Съдържание

[1. Въведение 3](#_Toc64316656)

[2. Анализ на предметната област 5](#_Toc64316657)

[2.1. Основни термини в Интернет икономиката 5](#_Toc64316658)

[2.2. Интернет икономика (модел) 7](#_Toc64316659)

[2.3. Бизнес модели за електронна търговия 8](#_Toc64316660)

[2.3.1. Определение на бизнес модела 8](#_Toc64316661)

[2.3.2. Верига на добавената стойност и бизнес модели 8](#_Toc64316662)

[2.3.3. Видове бизнес модели за онлайн търговия 10](#_Toc64316663)

[2.4. Електронна търговия 12](#_Toc64316664)

[2.4.1. Определение за търговия 12](#_Toc64316665)

[2.4.2. Определение за електронна търговия 13](#_Toc64316666)

[2.4.3. Електронна търговия- елементи 14](#_Toc64316667)

[2.4.4. Предимства, постигани чрез използването на електронна търговията 15](#_Toc64316668)

[2.4.5. Разновидности на електронната търговия 16](#_Toc64316669)

[2.5. Фактори, подпомагащи реализирането на успешен онлайн магазин 18](#_Toc64316670)

[3. Проектиране на системата 20](#_Toc64316671)

[3.1. Архитектура на системата 20](#_Toc64316672)

[3.2. Бизнес логика 21](#_Toc64316673)

[3.3. Прокетиране на базата данни 22](#_Toc64316674)

[3.4. Интерфейс 26](#_Toc64316675)

[4. Програмна реализация на системата 33](#_Toc64316676)

[4.1. Използвани технологии 33](#_Toc64316677)

[4.1.1. ASP.NET 33](#_Toc64316678)

[4.1.2. C#- C sharp 34](#_Toc64316679)

[4.1.3. HTML (Hyper Text Markup Language) 35](#_Toc64316680)

[4.1.4. CSS (Cascading Style Sheets) 36](#_Toc64316681)

[4.1.5. SQL (Structured Query Language) 37](#_Toc64316682)

[4.2. Реализация на базата данни 38](#_Toc64316683)

[4.2.1. Създаване на таблици и колони (заедно с техните ключове) в базата данни 38](#_Toc64316684)

[4.3. Реализация на приложението 42](#_Toc64316685)

[4.3.1. Създаване на дизайн 42](#_Toc64316686)

[4.3.2. Реализиране на бизнес логиката 44](#_Toc64316687)

[5. Верификация и тестване 68](#_Toc64316688)

[6. Анализ на резултатите 69](#_Toc64316689)

[7. Фигури 70](#_Toc64316690)

[8. Източници: 70](#_Toc64316691)

# Въведение

Идеята за World Wide Web (WWW), наричана още „световната мрежа“, е предложена през 1989 г. от английския инженер Тим Барнърс-Лий, който ръководи и нейната първа реализация. През 1989 г. той предлага да се създаде хипертекстова система „Mesh“, която да обслужва информационните нужди на CERN. Към края на 1990 г. основните компоненти на системата са готови: мрежов протокол HTTP, маркиращ език HTML, пъривят уеб браузър и редактор World Wide Web, първият сървър CERN http и първите уеб страници.

Ако първоначалната идея на световната мрежа е била с цел достъп и обмен на информация, то в днешно време с развитието на технологиите това далеч не е така. Хората започват да използват World Wide Web с различни цели: развлечения, достъп до информация, комуникация, маркетинг и реклама, складиране на данни и търговия. От гледна точка на използването на Интернет с търговска цел по отношение на комерсиалните дейности в мрежата се употребяват две понятия.

* Електронен бизнес (e-Business)
* Електронна търговия (e-Commerce)

***Електроният бизнес***- включва в себе си електронната търговия, също така и всяка комерсиална дейност, която се извършва през мрежата. Практически, почти всеки сайт в Интернет, независимо от неговата тематика, може да представлява някаква форма на електронен бизнес.

Електронният бизнес подразбира използването на информационните технологии и на глобалните компютърни мрежи за правене на бизнес. Благодарение на широкото разпространение на технологията „www“ в течение на много години, Интернет от академична мрежа се превръща в популярна среда за общуване, реклама и бизнес. Макар, че електронният бизнес съществува и по-рано, но именно популярността и достъпността на Интернет прави възможно широкото му използване. Електронният бизнес позволява да се изменят практически всички процеси, протичащи в съвременния бизнес, интегрирайки ги в едно цяло. Потребителите могат да търсят, да поръчват и заплащат стоки, използвайки Интернет, да се обменя информация за стоки и услуги с други потребители.

Или най-общо електронният бизнес се дефинира като процес на използване на електронните технологии за осъществяване на бизнес.

***Електронната търговия*** е по-специализираното понятие и представлява част от по-широкото понятие електронен бизнес. Под „електронна търговия“ се разбира преди всичко продажбата на стоки и услуги през различни видове електронни магазини, складове и аукциони.

Целта на дипломния проект е разбработване на уеб базирана система за онлайн търговия, която всъщност представлява електронен магазин. Магазинът се състои от една основна част – потребителска.

Потребителксата част предоставя възможност (след регистрация) на потребителите да предлагат стоки, както и да търсят и закупуват други стоки. Също така тези потребители, които желаят да закупят продукт, получават информация за него (количество, описание, цена и т.н).

# 2. Анализ на предметната област

## 2.1. Основни термини в Интернет икономиката

С развитието на световната компютърна мрежа и неизбежната й комерсиализация в ежедневието на хората се появиха множество нови понятия. Термини, като електронна търговия (E-commerce), електронен бизнес (E-business), електронна услуга (E-service) и електронно обучение (E-learning), придобиват все по-голяма популярност в съвременното общество. Самата комбинация от думи подсказва основата, която става причина за това бурно развитие и поява на нови направления в икономиката - виртуалната среда на Интернет.

В Националната стратегия за електронна търговия, одобрена от Министерския съвет на Република България през 2000 г. е формулирано следното определение: **"Електронната търговия е непрекъснат цикъл от обработка и обмен на данни, чрез които се осъществява унифицирано и интегрирано информационно осигуряване на участниците в цялостната търговска транзакция независимо от сферата на дейност, отрасъла, държавата и пр."**

**Е-търговия** (е-commerce)- състои се от купуването и продаването на продукти и услуги чрез електронни системи като Интернет или други комютърни мрежи. Количеството на търговията, която се осъществява електронно, нараства с широкото разпространение на Интернет. Електронната търговия е предоставянето на услуги на информационното общество, т.е. такива услуги, които обикновено са възмездни и се предоставят чрез електронни средства след изрично изявление от страна на получателя на услугата. При тези услуги всяка от страните използва устройства за електронна обработка, включително цифрово компресиране и съхраняване на информацията, като услугата изцяло се осъществява чрез използване на проводник, радиовълни, оптичен кабел или други средства за пренос на електромагнитни вълни.

Доверието е в основата на е-търговията. В традиционната търговия голяма част от механизмите за изграждане на доверие са свързани с формата на комуникация между участниците – средства, свързани с контакта лице в лице. При отдалечения достъп много от традиционните механизми за установяване на доверие между участниците не съществуват. Затова се използват варианти, които да подсилят останалите или да въведат нови механизми за изграждане на доверие.

**Е-бизнес-** обхваща вътрешните процеси като производство, управление, развитие на продукта, маркетинг, риск мениджмънт, финанси, управление на познанието, човешки ресурси и е-търговия. Стратегията на бизнеса е по- сложна и е насочена предимно към вътрешните процеси.

Много хора не могат да разграничават е-търговия и е-бизнес. Казано накратко- е-бизнесът е повече от е-търговия. При е-бизнеса, технологиите не се използват само за продажби (което е основно за е-търговията), а за да се подобрят всички аспекти от бизнес процесите. В тази връзка може да се обобщи, че е-търговията обхваща отворените навън процеси, свързани с клиенти, доставчици и външни партньори. Тук се включват: маркетинг, приемане на поръчки, доставка, обслужване на клиентите, закупуване на суровини и материали за производството и снабдяване. При е-търговията се създават нови бизнес модели и потенциал за увеличаване на печалбата или реализиране на загуби за сметка на новопоявили се конкуренти.

**Е-услуга-** всяка услуга, която може да бъде предоставена по електронен път или казано накратко- най- разширеното понятие, което включва е- търговия и е- бизнес.

**Е-магазин-** термин от електронната търговия за уебсайт за онлайн продажби на дребно. С все по-мащабното навлизана на Интернет в света на хората, неизменно се появяват нови неща, които улесняват живота на обикновените потребители. Сред тях е и пазаруването в Интернет, което спестява много време на обикаляне и търсене по магазините, спестява пари, а закупените стоки са доставяни до вратата на получателя. Тук отношенията търговец-клиент не се различават съществено от тези при традиционната търговия. Единствената разлика е, че тези отношения се развиват във виртуална среда. Някои от предимствата са: възможност на клиента за проучване на пазара, значително по-ниски цени на продукта, реклама, която в мрежата е значително по евтина и удобството за клиента да пазарува от вкъщи и да получава закупената стока до вратата.

## Интернет икономика (модел)

Интернет икономика се отнася до правенето на бизнес чрез пазари, чиято инфраструктура е базирана на Интернет и Световната мрежа. Интернет икономиката се различава от традиционната икономика по множество начини, включително: комуникация, сегментация на пазара, стойности на разпределение и цена. Гош определя, че бизнесите не могат да избегнат Интернет икономиката. Те трябва да отчетат и разберат, че има едновременно глобални възможности на разположение, както и рискове от неучастие. И трябва да отчетат, че чрез Интернет, всеки участник във веригата на стойността може да узурпира ролята на който и да е друг участник.

Поради огромното количество свързани потребители, невероятната скорост на разпространение на информацията и на липсата на определящо значение на разстоянията, фирмите могат да предлагат стоки и услуги не само локално, но и на потенциални клиенти по цялото земно кълбо. Както Грегори Манкю определя през 2003 г. „преимуществата на информационните технологии като Интернет са неимоверни и повлияват много дялове на икономиката“.

За да изследват икономическите параметри на Интернет икономиката, Бостънският университет и Cisco Systems разработват изследване на главните пазарни участници в нея. Те си поставят за цел да си отговорят на следните въпроси:

* Кои бизнес сектори, продукти и услуги трябва да бъдат включени в тази икономика
* Каква методология е подходяща и приложима за измерване на дейностите в Интернет света
* Колко голяма е Интернет икономиката
* Колко бързо се развива

За целта се въвеждат т.нар. „индикатори на Интернет икономиката". Те имат за цел да изяснят главните характеристики и чрез измерване на различни нейни компоненти. Тези индикатори - "индикатори на приходите" и "индикатори на заетостта в Интернет икономиката" са изградени на базата на анализа на четири нейни слоя:

Слой на инфраструктурата

Слой на Интернет приложенията

Интернет посредници

Интернет търговия

## Бизнес модели за електронна търговия

### Определение на бизнес модела

Бизнес моделът сам по себе си не гарантира разбирането как той ще допринесе за реализирането на бизнес мисията на всяка от компаниите, която участва в дадения модел. Трябва да бъде известна също и маркетинговата стратегия на компанията, за да успеем да оценим търговската жизнеспособност и да отговорим на въпроси като: как се постига конкурентно предимство, каква е позицията, какъв е маркетинговият микс, коя пазарна стратегия се следва.

### Верига на добавената стойност и бизнес модели

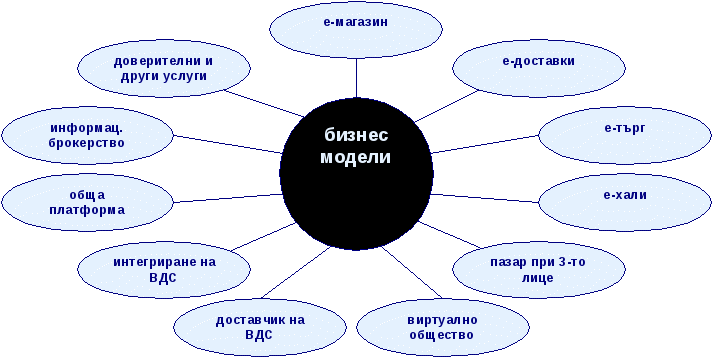
Методологията за определяне структурата на бизнес моделите може да се базира на декомпозиране и преструктуриране на веригата на добавената стойност. Това включва идентифициране на елементите на веригата и определяне на възможните начини за интегриране на информационните технологии в различните и звена. В случая се взема под внимание възможността за използване на електронни пазари. Те могат да бъдат изцяло отворени, което означава, че имат произволен брой купувачи и продавачи, или полуотворени- с един купувач и много продавачи, или обратното. Схемата е следната:

* Декомпозирането на веригата на добавената стойност означава да се идентифицират елементите й, както е направено от Майкъл Портър (Porter, Michael E., (1985). "Competitive Advantage", Ch.1), който е разграничил девет елемента. Основните включват вътрешна и външна логистика, преработка, маркетинг и продажби, поддръжка. Вторичните елементи са: технологично развитие, снабдяване, мениджмънт на човешките ресурси, корпоративна инфраструктура.
* Модели на взаимодействие, които могат да бъдат няколко типа: 1 към 1; много към 1; 1 към много; много към много. В случая 1 към 1 трябва да се разбира като брой на компаниите, включени в пазара, а много означава, че се обединява информацията от няколко компании.
* Преструктуриране на веригата на добавената стойност, което включва интегриране на информационните процеси в различните етапи на веригата. Получените комбинации съдържат елементите, участващи в подобна интеграция.

Възможните структури за бизнес моделите се построяват чрез комбиниране на моделите на взаимодействие с интеграцията на веригата на добавената стойност. Например електронният магазин е "един играч" към "един играч"-модел за продажби и маркетинг. Електронните "хали" се състоят от N електронни магазина. Електронните "хали", които предлагат един и същ вид стока, са пример за „много към 1“ модел за продажби и маркетинг (информацията за стоката е обща за многото доставчици в халите). Електронният търг, където много купувачи наддават за стоките, предлагани от един доставчик, събира на едно място продажбите на доставчика с интересите на много купувачи, като се комбинира информацията за наддаването от многото купувачи.

Техническото обезпечение на всеки от бизнес моделите зависи от степента на развитие на информационните технологии. Това включва възможностите за интеграция, за реализация на отделни функции и поддръжката на моделите на взаимодействие. Комерсиалната страна на всеки от бизнес моделите е съвсем различен въпрос и следва да се третира отделно.

### Видове бизнес модели за онлайн търговия



Фигура 1- Видове бизнес модели

* Електронен-магазин

Това е модел на магазин, представен в електронна форма в уеб пространството на фирмата. Основните му функции са да се представят продуктите и услугите на фирмата. В последно време се добавят възможности за осъществяване на поръчки и възможности за разплащане. Предимствата за фирмата-продавач се изразяват в увеличаване на каналите за предлагане, ниска цена на реализацията в глобален мащаб, работа 24 часа, 7 дни и по-добра възвръщаемост на инвестициите. Предимствата за купувачите са по-ниските цени в сравнение с физическите магазини, по-големи възможности за информация и избор на продукти и услуги, удобства при избора, пазаруването и 24 часово работно време. При повтаряне на посещението в даден магазин се използват предимствата на директния маркетинг за купувача и продавача. Печалбата за продавача се получава от намалените разходи, увеличените продажби и реклама.

* Електронно купуване (Е-procurement)

Това е модел за електронни оферти и купуване на продукти и услуги. Той доразвива модела на електронния магазин чрез добавяне на инструменти за търсене и сравняване на стоките. Използва се от големите фирми и обществени организации. Предимствата за фирмата-продавач са повишените възможности за офериране и оформяне на поръчките, възможностите за глобални канали, ниски цени на оферирането и сътрудничество с другите представени фирми. Предимствата за купувачите са големия набор от доставчици, което се очаква да доведе до по-голям избор, ниски цени на стоките и повишаване на качеството. Намаляват се разходите за офертите. Печалбата за продавача се изразява в намаляване на цената на офертите и автоматизация на процесите на договорирането.

* Електронен търг (Е-auction)

Това е електронен модел на класическите търгове. Той използва предимствата на съвременните технологии за преодоляване на времето и пространството и възможностите за мултимедийни представяния. Няма ограничения за прилагане на етапите на електронната търговия и се използва при всеки от тях. Поради ниските разходи може да се използва и за малки количества на стоките. Предимствата за фирмата-продавач и за купувачите са повишаването на ефективността от дейностите и спестяване на време и разходи за транспорт. Печалбата за извършващите го се изразява в осъществяването на продажбата на технологична основа, редуциране на таксите за търга и рекламата.

* Виртуални общности

Този модел се развива от клиентите и партньорите на виртуална фирма. Те добавят своята информация като: маркетингова информация, профили на клиентите, лоялност на клиентите и получават обратна връзка. Обособяват се в специфичен пазарен сектор (например книгоразпространение).

* Информационни брокери (Information brokers)

Този модел включва множеството от нови информационни услуги, които разглеждат информацията и знанието като стока. Подобни услуги са търсене на информация, издаване на списания, класификация на стоки и дейност, потребителски профайлове, инвестиционни съвети и др. Печалбата за продавача се получава от таксите за абонамент, консултации и други информационни услуги.

При изследването на действащите в момента бизнес модели в Интернет можем да направим следните изводи:

Информационните и комуникационните технологии дават възможност за развитие на голям кръг от бизнес модели. Възможностите на техниката са само един от аспектите при подбирането на бизнес модел. Технологията сама по себе си не осигурява насоки за подбора на бизнес модел от гледна точка на търговията. Управлението на технологичното развитие може да се получи от дефиницията на нови модели. Много от възможните за изпълнение модели не са били тествани в реалността.

## Електронна търговия

Напоследък огромен брой фирми, независимо дали става въпрос за големи корпорации, малки предприятия или физически лица, предлагащи стоки и услуги, станаха част от една модерна и новаторска инфраструктура, наречена електронна търговия. Инвестициите, нужни за функционирането на този бизнес, са в размер на стотици милиарди, похарчени в най-различни валути, като тези средства се инвестират в редица важни компоненти нужни за осъществяването на подобен тип сделки – Интернет капацитет, сървъри, обучения на персонал, покупка на софтуерни продукти, ноу хау, електронна идентичност и т.н.

### Определение за търговия

Най-напред е важно да се схване цялата идея, стояща зад процеса на търговия изобщо. След запознаване с основните понятия, използвани в света на търговията може да се заключи, че Интернет търговията не е нещо ново, а просто нова форма на търговията, реализирана в Интернет.

Според тълковния речник думата „търговия” изразява размяната, продажбата или покупката на стоки и изискването за тяхното транспортиране от едно място до друго. Търговията в контекста на нашия живот се определя като обмен на стоки и услуги срещу финансови средства- пари.

Ако вие притежавате компания за производство на стоки например, тя вече е част от голямата верига на търговията. При подобно разглеждане на понятието търговия може да се разграничат няколко типа търговски субекти:

**Купувачи** – лицата, които имат за цел срещу определена цена, заплатена от тях, да закупят стока или услуга.

**Продавачи** – лицата, предлагащи на купувачите възможността да закупят стоки и услуги, и получаващи определена сума срещу това. Те биват два вида: крайни търговци – това са продавачите, предоставящи стоките и услугите на крайни купувачи, и търговци на едро (дистрибутори) – това са продавачите, предоставящи стоки и услуги на крайните търговци или на други фирми.

**Производители** – това са икономически субекти, създаващи стоките, които в последствие се предлагат от търговците (продавачите) на купувачите. Производителят на стоки е и продавач по необходимост, тъй като се налага той самият да продава произведените от него стоки на продавачите, а понякога и директно на крайните купувачи.

### Определение за електронна търговия

Всъщност електронната търговия не е толкова мистериозна и трудна за разбиране, колкото изглежда в очите на незапознатите с нейните концепции. В общите си принципи тя по нищо не се различава от обикновената търговия. В случай, че вие вече сте част от търговията по един или друг начин, то това означава, че вие сте на 80-90% подготвен, за да стартирате собствен онлайн сайт за електронна търговия. Дори да не разполагате със задълбочени технически познания или да не разполагате с голям финансов ресурс, винаги ще откриете подходящи стоки и услуги, които да предложите на вашите клиенти.

### Електронна търговия- елементи

Елементите на електронната търговия, и по-конкретно свързаните с търговските операции, налагат да се имат предвид конкретни детайли. Всички стъпки, които трябва да се направят, са описани по-долу, като изхождаме от гледна точка на това, че електронната търговия се изразява в продажбата на стоки и услуги на крайните потребители.

* В случай, че ще искате да продавате определени стоки на конкретна група потребители, то именно този елемент е съществен за цялата ви идея. Вие трябва да имате стоки, които да предлагате на вашите потребители, за да могат те да ги закупят и да генерират печалба. Стоките може да са собствено производство или закупени на едро, защото все пак вашата таргет група са крайните потребители или по-малки фирми дистрибутори.
* Важно е да разполагате с място, където ще се помещава извършваната търговия и ще се съхраняват продуктите, които предлагате. При обичайните търговски практики това място е магазинът, където продавате стоките, а клиентът може да ги разгледа и да прецени дали да ги закупи или не.
* Съществуват места, които имат чисто виртуално предназначение- телефон за приемане на поръчките на потребителите например. Купувачите и продавачите при подобно разпределение на търговията не се срещат и не се виждат, а всичко се извършва посредством комуникационни практики – Интернет, телефон и т.н. Естествено за този бизнес е това, че купувачът знае какво да си закупи, защото той вече е проучил продукта, който желае. При двата широко разпространени начини за осъществяване на търговски взаимоотношения двата пазарни субекта – купувач и продавач, се ангажират в еднакъв момент от времето.

Тук се появява и първата специфична особеност на електронната търговия, защото тя няма физическо място за извършване на търговските взаимоотношения между търговските субекти, а каталогът, от който клиентите разглеждат и проучват продуктите, е в електронен вид, качен на уеб сайта на фирмата, която предлага тези стоки. При положение, че потребителят избере определена стока, то той решава да я закупи и го прави.

### Предимства, постигани чрез използването на електронна търговията

Предимствата, постигани чрез използването на електронна търговията, са няколко:

* Клиентите не се разкарват до магазина, а правят всичко от своя дом, без да стават от компютъра
* Нито магазинът, нито клиентите се съобразяват с работно време, а това дава възможност вашият магазин да бъде 24 часа в денонощието,7дни в седмицата и 365 дни в годината отворен за клиентите
* Още нещо важно, което трябва да се отбележи, е нуждата от намиране на начин, по който да насочвате клиенти да посещават вашия уеб сайт за електронна търговия – процес, познат в обикновената търговия, като маркетинг. Разликата при провеждането на рекламни кампании в обикновената търговия и електронната е, че при електронния маркетинг (e-marketing) целевата група или таргет групата (target audience), както също е известна, се състои от хора, които вече са потребители на Интернет. Все пак, ако никой не знае за вашия уеб сайт за електронна търговия – вероятно няма да продадете нищо.
* Ще имате нужда и от начин, по който да приемате поръчките от вашите потребители. При изграждането на решение за електронна търговия, това най-често се явява електронен формуляр (online form), който описва редица данни за потребителите и техните поръчки, като изискваната информация зависи от това какво предлагате на вашите клиенти. Част от информацията, която изисквате, е все пак задължителен компонент. Тази част включва адресната информация, нужна за доставяне на стоката; стоките, които предлагате и съответните им цени; общата стойност на поръчките; информация за изготвянето на фактура; форма на плащане; доставка.
* Трябва да обезпечите и начините за приемане на плащания, като те най-често се осъществяват посредством финансови средства под формата на пари. В случаите, когато приемате разплащания чрез банкови карти, а не трябва да забравяме, че това е най-разпространеният начин за подобна търговия, трябва да имате изградена уеб страница с повишено ниво на защита, т. нар. secure web page и връзка към банката. При използването на банкови карти, както вече подчертахме, е нужно да си създадете специална сметка за електронни разплащания.
* Естествено трябва да осигурите и начина, по който купувачът ще получава своята стока, която е поръчал от уеб сайта ви за електронна търговия. При електронния магазин клиентът идва в уеб сайта, пазарува и след това очаква вие да се погрижите да му изпратите стоките, които е поръчал.
* Редица съвременни стоки са сложни за обслужване поради техническо или друго естество, като клиентите в подобни случаи се нуждаят от компетентна поддръжка, за да могат да се консултират с вас относно поръчаните от тях продукти. Всеки клиент ще ви бъде безкрайно благодарен, ако сте поместили на вашия уеб сайт подробно ръководство за инсталация или за експлоатация на технологично сложния продукт, който те са си закупили. Друга възможност е да предоставите на клиентите възможността за обратна връзка с вас посредством електронен формуляр или опцията да подадат заявка за техническо обслужване.

В крайна сметка се оказва, че основните принципи на работа при стандартната търговия и при електронната търговия са едни и същи, а там, където се различават, то е, за да се постигне по-висока ефективност в работата на електронния магазин.

### Разновидности на електронната търговия

Наред с е-понятията, за които стана въпрос, има и някои други абревиатури, които звучат най-малкото объркващо за непросветения потребител. Става въпрос за съкращения от типа B2B (business-to-business), B2C (business-to-consumer), C2C (consumer-to-consumer) и други подобни. Термини от такъв тип се използват за отграничаване на един вид е-бизнес от друг. За целта на настоящата разработка ще спрем вниманието си само на бизнес отношенията, в които участват двете основни фигури в икономическите процеси - клиент и продавач. Все пак е необходимо да се отбележи, че освен тях има и други направления - B2G (business-to-government), B2E (business-to-employee) и т.н.

* Търговски отношения по частни компютърни мрежи

Частните мрежи се изграждат от и за нуждите на една организация и всичките им ресурси (компютри, средства и канали за предаване на данни т.н.) принадлежащи на тази компания. Само комуникационните линии за много отдалечен достъп се наемат от обществените комуникационни мрежи, като телефонните компании. Тези мрежи имат за цел да обслужват целите само на организацията, която ги е изградила. Мрежите могат да бъдат със всякаква архитектура и географско местоположение. Последното зависи най-вече от големината и това къде се намират офисите на организацията. Но каквото и да е, частната мрежа е с ограничен достъп само за оторизирани лица - персонал, управители, собственици. Също така е строго регламентирано кой потребител до какви ресурси има достъп. Този тип мрежи могат лесно да имат връзка с други мрежи, но това не прави достъпа до тях по-лесен. Някой частни мрежи са създадени изцяло с цел да се печели от тяхното използване. Пример за това са частните телефонни компании и мрежите, които предлагат пренос на данни срещи заплащане на услугата.

* Отношения Бизнес - Бизнес (Business-to-Business, B2B)

B2B или бизнес-към-бизнес е най-широко разпространеният тип е-бизнес. Той обхваща взаимоотношенията между различните компании и съответно неговият обем е най-голям, а и разнообразието и сложността му също. Той е атрактивен за фирмите, тъй като оползотворява по най-добрия начин предимствата на електронните мрежи - спестява време, съкращава разходи и минимизира неудобствата от фактора "географско положение". В B2B се включва всичко - връзки между продавачи и купувачи, производство и предлагане на услуги.

* Отношения Бизнес - Потребител (Business-to-Consumer, B2C)

Тези отношения наследяват обичайните връзки-клиент продавач. Тук те обаче са поставени на нови основи, поради спецификата на Интернет като среда за пазаруване и виртуален пазар.

Най-типични представители тук са продавачите на дребно като Amazon, BarnsAndNoble, Sears&Sears и др.

* Отношения Потребител - Потребител (Consumer-to-Consumer, C2C)

Тези отношения се срещат изключително рядко в реалния физически свят и това е още едно доказателство за потенциалната мощ, която крие Интернет като средство за търговски отношения.

Те възникват когато се съберат две страни, които не са обвързани с дадена търговска фирма и решават да направят транзакция на стоки и услуги срещу пари или други изгоди. Това се среща на специализирани, направени специално за целта брокерски сайтове. Това са аукционите (http://www.ebay.com/) или др. подобни места. Не е пречка срещата да стане на предназначен за съвсем друга цел сайт, но тези срещи са по-скоро случайни отколкото целенасочени както е в брокерските сайтове.

Уточняването на цената често е на принципа на търговете или свободното договаряне. Поради тази причина тези места служат като фини индикатори на пазара.

## Фактори, подпомагащи реализирането на успешен онлайн магазин

* Основната цел на един онлайн магазин е да печели пари.

Всичко, което се прави по онлайн магазин, трябва да не пречи на това първо правило. Една от най-важните стъпки, която често се пропуска – това би обяснило и големия процент на провалените или недостатъчно успешни бизнеси– е изготвянето на ясен бизнес модел. Още в началото трябва да се реши дали се иска да се произвежда или предлага, да се внасят продукти от чужбина, да се използва dropshipping или affiliate или да се заложи на хибриден вариант. Чак след това може да се започне с изготвянето на бизнес плана, който ще бъде една стабилна основа на прохождащия онлайн магазин.

* Избор на система за изграждане на онлайн магазин.

Основните варианти са три: поръчкова платформа (custom код), платформа с отворен код (open source) и софтуер като услуга (SAAS). Ще се спрем на платформите с отворен код, защото това е най-популярният вариант в момента за изграждане не само на онлайн магазин, но и на корпоративен уебсайт, блог и други. Тези платформи предоставят голяма свобода на избор– разполага се с много повече възможности (теми, разширения) и се получава продукт, който следва общностния модел на тази култура, но и който може да се моделира според изисквания. Освен това, те са изключително динамични и се обновяват ежедневно, разполагат с постоянно тестване за бъгове и пропуски в сигурността, а и не на последно място, са най-бюджетната възможност. Рядко срещана ситуация е, при която най-добрата възможност да е и най-изгодната откъм цена.

* Създаване на качествено и полезно съдържание.

Типовете съдържание са много: от описания на продукти, през снимки, видео и инфографики, до статии, е-книги и презентации. Но целта на съдържанието не е просто да се класира по-напред в класацията на търсещите машини (това е само едно от многобройните му предимства), а да се предаде послание. Качественото и оригинално съдържание е една от най-важните предпоставки за успешен бизнес и добра репутация. Старае се винаги да се предлага интересна, истинска и интерактивна информация. Старае се да се образоват клиентите и винаги да се предлагат информативни текстове с допълнителна стойност.

И въпреки че съдържанието е в сърцето на цялата дейност, то не трябва да се ограничава само с нея.

* Стратегия за дигитален маркетинг.

Дигиталният маркетинг е в основата на успешното бизнес развитие. Дори да се представя само качествено съдържание и да има силен бранд, ако те не достигат до потребителите от целевата група, то те няма да помогнат с нищо за развитието на потенциала от възможности за бизнеса. Трябва да се намери най-подходящото решение за популяризирането на бизнеса, предвид индустрията, в която се намира.

* Маркетинг в социалните мрежи– естествено, най-очевидният избор е Facebook. И въпреки че там възможностите изглеждат неизчерпаеми, още повече подкрепени с факти като „1 милион потребители дневно са във Facebook”, най-важното, което трябва да се запомни е, че социалните мрежи, и в частност Facebook, не са онлайн магазини, а канали за комуникация с клиентите. Дори използването му за бизнес цели да нараства ежедневно, голяма част от хората продължават да използват Facebook като начин за разтоварване.
* E-mail маркетинг– дигиталният маркетинг не се изчерпва само с реклама в социалните мрежи. E-mail маркетингът е онази част от него, която съчетава малка доза креативност със максимална точност за достигане на целева група и страхотна възможност и скорост за реализация. Той предоставя възможност за персонализирани и таргетирани съобщения и начини да се достигне емоционално до целевата група.

Google Adwords и Facebook Ads – платеният начин да се достигне до максимално голям брой от (потенциални) клиенти. Чрез платените рекламни кампании във Facebook или Google има възможност да се предоставят позитивните страни на услуги/стоки само на хора, които биха се заинтересували от тях.

* Оптимизация за търсещите машини

SEO оптимизацията е процес за генериране на трафик чрез органични резултати в различните търсачки. Основната цел на търсачките е намирането на съдържание, което най-много се доближава до търсенето. В тази връзка, трябва да се вземат под внимание доста компонента при оптимизирането на един уебсайт – авторитет (linkbuilding and guest blogging), изживяване на потребителя и user interface, но отново се връщаме до изключителната важност най-вече на съдържанието и бранда. Без съдържание няма какво да се оптимизира.

Съществува огромно разнообразие от SEO инструменти, които може да се използват и да направят оптимизирането на онлайн магазин занимание, носещо само приятни емоции и допълнителни позитиви за бизнеса.

# Проектиране на системата

## Архитектура на системата

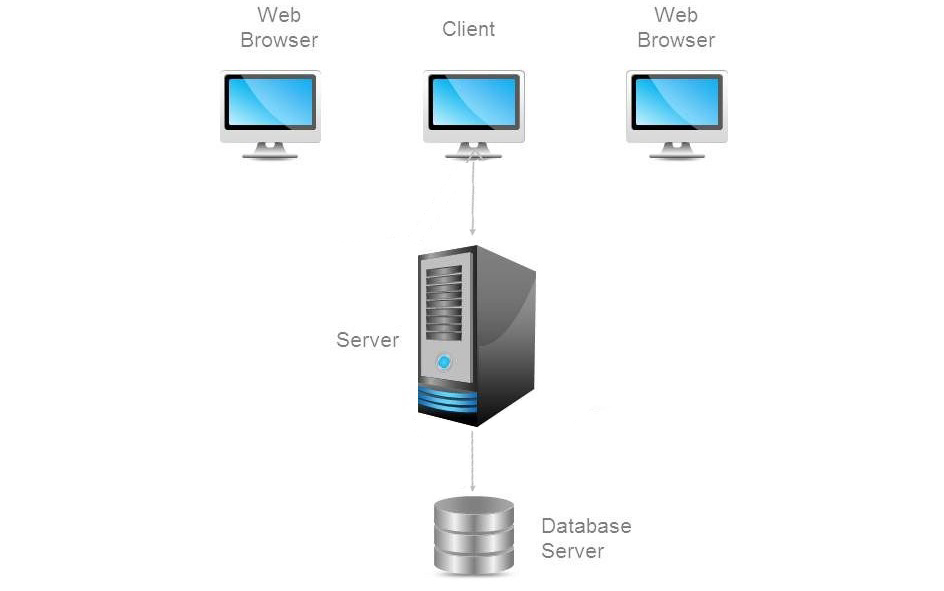
За архитектура на системата е избрана Клиент- сървър архитектура (Client- Server), тъй като тя най- добре пасва спрямо нуждите и изискванията на системата.

Клиент-сървър е тип мрежова архитектура, която отделя клиента от сървъра и най-често се използва в компютърни мрежи. Всеки клиент или сървър, свързан с мрежата, може също така да бъде използван като възел.

Най-елементарната типова клиент-сървър архитектура се състои от две части- от едната страна е сървърът, а от другата страна е клиентът. Постигането на определен резултат при използване винаги е свързано с комуникация между двете страни.

Характеристики на Клиента- подава заявки, изчаква отговор, свързва се до малък брой сървъри едновременно, взаимодейства си с крайните потребители чрез графичен интерфейс.

Характеристика на Сървъра- пасивност, чака за заявки от клиенти, при получаване на заявки ги обработва и след това отговаря, получава заявки от голямо количество клиенти, не контактува директно с крайния потребител



Фигура 2- Архитектура на системата

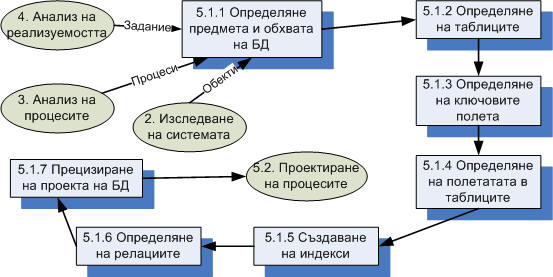
## Бизнес логика

Системата, която ще разработваме се състои от потребителска и администраторска част. При влизане в системата, на нерегистриран потребител, на началната страница ще види най-новите продукти, които са добавени. Ще може да търси артикули по тяхната категория. При желание да се видят детайли за дадена стока и/или тя да бъде закупена, потребителите ще трябва да се регистрират. Ако вече имат регистрация, то те трябва да влязат в профила си. Чрез профилът си, потребителите могат да бъдат продавачи и купувачи на стоки (не е нужна отделна регистрация за различните дейности, които ще се извършват).

За администраторската част е създаден специален потребител (admin), чрез който ще може да се администрира системата. Администраторът ще има възможността да променя данните на вечерегистрираните потребители, както и да премахва продукти от системата.

## Прокетиране на базата данни

Тази стъпка от проектирането на системата е може би е една от най-важните за успешната реализация. Този етап може да се раздели на няколко подетапа:



Фигура 3- Схема за проектиране на БД

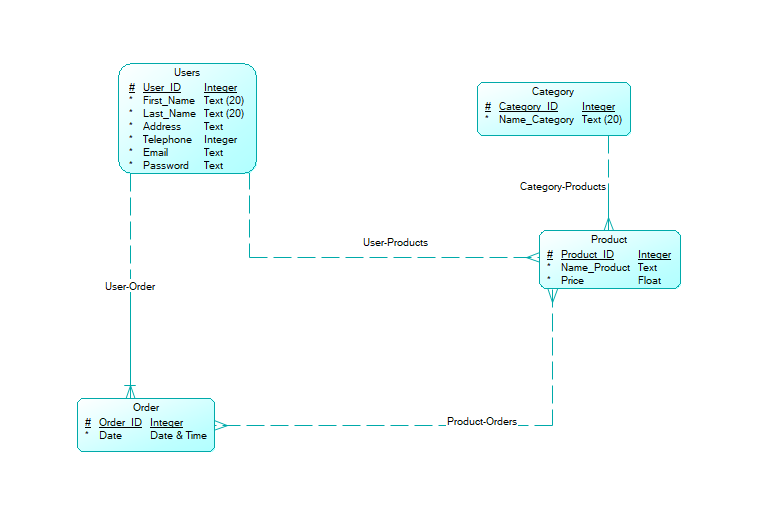
* Проектиране на концептуален модел

Концептуалният модел на данните отразява структурата на информацията, която ще се проведе в базата данни. Един общ подход към това е да се разработи модел на обектни взаимовръзки, често с помощта на инструменти за рисуване. Един успешен модел на данните акуратно ще отрази възможното състояние на външния свят, който е обект на моделирането: например ако хората могат да имат повече от един телефонен номер моделът ще позволи на тази информация да бъде съхранена. Проектирането на добър концептуален модел на данните изисква добро разбиране на областта на приложение. Обикновено е свързано със задаването на сериозни въпроси за нещата, които представляват интерес за дадена организация, като например „може ли клиентът да бъде едновременно и доставчик?“ или „ ако даден продукт се продава с два различни вида опаковки, това един и същ продукт ли е или различни продукти?“. Отговорите на тези въпроси създават дефиниции на терминологията, използвана за обектите и техните взаимовръзки и атрибути.

Производството на концептуалния модел на данни понякога включва входни данни от бизнес процесите или анализи на работния процес в организацията. Това може да помогне да се установи каква точно информация е необходима в базата данни и кое е ненужно. Например може да помогне в решаването дали базата данни трябва да съдържа исторически данни освен актуалните данни.

Основни компоненти (отношения, релации) на концептуалния модел на данните:

* Типовете обекти
* Типовете връзки или взаимодействия
* Атрибути и ключове
* Структурните ограничения

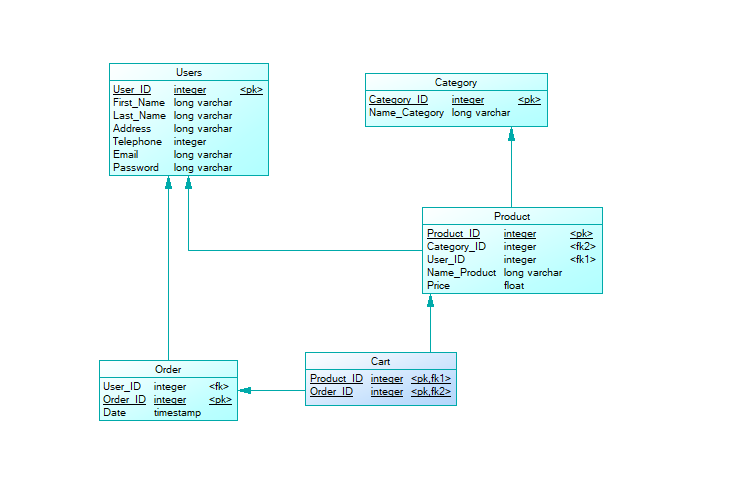


Фигура 4- Концептуален модел на БД

След като е произведен концептуален модел на данните, от който потребителите са доволни, следващият етап е това да се превърне в схема, която имплементира съответните структури от данни в базата данни.

* Логически модел

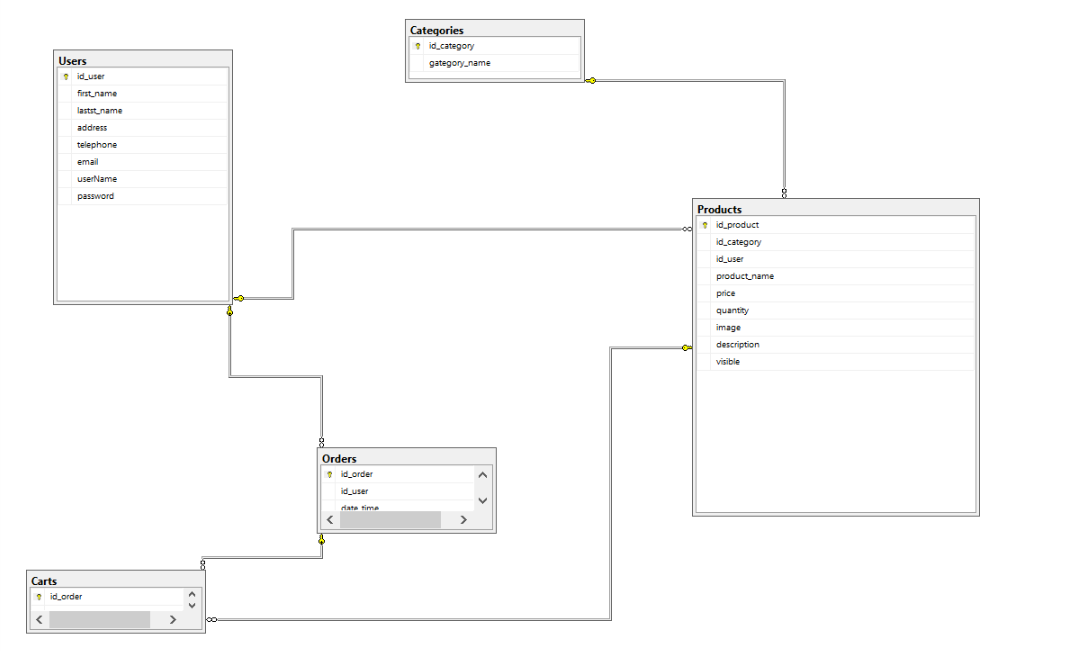
Логическият дизайн на базата данни е изразен под формата на схема. Докато концептуалния модел на данни е (поне на теория) независим от избора на технология на базата данни, то логическия модел на данните ще бъде изразен по правилата на конкретната технология на системата за управление на базата данни. (Термините „модел на данните“ и „модел на базата данни“ често се използват взаимозаменяемо, но в случая използваме модел на данните за проектирането на специфична база данни, а модел база данни за нотация в моделирането, използвана за изразяването на този дизайн.)



Фигура 5- Логически модел на БД

* Физически модел

Физическият модел представлява методите за съхранение и достъп до данните.



Фигура 6- Физически модел на БД

* Определяне на целите:

Изграждането на база данни зависи от целите, които си поставяме. Базата данни, която ще създадем трябва да пази информация за регистрираните потребители. Ако добавят артикули, то тези артикули трябва да се запазват под формата потребител- добавен продукт. Също така е необходимо пазенето на артикулите, които са вече добавени и подробна информация за тях. Друга основна цел на БД е да предлага възможност на потребителя да добавя избраните стоки в количка, както и история на направени предишни поръчки.

* Определяне на обектите в предметната област (таблици в БД):

Обикновено в една таблица се записват данните, които се отнасят за един клас от обекти. Колко таблици ще има в една РБД, зависи от броя на описваните класове от обекти. Нашата база ще има 5 обекта: потебители, продукти, категория продукти, поръчки и количка. Това ще са и таблиците в нея.

* Определяне на характеристиките на обектите (полетата):

На тази стъпка се определя какви полета (свойства) ще има всеки обект (таблица). Също така се определя какъв ще е и типът на данните.

* Определяне на първичните и външните ключове:

На тази стъпка чрез първичните ключове се опрделя кое ще е уникалното за всеки един обект, а чрез външните ключове се решава по какъв начин обектите ще си взаимодействат помежду си.

## Интерфейс

При посещение на сайта (responsive design) ще се зарежда началната страница, на коята ще се визуализират всички добавени продукти. На всички страници от системата ще се визуализира „header“ с лого, „footer“ и меню със следните елементи:

* Registration- страница за регистрация
* Login- страница за вход

***Страница за регистрация (Register).***

Чрез тази страница потребителите, които нямат регистрация, ще могат да си направят такава. На страницата се визуализират текстови полета, в които потребителите трябва да въведат:

* User Name- потребителско име
* Password- парола
* Confirm Password- потвърждение на паролата
* First Name- име
* Last Name- фамилия
* Address- адрес
* E-mail address- емайл адрес
* Telephone- телефон
* Бутон за регистрация (Register)
* Бутон за нулиране на данните (Reset)

След коректно въведени данни, за осъществяване на регистрацията, потребителите трябва да кликнат върху бутон „Register“.

***Страница за вход (Login).***

На тази страница потребителите, които вече са си направили регистрация в сайта, ще могат да влизат в системата. Страницата се състои от следните контроли:

* Текстово поле, в което потребителят да въведе потребителското си име
* Текстово поле, в което потребителят да въведе паролата си
* Бутон „Login“ – бутон за вход в системата.
* Хиперлинк- линк, който препраща към страницата за регистрация, ако посетителят на сайта няма такава.

При коректно въведени данни потребителят ще влезе в профила си и ще бъде препратен към началната страница на сайта.

След успешна регистрация или успешен вход в системата, менюто се променя по следния начин:

* Home- начална страница
* Categories- падащ списък от различните категории продукти
* Add product- добавяне на продукт (ако потребителят реши да продава стока)
* My products- страница с продаваните от потребителя продукти
* My orders- история на поръчките извършени от потребителя
* Cart- стоки, които потребителят желае да закупи, но все още не е финализирал поръчката си
* Logout- изход от системата

***Съдържанието на началната страница (Home) се състои от:***

* “Label”- приветствено съобщение с името на потребителя
* „Grid View“- таблица с информация за всички добавени продукти в системата (име на продукта, цена, изображение, номер на продукта и бутон за избор)

***Страница детайли (Details).***

След натискане на бутона „Select“ се визуализира страница с детайли за избрания продукт. Съдържанието на страницата се състои от:

* „Label“- име на продукт (Product Name)
* Текстово поле с името на продукта
* „Label“- категория (Category)
* Текстово поле с категорията на продукта
* „Label“- цена (Price)
* Текстово поле с цената на продукта
* „Label“- количество (Quantity)
* Текстово поле с количеството на продукта
* Снимка на продукта
* Описание на продукта
* Бутон за стартиране на поръчка (Make Order)

***Страница поръчка (Order).***

След натискане на бутона „Make Order“ се визуализира аналогична страница на страница детайли, но вече с поле за въвждане на желаното количество от дадения продукт. Съдържанието на страницата се състои от:

* „Label“- име на продукт (Product Name)
* Текстово поле с името на продукта
* „Label“- категория (Category)
* Текстово поле с категорията на продукта
* „Label“- цена (Price)
* Текстово поле с цената на продукта
* „Label“- количество (Quantity)
* Текстово поле- за въвеждане на цифри
* Снимка на продукта
* Описание на продукта
* Бутон за калкулация на сумата, която потребителят би платил, ако поръча това количество от този артикул (Calculate) (ако желаните количества от страна на потребителя надвишават наличните- изписва се съонщение за грешка, след което потребителят трябва да ги коригира)
* Бутон за добавяне на артикула в количката (Add to cart)

***Страница количка (Cart).***

Всеки избран от потребителя продукт се добавя в потребителска количка с цел по–бързо и лесно поръчване и преглед на продуктите в нея. Съдържанието на страницата се състои от:

* „Labe“- който показва общата стойност на всички артикули добавени в количката
* „Grid Viеw“- таблица с количествата и имената на избраните продукти
* Бутон за потвърждение и финализиране на поръчката (Confirm)
* Бутон за отказ от поръчката (Cancel)

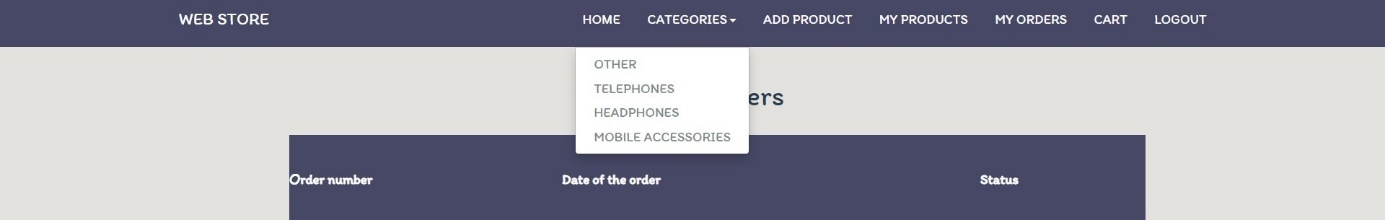
***Страница моите поръчки (My orders).***

След натискане „завършване на поръчката“ се визуализира страница с всички извършени от потребителя поръчки. Съдържанието на страницата се състои от:

* „Grid View“- таблица, визуализираща номерата на поръчките, дата на която са извършени поръчките и техният статус.

***Страница категории (Categories).***

Чрез този падащ списък, потребителите могат да изберат конкретна категория от продукти. След като изберат желаната от тях, те ще бъдат препратени към страница, съдържаща само този вид артикули.



Фигура 7- Проект на меню "Категории"

***Страница добавяне на продукт (Add product).***

Чрез тази страница, потребителите, които желаят да продават стока могат да добавят артикули в системата. Страницата се състои от следните контроли:

* „Label“- име на продукт (Product Name)
* Текстово поле - за въвеждане на текст
* „Label“- цена (Price)
* Текстово поле - за въвеждане на цифри
* „Label“- количество (Quantity)
* Текстово поле за въвеждане на цифри
* „Label“- категория (Category)
* „Drop Down Menu“- падащ списък за избор на категория
* „Label“- добави снимка (Add Image)
* „File Upload Images“- контролер за добавяне на файлове
* „Label“- описание (Description)
* Текстово поле- за въвеждане на текст
* Бутон за добавяне на продукт (Add product)(при натискането му, ако полетата са попълнени коректно се изписва съобщение за успешно добавен продукт в системата)
* Бутон за отказ (Cancel) (препраща към началната страница)

***Страница моите продукти (My product).***

Чрез тази страница потребителите, които продават стоки могат да следят датайли по своите продукти (като количество). Също така те могат, ако решат да спрат продажбата на предлаган от тях артикул, като го деактивират. Страницата се състои от следните контроли:

* „Label“- име на продукт (Product Name)
* Текстово поле с името на продукта
* „Label“- категория (Category)
* Текстово поле с категорията на продукта
* „Label“- цена (Price)
* Текстово поле с цената на продукта
* „Label“- количество (Quantity)
* Текстово поле- за въвеждане на цифри
* Снимка на продукта
* Описание на продукта
* Бутон за спиране на продажбите от този продукт (Disable)
* Бутон за отказ от редакция (Cancel)

***Вход като Администратор***

След успешен вход в системата като администратор, менюто се променя по следния начин:

* Home- начална страница
* Users- страница с регистрираните потребители
* Products- страница с продаваните от потребителите продукти
* Logout- изход от системата

***Съдържанието на началната страница (HomeAdmin) се състои от:***

* “Label”- приветствено съобщение
* „Grid View“- таблица с информация за всички добавени продукти в системата (име на продукта, цена, изображение, номер на продукта и бутон за избор)

***Страница потребители (Users) се състои от:***

* „Grid View“- таблица с информация за всички потребитрли в системата (име на потребителя, име, фамилия, адрес, телефон, имейл и бутон за избор)

***Страница потребителски детайли(User Details).***

Чрез тази страница администраторът може да променя данните на вече регистрираните потребители. Страницата се състои от следните контроли:

* „Label“- име на потребителя (User Name)
* Текстово поле с името на потребителя
* „Label“- парола (Password)
* Текстово поле за парола
* „Label“- първо име на потребителя (First Name)
* Текстово поле с първото име на потребителя
* „Label“- фамилия на потребителя (Last Name)
* Текстово поле с фамилията на потребителя
* „Label“- адрес (Address)
* Текстово поле с адреса на потребителя
* „Label“- телефонен номер (Telephone number)
* Текстово поле с телефония номер на потребителя
* „Label“- имейл адрес (Email)
* Текстово поле с имейл адресa на потребителя
* Бутон за редакция на потребителските данни (Edit user)
* Бутон за отказ от редакция (Cancel)

***Страница продукти ( Products).***

Чрез тази страница администраторът, може да следи датайли по продуктите (като количество) в системата. Също така може, ако реши да спре продажбата на предлаган артикул, като го деактивира. Страницата се състои от следните контроли:

* „Label“- име на продукт (Product Name)
* Текстово поле с името на продукта
* „Label“- категория (Category)
* Текстово поле с категорията на продукта
* „Label“- цена (Price)
* Текстово поле с цената на продукта
* „Label“- количество (Quantity)
* Текстово поле- за въвеждане на цифри
* Снимка на продукта
* Описание на продукта
* Бутон за спиране на продажбите от този продукт (Disable)
* Бутон за отказ от редакция (Cancel)

# Програмна реализация на системата

## Използвани технологии

### ASP.NET

Технологията ASP.NET е избрана като средство за разработка на системата.

ASP.NET е технология за създаване на уеб приложения и уеб услуги, разработена от Microsoft. За първи път е публикувана през януари 2002 г. с версия 1.0 на .NET Framework, и е наследник на Microsoft Active Server Pages (ASP) технологията, но не е подобрена версия на ASP. ASP.NET е изградена въз основа на Common Language Runtime (CLR), което позволява на програмистите да пишат ASP.NET код като използват .NET език по избор.

ASP.NET цели производителност спрямо останалите скрипт-базирани технологии (включително класическия ASP), като компилира сървърно кода в един или повече DLL файлове на Уеб сървъра. Тази компилация става автоматично, когато страницата бива заредена за пръв път (което от своя страна означава, че програмистът не трябва да изпълнява отделни компилации за страниците). Това комбинира лекотата на разработване, предлагана от скриптовите езици, с производителността на бинарните операции. Трябва да се има предвид обаче, че самата компилация може да причини забележимо забавяне при потребителя, когато редактираната страница бива изискана за първи път от Уеб сървъра, но това забавяне не би се появило отново преди следваща промяна.

ASPX и необходимите ресурси се поставят на виртуален хост на Internet Information Services сървър (или друг сървър, съвместим с ASP.NET). Когато потребителят за първи път поиска да зареди страница, .NET Framework парсва и компилира файловете в .Net асембли и изпраща отговор; при последвалите зареждания се извикват от DLL файловете ASP.NET има възможността да компилира цели сайтове на части от по 1000 файла при първо зареждане. Ако забавянето е значително или причинява проблеми, големината на самите части може да бъде променяна.

Програмистите могат също да изберат да компилират предварително своите файлове, преди поставянето им на сървъра, използвайки MS Visual Studio, елиминирайки нуждата от първоначалното компилиране в уеб средата. Това елиминира и нуждата от качване на изходен код на уеб сървъра. Той поддържа и предварително компилиран текст.

### C#- C sharp

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 един 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# (csc) до изпълним код и в резултат се получават асемблита – файлове със същото име, но с различно разширение (.exe или .dll).

Първата версия на C# е разработена от Microsoft в периода 1999 – 2002 г. и е пусната официално в употреба през 2002 г., като част от .NET платформата, която има за цел да улесни съществено разработката на софтуер за Windows среда чрез качествено нов подход към програмирането, базиран на концепциите за „виртуална машина“ и „управляван код“. По това време езикът и платформата Java, изградени върху същите концепции, се радват на огромен успех във всички сфери на разработката на софтуер и разработката на C# и .NET е естествения отговор на Microsoft срещу успехите на Java технологията. Едно от най-големите предимства на .NET Framework е вграденото автоматично управление на паметта. То предпазва програмистите от сложната задача сами да заделят памет за обектите и да търсят подходящия момент за нейното освобождаване. Това сериозно повишава производителността на програмистите и увеличава качеството на програмите, писани на C#.

### HTML (Hyper Text Markup Language)

HTML е основният маркиращ език за описание и дизайн на уеб страници. HTML е стандарт в Интернет, а правилата се определят от международния консорциум W3C. Текущата версия на стандарта е HTML 5.0 (от 28 октомври 2014 г.)

Описанието на документа става чрез специални елементи, наречени HTML елементи или маркери, които се състоят от етикети или тагове (HTML tags) и ъглови скоби (като например елемента <html>). HTML елементите са основната градивна единица на уеб страниците. Чрез тях се оформят отделните части от текста на една уеб страница, като заглавия, цитати, 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 файлове и се хоства на сървъри, свързани към Интернет. Тези файлове съдържат текстово съдържание с маркери – инструкции за браузъра за това как да се показва текстът. Например <маркер> Някакъв текст. </край на маркера>. Предназначението на уеб браузърите е да могат да прочетат HTML документите и да ги превърнат в уеб страници. Браузърите не показват HTML таговете, а ги използват, за да интерпретират съдържанието на страницата.

Основното предимство на HTML е, че документите, оформени по този начин, могат да се разглеждат на различни устройства, а не само на екрана. Документът може да бъде правилно оформен и върху монитора на персонален компютър, и върху миниатюрния дисплей на пейджър или мобилен телефон.

HTML може да прикрепя скриптове писани на езици като JavaScript, което променя поведението на уеб страницата. Може да се използва Cascading Style Sheets (CSS), който определя изгледа и оформлението на текста и други материали. World Wide Web Consortium (W3C) поддържа и двете CSS и HTML и насърчава използването на CSS в HTML страниците от 1997. Това допринася за разделяне съдържанието и структурата на уеб страниците от тяхното визуално представяне.

Създаването на HTML-базирана уеб страница може да се извърши с помощта на обикновен текстов редактор. Този начин изисква познаване на HTML тагове, така че те да бъдат интегрирани в текста, който ще се показва на страницата. Също така, често срещани са по-приятелски настроените инструменти, които не изискват от потребителя да притежава познания по HTML, което му позволява да създаде страница по метода WYSIWYG. Основен инструмент за тази цел е текстообработваща програма Word, която позволява да запазите документ като HTML и да го редактирате. Специализирани инструменти за създаване на HTML страници са Microsoft, FrontPage, Macromedia Dreamweaver, Notepad, Notepad++,Sublime Text и други.

### CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) е език за описание на стилове (език за стилови листове, style sheet language) – използва се основно за описване на представянето на документ, написан на език за маркиране. Най-често се използва заедно с HTML, но може да се приложи върху произволен XML документ. Официално спецификацията на CSS се поддържа от W3C.

CSS е създаден 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 кода 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.

CSS позволява да се определя как да изглеждат елементите на една HTML страница – шрифтове, размери, цветове, фонове, и др. CSS кодът се състои от последователност от стилови правила, всяко от които представлява селектор, последван от свойства и стойности.

Има три начина на използване на CSS. Първият е да се използва външен CSS файл. В този случай всичкият CSS код се пише във външен файл, към който по-късно се дава линк във HTML страницата. Този начин обикновено се ползва, когато са налице по-дълги и по-обемни CSS кодове. Вторият начин е да се използват извадени CSS кодове, които да бъдат поставени в HTML файловете. Това означава, че всяка страница има свой собствен CSS код, който се намира в нея. И третият начин е да се използва CSS директно на местата където трябва, директно при останалия HTML код.

### SQL (Structured Query Language)

SQL е популярен език за програмиране, предназначен за създаване, видоизменяне, извличане и обработване на данни от релационни системи за управление на бази данни. Стандартизиран е от ANSI / ISO.

Появяването на SQL се свързва с възникването на релационни модели за бази от данни. Публикацията „A Relational Model of Data for Large Shared Data Banks“ на Едгар Ф. Код е публикувана през юни 1970, като черновите били разпространявани вътрешно в IBM през 1969. В нея той описва релационен модел, който впоследствие е приет като „базов“ модел за релационни системи за управление на бази данни. През 70-те години, група от центъра за разработки на IBM в Сан Хосе, разработва комерсиална система за бази от данни „System R“, която се базира на модела на Код, но не много стриктно. Доналд Д. Чембърлейн и Рейнолд Ф. Бойс от IBM разработват езика „SEQUEL“ (Structured English Query Language), който да управлява и извлича данните от System R. Акронимът SEQUEL след това бива съкратен до SQL, защото думата 'SEQUEL' е търговска марка на Hawker-Siddeley- авиационна компания от Обединеното кралство.

Първата некомерсиална, релационна база от данни, която е не-SQL, е Ingres - разработена през 1974г. в Калифорнийски университет, Бъркли.

През 1978, IBM започва методично тестване в клиентски тестови центрове успешно демонстрирайки ползата и практичността на системата. В резултат на това IBM започва да разработва комерсиални продукти базирани на прототипа на тяхната System R осъществяваща SQL, като System/38 (известена през 1978 и достъпна от август 1979), SQL/DS (представена през 1981), и DB2 (през 1983).

По същото време Relational Software, Inc. (сега Oracle Corporation) забеляза потенциала на концепцията, описана от Чембърлейн и Бойс и разработва своя версия на РСУБД за флота на САЩ, ЦРУ и др. През лятото на 1979 компанията Relational Software, Inc. представя Oracle V2 (Версия 2) за VAX миникомпютри, като първата достъпна на пазара реализация на SQL. Oracle често погрешно е хвалена че е изпреварила IBM с две години, докато всъщност те са изпреварили представянето на IBM - System/38 с няколко седмици. Следвайки големия интерес на обществеността много скоро и други доставчици започват да разработват свои версии.

SQL е приет като стандарт от ANSI (American National Standards Institute) през 1986 и от ISO (Международна организация по стандартизация) през 1987.

## Реализация на базата данни

След създаването на концептуалния и логическия модел на базата данни, дойде и времето за направата на физическия модел. Както споменахме по- горе, технологията, която ще използвам е SQL. Чрез SQL заявки ще бъдат създадени таблиците и колоните.

* Създаване на БД

Create database Store;

go

Чрез тези два реда код се създава базата от данни „Store“. Всъщност тя е създадена, но е празна (без никакви таблици и колони в нея)

### Създаване на таблици и колони (заедно с техните ключове) в базата данни

* ***Таблица Users***

use Store;

create table Users(id\_user int identity Primary key,

first\_name varchar(50) Not null,

lastst\_name varchar(50) Not null,

address varchar(50) Not null,

telephone int Not null,

email varchar(50) Not null,

userName varchar(50) Not null,

password varchar(50) Not null)

go

Чрез този код се създава таблица Users, в която ще се съхранява информацията за регистрираните потребители. Тя се състои от следните колони:

* id\_user- уникален номер за всеки регистриран потребител
* first\_name- текст с дължина до 50 символа, задължителен за въвеждане
* last\_name- текст с дължина до 50 символа, задължителен за въвеждане
* address- текст с дължина до 50 символа, задължителен за въвеждане
* telephone- цифри, задължителни за въвеждане
* email- текст с дължина до 50 символа, задължителен за въвеждане
* userName- текст с дължина до 50 символа, задължителен за въвеждане
* password- текст с дължина до 50 символа, задължителен за въвеждане
* ***Таблица Categories***

use Store;

create table Categories(

id\_category int identity Primary key,

gategory\_name varchar(50) Not null)

go

Чрез този код се създава таблица Category, в която ще се съдържат категориите от продукти. Чрез нея потребителите ще могат да добавят/търсят продукти по- лесно. Тя се състои от следните колони:

* id\_category- уникален номер за всяка добавена категория
* category\_name- текст с дължина до 50 символа, задължителен за въвеждане
* ***Таблица Products***

use Store;

create table Products

(id\_product int identity Primary key,

id\_category int Not null,

id\_user int Not null,

product\_name varchar(50) Not null,

price float Not null,

quantity int Not null,

image text,

description text)

go

alter table Products

add constraint FK\_Products\_REFERENCE\_Cat foreign key(id\_category)

references Categories (id\_category)

alter table Products

add constraint FK\_Products\_REFERENCE\_User foreign key(id\_user)

references Users (id\_user)

go

Чрез този код се създава таблица Products, която служи за съхранение на добавените от потребителите продукти в базата данни. Тя се състои от следните колони:

* id\_product- уникален номер за всяка добавена категория
* id\_category- уникалният номер от таблица Category (външен ключ). По този начин продуктите ще се свързват с кетегориите
* id\_user- уникалният номер от таблица Users (външен ключ). По този начин продуктите ще се свързват с потребителите, които са ги добавили или които искат да ги купят.
* product\_name -текст с дължина до 50 символа, задължителен за въвеждане.
* price- цифри, задължителни за въвеждане
* quantity- цифри, задължителни за въвеждане
* image- текст
* description- текст
* ***Таблица Orders***

use Store;

create table Orders

(id\_order int identity Primary key,

id\_user int Not null,

date\_time date,

status text)

go

alter table Orders

add constraint FK\_Orders\_REFERENCE\_Oders\_P foreign key(id\_user)

references Users (id\_user)

go

Чрез този код се създава таблица Orders, която ще съхранява информация за направените поръчки от всеки един потребител. Тя се състои от следните колони:

* id\_order- уникален номер за всяка добавена поръчка
* id\_user- уникален номер от табица Users (външен ключ). По този начин поръчките ще се свързват с потребителите, които са ги направили.
* data\_time- дата
* status – текст
* ***Таблица Carts***

use Store;

create table Carts

(id\_order int Primary key,

id\_product int Not null,

quantity int,

productName text)

go

alter table Carts

add constraint FK\_Carts\_REFERENCE\_Orders foreign key(id\_order)

references Orders (id\_order)

Чрез този код се създава междинната таблица Carts. Тя служи за временно съхранение на продуктите, които потребителят би желал да закупи, но все още не е финализирал своята поръчка. Тя се състои от следните колони:

* id\_order- уникален номер от таблица Orders (външен ключ). По този начин количката ще се свързва с поръчката на потребителя.
* id\_product- уникалният номер от таблица Products (външен ключ). По този начин продуктите ще се свързват с количката на потребителя.
* quantity- цифри, задължителни за въвеждане
* productName- текст

## Реализация на приложението

Въз основа на схематичното представяне на различните елементи, схеми и създадената база данни се започва реализирането на приложението.

Връзка с базата данни. Тъй като за програмирането отговаря една програма (Visual Studio 2019), а за базите данни отговаря друга програма (SQL Server 2014) трябва да се направи връзка между тях. Това обикновено става чрез така наречения connection string.

Автентикация. За автентикация на потребителите ще се работи с така наречените сесии (когато потребител има регистрация и неговите данни съществуват в базата, при влизане в системата ще се създава сесия, която ще „помни“ неговите данни).

Изображения. Всички изображения на продуктите ще се намират в папка „upload“.

### Създаване на дизайн

Дизайнът на цялото приложение е базирано на три шаблона- „master page“.

Main.Master служи за визуализиране на даннити, когато потребителите не са регистрирани, Main2.Master визуализира данните след успешен вход в системата, а AdminMain.Master служи за администраторския панел. В тях се включват следните компоненти:

* Меню
* „Header“
* „Footer“
* Два „ContentPlaceHolder“-а, в които ще се зареждат динамично данните на всички страници.

За да се визуализират горе изброените компоненти във всяка страница се използва следния код:

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Main2.master.cs" Inherits="webStore.Main2" %>

<!DOCTYPE html>

<html>

<head runat="server">

<title></title>

<link href="style/style.css" rel="stylesheet" />

<link href="style/MyCSS.css" rel="stylesheet" />

<script src="script/jquery-3.3.1.js"></script>

<script src="script/bootstrap.js"></script>

<asp:ContentPlaceHolder ID="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

<header>

<div class="navbar navbar-default navbar-fixed-top text-uppercase">

<div class="container">

<div class="navbar-header">

<a href="home.aspx" class="navbar-brand my">Web Store</a>

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

<div class="navbar-collapse collapse">

<ul class="nav navbar-nav navbar-right">

<li class="my"><a href="home.aspx">Home</a></li>

<li class="my dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Categories<span class="caret"></span></a>

<ul class="dropdown-menu" role="menu">

<li><a href="other.aspx">Other</a> </li>

<li><a href="telephones.aspx">Telephones</a> </li>

<li><a href="headphones.aspx">Headphones</a> </li>

<li><a href="mobileAccessories.aspx">Mobile Accessories</a> </li>

</ul>

</li>

<li class="my"><a href="create.aspx">Add Product</a></li>

<li class="my"><a href="myproducts.aspx">My Products</a></li>

<li class="my"><a href="myorder.aspx">My Orders</a></li>

<li class="my"><a href="cart.aspx">Cart</a></li>

<li class="my"><a href="homeNoRegister.aspx">Logout</a></li>

</ul>

</div>

</div>

</div>

</header>

<main>

<asp:ContentPlaceHolder ID="main" runat="server">

</asp:ContentPlaceHolder>

</main>

<footer>

<div class="container modal-footer">

<p>&copy; 2021 - Technical University- Sofia </p>

</div>

</footer>

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</body>

</html>

Този код включва технологията HTML, Bootstrap и CSS

Чрез HTML и CSS се създава визуалното оформление на страниците, a чрез Bootstrap се създава Responsive Design на приложението.

### Реализиране на бизнес логиката

При стартиране на платформата се визуализира началната страница- „Home“ (home.aspx). Тя е базирана на шаблона- „Main.Master“ и използва всички негови функционалности. Допълнителни елементи на началната страница са:

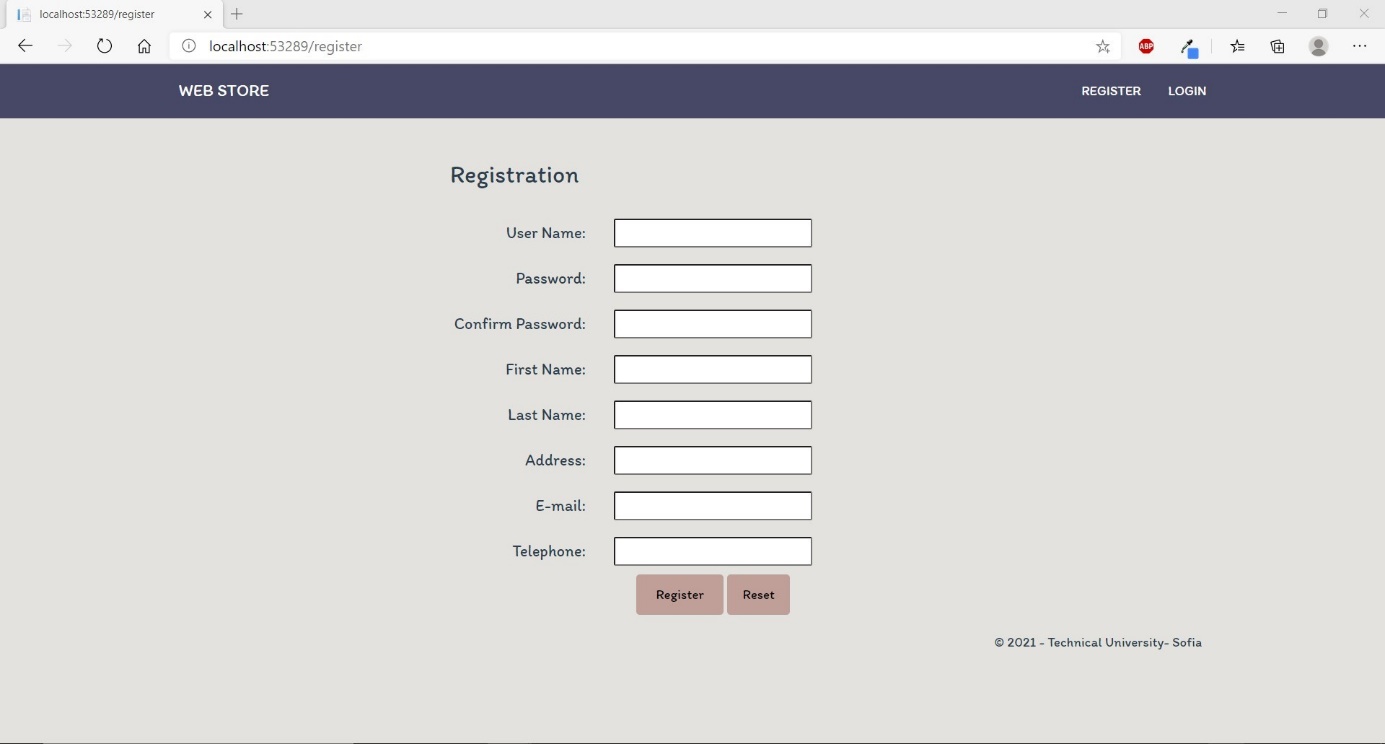
* „Label“- приветствено съобщение
* Grid View- таблица с информация за всички налични продукти в сайта (име на продукта, цена, номер, изображение и бутон за избор (Select))

При кликване на бутона „Select“, платформата препраща към страницата за вход (Login.aspx), която подканя потребителите да влязат в профила си или, ако нямат такъв, съответно да си създадат.

***Страница за регистриране на потребители- „Register“ (register.aspx)***

Базиранa е на шаблона „Main.Master“ и използва всички негови функицоналности. Допълнителни елементи на началната страница са:

* Осем текстови полета, в които потребителят трябва да въведе:
* User Name- потребителско име
* Password- парола
* Confirm Password- потвърждение на паролата
* First Name- име
* Last Name- фамилия
* Address- адрес
* E-mail address- емайл адрес
* Telephone- телефон



Фигура 8- Активна страница за "Регистрация"

* Бутон за нулиране на полетата (Reset)
* Бутон за регистрация (Register)- който при кликване, след коректно въвеждане на всички данни, осъществява регистрацията на потребителя (като чрез connection string осъществява попълването на иформация в базата данни) със следния код:

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["RegistrationStoreConnectionString"].ConnectionString);

conn.Open();

string insertUserQuery= "insert into Users(first\_name,lastst\_name,address,telephone,email,userName,password)" +

"values(@firtName,@lastName,@address,@telephone,@email,@userName,@password)";

comm = new SqlCommand(insert, conn);

comm.Parameters.AddWithValue("@firtName", TextBoxFirstN.Text);

comm.Parameters.AddWithValue("@lastName", TextBoxLastN.Text);

comm.Parameters.AddWithValue("@address", TextBoxAddress.Text);

comm.Parameters.AddWithValue("@telephone", TextBoxTelephone.Text);

comm.Parameters.AddWithValue("@email", TextBoxMail.Text);

comm.Parameters.AddWithValue("@userName", TextBoxUserName.Text);

comm.Parameters.AddWithValue("@password", TextBoxPassword.Text);

comm.ExecuteNonQuery();

Response.Write("Registration is successful");

Response.Redirect("home.aspx");

conn.Close();

Чрез инициализирането на променливата „conn“ от тип „SqlConnection“ се създава връзката с базата данни (connection string). След успешната инициализация, за да може да се осъществи предаване на данни с базата е необходимо тази връзка да има начало (да бъде отворена чрез „conn.Open()“). Променливата „insertUserQuery“, която е от тип „string“, представлява заявката, която ще се извършва, а именно добавяне на въведените данни от потребителя.

След като вече успешно сме се регистрирали, се изършва нова заявка към базата. Чрез нея се взимат данни за името на потребителя и неговият идентификационен номер. Тази информация се използва за създаването на сесии. След приключване на процеса, връзката с базата данни трябва да бъде затворена („conn.Close();“) и потребителят бива препратен към началната страница.

string checkUserIdQery = "select id\_user from Users where userName='"+ TextBoxUserName.Text + "'";

SqlCommand idUserComm = new SqlCommand(checkUserIdQery, conn);

string userID = idUserComm.ExecuteScalar().ToString().Replace(" ","");

Session["UserName"] = TextBoxUserName.Text;

Session["UserId"] = userID;

Response.Redirect("home.aspx");

conn.Close();

След влизане в профила, някои от страниците на лентата с менюто се заменят от други. Основен шаблон става „Main2.Master“. Визуализират се и нови допълнителни страници:

***Страница за влизане в профила – „Login“ (login.aspx)***

Страницата също е базирана на шаблона „Main.Master“ и използва всички негови функционалности. Допълнителни елементи на тази страница са:

* Две текстови полета
* User Name- потребителско име
* Password- парола
* Бутон „Login“ – бутон за вход в системата.
* Хиперлин- линк, който препраща към страницата за регистрация, ако посетителят на сайта няма такава.

След кликване на бутона за вход, чрез методи се проверява дали в базата данни съществува такъв потребител с такава парола. Ако потребителското име и/или парола са грешни или несъществуват, на екрана се показва съобщение за некоректно въведени данни. След успешен вход в системата се създава сесия, която служи за автентикация на потребителя.

За да се реализира функционалността на менюто и автентикацията на потребителите се използва следния код:

public partial class login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Create session

Session["UserName"] = null;

Session["Price"] = 0.0;

}

protected void ButtonLogin\_Click(object sender, EventArgs e)

{

//Create db connection for new user registration

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["RegistrationStoreConnectionString"].ConnectionString);

conn.Open();

string selectUsersQuery = "select count(\*) from Users where userName='" + TextBoxUserName.Text + "'";

SqlCommand comm = new SqlCommand(selectUsersQuery, conn);

int userExistsValue = Convert.ToInt32(comm.ExecuteScalar().ToString());

conn.Close();

//Verify user

if (userExistsValue == 1)

{

LabelPass.Visible = false;

LabelUser.Visible = false;

conn.Open();

string checkPasswordQery = "select password from Users where userName='" + TextBoxUserName.Text + "'";

SqlCommand passComm = new SqlCommand(checkPasswordQery, conn);

string password = passComm.ExecuteScalar().ToString().Replace(" ", "");

string checkUserIdQery = "select id\_user from Users where userName='" + TextBoxUserName.Text + "'";

SqlCommand idUserComm = new SqlCommand(checkUserIdQery, conn);

string userID = idUserComm.ExecuteScalar().ToString().Replace(" ", "");

if (password == TextBoxPassword.Text)

{

//Create Session

Session["UserName"] = TextBoxUserName.Text;

Session["UserId"] = userID;

if (TextBoxUserName.Text == "admin")

{

//Redirect to admin panel

Response.Write("Password is correct");

Response.Redirect("homeAdmin.aspx");

}

else

{

//Redirect to home page

Response.Write("Password is correct");

Response.Redirect("home.aspx");

}

}

else

{

LabelPass.Visible = true;

}

}

else

{

LabelUser.Visible = true;

}

}

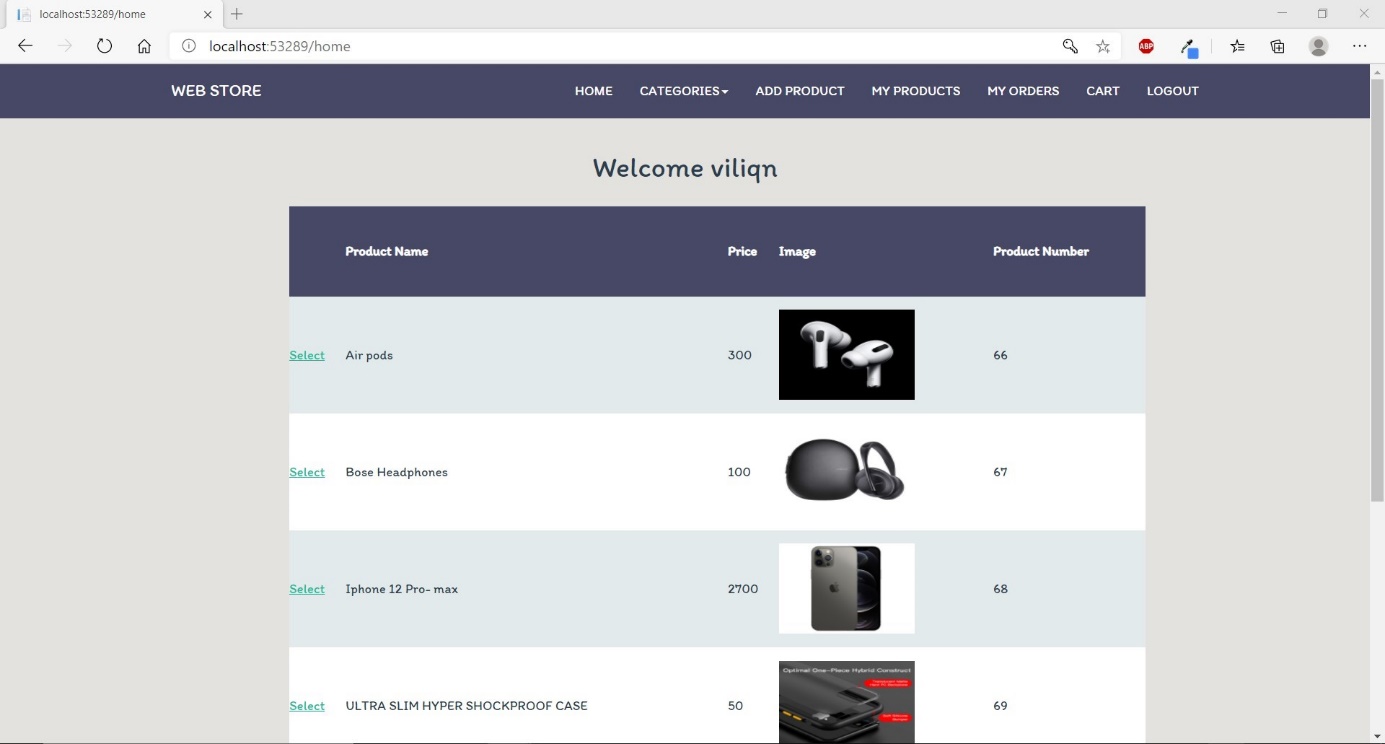
Чрез инициализирането на променливата „conn“ от тип „SqlConnection“ се създава връзката с базата данни (connection string). Чрез променливите „checkUserIdQuery“ и „checkPasswordQuery“ се осъществяват селект заявки, които проверяват дали въведените потребителско име и парола съществуват в базата данни. Ако съществуват, се създават сесии „Session["UserId"] = userID“ и „Session["UserName"] = TextBoxUserName.Text“, които запазват данни за потребителя и служат за автентикацията.

След влизане в профила, някои от страниците на лентата с менюто се заменят от други. Основен шаблон става „Main2.Master“. Визуализират се и нови допълнителни страници:

* Вместо страница „Login“ се визуализира „Logout“, която след кликване осъществява изход от профила
* Вместо страница „Registration“ се визуализира страница „Cart“
* Новодобавената страница „Home“
* Новодобавената страница „Categories”е падащо меню
* Новодобавената страница „Add Product“
* Новодобавената страница „My products“
* Новодобавената страница „Orders“

***Начална страница (home.aspx)***

След успешен вход в системата тази страница вече е базирана на шаблона- „Main2.Master“. Тя служи за визуализиране на всички налични продукти в системата. Благодарение на автентикацията, вече приветственото съобщение съдържа и името на потребителя.



Фигура 9- Активна страница "Начало"

За да се извлекат всички необходими данни за визуализиране на продуктите, се използва контролера „asp:SqlDataSource“, който представлява SQL connection. След това данните се поставят в друг контролер „asp:GridView“, където се визуализират.

След кликване на бутона Select, отново благодарение на сесията този път системата не ни препраща към login.aspx, а към страница с детайлите на конкретен продукт.

***Страница за детайли на продукт (details.aspx)***

Базиранa е на шаблона- „Main2.Master“ и използва всички негови функицоналности. Допълнителни елементи на страницата са:

* Пет текстови полета:
* Product Name- име на продукт
* Category- категория
* Price- цена
* Quantity- количество
* Description- описание
* Контролер за визуализиране на изображения
* Asp:Image

При зареждането на тази страница се случват следните процеси:

// Verify auth

if (Session["UserName"] == null || Session["UserId"] == null)

{

Response.Redirect("login.aspx");

}

// Verify product select

if (Session["ProductNumber"] == null)

{

Response.Redirect("home.aspx");

}

* Проверява се чрез сесията дали потребителят е регистриран. Ако сесията е null, препраща към страницата за вход (login.aspx)
* Проверява се чрез сесия, дали съществува в базата данни, избраният от потребителя продукт. Ако не съществува, препраща към началната страница (home.aspx)

След като тези проверки минат успешно, зареждането на страницата продължава:

// Load data from db

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ProductsConnectionString"].ConnectionString);

int productID = int.Parse(Session["ProductNumber"].ToString());

conn.Open();

// Get Product Name

string selectCommand = "select product\_name from Products where id\_product='" + productID + "'";

SqlCommand comm = new SqlCommand(selectCommand, conn);

string name = comm.ExecuteScalar().ToString();

//Get Price

selectCommand = "select price from Products where id\_product='" + productID + "'";

comm = new SqlCommand(selectCommand, conn);

string price = comm.ExecuteScalar().ToString().Replace(" ", "");

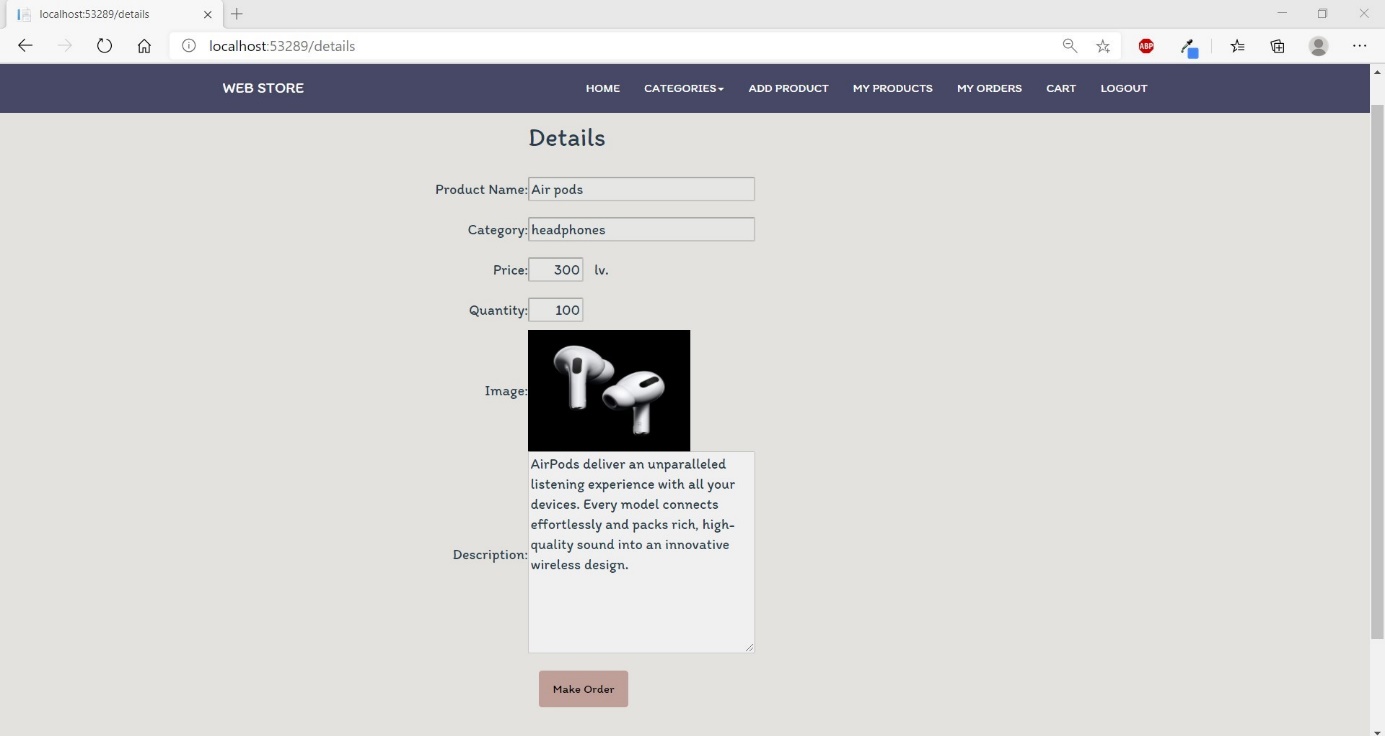
//Get Quantity

selectCommand = "select quantity from Products where id\_product='" + productID + "'";

comm = new SqlCommand(selectCommand, conn);

string quantity = comm.ExecuteScalar().ToString().Replace(" ", "");

Този код е част от селект заявка, която взема информация за конкретен продукт от базата данни, в резултат на което на екрана се визуализира следното:

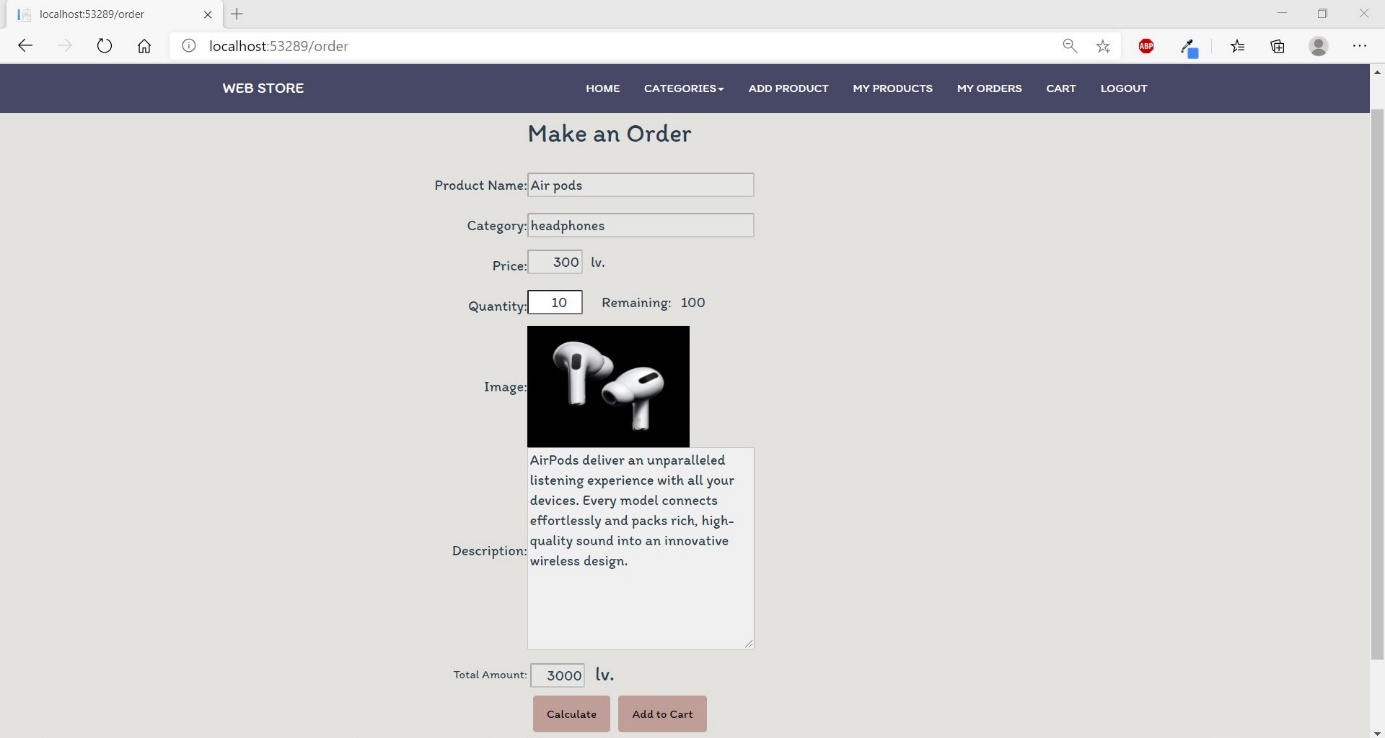


Фигура 10- Активна страница за детайли на продукт

При натискане на бутона „Make Order“ се създава нова сесия, която вече служи за запомняне на номера на поръчката, след което се препраща към страница „поръчка“ (order.aspx)

***Страница „Поръчка“ (Order.aspx)***

Базиранa е на шаблона „Main2.Master“ и използва всички негови функицоналности. Визуализира се на екрана аналогична страница на (details.aspx), но с възможност за въвеждане на желано количество от потребителя. Също така и с нови два бутона „калкулация“ (Calculate) и „добавяне в количка“ (Add to cart).



Фигура 11- Активна страница за поръчка на продукт

Зареждането на информацията в страницата е аналогично с това на details.aspx.

При натискане на бутона „калкулация“ (Calculate), в зависимост от наличната информация за количествата и въведените от потребителя се прави следната проверка :

int realQuantity = int.Parse(quantity) - int.Parse(orderedQuantity);

int needQuant = int.Parse(TextBoxQuantity.Text);

double money = double.Parse(price);

if (needQuant > realQuantity)

{

Label7.Visible = true;

return;

}

else

{

double sum = money \* needQuant;

TextBoxTotal.Text = sum.ToString();

TextBoxTotal.Visible = true;

Label5.Visible = true;

Kъдето, ако променливата „neеdQuant“ (въведена от потребителя) е по- голяма от променливата „realQuantity“- на екрана се визуализира съобщение за грешка. Ако двете променливи са равни или „neеdQuant“ е по- малко от „realQuantity“, то тогава се извършва успешна калкулация и на екрана се визуализира сумата, която потребителят би трябвало да плати.

При натискането на бутона „добави в количка“ (Add to cart) се прави споменатата по- горе проверка.

//Session for Total Price

double totalForThisProducts = sum;

double temp = double.Parse(Session["Price"].ToString());

double sumTotal = totalForThisProducts + temp;

Session["Price"] = sumTotal;

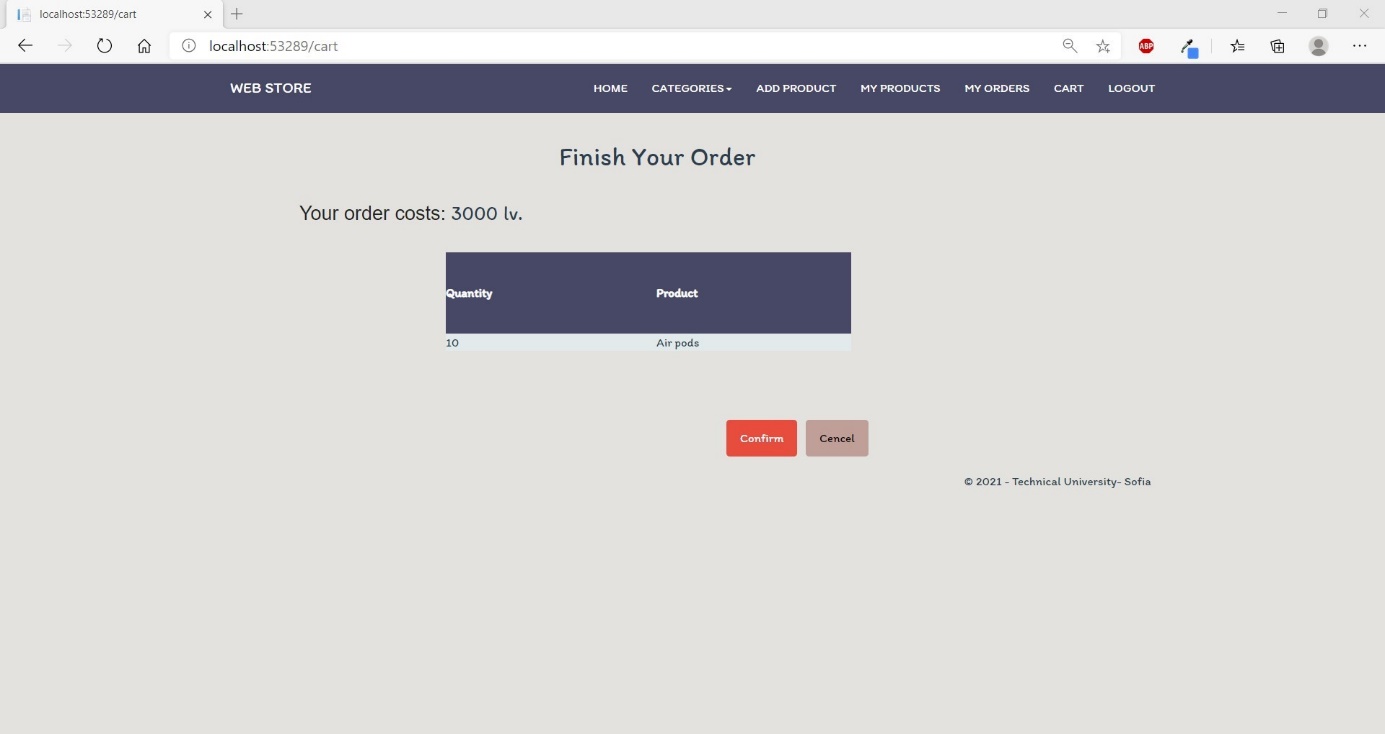
След като проверката е минала успешно се създава сесия, която запазва сумата, количеството желани продукти и името на желания продукт. След това тези стойности се запазват в базата данни (таблица Cart) и се препраща към началната стрaница, където потребителят може да продължи да избира други продукти за своята количка.

***Страница „Количка“ (cart.aspx)***

Базиранa е на шаблона „Main2.Master“ и използва всички негови функицоналности. Допълнителни елементи на страницата са:

* „Label“- който показва общата стойност на всички артикули добавени в количката
* „Grid Viеw“- таблица с количествата и имената на избраните продукти
* Бутон за потвърждение и финализиране на поръчката (Confirm)
* Бутон за отказ от поръчката (Cancel)

След като потребителят реши, че няма да добавя повече продукти в количката, при натискане на „Cart“ от менюто на екрана се визуализира следната страница:



Фигура 12- Активна страница "Количка"

LabelFinalPrice.Text = Session["Price"].ToString();

double price = double.Parse(Session["Price"].ToString());

Отново чрез сесиите, които бяха създадени по-горе, в „Label“ се изписва и крайната дължима сума от потребителя.

Таблицата с количествата и имената на продуктите също е резултат на тези сесии, както и на Select заявки от базата данни.

При натискане на бутона отказ „Cancel“, потребителят се отказва от своята поръчка. В резултат на това събитие се изпълнява следния код:

//Delete carts

int orderID = int.Parse(Session["Order"].ToString());

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["CartConnectionString"].ConnectionString);

conn.Open();

string comand = "delete from Carts where id\_order='" + orderID + "'";

SqlCommand comm = new SqlCommand(comand, conn);

comm.ExecuteNonQuery();

conn.Close();

//Delete order

conn.Open();

comand = "delete from Orders where id\_order='" + orderID + "'";

comm = new SqlCommand(comand, conn);

comm.ExecuteNonQuery();

conn.Close();

// Clear sessions

Session["Price"] = 0.0;

Session["Order"] = null;

Response.Redirect("home.aspx");

Сесията, служеща за запомняне на общата крайна цена се занулява (Session["Price"] = 0.0;), както и сесията за добавените в количката продукти (Session["Order"] = null;). Също така се извършват и Delete заявки, които изтриват запазената информация за тази поръчка.

При натискането на бутона за потвърждение „Confirm“, потребителят потвърждава своата поръчка. В резултат на това се изпълнява следния код:

int orderID = int.Parse(Session["Order"].ToString());

string status = "confirm";

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["OrderConnectionString"].ConnectionString);

conn.Open();

string comand = "update Orders set status='" + status + "' where id\_order='" + orderID + "'";

SqlCommand comm = new SqlCommand(comand, conn);

comm.ExecuteNonQuery();

conn.Close();

Session["Price"] = 0.0;

Session["Order"] = null;

Response.Redirect("myorder.aspx");

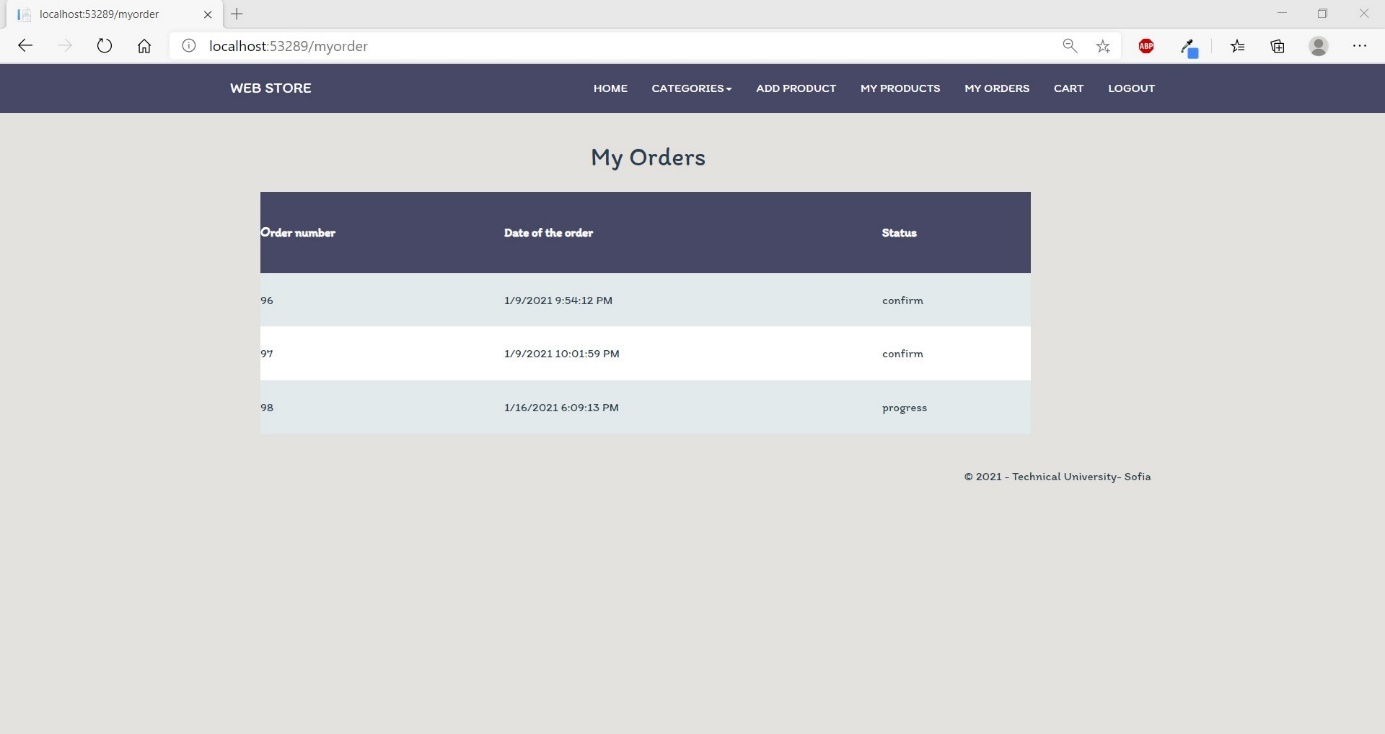
В резултат на този код се осъществява крайното записване в базата данни за конкретната поръчка. Също така тук се извършва изваждането на желаните количества от наличните за даден продукт. Ако след това изваждане, наличните количества са равни на нула, то автоматично обявата за тази стока се деактивира и тя вече не е видима за потребителите. Друг важен момент е, че сесията, служеща за запомняне на общата крайна цена се занулява (Session["Price"] = 0.0;), както и сесията за добавените в количката продукти (Session["Order"] = null;).

След приключването на тези действия се препраща към страницата „Мойте поръчки“ (My Order)( Response.Redirect("myorder.aspx");

***Страница „Мойте поръчки“ (myorder.aspx)***

Базиранa е на шаблона „Main2.Master“ и използва всички негови функицоналности. Допълнителни елементи на страницата са:

* „Grid View“- таблица, визуализираща номерата на поръчките, дата на коята са извършени поръчките и техният статус



Фигура 13- Активна страница "Мойте поръчки"

Тази страница има за цел да информира потребителя за историята на неговите поръчки. За да се извлекат всички необходими данни за визуализиране на тази информация, се използва контролера „asp:SqlDataSource“, който представлява SQL connection. След това данните се поставят в друг контролер- „asp:GridView“, където се визуализират .

***Страница „Добавяне на продукт“ (create.aspx)***

Базиранa е на шаблона „Main2.Master“ и използва всички негови функицоналности. Стрницата е предназначена за потребители, които искат да развиват търговия и да продават своите продукти. Допълнителни елементи на страницата са:

* Четири текстови полета:
* Product Name- Име на продукт
* Price- Цена (използва „Regular Expression“- въвеждането на различна стойност от числа, би довело до извеждане на екрана за некоректно въведени данни)

//Price validation

string patern = @"(^\d+(,\d{2})\*(\.\d{1,2})?$)";

* Quantity- Количество
* Description- Описание
* Контролер asp:FileUploadImages- Чрез този контролер се осъществява добавянето на снимки в системата

// Insert into images

if (this.FileUploadImages.HasFile)

{

double fileSize = FileUploadImages.PostedFile.ContentLength;

double check = fileSize / 1024 / 1024;

if (check > 1)

{

this.Label1.Text = "File size exceeds 1 megabyte!";

Label1.Visible = true;

}

else

{

Label1.Visible = false;

string APP\_PATH = System.Web.HttpContext.Current.Request.ApplicationPath.ToLower();

if (APP\_PATH == "/") //a site

APP\_PATH = "/";

else if (!APP\_PATH.EndsWith(@"/")) //a virtual

APP\_PATH += @"/";

string it = System.Web.HttpContext.Current.Server.MapPath(APP\_PATH);

if (!it.EndsWith(@"\"))

it += @"\";

string fileName = Guid.NewGuid().ToString(); //това генерира уникално име за файла

string ext = Path.GetExtension(this.FileUploadImages.FileName);

string savePath = it + @"upload\" + fileName + ext;

this.FileUploadImages.SaveAs(savePath); //това записва файла на диска

string savep = @"upload/" + fileName + ext; //това е за базата данни стойността

* Падащ списък- Този падащ списък е базиран на контролера „asp:SqlDataSource“, който представлява SQL connection и служи за избор на категория. Даните, които се съдържат в този списък се извличат от таблица „Categories“ от базата данни.
* Бутон „Отказ“ (Cancel)- при натискане на този бутон, потребителят отказва да добавя продукт, след което бива препратен към началната страница

(onclick="window.location.href='home.aspx'")

* Бутон „Добави продук“ (Add product)- който при кликване, след коректно въвеждане на всички данни, осъществява добавянето на продукт в базата данни (като чрез connection string се осъществяват Insert заявки) чрез следния код:

// Insert into Products table

int visible = 0;

conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ProductsConnectionString"].ConnectionString);

conn.Open();

string insertQuery = "insert into Products(id\_category,id\_user,product\_name,price,quantity,image,description,visible)" +

"values(@categoryId,@userId,@productName,@price,@quantity,@image,@description,@visible)";

SqlCommand comm = new SqlCommand(insertQuery, conn);

comm.Parameters.AddWithValue("@categoryId", categoryID);

comm.Parameters.AddWithValue("@userId", userID);

comm.Parameters.AddWithValue("@productName", TextBoxProdName.Text);

comm.Parameters.AddWithValue("@price", price);

comm.Parameters.AddWithValue("@quantity", TextBoxQuantity.Text);

comm.Parameters.AddWithValue("@image", savep);

comm.Parameters.AddWithValue("@description", TextBoxDescription.Text);

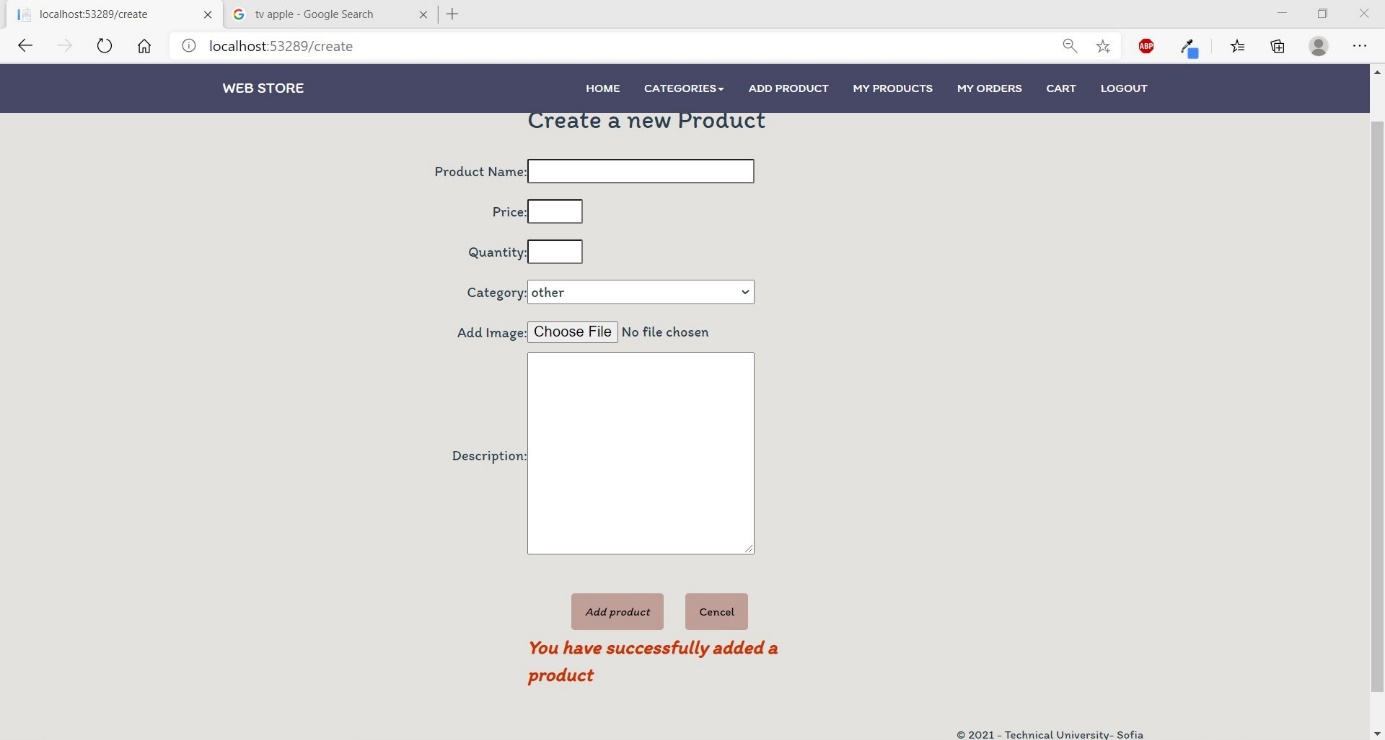
comm.Parameters.AddWithValue("@visible", visible);

comm.ExecuteNonQuery();

conn.Close();

Алгоритъмът за добавяне в таблицата е аналогичен с този от страницата за регистрация описан по- горе.

След като потребителят успешно е добавил продукт в системата, на екрана се визуализира съобщение, което гласи, че всичко е минало успешно.



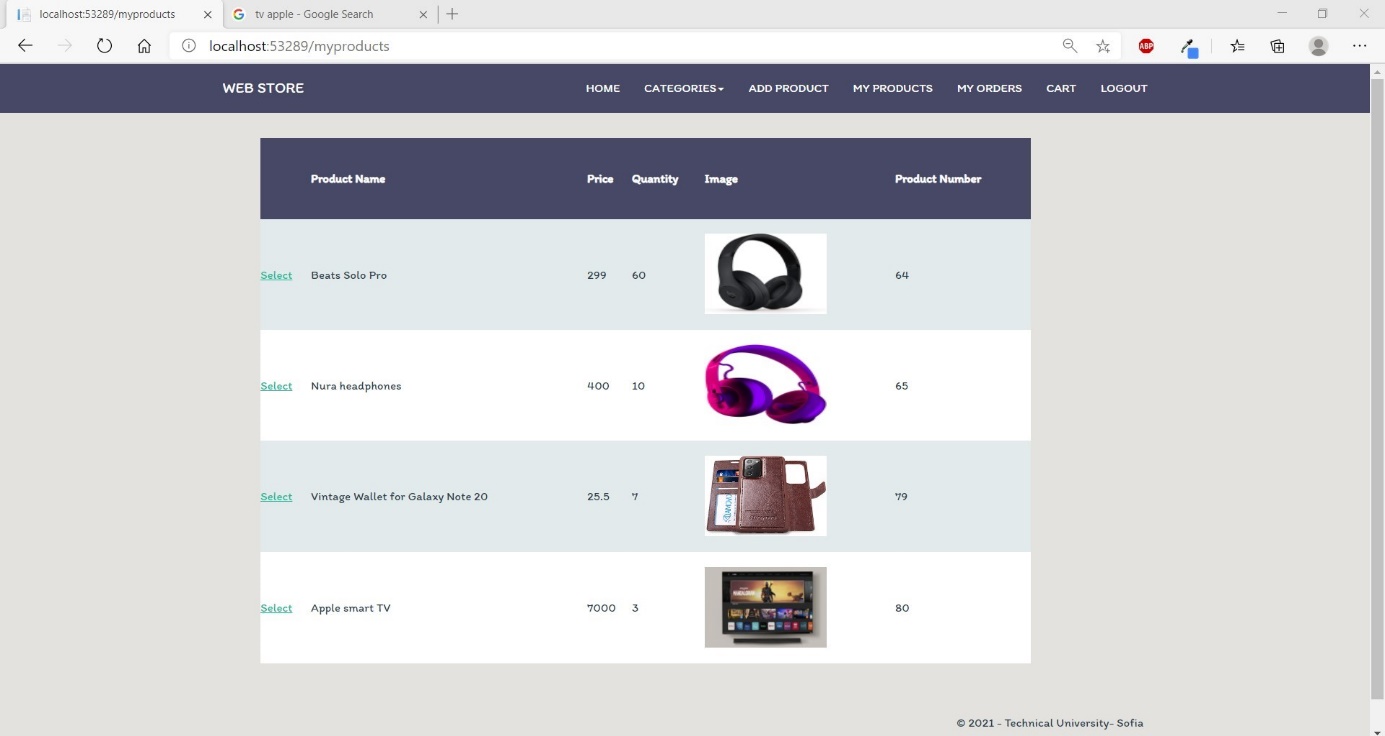
Фигура 14- Активна страница "Добавяне на продукт"

***Страница „Моите продукти“ (myproducts.aspx)***

Базиранa е на шаблона „Main2.Master“ и използва всички негови функицоналности. Стрницата е предназначена за потребителите, които имат добавени продукти в системата. Тя визуализира всички добавени обяви, които са активни. Допълнителни елементи са:

* „Grid Viеw“- таблица с информация за активните продукти, добавени от потребителя (име на продукта, цена, номер, изображение и бутон за избор(Select))

За да се извлекат всички необходими данни за визуализиране на продуктите, се използва контролера „asp:SqlDataSource“, който представлява SQL connection. След това данните се поставят в друг контролер- „asp:GridView“, където се визуализират.



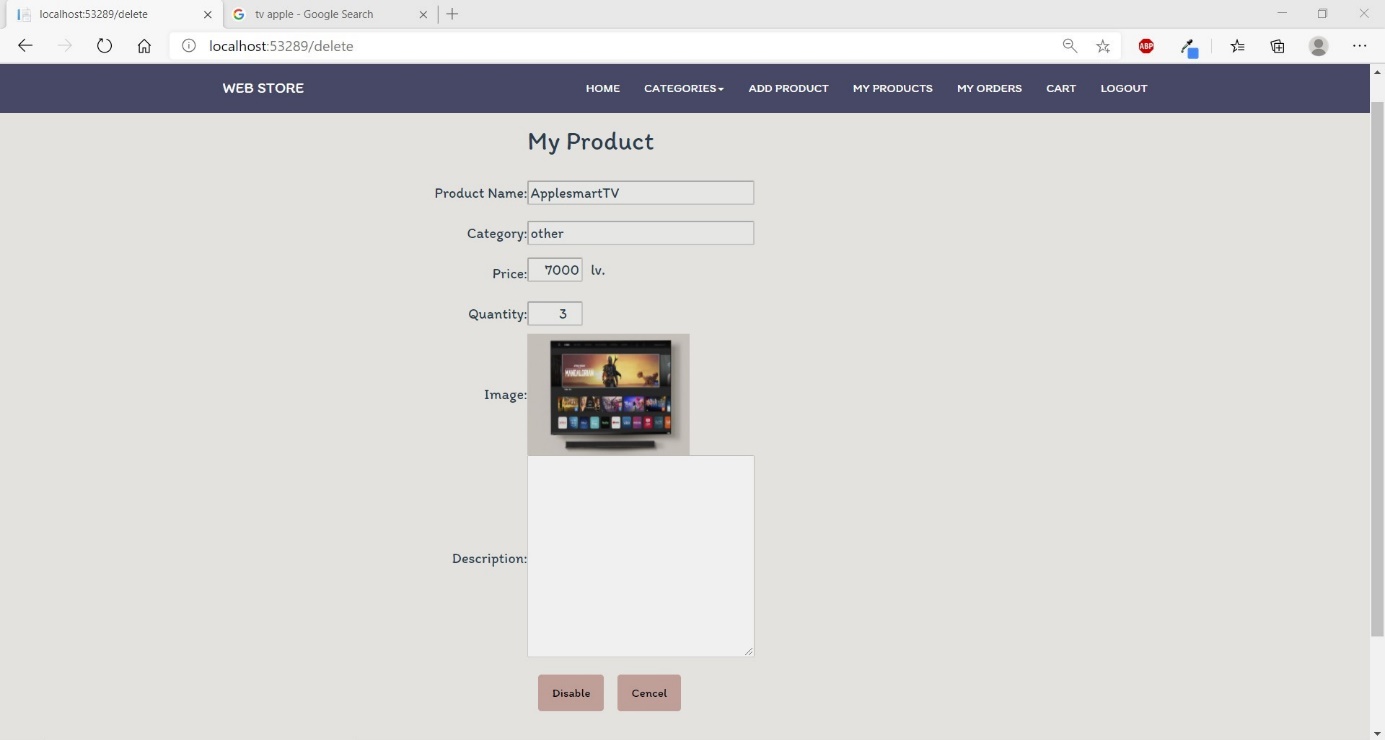
Фигура 15- Активна страница "Мойте продукти"

След кликване на бутона Select, отново благодарение на сесията, се визуализира страницата delete.aspx

***Страница за контрол на обявите (delete.aspx)***

Базиранa е на шаблона „Main2.Master“ и използва всички негови функицоналности. Стрницата е предназначена за визуализиране на конкретен продукт, добавен от потребителя. Чрез нея може да се следи състоянието на продукта (наличните количества). Допълнителни елементи на страницата са:

* Пет текстови полета:
* Product Name- Име на продукта
* Category- Категория
* Price- Цена
* Quantity- Количество
* Description- Описание
* Контролер за визуализиране на изображения
* Asp:Image
* Бутон за отказ от промени „Cancel“- след кликването на този бутон, потребителят бива препращан към началната страница на приложението (onclick="window.location.href='home.aspx'")
* Бутон за деактивиране на обявата „Disаblе“- при натискане на този бутон обявата се деактивира и тя вече не се визуализира в системата



Фигура 16- Активна страница "Моите продукти- детайли"

int productID = int.Parse(Session["ProductNumber"].ToString());

int visible = 1;

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ProductsConnectionString"].ConnectionString);

conn.Open();

string updateQuery = "update Products set visible='" + visible + "' where id\_product='" + productID + "'";

SqlCommand comm = new SqlCommand(updateQuery, conn);

comm.ExecuteNonQuery();

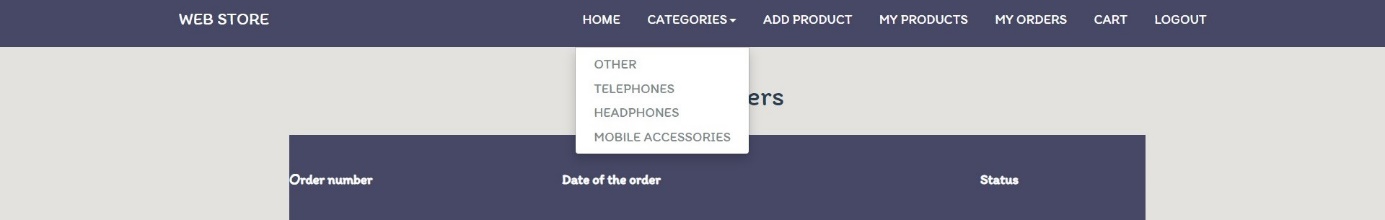
conn.Close();

Response.Redirect("myproducts.aspx");

Чрез този код се осъществява деактивирането на обява, където чрез Update заявка (string comand = "update Products set visible='" + visible + "' where id\_product='" + idProduct + "'") данните в таблицата „Products“ се манипулират. Чрез променливата „int visible = 1;“ се променя стойността на избраният продукт и се деактивира.

***Categories***

Categories представлявападащ списък, който служи като филтър за търсене на продукти по категория. При избиране на някой елемент от списъка, на екрана се визуализира аналогична страница на началната (Home.aspx). Разликата е в това, че вместо да се показват всички продукти в системата, се зареждат само продукти от избраната категория.



Фигура 17- Активно падащо меню за "Категории"

За създаването на такъв списък, промени се правят по шаблонът- „Main2.Master“ чрез следния HTML код:

<ul class="nav navbar-nav navbar-right">

<li class="my"><a href="home.aspx">Home</a></li>

<li class="my dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Categories<span class="caret"></span></a>

<ul class="dropdown-menu" role="menu">

<li><a href="other.aspx">Other</a> </li>

<li><a href="telephones.aspx">Telephones</a> </li>

<li><a href="headphones.aspx">Headphones</a> </li>

<li><a href="mobileAccessories.aspx">Mobile Accessories</a> </li>

</ul>

***Вход като Администратор***

След въвеждането на специални потребител и парола (userName: admin, Password: admin), се прави проверка, която препраща към администраторския панел

//Create Session

Session["UserName"] = TextBoxUserName.Text;

Session["UserId"] = userID;

if (TextBoxUserName.Text == "admin")

{

//Redirect to admin panel

Response.Write("Password is correct");

Response.Redirect("homeAdmin.aspx");

}

else

{

//Redirect to home page

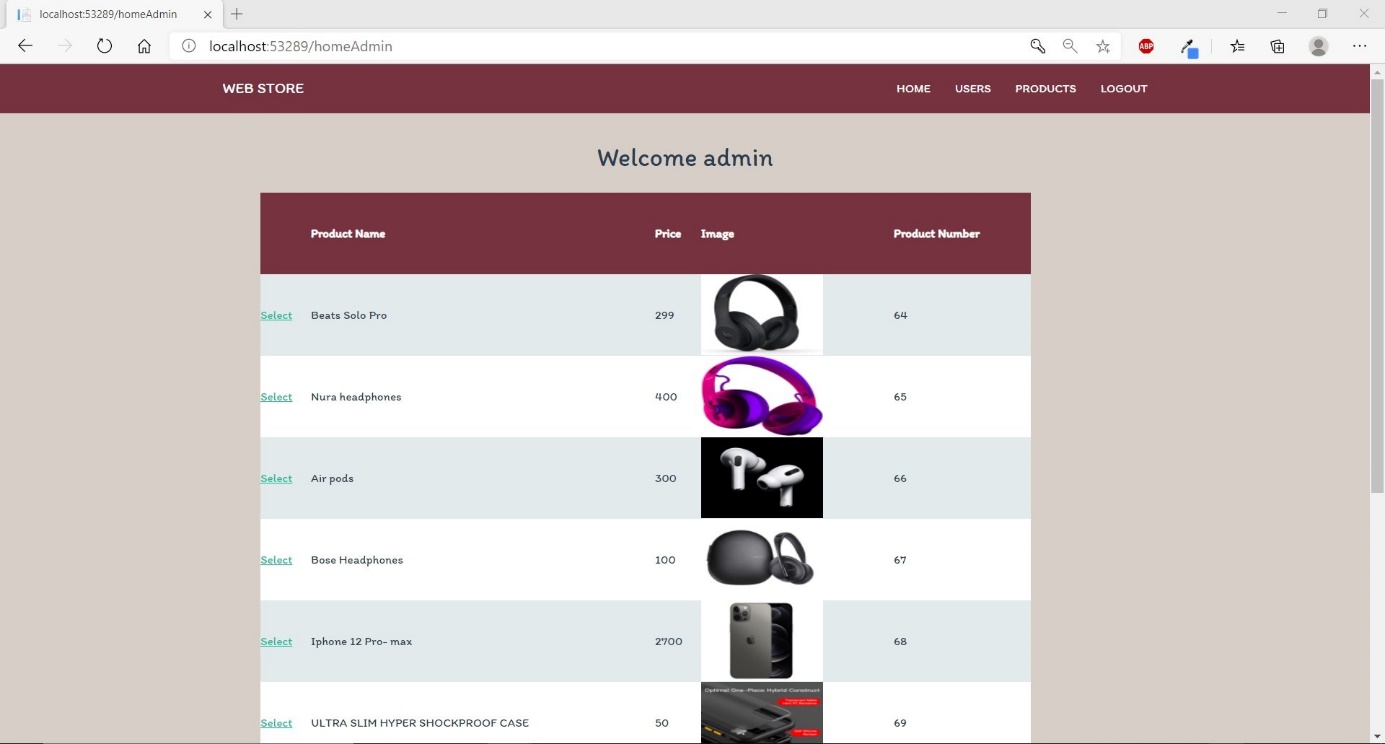
Response.Write("Password is correct");

Response.Redirect("home.aspx");

}

***Начална страница администраторски панел (homeAdmin.aspx)***

След успешен вход в системата тази страница вече е базирана на шаблона- „AdminMain.Master“. Тя служи за визуализиране на всички налични продукти в системата.



Фигура 18- Активна страница "Начало админ панел"

За да се извлекат всички необходими данни за визуализиране на продуктите, се използва контролера „asp:SqlDataSource“, който представлява SQL connection. След това данните се поставят в друг контролер „asp:GridView“, където се визуализират.

След кликване на бутона Select, отново благодарение на сесията ни препраща към страница с детайлите на конкретен продукт.

***Страница за детайли на продукт администраторски панел (detailsProduct.aspx)***

Базиранa е на шаблона- „AdminMain.Master“ и използва всички негови функицоналности. Допълнителни елементи на страницата са:

* Пет текстови полета:
* Product Name- име на продукт
* Category- категория
* Price- цена
* Quantity- количество
* Description- описание
* Контролер за визуализиране на изображения
* Asp:Image
* Проверява се чрез сесия, дали съществува в базата данни, избраният от администратора продукт. Ако не съществува, препраща към началната страница (homeAdmin.aspx)

След като тези проверки минат успешно, зареждането на страницата продължава:

// Load data from db

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ProductsConnectionString"].ConnectionString);

int productID = int.Parse(Session["ProductNumber"].ToString());

conn.Open();

// Get Product Name

string selectCommand = "select product\_name from Products where id\_product='" + productID + "'";

SqlCommand comm = new SqlCommand(selectCommand, conn);

string name = comm.ExecuteScalar().ToString();

//Get Price

selectCommand = "select price from Products where id\_product='" + productID + "'";

comm = new SqlCommand(selectCommand, conn);

string price = comm.ExecuteScalar().ToString().Replace(" ", "");

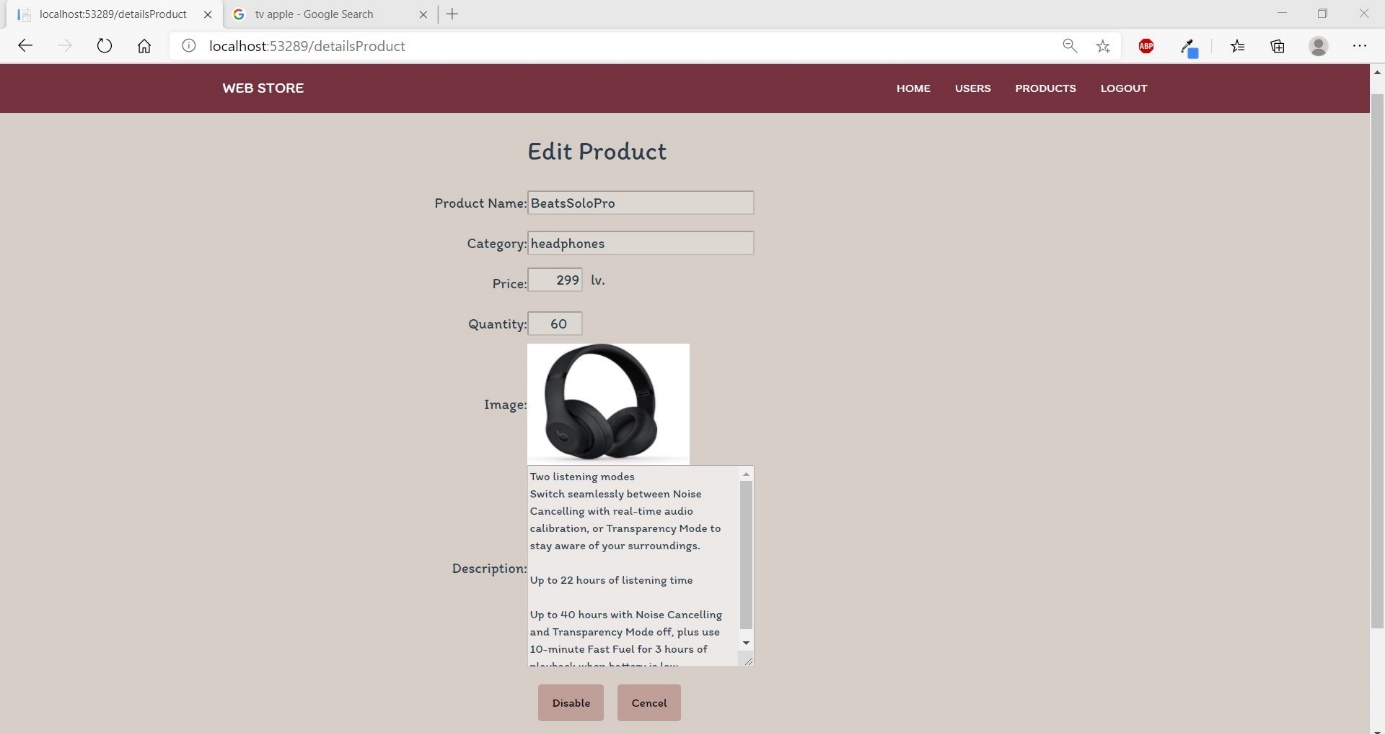
//Get Quantity

selectCommand = "select quantity from Products where id\_product='" + productID + "'";

comm = new SqlCommand(selectCommand, conn);

string quantity = comm.ExecuteScalar().ToString().Replace(" ", "");

Този код е част от селект заявка, която взема информация за конкретен продукт от базата данни, в резултат на което на екрана се визуализира следното:



Фигура 19- Активна страница за детайли на продукт- администраторски панел

При натискане на бутона „Disable“ администраторът може да деактивира продукта. Логиката и изпълнението на тази функционалност е аналогична както при страницата delete.aspx

***Страница потребители (users.aspx)***

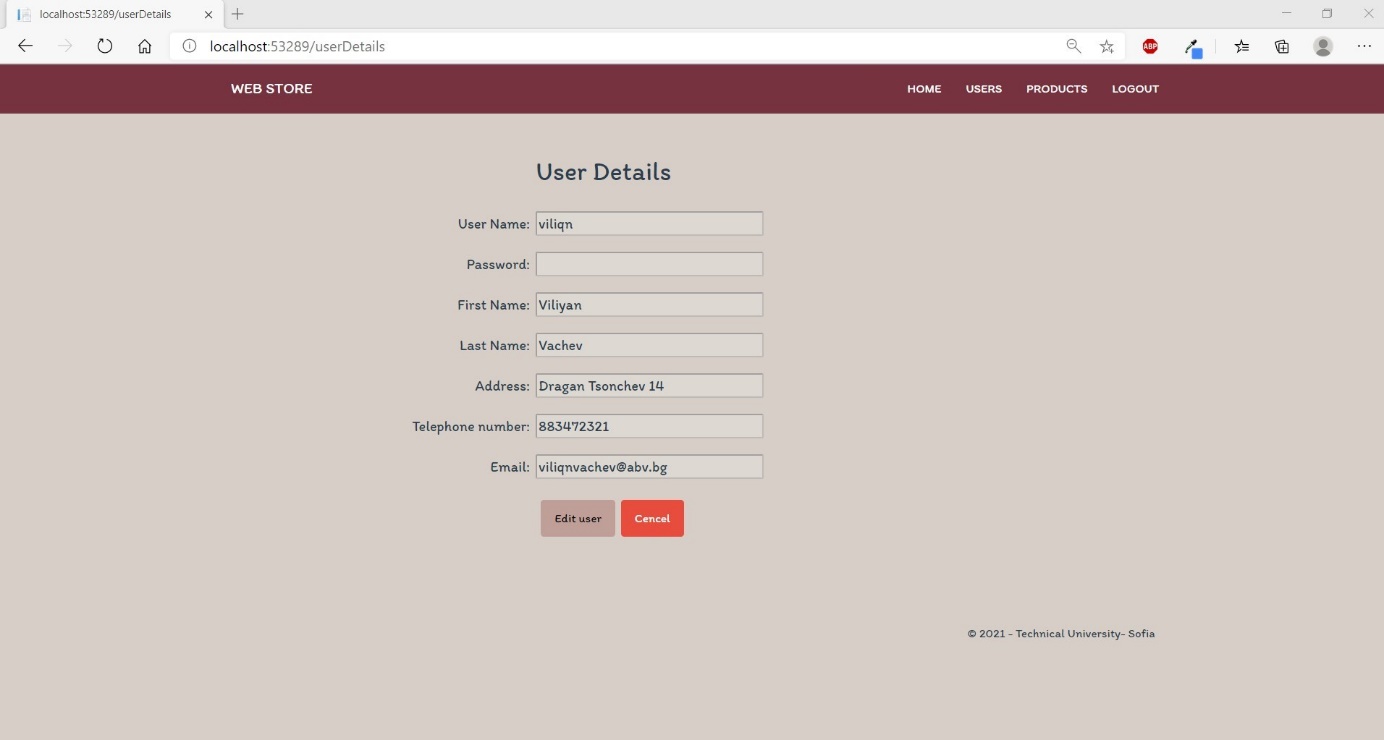
Базиранa е на шаблона- „AdminMain.Master“ и използва всички негови функицоналности. Тя служи за визуализиране на всички налични потребители в системата. За да се извлекат всички необходими данни за визуализиране на потребителите, се използва контролера „asp:SqlDataSource“, който представлява SQL connection. След това данните се поставят в друг контролер „asp:GridView“, където се визуализират.

След кликване на бутона Select, отново благодарение на сесията ни препраща към страница с детайлите на конкретения потребител.

***Страница детайли на потребителя (userDetails.aspx)***

Базиранa е на шаблона- „AdminMain.Master“ и използва всички негови функицоналности. Тя служи за визуализиране на всички налични данни за конкретен потребител в системата. Допълнителни елементи на страницата са:

* Седем текстови полета:
* User Name- име на потребителя
* Password- парола
* First Name- собствено име
* Last Name- фамилия
* Address- адрес
* Telephone number- телефонен номер
* Email- имейл адрес
* Бутон за промяна на потребителя „Edit user“
* Бутон за отказ „Cancel“



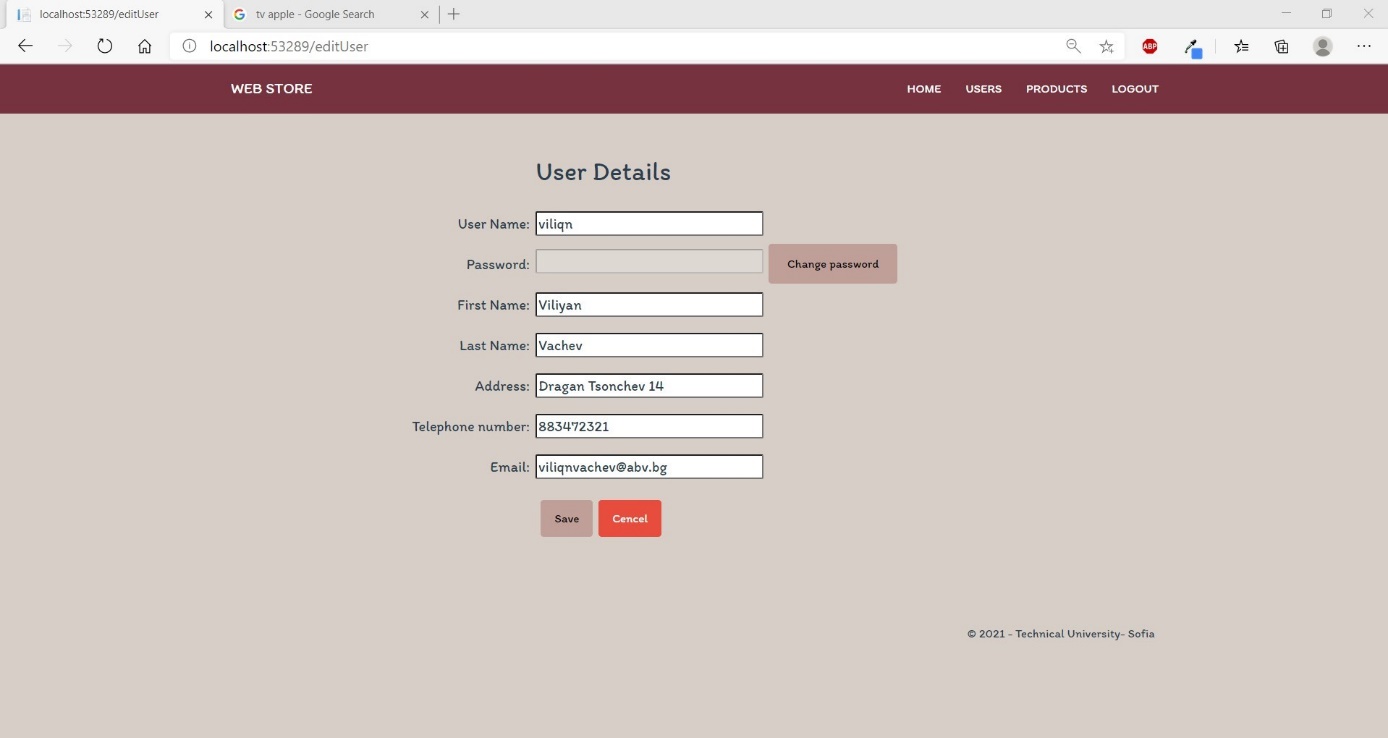
Фигура 20- Активна страница за детайли на потребител- администраторски панел

При натискане на бутона „Edit user“ се зарежда страницата editUser.aspx

***Страница за редакция на потребителя (editUser.aspx)***

Базиранa е на шаблона- „AdminMain.Master“ и използва всички негови функицоналности. Тя служи за редактиране на всички налични данни за конкретен потребител в системата. Допълнителни елементи на страницата са:

* Седем текстови полета:
* User Name- име на потребителя
* Password- парола
* First Name- собствено име
* Last Name- фамилия
* Address- адрес
* Telephone number- телефонен номер
* Email- имейл адрес
* Бутон за промяна на потребителя „Save“
* Бутон за отказ „Cancel“
* Бутон за отказ „Change password“



Фигура 21- Активна страница за редактиране на потребител- администраторски панел

От съображения за сигурност паролата на потребителите не се визуализира. При натискане на бутона „Change password“ се изпълнява следния код:

PasswordTextBox.Enabled = true;

Чрез този код полето за парола става активно, ако администратора реши да я смени.

След като промените по дадения потребител са попълнени, трябва да се натисне бутонът „Save“, който изпълнява следния код:

string selectedUserName = Session["SelectedUserName"].ToString();

//Get data from page

string userName = UserNameTextBox.Text;

string password = PasswordTextBox.Text;

string firstName = FirstNameTextBox.Text;

string lastName = LastNameTextBox.Text;

string address = AddressTextBox.Text;

string telephone = TelephoneTextBox.Text;

string email = EmailTextBox.Text;

string updateQuery;

//Ckeck password

if (password == null || password == "")

{

updateQuery = $"update Users set first\_name='{firstName}', lastst\_name='{lastName}', address='{address}', telephone='{telephone}', email='{email}', userName='{userName}' where userName='{selectedUserName}'";

}

else

{

updateQuery = $"update Users set first\_name='{firstName}', lastst\_name='{lastName}', address='{address}', telephone='{telephone}', email='{email}', userName='{userName}', password='{password}' where userName='{selectedUserName}'";

}

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["RegistrationStoreConnectionString"].ConnectionString);

conn.Open();

SqlCommand comm = new SqlCommand(updateQuery, conn);

comm.ExecuteScalar();

Session["SelectedUserName"] = null;

Response.Redirect("homeAdmin.aspx");

Чрез стрингови променливи се взимат данните от вече променените полета, след което се конструира заявка, която да промени данните в базата данни. Накрая сесията се занулява и се препраща към началната страница

***Страница продукти- администраторски панел (products.aspx)***

Базиранa е на шаблона- „AdminMain.Master“ и използва всички негови функицоналности. Тя служи за визуализиране на всички продукти в системата.

За да се извлекат всички необходими данни за визуализиране на продуктите, се използва контролера „asp:SqlDataSource“, който представлява SQL connection. След това данните се поставят в друг контролер „asp:GridView“, където се визуализират.

Аналогично на начлната страница, при натискане на бутона „Select“, се зареждат детайли за съответния продукт, след което администраторът има възможността да изключи обявата.

# Верификация и тестване

Тестването на софтуер е изследване, което се прави с цел да се даде информация на всички заинтересовани страни за качеството на продукта или услугата. То трябва да даде обективна и безпристрастна оценка на софтуера, която да позволи на бизнеса да прецени и разбере рисковете от употребата му. Техниките за тестване могат да включват изпълнение на програмата с цел откриването на грешки (бъгове). Тестването на софтуер може да се разглежда като процес на валидация и верификация, че дадена компютърна програма/приложение/продукт:

* Отговарят на изискванията, залегнали при неговия проект и разработка
* Работи според очакванията
* Може да се въведе в употреба при тези си параметри
* Отговаря на нуждите на всички заинтересувани страни

За да се гарантира, че системата работи според изискванията е необходимо тя да бъде тествана. От към функционална страна, бяха направени редица тестове свързани с регистрация на потребители, създаване на продукти, изтриване на продукти, купуване на продукти, финализиране на поръчки, въвеждане на невалидни данни и други. От към пърформанс страна, бяха направени редица стрес тестове свъразни с голямо натоварване на системата. След изпълнението на тестовете и анализиране на резултатите от тях, може да кажем, че системата се държи спрямо първоначалните очаквания и изисквания.

# Анализ на резултатите

Целите на дипломната работа са постигнати успешно. Разполагаме с напълно функционален и лесен за използване „онлайн магазин“, чрез който бъдещите потребители могат бързо да направят своя избор и да поръчат избраните от тях продукти или да направят продажба на такива.

В хода на изпълнение на дипломния проект, след направения анализ на предметната област, бизнес процесите и бизнес логиката, е изградена функционална база данни. На базата на тези анализи е създадена и уеб – платформата.

# Фигури

[Фигура 1- Видове бизнес модели 10](#_Toc64317888)

[Фигура 2- Архитектура на системата 21](#_Toc64317889)

[Фигура 3- Схема за проектиране на БД 22](#_Toc64317890)

[Фигура 4- Концептуален модел на БД 23](#_Toc64317891)

[Фигура 5- Логически модел на БД 24](#_Toc64317892)

[Фигура 6- Физически модел на БД 25](#_Toc64317893)

[Фигура 7- Проект на меню "Категории" 29](#_Toc64317894)

[Фигура 8- Активна страница за "Регистрация" 45](#_Toc64317895)

[Фигура 9- Активна страница "Начало" 49](#_Toc64317896)

[Фигура 10- Активна страница за детайли на продукт 51](#_Toc64317897)

[Фигура 11- Активна страница за поръчка на продукт 52](#_Toc64317898)

[Фигура 12- Активна страница "Количка" 54](#_Toc64317899)

[Фигура 13- Активна страница "Мойте поръчки" 56](#_Toc64317900)

[Фигура 14- Активна страница "Добавяне на продукт" 58](#_Toc64317901)

[Фигура 15- Активна страница "Мойте продукти" 59](#_Toc64317902)

[Фигура 16- Активна страница "Моите продукти- детайли" 60](#_Toc64317903)

[Фигура 17- Активно падащо меню за "Категории" 61](#_Toc64317904)

[Фигура 18- Активна страница "Начало админ панел" 62](#_Toc64317905)

[Фигура 19- Активна страница за детайли на продукт- администраторски панел 64](#_Toc64317906)

[Фигура 20- Активна страница за детайли на потребител- администраторски панел 65](#_Toc64317907)

[Фигура 21- Активна страница за редактиране на потребител- администраторски панел 66](#_Toc64317908)

# Източници:

<http://tuj.asenevtsi.com/APIS/APIS05.htm>

<http://www.intel-bg.com/elt/lekcia1.pdf>

<https://host.bg/bg/info/kakvo-e-e-biznes-kakvo-e-e-tyrgoviya/>

<http://www-it.fmi.uni-sofia.bg/courses/BonI/chapter3.html#3.3.1>

<https://bg.wikipedia.org>