненти в SQL Server. То съчетава широка група от графични инструменти с голям брой текстови редактори, осигуряващи на разработчиците и администраторите всички нива на достъп до сървъра. Водещ елемент в SSMS е [Object Explorer](https://bg.wikipedia.org/w/index.php?title=Object_Explorer&action=edit&redlink=1), който позволява на потребителя да търси, избира и да работи с всеки от обектите на сървъра. Приложението има и „експресна“ версия, която може да бъде изтеглена безплатно.

**Azure Web Apps** е базирана на облачни изчисления платформа за хостване на уебсайтове, създадена и управлявана от Microsoft. Това е платформа като услуга (PaaS), която позволява публикуване на уеб приложения, работещи на множество рамки и написани на различни езици за програмиране (.NET, node.js, PHP, Python и Java), включително собствени на Microsoft и такива на трети страни. Той предлага автоматично мащабиране и висока достъпност, поддържа както Windows, така и Linux и позволява автоматизирани внедрявания от GitHub, Azure DevOps или всяко Git repo.

**Статичният IP адрес** е IP адрес, който не се променя с течение на времето. IP (интернет протокол) адресите са цифрови знаци, които позволяват пакети с данни да бъдат изпращани и получавани от нашите мрежи и устройства. Повечето IP адреси са динамични, което означава, че се променят от време на време. Статичният IP, от друга страна, винаги е една и съща последователност от числа. Статичните IP адреси обикновено се използват от сървъри или друго важно оборудване. IP адресът е уникален номер, присвоен на всяко устройство в TCP/IP мрежа. IP адресите идентифицират компютри и устройства и им позволяват да комуникират помежду си.

**Система за имена на (DNS)** представлява [разпределена база от данни](https://bg.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8_%D0%B1%D0%B0%D0%B7%D0%B8_%D0%BE%D1%82_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8) за компютри, услуги или други ресурси свързани към [интернет](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) или частни мрежи, с чиято помощ се осъществява преобразуването на [имената](https://bg.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%B9%D0%BD) на хостовете в [IP адреси](https://bg.wikipedia.org/wiki/IP_%D0%B0%D0%B4%D1%80%D0%B5%D1%81). Това улеснява работата на потребителите на интернет услуги. Вместо да въвежда IP адрес (комбинация от цифри), за да достигне до даден ресурс в мрежата, потребителят може просто да въведе неговото име (домейн).

Информацията за IP адресите и имената на домейни се съхранява на DNS сървърите. DNS е разпределена дървовидна система от обвързани чрез логическа йерархия [сървъри](https://bg.wikipedia.org/wiki/%D0%A1%D1%8A%D1%80%D0%B2%D1%8A%D1%80). В основата на тази структура са сървърите, съхраняващи:

* [*домейни от първо ниво*](https://bg.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%B9%D0%BD%D0%B8_%D0%BE%D1%82_%D0%BF%D1%8A%D1%80%D0%B2%D0%BE_%D0%BD%D0%B8%D0%B2%D0%BE) (*top-level domains*) – например [.com](https://bg.wikipedia.org/wiki/.com), [.org](https://bg.wikipedia.org/wiki/.org), [.edu](https://bg.wikipedia.org/wiki/.edu) и т.н. и
* множество *домейни на държавно ниво* (*country-level domains*) – [.bg](https://bg.wikipedia.org/wiki/.bg) (за България), [.fi](https://bg.wikipedia.org/wiki/.fi) (за Финландия), [.fr](https://bg.wikipedia.org/wiki/.fr) (за Франция) и т.н.

Следващото ниво образуват *регистрираните домейни* (*registered domains*) – about.com, abv.bg, pirin.com и т.н. *Местните домейни* (*local domains*), наричани още *поддомейни* (*subdomains*), като compnetworking.about.com, sdyn.pirin.com, се определят и администрират от собствениците на съответните главни домейни. За разделяне на различните равнища се ползва точка

Дизайнът на потребителския интерфейс, както и използваните технологии са съобразени с функционалността на приложението. Уеб платформата се състои от публична част, в която са включени:

* ***Начална страница***, която представя обща информация за идеята на сайта. Обяснена е кратка информация относно това, какво представлява хранителният режим. Също така са представени някои от треньорите и техните професии. Накрая на страницата е показан пример с резултатите на клиент, който е пробвал някой от плановете.
* ***Страница ,,Контакти”-***Когато дадено лице иска да зададе своя въпрос му даден формуляр, който е с полета за имейл, парола, съобщение(въпроса, който иска да зададе), пол и държава.
* ***Страница ,,За нас”*** -в тази страница е представена подробна информация за екипа. За работата на екипа, целите.
* ***Страница ,,Треньори”-*** в тази страница е представен подробна информация за треньорите и техните програми.