ДИПЛОМЕН ПРОЕКТ

**професия код 481030 „Приложен програмист“   
специалност код 4810301 „Приложно програмиране“**

Кристина Владиславова Велинова , 12б , N15

**ТЕМА**

Разработка на Уеб сайт за цветарски магазин „Роза“

**Изготвил**: Кристина Велинова  **Рецензент**: Иван Василев

Перник, 2022 г.

Съдържание

[Увод 2](#_Toc98590299)

[Първа глава 4](#_Toc98590300)

[1. Проучване 4](#_Toc98590301)

[2. Трислойна архитектура 5](#_Toc98590302)

[3. Използвани технологии и среди 6](#_Toc98590303)

[3.1. C# език за програмиране 6](#_Toc98590304)

[3.1. Asp.Net Core MVC 7](#_Toc98590305)

[4. Bootstrap 10](#_Toc98590306)

[5. База данни 11](#_Toc98590307)

# Увод

Живеем във времена, в които технологиите са неизменна част от нашето ежедневие. Използваме ги постоянно, разговаряме лесно с хора, които са далеч от нас, работим от тях, търсим полезна информация в сайтове като „Google” , които улесняват много ежедневието ни, както и за много други неща, спрямо нуждите ни.

Забелязваме и нарастване на обвързаността между бизнеси и новите технологии. Организациите се стремят да се разрастват, а интернет, новите тенденции и уеб сайтовете помагат точно за това.

Зад всички уеб сайтове, които използваме, стоят програмисти. Всички сме наясно, че компютърното програмиране набира огромна популярност през последните години и дори десетилетия. Какво е важно да знаем за него?

Програмирането е компютърна наука и дял от математиката-информатиката, която се занимава с методите и средствата за създаване на компютърни програми.

Описано по начин, разбираем от всички може да се определи като „казване на компютъра какво да прави“ или по-точно програмистът предоставя набор от инструкции, които са написани на език за програмиране. Както при хората , така и при програмирането има различни езици. Някои от най-популярните езици за програмиране са C, C++ , C# , SQL , Java , JavaScript, Python и много други. Целта на компютърното програмиране е да създаде продукт, който може да бъде уеб страница, уебсайт, софтуерна програма, компютърна игра или дори изображение.

Защо е толкова важно да има хора, които да създават уеб сайтове?

Технологиите се развиват изключително бързо. Пандемията с „Ковид-19“ също допринесе за масовото използване на интернет и вече можем да правим почти всичко онлайн. Хората започнаха да работят от вкъщи през компютрите си, учениците и студентите започнаха да се обучават онлайн. Наскоро създадоха приложения, които да улеснят поръчването на храна до вкъщи, вече започнаха и доставки от големи вериги като „Кауфланд“ , „Лидл“ , „ДМ“ и много други магазини.

С днешната пандемия, работата стана много по-трудна и драстично намаля, за това онлайн сайтовете помагат на малките бизнеси да не умрат, а точно обратното – да се разраснат. Всичко е направено с цел улеснение за хората, спестяване на време както и една част от всичко за „предпазване“ от вируса.

Цели на проекта

Уеб приложенията станаха доста популярни и с помощта на програмисти, различни бизнеси превръщат работата си доста по-мащабна. Има много създадени уеб сайтове за дрехи например. Те улесняват хората със забързано ежедневие, които нямат време да обикалят по молове и магазини на място, да пазаруват онлайн. Това помага изключително много и на всеки друг бизнес. По този начин има много повече клиенти и поръчки.

Целта на моята дипломна работа е да разработя Уеб сайт за веригата ни цветарски магазини „Роза“. По този начин ще допринесем на потребителите с удобство и спестяване на време. Чрез нашият сайт, клиентите ни ще имат възможността да поръчат букети и цветя онлайн, без да се налага да търсят магазин. Също така в много градове в България все още нямаме обекти на място и понеже много често получаваме запитвания в социалните ни мрежи, решихме да пуснем доставки из цяла България. Това ще улесни потребителите ни да не се ангажират и да не отнемат от ценното си време в търсене на цветя или подарък за половинките си или пък за някой друг близък човек, а да могат да направят всичко много по-лесно и бързо през нашият уеб сайт.

За да постигнем всичко това трябва да изпълним следните задачи:

* Проучване на конкурентни сайтове и услугите, които предлагат те
* Да предоставим коректна информация за продуктите, които ще предлагаме
* Да създадем различни функционалности, като: форма за регистрация,
* Да се създаде приятен интерфейс
* Съвременен и адаптивен дизайн
* Лесна форма за поръчки

# Първа глава

# Проучване

Направихме проучване за други уеб сайтове в интернет за цветарски магазини и това какво предоставят те. Забелязахме, че в нито един сайт няма информация за това колко останали броя има от дадения артикул или дали е наличен или резервиран. В нашето уеб приложение ще имате тази информация, както и функцията да резервирате букет. Това ще ви бъде от полза, за да не изкупят всички артикули преди вас.

 Разгледахме статични и динамични уеб сайтове за цветя. Нашият сайт ще бъде динамичен и ще използваме база данни. Статичните уебсайтове са основни уебсайтове без персонализирано кодиране и бази данни, докато динамичните уебсайтове показват различно съдържание и са по-сложни и интерактивни. Статичен уебсайт е уебсайт, чиито уеб страници са кодирани в HTML, а съдържанието на всяка страница е фиксирано и не се променя, освен ако не се редактира и преиздава. Динамичен уебсайт е уебсайт, чиито уеб страници се генерират в реално време. Това е основната разлика между статичния и динамичния уебсайт. По сложност статичните уебсайтове са прости и лесни за програмиране, докато динамичните уебсайтове са по-сложни и трудни за програмиране. Освен това статичните уебсайтове нямат достъп до бази данни, докато динамичните уебсайтове получават достъп до информация от база данни. За това ние сме избрали нашият уеб сайт да бъде динамичен.

# Трислойна архитектура

Какво е трислойната архитектура?

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.

* При нея приложението е разпределено на слоеве
* Всеки слой има строго определена задача
* Във Visual Studio можем да създадем такова приложение, създавайки различни проекти в рамките на Solution-a ни /или различни папки в един проект/

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:

* Презентационният слой е 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. Бизнес слоят служи за обработка на данните и работните процеси. Той комуникира както с презентационния слой, така и с базите данни. Освен, че служи комуникира с останалите слоеве, презентационният слой предоставя различни видове информация на потребителя.

* Слой за бизнес логика (междинен слой, слой за обработка на данни)

Този слой е изтеглен от презентационния слой, и като отделен такъв, контролира функционалността на приложението като извършва различни процеси по обработката на данните.

* Слой за данните - Този слой се състои от сървър база данни. Тук информацията се съхранява и чете. В слоя за бази данни информацията се съхранява независима от бизнес логиката или сървърът за приложения. Когато данните се съхраняват в отделен слой се увеличава мащаба и се подобрява производителността. Слоят за данни комуникира само с бизнес логиката и служи за съхранение данни и тяхното използване.



# Използвани технологии и среди

# C# език за програмиране

За реализирането на нашият уеб сайт ще използваме C# език за програмиране. Избрали сме него, защото има много предимства пред останалите езици.

Езикът C# е удобен и предоставя възможност за много действия, които да се реализират с наистина малко код. Това е така, защото C# е програмен език от високо ниво. „Високо ниво“ е условно понятие, което означава, че даден език акцентира повече върху писането на програмната логика, предлагайки повече автоматизация и абстракции за управление на техническата част. За същата логика, в други езици за постигане на същия резултат ще ви бъде необходим доста по-голям обем от код.

C# обработва повечето сложни детайли на компютъра, така че да се съсредоточим върху програмирането, вместо да се тревожим за малките подробности.

При другите езици изграждането на едно пълноценно приложение, често пъти може да изисква съчетаването на два или повече езика/технологии. Със C# можете да създавате всичко, което ви дойде наум: от уеб сайтове, до десктоп приложения и игри, стигайки дори до мобилни приложения.

C# също така е удобен за разработката на уеб приложения, най-вече в съчетание с MVC.

# Asp.Net Core MVC

За проекта ще използваме няколко технологии, а основно Asp.Net Core MVC. Платформата използва C#, HTML, CSS, JavaScript и бази данни, а в съчетание със C#, можем да създаваме уеб приложения, базирани на шаблона за дизайн model-view-controller (MVC).

Модел (Model)

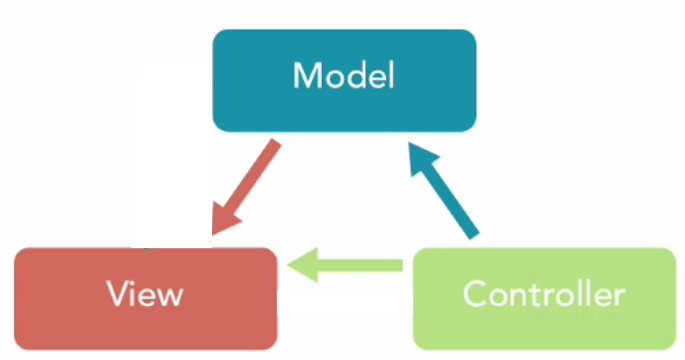
Тази част от модела на дизайна е основната част и съдържа чисто информация за приложението. Контролира логиката и правилата на приложението.

Преглед (View)

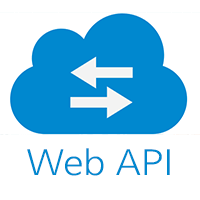
Тази част помага на потребителя да види данните на модела. Основната грижа на тази част е достъпът до данните на модела и е визуализация на информацията, която приложението съдържа.

Контролер (Controller)

По-голямата част от работата се извършва от контролера. Той следи за въведената информация от страна на потребителя. Използва се между модела и частта за преглед.

Някои от най-основните ѝ предимства на Asp.Net Core са скоростта, широкият спектър от поддържани езици и фактът, че е безплатна.

Уебсайтовете и апликациите, построени с ASP.NET Core са по-бързи от тези, написани на PHP например, тъй като кодът се компилира, след което се изпълнява.

 Web API

Най-общо казано API или “приложен програмен интерфейс” е набор от инструкции за програмиране, които казват на уеб базираните софтуерни приложения как да взаимодействат и общуват помежду си и това дава достъп до услугите на приложението, които разработчиците могат да интегрират в собствените си уеб продукти.

Razor Pages

Razor Pages са алтернатива на стандартния MVC модел. Може да направи кодирането по-лесно и по-продуктивно, отколкото използването на контролери и изгледи

SignalR може да се използва за добавяне на всякакъв вид уеб функционалност "в реално време" към вашето ASP.NET приложение. Уеб функционалността в реално време е възможността сървърният код да изпраща съдържание към клиентите веднага, когато стане достъпно, вместо сървърът да изчака клиентът да поиска нови данни.

Blazor позволява изграждането на клиентски уеб потребителски интерфейс с .NET.

Visual Studio

За създаването на нашето уеб приложение ще използваме Visual Studio 2019, защото имаме няколко предимства в него.

Когато Microsoft пусна нова версия (2019 година) - Интегрирана среда за развитие (IDE) е софтуер, който ще увеличи производителността на програмистите, докато разработват софтуера и приложения.

Visual Studio е едно от най-известните IDE, което използва през последните няколко години. Използва се за създаване на компютърна програма, уеб приложения, EXE файлове и др.

Visual Studio ще подобрят производителността на програмистите и екипното сътрудничество в разработката на софтуер в сравнение с предишните версии.

Новият потребителски интерфейс ще позволи на потребителите да създават приложения много ефективно, а подобрената лента за търсене осигурява точни резултати за разработчиците, докато разработват софтуера.

Отстраняването на грешки е с 50% по-бързо от предишните версии.

В крайна сметка, тези разширени функции в Visual Studio ще повлияят значително на софтуера и разработване на приложения.

# Bootstrap

Софтуерът Bootstrap е среда с [отворен код](https://bg.wikipedia.org/wiki/%D0%9E%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD_%D0%BA%D0%BE%D0%B4), която съдържа набор от инструменти за създаване на уеб приложения и уебсайтове.

В допълнение към обичайните HTML компоненти, Bootstrap има много други компоненти, написани на CSS и JavaScript, проектирани така, че да улеснят разработчиците да създадат висококачествен потребителски интерфейс.

Компонентите включват: бутони, изображения, прогрес бар и съобщения.

Състои се от три части:

CSS файлове - Има няколко компонента, които често могат да се видят, че се използват на днешните уебсайтове и други компоненти. Компонентите като тези включват разширяване на елементи, бутони и разширени функции като групиране, меню за страници и навигация, напредък. s ленти или съобщения.

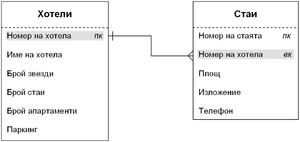
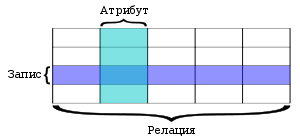
JavaScript - За Bootstrap съществуват и много [скриптове](https://bg.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) във формата на [JQuery](https://bg.wikipedia.org/wiki/JQuery) плъгини, които добавят разширени опции като изскачащи прозорци или графики.

Наред с другите неща, плъгините могат да разширят функционалността на съществуващите компоненти, например, за да създадат текстово поле, което допълва текста, записан в него.

Освен стандартния дизайн на Bootstrap, може да персонализирате всяка част от дизайна, като редактирате файловете CSS и JavaScript.

Има редица сайтове, които позволяват свободно изтегляне на дизайна на Bootstrap, например [Bootswatch.](https://bootswatch.com/)

# База данни

За нашето уеб приложение ще използваме релационна база данни. Тя е тип база данни, която съхранява множество данни във вид на релации, съставени от записи и атрибути (полета) и възприемани от потребителите като таблици. Релационните бази данни понастоящем преобладават при избора на модел за съхранение на финансови, производствени, лични и други видове данни.

Фиг. 5

Фиг. 4

Ще използваме SQL Server за създаването на базата данни. Това е системата за управление на релационни бази данни. SQL Server поддържа по-големи количества данни, повече едновременни потребители и разполага с по-голям капацитет от ядрото на базата данни JET/ACE. Казано по-просто, той ни предлага платформа, където можем да актуализираме, променяме и управляваме данните. Тя се нарича система за управление на релационни бази данни поради своето естество да съхранява данните в таблици, където таблиците съхраняват данните за едно и също образувание.

Той има много лесен интерфейс, който помага на разработчика на бекенда да се съсредоточи много върху грижата за данните, а не да се страхува от работата му. Тъй като е продукт на Microsoft, .Net рамката може лесно да бъде интегрирана с нея, тъй като тя също принадлежи на същата организация. В сравнение с другото средство за съхранение на данни като Excel, текст и така нататък, базата данни винаги се предпочита поради своята великолепна сила да обработва данни, да предлага висока сигурност и главно поради капацитета си да съхранява големи данни.

С много кратки и ясни думи, SQL сървърът е инструментът, използван за прилагане на механизма на система за управление на релационни бази данни. Тя позволява на разработчиците да работят с данни, за да предложат добро изживяване на потребителя. В организациите SQL сървърът е най-предпочитаното средство за обработка на данни, тъй като е много способен да се справи с голям пул от данни.

За разглеждането на нашата база данни ще използваме Microsoft SQL Management Studio.