Дипломна работа

на тема

„Разработване на онлайн книжарница”



Студент: Калин Емилов Ценков

Факултет: ФКСТ

Специалност: ИТИ

Факултетен номер: 501218032

Дипломен ръководител:

Група: 42

Доц. д-р инж. Иван Станков

**Съдържание**

[1. Анализ на проблемната област 3](#_Toc138143735)

[2. Цел и задачи на дипломната работа 3](#_Toc138143736)

[3. Въведение в информационните системи 4](#_Toc138143737)

[3.1 Какво представлява информационната система? 4](#_Toc138143738)

[3.2 Компоненти 5](#_Toc138143739)

[3.3 Видове информационни системи 10](#_Toc138143740)

[1. Какво е система за подкрепа на вземане на решения (DSS)? 10](#_Toc138143741)

[2. Какво представляват „социалните информационни системи“? 12](#_Toc138143742)

[3. Системи за управление на процеса (PCS) 12](#_Toc138143743)

[4. Информационна система за управление (MIS) 13](#_Toc138143744)

[5. Корпоративни системи 14](#_Toc138143745)

[6. Склад за данни 14](#_Toc138143746)

[7. Планирането на корпоративните ресурси (ERP) 15](#_Toc138143747)

[8. Експертна система 16](#_Toc138143748)

[9. Търсачка 16](#_Toc138143749)

[10. Географска информационна система (ГИС) 17](#_Toc138143750)

[11. Глобална информационна система 19](#_Toc138143751)

[12. Мултимедийна информационна система 20](#_Toc138143752)

[13. Офисната автоматизация 20](#_Toc138143753)

[3.4 Разработване на информационна система 21](#_Toc138143754)

[4. Проектиране на системата 23](#_Toc138143755)

[5. Програмна реализация 28](#_Toc138143761)

[6. Ръководство на потребителя 41](#_Toc138143769)

[7. Заключение 52](#_Toc138143770)

[8. Използвана литература: 53](#_Toc138143771)

# **Анализ на проблемната област**

В света на технологиите и интернета се наблюдава растеж на популярността на онлайн пазаруването, включително и за книги. Онлайн книжарницата предоставя възможност на хората да разглеждат и поръчват книги от удобството на своите домове или работни места, без да се налага да отделят време за посещение на физически магазин.

Моята идея е да създам уеб сайт за онлайн книжарница с максимално улеснен интерфейс, който да бъде достъпен и интуитивен дори за най-обикновения човек. Чрез този уеб сайт бих искал да предоставя на потребителите голямо разнообразие от книги, които могат да разглеждат и поръчват с лекота. Системата за търсене и филтриране ще позволява бързо и лесно намиране на желаната книга, а възможността за добавяне в кошницата и онлайн плащания ще гарантира удобство и безопасност при пазаруването.

Освен това, моята онлайн книжарница ще предоставя информация за най-новите издания, авторски събития, промоции и отзиви от читатели. Това ще позволи на потребителите да бъдат винаги информирани за актуалните литературни тенденции и да открият нови и вълнуващи заглавия.

Вярвам, че чрез създаването на уеб сайт за онлайн книжарница бих спомогнал за популяризирането на четенето и улеснил достъпа до литературата за хората. Онлайн книжарницата предлага удобство, бързина и разнообразие, които са от съществено значение за съвременния начин на живот.

# **Цел и задачи на дипломната работа**

Целта на дипломната работа "Разработване на онлайн книжарница" е да проучи и анализира проблематичните аспекти, свързани със създаването и управлението на успешна онлайн книжарница, и да предложи решения и препоръки за тяхното преодоляване. Главната цел е да се представи обща представа за основните предизвикателства и важни аспекти, които трябва да се разгледат и се прилагат при разработването на онлайн книжарница.

Задачи на дипломната работа:

1. Проучване на литературата и анализ на съществуващи изследвания в областта на онлайн книжарниците.
2. Анализ на проблематичните аспекти на онлайн книжарниците, включително управление на инвентара, сигурност на данните, потребителски интерфейс и изживяване, както и конкурентно предимство.
3. Идентифициране на основни предизвикателства и трудности, които могат да се срещнат при разработването на онлайн книжарница.
4. Предоставяне на препоръки и решения за преодоляване на проблематичните аспекти и предизвикателствата, свързани с разработването на онлайн книжарница.
5. Проектиране и разработване на модел или прототип на онлайн книжарница, който да демонстрира прилагането на предложените решения и препоръки.
6. Оценка и анализ на ефективността на предложените решения и препоръки за разработване на успешна онлайн книжарница.
7. Създаване на заключение, което обобщава основните резултати и изводи от дипломната работа, както и предлага възможности за бъдещо развитие и подобрения в областта на онлайн книжарниците.

Задачите на дипломната работа са насочени към изследване, анализ, проектиране и предложение на практически решения за успешното разработване и управление на онлайн книжарница, с фокус върху проблематичните аспекти, които възникват в процеса.

# **Въведение в информационните системи**

## **Какво представлява информационната система?**

**Информационна система** е комбинацията от информационни технологии и действията на хората, които ги прилагат за управлението на процеси, вземане на решения и други с помощта на компютърни системи. Системата е предназначена за използване от организация или физическо лице и дава възможност за съхранение на бази данни, управление и обработка на цялата информация или на част от нея. Съществуват различни информационни системи: финансови, промишлени, географски, и други подобни като за тях ще задълбаем по-нататък. Като цяло информационните системи могат да бъдат разделени на такива, предназначени да извършват определена операция (обработка на транзакции, Transaction Processing) и на такива, които са предназначени за събиране на данни, необходими при вземането на решения (Decision Support). Според това на какво ниво в организацията се ползват, информационните системи могат да бъдат разделени на четири нива, които от долу нагоре са:

- Системи за обработка на транзакциите – ползвани основно от оперативните работници. Това са всички системи, които се използват в ежедневната работа на дадена компания, било то географска информационна система, системи за CAD, графично оформление и онлайн публикуване, подпомагане на телефонни услуги или други.

- Управленски (или мениджърски) информационни системи – ползвани от оперативния мениджмънт. Примери са системите за управление на взаимоотношенията с клиентите (CRM), за планиране на ресурсите на предприятието (ERP) системите и за управление на съдържанието.

- Системи за подпомагане на решения – системи за подпомагане на

колективното управление и взимане на решения. Подобни приложения са групуер системите.

- Директорски информационни системи – предоставят редовни отчети и информация за прогреса по работата в организацията.

## **Компоненти**

Една информационна система съдържа следните основни компоненти:

**• Хардуер:**

Терминът хардуер ни насочва към машинната част на информационната система. Хардуерът е вид техника. Изграден е от електрически вериги. Думата е чуждица в българския език. Произлиза от английската дума hardware (означава железария; техническа апаратура; изчислителна техника; устройства, изграждащи по-голям апарат или машина). Преди появата на тази дума в българския език се е използвал терминът апаратно осигуряване. Хардуерът, изграждащ компютъра, се състои от процесор, дънна платка, видео карта, RAM памет, захранващ блок, твърд диск/ССД, компютърна кутия и входно-изходни устройства.

**• Софтуер:**

Софтуерът (програмно осигуряване, програмно обезпечение, на английски: software, буквално „меко изделие“) е съвкупността от цялата информация от инструкции и данни, необходими за работата на всяка електронноизчислителна машина.

Обикновено инструкциите се задават като съвкупност от алгоритми, групирани като програми с различно предназначение. Освен самите алгоритми, за изпълнението на програмите са необходими и начални данни. Резултатът от действието на даден алгоритъм може да служи като начални данни за стартирането на друг и т.н., обединявайки програмите в едно. В този смисъл все повече се налага и терминът софтуер, който исторически се е наложил като антоним на хардуера. Все пак границата между софтуер и хардуер се размива, когато се отчете, че програмното осигуряване има все пак някакви физически носители, от които изчислителната машина чете програмите. Тези носители, както и инструкциите за работа с програмите и тяхната поддръжка (документация), са спомагателни елементи от софтуера.

В масовите електронноизчислителни машини (компютри) голяма част от софтуера е разположена на външна памет и може лесно да бъде променяна от потребителите според нуждите им. Инструкциите в централните процесори и специализираните интегрални схеми обаче се задават при програмирането им, и не могат да бъдат променяни директно от потребителя.

По принцип хардуерът е неизползваем без програмно осигуряване или софтуер. Най-ясно разликата между софтуер и хардуер се обяснява така: Хардуерът е всяко едно видимо и осезаемо устройство, както самият компютър, така и всяко едно периферно устройство. Софтуерът е само видим, но не и осезаем, когато работи – това са именно програмите.

Със създаването на алгоритми и писането на програми се занимават програмистите. Според нивото на действие на написаните от тях инструкции има език за програмиране на ниско ниво, на средно ниво и на високо ниво. Езиците на високо ниво позволяват лесно редактиране от програмист, и това се нарича изходен код (програмен код). За да се превърне в изпълними инструкции от машината, той се компилира (превежда) към обектен код или машинен език.

Софтуерът бива два вида - системен и приложен.

***Системен***

Към системния софтуер се отнасят програми, изпълняващи системни функции и грижещи се за компютърния хардуер и връзката между него и потребителя. Предназначението на системния софтуер е да освободи авторите на приложни програми от пряко взаимодействие с компютърния хардуер, който е много разнороден. Той дава достъп хардуера, който е унифициран и независим от начина на работа на конкретните устройства.

***Приложен***

Потребителски приложения и програми, работещи като краен продукт и тясно свързващи потребителя със софтуер от по-ниско ниво. Това са най-известните сред потребителите програми, с който те имат пряк контакт и използват най-често. Пример за такъв софтуер са компютърните игри, текстовите редактори, уеб браузърите и т.н.

**• Данни:**

Данни (на английски - data) са неструктурирани факти за обект, които се съхраняват без да се използват. В случай че се появи необходимост, тези данни се използват (обработват) с някаква цел (за намаляване неопределеността на нещото). Преобразуваните данни се превръщат в информация. Данните често са възприемани като най-ниското ниво на абстракция, от което информацията и знанието произхождат.

Съществуват няколко форми за представяне на данни.

***Символна форма***

*Символ* – служи за условно обозначаване на някакво понятие, явление или процес. Символите могат да бъдат алфавит, знаци, управляващи, графични, специални и т.н.

Символите се кодират с кодови таблици, като за всеки символ се определя точно определена двоична комбинация, в зависимост от избрания метод за кодиране:

ASCII – американски стандартен код за информационен обмен;

EBCDIC – разширен двоично-десетичен код;

Unicode – универсален код.

***Текст***

Форма за представяне на данните във вид на последователност от символи, на избран език, съдържателно разглеждана като едно цяло.

В текста основно се използва естествен език и неговата азбука.

При писане на програми се използват езици за програмиране.

**Видове текстово представяне:**

Текст – обикновен текст, чрез който се съставят документи.

Телетекст – текст, предаван по телевизионната мрежа.

Хипертекст – свързан чрез ключови думи (хипервръзки) текст, чрез който се създават верижно свързани документи. На тази база се изгражда Web-пространството в Интернет.

Метатекст – текст, който в себе си съдържа данни за собствената структура и съдържание. Той е основа за бъдещето развитие на Интернет като семантична мрежа.

***Звукова форма***

Звук – образуван е от вибрации на въздуха, които съответстват на периодични сигнали със сложна форма. Звуците в природата не са единична синусоидална честота, а набор от много – сходни и несходни сигнали с различна форма. Всеки звук може да бъде съставен, като се съберат нужните честоти във вярна комбинация.

Звукова форма – форма за представяне на данните във вид на последователност от битове, получени вследствие на аналого-цифрово преобразуване на реален звук.

В зависимост от броя битове, които се заделят за измерване на един квант от звука са възможни 4, 8 и 16 битова форма на представяне на звука.

***Графична форма***

Изображение – форма на представяне на информацията, предназначена за зрително възприемане. Изображението е много нагледна и обемна форма за представяне на информацията. Възприемането му зависи от мозъка на човека и от възможностите на неговите органи на зрение. Човек леко оперира със зрителни образи.

Има две форми за представяне на изображение:

Растерно изображение – формира се от отделни точки, характеризиращи се с цвят и яркост.

Векторно изображение – характеризира се с голямо число вектори (къси прави), които имат определено направление, цвят и координати на точки, всяка от които има определено направление, цвят и координати на началната и крайна точки.

***Комбинирана форма***

С развитие на мултимедийните технологии се създава възможност за изграждане на мултисреда включваща всички форми за представяне на данни. Интеграцията им е осъществена чрез единна технология за обработка, съхраняване и предаване на данните.

* **Процедури:**

Процедурата е последователност от действия или операции, които трябва да бъдат извършени по един същ начин, с цел получаването на един и същ резултат, при еднакви обстоятелства. Те са политиката, която следва компютърната система. Една аналогия, по която често се използва, за да покаже ролята на процедурите в системата, е „Процедурите за хората са като програмите за машината“.

* **Хора:**

Всяка система се нуждае от хора, ако иска да е полезна. Много често най-пренебрегнатата част от системата са хората, въпреки че те са тези, които играят най-важна роля в успеха или провала на информационната система. Това включва не само потребителите, но и тези, които поддържат и работят с компютрите.

* **Обратна връзка:**

Обратната връзка(на английски - feedback) представлява най-общо казано събраната или получена информация от въздействието на определен обект, програма или процес. Терминът се използва в различни сфери на човешкото познание. Това е друг компонент от операционната система, който определя дали системата ще поддържа обратна връзка с потребителя. Тази част не е задължителна.

## **Видове информационни системи**

### **Какво е система за подкрепа на вземане на решения (DSS)?**

Системата за подкрепа на вземане на решения (DSS) е компютъризирана програма, използвана за подпомагане на решения, преценки и курсове на действие в организация или бизнес. DSS пресява и анализира огромни количества данни, като съставя изчерпателна информация, която може да се използва за решаване на проблеми и при вземане на решения. Типичната информация, използвана от DSS, включва целеви или прогнозирани приходи, данни за продажби или минали такива от различни времеви периоди и други данни, свързани с инвентара или операциите.

Системата за подпомагане на вземането на решения (DSS) е компютъризирана система, която събира и анализира данни, синтезирайки ги за създаване на изчерпателни информационни отчети. Тя се различава от обикновеното операционно приложение, чиято функция е само да събира данни.

Системите за подкрепа на решения позволяват по-информирано вземане на решения, навременно решаване на проблеми и подобрена ефективност при справяне с проблеми или операции, планиране и дори управление.

DSS може да бъде напълно компютъризиран или захранван от хора. В някои случаи може да комбинира и двете. Идеалните системи анализират информацията и действително вземат решения за потребителя. Най-малкото, те позволяват на потребителите да вземат по-информирани решения с по-бързи темпове.

DSS може да се използва от управлението на операциите и други отдели за планиране в организацията, за да събира информация и данни и да ги синтезира в действащо разузнаване. Всъщност тези системи се използват предимно от управление на средно и по-високо ниво. Например, DSS може да се използва за прогнозиране на приходите на компанията през следващите шест месеца въз основа на нови предположения за продажбите на продукти. Поради голям брой фактори, които заобикалят прогнозните приходи, това не е просто изчисление, което може да се направи ръчно. Въпреки това, DSS може да интегрира всички множество променливи и да генерира резултат и алтернативни резултати, всички базирани на данни за минали продажби на продукти на компанията и текущи променливи. DSS може да бъде пригодена за всяка индустрия, професия или домейн, включително медицинската област, правителствени агенции, селскостопански концерни и корпоративни операции.

Основната цел на използването на DSS е да представи информация на клиента по лесен за разбиране начин. DSS системата е полезна, защото може да бъде програмирана да генерира много видове отчети, всички базирани на потребителски спецификации. Например, DSS може да генерира информация и да извежда информацията си графично, като в диаграма, която представлява прогнозирани приходи или като писмен отчет.

Тъй като технологията продължава да напредва, анализът на данни вече не се ограничава до големи, обемисти мейнфрейм компютри. Тъй като DSS по същество е приложение, той може да бъде зареден на повечето компютърни системи, независимо дали на настолни компютри или лаптопи. Някои DSS приложения са достъпни и чрез мобилни устройства.

Гъвкавостта на DSS е изключително полезна за потребителите, които пътуват често. Това им дава възможност да бъдат добре информирани по всяко време, като им предоставя възможността да вземат най-добрите решения за своята компания и клиенти в движение или дори на място.

В организациите системата за подпомагане на вземането на решения (DSS) анализира и синтезира огромни количества данни, за да подпомогне вземането на решения. С тази информация той изготвя отчети, които могат да прогнозират приходи, продажби или управляват инвентара. Чрез интегрирането на множество променливи, DSS може да произведе редица различни резултати въз основа на предишни данни на компанията и текущи входове.

Много различни индустрии, от медицината до селското стопанство, използват системи за подкрепа на вземане на решения. За да подпомогне диагностицирането на пациент, медицинският лекар може да използва компютъризирана система за подпомагане на вземането на решения за диагностика и предписване. Комбинирайки данните от клинициста и предишни електронни здравни досиета, системата за подкрепа на вземане на решения може да помогне на лекар при диагностицирането на пациент.

Най-общо казано, системите за подпомагане на вземането на решения помагат при вземането на по-информирани решения. Често използвани от ръководството на по-високо и средно ниво, системите за подкрепа на вземане на решения се използват за вземане на решения, които могат да бъдат предприети, или за създаване на множество възможни резултати въз основа на текущи и исторически данни на компанията. В същото време системите за подкрепа на вземане на решения могат да се използват за изготвяне на отчети за клиенти, които са лесно смилаеми и могат да бъдат коригирани въз основа на спецификациите на потребителя.

### **Какво представляват „социалните информационни системи“?**

По същество социалните информационни системи са информационни системи, базирани на социални технологии и открито сътрудничество. Социалните информационни системи са различни от традиционните бизнес информационни системи, т.е терминът "информационна система" често се използва имплицитно.

Социалните информационни системи се използват все по-често както по хедонистичен, така и по утилитарен начин - използването на социални медии се е превърнало в любимо занимание за свободно време за интернет потребителите. За разлика от другите форми на медиите, потребителите могат активно да допринасят и съвместно създават в социалните медии. Следователно виждаме различни форми на автономно съвместно създаване на потребители, възникващи в Интернет, което ефективно представлява нов и независим от фирмата метод на производство.

### **Системи за управление на процеса (PCS)**

Системите за управление на процеса (PCS), понякога наричани системи за индустриален контрол (ICS), функционират като части от оборудване по производствената линия по време на производството, които тестват процеса по различни начини и връщат данни за наблюдение и отстраняване на неизправности. Съществуват много видове системи за управление на процесите, включително надзорен контрол и събиране на данни (SCADA), програмируеми логически контролери (PLC) или разпределени системи за управление (DCS) и работят за събиране и предаване на данни, получени по време на производствения процес.

PCS може да бъде сравнително прост елемент със сензор, често наричан първичен преобразувател, който получава вход, заедно с контролер, който обработва входа, и приемник, който обработва изход.

По-сложните PCS устройства са роботизирани и изпълняват много задачи. PCS устройствата могат да предават своите данни към компютърното приложение за планиране на корпоративните ресурси (ERP) на компанията чрез софтуер за междинен софтуер, наречен производствена система за изпълнение (MES).

### **Информационна система за управление (MIS)**

Информационната система за управление (MIS) е информационна система, използвана за вземане на решения и за координация, контрол, анализ и визуализация на информация в организация. Изучаването на информационните системи за управление включва хора, процеси и технологии в организационен контекст.

В корпоративна среда крайната цел на използването на управленска информационна система е повишаване на стойността и печалбите на бизнеса. Това се постига чрез предоставяне на мениджърите на навременна и подходяща информация, която им позволява да вземат ефективни решения в по-кратък период от време.

Следват някои от предимствата, които могат да бъдат постигнати с помощта на MIS:

* Подобряване на оперативната ефективност на организацията, добавяне на стойност към съществуващите продукти, пораждане на иновации и разработване на нови продукти и помагане на мениджърите да вземат по-добри решения.
* Компаниите са в състояние да идентифицират своите силни и слаби страни поради наличието на отчети за приходите, записи за представяне на служителите и т.н. Идентифицирането на тези аспекти може да помогне на компанията да подобри своите бизнес процеси и операции.
* Даване на цялостна картина на компанията.
* Действа като средство за комуникация и планиране.
* Наличието на клиентски данни и обратна връзка може да помогне на компанията да приведе своите бизнес процеси в съответствие с нуждите на своите клиенти. Ефективното управление на данните за клиентите може да помогне на компанията да извършва дейности по директен маркетинг и промоция.
* MIS може да помогне на компанията да спечели конкурентно предимство.
* Отчетите за MIS могат да помогнат при вземането на решения, както и да намалят времето за престой за елементи, които могат да бъдат предприети.

Някои от недостатъците на MIS системите:

* Извличането и разпространението зависят от технологичния хардуер и софтуер.
* Възможност за неточна информация.

### **Корпоративни системи**

Корпоративните системи са мащабни софтуерни пакети, които могат да проследяват и контролират всички сложни операции на бизнеса. Тези системи се използват като централен команден център за автоматизиране на бизнеса и улесняване на отчитането и вземането на решения.

### **Склад за данни**

Складът за данни е вид система за управление на данни, която е проектирана да позволява и поддържа дейности по бизнес разузнаване (BI), особено анализи. Складовете за данни са предназначени единствено за извършване на заявки и анализи и често съдържат големи количества исторически данни. Данните в склад за данни обикновено се извличат от широк спектър от източници, като регистрационни файлове на приложения и приложения за транзакции.

Склад за данни централизира и консолидира големи количества данни от множество източници. Неговите аналитични възможности позволяват на организациите да извличат ценни бизнес прозрения от своите данни, за да подобрят вземането на решения. С течение на времето той изгражда исторически записи, които могат да бъдат безценни за учените по данни и бизнес анализаторите. Поради тези възможности, складът за данни може да се счита за „единствен източник на истина“ на организацията.

### **Планирането на корпоративните ресурси (ERP)**

Планирането на корпоративните ресурси (ERP) се отнася до вид софтуер, който организациите използват за управление на ежедневните бизнес дейности като счетоводство, поръчки, управление на проекти, управление на риска и съответствие, както и операции по веригата на доставки. Пълният ERP пакет включва също управление на ефективността на предприятието, софтуер, който помага за планиране, бюджет, прогнозиране и отчитане на финансовите резултати на организацията.

ERP системите свързват множество бизнес процеси и позволяват потока от данни между тях. Чрез събирането на споделени транзакционни данни на организацията от множество източници, ERP системите елиминират дублирането на данни и осигуряват целостта на данните с един източник на истина.

Днес ERP системите са от решаващо значение за управлението на хиляди предприятия от всякакъв размер и във всички индустрии. За тези компании ERP е също толкова незаменим, колкото и електричеството, което поддържа осветлението.

Как тези решения могат да управляват ежедневните бизнес дейности на организациите, като счетоводство, финанси, обществени поръчки, управление на проекти, верига за доставки и производство? Системите за планиране на корпоративните ресурси са завършени интегрирани платформи, локални или в облака, управляващи всички аспекти на базиран на производство или дистрибуторски бизнес. Освен това, ERP системите поддържат всички аспекти на финансовото управление, човешките ресурси, управлението на веригата за доставки и производството с вашата основна счетоводна функция.

ERP системите също така ще осигурят прозрачност на пълния ви бизнес процес, като проследяват всички аспекти на производството, логистиката и финансите. Тези интегрирани системи действат като централен център на бизнеса за работен поток и данни от край до край, позволявайки достъп на различни отдели. ERP системите и софтуерът поддържат множество функции в предприятието, средния или малкия бизнес, включително персонализации за вашата индустрия.

### **Експертна система**

Експертна система е компютърна програма, която използва технологии с изкуствен интелект (AI), за да симулира преценката и поведението на човек или организация, която има експертни познания и опит в определена област.

Обикновено експертната система включва база от знания, съдържаща натрупан опит и машина за изводи или правила - набор от правила за прилагане на базата от знания към всяка конкретна ситуация, която е описана в програмата. Възможностите на системата могат да бъдат подобрени с допълнения към базата от знания или към набора от правила. Настоящите системи могат да включват възможности за машинно обучение, които им позволяват да подобрят производителността си въз основа на опит, точно както правят хората.

Концепцията за експертни системи е разработена за първи път през 70-те години на миналия век от Едуард Файгенбаум, професор и основател на лабораторията за системи на знания в Станфордския университет. Файгенбаум обяснява, че светът се движи от обработка на данни към „обработка на знания“, преход, който се осъществява от новите процесорни технологии и компютърни архитектури.

Експертните системи изиграха голяма роля в много индустрии, включително във финансовите услуги, телекомуникациите, здравеопазването, обслужването на клиенти, транспорта, видеоигрите, производството, авиацията и писмената комуникация. Две ранни експертни системи пробиха почвата в областта на здравеопазването за медицински диагнози: Dendral, която помогна на химиците да идентифицират органични молекули, и MYCIN, която помогна да се идентифицират бактерии като бактериемия и менингит и да се препоръчат антибиотици и дози.

### **Търсачка**

Търсачката е софтуерна система, която е предназначена да извършва търсене в мрежата. Те търсят в World Wide Web по систематичен начин конкретна информация, посочена в текстова заявка за търсене в мрежата. Резултатите от търсенето обикновено се представят в ред с резултати, често наричани страници с резултати от търсачките (SERP). Информацията може да бъде комбинация от връзки към уеб страници, изображения, видеоклипове, инфографики, статии, научни статии и други видове файлове. Някои търсачки също копаят данни, налични в бази данни или отворени директории. За разлика от уеб директориите, които се поддържат само от човешки редактори, търсачките също поддържат информация в реално време, като изпълняват алгоритъм на уеб робот. Интернет съдържание, което не може да бъде търсено от уеб търсачка, обикновено се описва като дълбока мрежа.

### **Географска информационна система (ГИС)**

Географската информационна система (ГИС) е компютърна система за улавяне, съхранение, проверка и показване на данни, свързани с позиции на земната повърхност. Чрез свързване на привидно несвързани данни, ГИС може да помогне на хората и организациите да разберат по-добре пространствените модели и взаимоотношения.

ГИС технологията е важна част от инфраструктурата за пространствени данни, която Белият дом определя като „технологията, политиките, стандартите, човешките ресурси и свързаните с тях дейности, необходими за придобиване, обработка, разпространение, използване, поддържане и запазване на пространствени данни“.

ГИС може да използва всякаква информация, която включва местоположение. Местоположението може да бъде изразено по много различни начини, като географска ширина и дължина, адрес или пощенски код.

Много различни видове информация могат да бъдат сравнени и противопоставени с помощта на ГИС. Системата може да включва данни за хора, като население, доходи или ниво на образование. Тя може да включва информация за ландшафта, като местоположението на потоци, различни видове растителност и различни видове почва. Тя може да включва информация за обектите на фабрики, ферми и училища или дъждовни канали, пътища и електропроводи.

С ГИС технологията хората могат да сравняват местоположенията на различни неща, за да открият как те се отнасят помежду си. Например, използвайки ГИС, една карта може да включва обекти, които произвеждат замърсяване, като фабрики, и обекти, които са чувствителни към замърсяване, като влажни зони и реки. Такава карта би помогнала на хората да определят къде водоснабдяването е най-застрашено.

ГИС приложенията включват както хардуерни, така и софтуерни системи. Тези приложения могат да включват картографски данни, фотографски данни, цифрови данни или данни в електронни таблици.

Картографските данни вече са под формата на карта и могат да включват такава информация като местоположението на реки, пътища, хълмове и долини. Картографските данни могат също да включват данни от проучвания и картографска информация, които могат да бъдат директно въведени в ГИС.

Фотографската интерпретация е основна част от ГИС. Интерпретацията на снимки включва анализиране на въздушни снимки и оценка на елементите, които се появяват.

В ГИС могат да се въвеждат и цифрови данни. Пример за този вид информация са компютърните данни, събрани от спътници, които показват използването на земята – местоположението на фермите, градовете и горите.

Дистанционното наблюдение предоставя друг инструмент, който може да бъде интегриран в ГИС. Дистанционното наблюдение включва изображения и други данни, събрани от спътници, балони и дронове.

И накрая, ГИС може също да включва данни под формата на таблица или електронна таблица, като например демографски данни на населението. Демографските данни могат да варират от възраст, доходи и етническа принадлежност до скорошни покупки и предпочитания за сърфиране в интернет.

ГИС технологията позволява всички тези различни типове информация, независимо от техния източник или оригинален формат, да бъдат насложени един върху друг върху една карта. ГИС използва местоположението като ключова променлива на индекса, за да свърже тези на пръв поглед несвързани данни.

Поставянето на информация в ГИС се нарича улавяне на данни. Данните, които вече са в цифрова форма, като повечето таблици и изображения, направени от спътници, могат просто да бъдат качени в ГИС. Картите обаче първо трябва да бъдат сканирани или преобразувани в цифров формат.

Двата основни типа GIS файлови формати са растерни и векторни. Растерните формати са решетки от клетки или пиксели. Растерните формати са полезни за съхраняване на ГИС данни, които се различават, като например надморска височина или сателитни изображения. Векторните формати са многоъгълници, които използват точки (наречени възли) и линии. Векторните формати са полезни за съхранение на ГИС данни с твърди граници, като училищни квартали или улици.

### **Глобална информационна система**

Глобална информационна система (ГИС) е всяка информационна система, която се опитва да достави съвкупността от измерими данни по целия свят в рамките на определен контекст.

Общото за този клас информационни системи е, че контекстът е глобална настройка, или за неговото използване, или за процес на разработка. Това означава, че тя е силно свързана с разпределени системи / разпределени изчисления, където разпространението е глобално. Терминът също така включва аспекти на глобалната разработка на софтуер и аутсорсинг (когато местата за аутсорсинг са глобално разпространени) и аспекти на офшоринг. Специфичен аспект на глобалните информационни системи е случаят (домейн) на глобалната разработка на софтуер. Основен изследователски аспект в тази област се отнася до координацията и сътрудничеството между виртуални екипи. Други важни аспекти са интернационализацията и езиковата локализация на системните компоненти.

Критични задачи при проектирането на глобални информационни системи са:

* Проектиране на процеси и системи: Как са организирани процесите между разпределените участници, как са разпределени/интегрирани системите.
* Техническа архитектура: Каква е техническата инфраструктура, която позволява на участниците да си сътрудничат?
* Механизми за подкрепа: Как се подпомагат участниците в процеса на комуникация, сътрудничество и сътрудничество?

Могат да се дадат различни примери. По принцип всеки многоезичен уебсайт може да се разглежда като глобална информационна система. Въпреки това, най-вече терминът GLIS се използва за обозначаване на конкретна система, разработена или използвана в глобален контекст.

### **Мултимедийна информационна система**

Мултимедийната информационна система е хранилище за всички видове информационни обекти. В компютърния контекст това означава всички видове цифрово представени данни като текст, изображения и глас. Такива системи са сложни и мултидисциплинарни по природа.

Мултимедийните информационни системи (MIS) съчетават възможности за управление на данни (ефективни и ефикасни методи за съхранение, индексиране, запитване и извличане) с управление на медиите (представяне на медии, компресиране на данни, стандартизация и предаване).

### **Офисната автоматизация**

Офисната автоматизация е общ термин, който описва различните видове компютърни системи и софтуер, които се използват за събиране на цифрови средства, съхраняване, прехвърляне, промяна и използване на офис информация за изпълнение на задачи. По същество автоматизацията на офиса помага да се управляват данни.

Офис автоматизацията позволява на данните да се движат без човешка намеса. Тъй като хората са изключени от уравнението, няма риск от ръчна грешка. Това, което някога е започнало с пишеща машина, се е превърнало в безброй автоматизация и електронни инструменти, които са променили начина на работа на хората.

Има много различни аспекти на автоматизацията на офиса, но те могат лесно да бъдат разделени на следните категории:

* Съхранение на информация: Това включва запис на информация, като формуляри, документи, файлове, изображения и електронни таблици. Съхранението на информация обикновено съществува във формати на текстови процесори или електронни таблици, но може да бъде и по-сложно като записи в CMS или софтуерен инструмент за автоматизация като SolveXia.
* Обмен на данни: Системите позволяват обмен на информация в реално време, като факс машини или имейли. Софтуерните инструменти за автоматизация също се вписват в тази категория, тъй като можете лесно да споделяте информация и да изпращате отчети между хората.
* Управление на данни: Автоматизацията на офиса също трябва да бъде лесно управляема между различни страни и съответната информация. Като такива, системите за автоматизация на офиса могат да обработват краткосрочни и дългосрочни данни, включително финансови планове, маркетингови разходи, управление на инвентара и т.н.

Бизнесът разчита на малко или много инструменти за автоматизация на офиса.

Независимо от степента, ползите са обширни, а именно:

* Намаляване на ръчните усилия за постигане на задачи
* Минимизиране на човешките и ръчните грешки
* Намалено време за обработка за изпълнение на задачата
* Повишена прозрачност и възможности за подобряване на процесите
* По-добро вземане на решения въз основа на данни и прогнози
* Подобрени показатели и мониторинг на KPI
* Повишена удовлетвореност и комуникация на служителите
* Резултатът е по-добро обслужване на клиентите

## **Разработване на информационна система**

Както подсказва името, разработването на информационна система или общоизвестно като SLC (жизнен цикъл на системите) или SLDC (жизнен цикъл на разработка на софтуер) е процес на създаване и промяна на системата и използвания модел и методология. С други думи, SDLC е подготовката на нова система, която да замени старата система, както изцяло, така и само частично.

Що се отнася до разработването на информационна система, свързаният екип се състои от няколко персонала, а именно координатор на проекта, системен анализатор и дизайн, мрежов дизайнер, програмист, техник (хардуер), администратор, софтуерен тестер, графичен дизайнер и документален.

Развитието на информационната система се състои от шест важни етапа, това е системно проучване, анализ на нуждите, проектиране, внедряване, тестване, промяна и поддръжка.

1. **Системно проучване**

Фазата SLDC също се състои от три основни точки: идентификация на системата, избор и планиране на системата.

* Идентификация на системата - Този процес е да идентифицира проблемите, пред които е изправена компанията и системата, която има. Екипът ще търси всякакви възможности, които могат да бъдат направени, за да преодолее това.
* Избор - Фазата на подбор ще приложи точки за оценка към проекта за разработка, за да се гарантира, че решенията са създадени в съответствие с очакваните цели на компанията.
* Системно планиране - Тази стъпка е стъпката за разработване на официален план за започване на работа и прилагане на концепцията за развитие на информационната система, която е избрана.

1. **Анализ на нуждите**

Анализът на системните изисквания е техника за решаване на проблеми чрез разлагане на компонентите на системата. Целта не е нищо друго освен да разберете повече за това как работи всеки компонент и взаимодействието между един компонент с други компоненти.

Някои аспекти, които трябва да бъдат насочени при анализа на нуждите при разработването на информационни системи, включват бизнес потребители, анализ на работни места, бизнес процеси, договорени правила, проблеми и решения, бизнес инструменти и бизнес планове.

1. **Дизайн**

Дизайнът или дизайнът на разработката на системата има за цел да предостави пълен план като насока за ИТ екипа (особено програмистите) при създаването на приложения. Така ИТ екипът вече не взема решения и не работи спорадично.

1. **Изпълнение**

Етапът на разработване на тази информационна система е да се работи по предварително проектирана разработка.

1. **Тестване**

Системата трябва да бъде тествана, за да се гарантира, че извършената разработка е подходяща или не с очакваните резултати. Тестовете, които се прилагат, са различни, като производителност, ефективност на входа, синтаксис (програмна логика), изход и т.н.

Този етап на развитие на информационната система изисква подготовка на различни поддържащи аспекти. В допълнение към приложенията, готовността на хардуера и няколко други свързани съоръжения също трябва да бъдат подготвени. Що се отнася до изпълнението, няколко извършени дейности включват миграция на данни (преобразуване), обучение за потребители и изпитания.

1. **Промяна и поддръжка**

Тази стъпка обхваща целия процес, за да се осигури непрекъснатост, гладкост и подобряване на системата. В допълнение към наблюдението на системата в определен момент, поддръжката включва и дейности за предвиждане на малки грешки (бъгове), подобрения на системата и предвиждане на някои рискове от фактори извън системата.

# **Проектиране на системата**

Приложението **Book Store** е изградено по максимално улеснен и разбираем начин. Има няколко функционалности, които са напълно достатъчни, за да покрият всичко необходимо за осъществяване на идеята на приложението, а именно приложението да предостави възможност на потребителите да откриват и закупуват книги онлайн.

## **Архитектура на проекта**

Архитектурата на back-end, която използвам за приложението е **Domain-driven design с Clean Architecture:**

* **Domain-driven design (DDD)** – е основен подход за проектиране на софтуер, фокусиран върху моделирането на софтуер, съответстващо на домейна според информацията от експертите в този домейн.

При домейн-ориентираното проектиране структурата и езикът на софтуерния код (имена на класове, методи на класове, променливи на класове) трябва да отговарят на бизнес домейна. Например, ако софтуерът обработва заявки за кредит, може да има класове като "заявка за кредит", "клиенти" и методи като "приемане на оферта" и "оттегляне".

Домейн-ориентираното проектиране се основава на следните цели:

* + да се постави основен фокус върху основния домейн и домейн логиката на проекта;
  + да се базират сложните дизайни на модел на домейна;
  + да се започне творческо сътрудничество между техническите и домейн експерти, за да се постепенно усъвършенства концептуалният модел, който решава конкретни проблеми от домейна.

Критиците на домейн-ориентираното проектиране твърдят, че разработчиците обикновено трябва да извършват значително изолиране и енкапсулация, за да запазят модела като чиста и полезна конструкция. Въпреки че домейн-ориентираното проектиране предлага предимства като поддържаемост, Microsoft го препоръчва само за сложни домейни, където моделът предоставя ясни ползи за формулиране на общо разбиране на домейна.

* **Clean Architecture** - е философия за проектиране на софтуер, която разделя елементите на дизайна на нива от концентрични обвивки. Важна цел на чистата архитектура е да предостави на разработчиците начин за организиране на кода по такъв начин, че да изолира бизнес логиката, като я отделя от механизма за доставка.

Картина, която съдържа текст, кръг, екранна снимка, диаграма

Описанието е генерирано автоматично

## **База данни**

Избрал съм релационната база данни MSSQL за приложението **Book Store**.

**Microsoft SQL Server (MSSQL)** е релационна база данни, разработена от Microsoft. Тя предоставя сигурно и надеждно съхранение и управление на структурирани данни. MSSQL се използва широко за разработка на бизнес приложения, уеб сайтове и системи за управление на данни.

Картина, която съдържа текст, диаграма, Паралелен, План

Описанието е генерирано автоматично

## **Клас диаграма**

Картина, която съдържа текст, екранна снимка, диаграма, Шрифт

Описанието е генерирано автоматично

## **Регистрация**

За да се осъществи това, първата стъпка, когато човек влезе в приложението, е да се регистрира. Това става с попълване на име, имейл адрес и парола.

## **Роли**

1. **Администратор**

В случай, че потребител е администратор, на екрана на потребителя ще се появи в горния десен ъгъл **Administration**. Натискайки там, се отваря администрацията на приложението. Там са предоставени всички функционалности нужни на администратора:

* преглед и редактиране на поръчки
* преглед, създаване, редактиране и изтриване на автори
* създаване, редактиране и изтриване на книги

1. **Клиент**

Достатъчно за администраторите. Нека разгледаме функционалностите, които се предлагат на клиентите. При регистрация те попълват име, имейл адрес и парола. След регистрация клиентите биват препращани към началната страница, която показва последните 10 добавени книги. Също така имат опция за разглеждане на всички книги, закупуване на книги и преглеждане на своя профил. Разглеждането на книгите може да стане чрез натискане на бутона **View all books**, който се намира на началната страница. След натискането му се отваря страница, на която клиентите могат да видят колекция от всички книги, както и да ги филтрират по заглавие, автор и жанр. След като клиентът отвори детайлите на някоя от книгите, той има опция за добавянето ѝ в кошницата си от бутона **Add to cart**.

От бутона **Cart**, който стои в горния десен ъгъл, клиентите могат да видят колекция от своите добавени книги и съответно могат да редактират количеството или да премахнат някоя от тях.

След натискане на бутона **Order**, който стои в долния десен ъгъл, клиентите могат да създадат поръчка.

# **Програмна реализация**

За програмната реализация на уеб приложението **Book Store**, което е предназначено за употреба при търсене на книги онлайн, е използвано Microsoft Visual Studio, ASP.NET Core и React.

Версията на .NET, с която е реализиран проектът, е .NET 6.0. Версията е сравнително нова – появява се през ноември 2021г. и се очаква да бъде заменена с .NET 7.0 през ноември 2022г.

Картина, която съдържа текст, екранна снимка, номер, Шрифт

Описанието е генерирано автоматично

ASP.NET Core е безплатна рамка за уеб приложения с отворен код, разработена от компанията Microsoft. Рамката ASP.NET Core поддържа успоредно управление на версиите, така че различните приложения, които се разработват на една машина, могат да се насочат към различни версии на ASP.NET Core. Това не е възможно с предишни версии на ASP.NET.

Ето и малко от нейните характеристики:

* Компилацията е непрекъсната, тоест разработчикът не трябва да извиква команда за компилиране
* Модулна рамка, разпространявана като NuGet пакети
* Оптимизирано за интернет време за изпълнение
* Може да работи в IIS(Internet Information Services) или самостоятелно
* Готова за облак конфигурационна система
* Лека и модулна поредица от HTTP заявки
* Създаване и стартиране на мултиплатформени приложения на Windows, Mac, Linux
* С отворен код и фокусирана върху общността
* Успоредно управление на версиите на приложението при насочване към .NET
* Вградена поддръжка за “dependency injection” – предоставяне на обектите, от които даден обект се нуждае, вместо да го кара да конструира самите обекти. Това е полезна техника, която прави тестването много по-лесно.

ASP.NET Core позволява използване на езика за програмиране С# в пълния му потенциал. В основата на ASP.NET Core стои концепцията на "модел-изглед-контролер" (MVC). MVC е архитектурен шаблон за уеб приложения, разработен отново от Microsoft, който позволява разделение на начините на представяне на информацията. В случая на Web API (Application Programming Interface), няма изрично използване на изгледи (views), тъй като API-то предоставя данни и резултати вместо представления на информация пред потребителя.

В ASP.NET Core Web API:

* Класовете модели представляват структури от данни, които се използват за представяне на информацията, която API-то обработва или връща. Моделите управляват данните, логиката и правилата на приложението.
* Контролерите са класове, които приемат HTTP заявките от клиента и ги обработват. Те съдържат методи, които се изпълняват при получаване на определени HTTP заявки и извършват нужната обработка върху моделите и другите компоненти на приложението.

ASP.NET Core Web API се фокусира върху предоставянето на данни и функционалност през уеб интерфейса, вместо на изгледи (views). Това означава, че API-то осигурява достъп до ресурсите и операциите, които могат да бъдат извършвани върху тях, като например четене, създаване, актуализиране и изтриване. Резултатите от изпълнението на заявките обикновено се връщат в желания формат, като JSON или XML, в зависимост от конфигурацията и предпочитанията на приложението.

Освен Web API, за разработката на уеб приложението **Book Store** съм използвал MSSQL (Microsoft SQL Server) за база данни, където да се съхраняват всички записи, и React за начина, по който ще се визуализират различните компоненти от приложението. За връзка с базата данни ни помага Connection String, а за работа с базата – Entity Framework Core. Използван е Code First подходът, което означава, че във Visual Studio дефинираме своите обекти и изисквания към тях и след това с помощта на миграции се създава базата данни. Възможен е и обратния подход – първо да се започне от изграждане на базата данни, но за целта на **Book Store** по-подходящо е да се използва споменатия вече Code First.

React е фронтенд рамка за уеб разработка, която се използва за създаване на интерактивни и ефективни потребителски интерфейси. Тя е разработена от Facebook и се характеризира със своя декларативен подход към създаването на компоненти.

Ето някои от основните характеристики на React:

* Компонентна базирана архитектура: React разделя потребителския интерфейс на независими компоненти, които могат да бъдат създавани, комбинирани и преизползвани по лесен начин. Всяка компонента има своя собствена логика и изглед, което улеснява поддръжката и разширяването на приложението.
* Виртуален DOM (Document Object Model): React използва виртуален DOM, който е лека копие на реалния DOM. Това позволява на React да оптимизира производителността, като минимизира пренарисуването на елементите на потребителския интерфейс и по-ефективно управлява обновленията.
* Декларативен синтаксис: React използва JSX (JavaScript XML), което е разширение на JavaScript, за създаване на компоненти и шаблони. Това позволява разработчиците да съчетават HTML-подобен синтаксис с JavaScript логика, което прави кода по-четим и лесен за разбиране.
* Еднопосочно свързване на данни: React препоръчва еднопосочното свързване на данни, където данните се предават от родителски компонент към децата му чрез пропс (props). Това улеснява следенето на потока на данни и прави приложението по-предсказуемо и лесно за отстраняване на грешки.
* Възможност за използване на други библиотеки и инструменти: React е гъвкав и позволява на разработчиците да използват други библиотеки и инструменти, които допълват функционалността му. Например, с React може да се използва Redux за управление на състоянието или React Router за навигация между различни страници.
* Общност и екосистема: React има голяма и активна общност от разработчици, което означава, че има множество ресурси, библиотеки и инструменти на разположение. Това улеснява обучението, разработката и решаването на проблеми по време на процеса на разработка.

React е широко използвана рамка в индустрията на уеб разработката, като позволява на разработчиците да създават мощни, мащабируеми и бързи потребителски интерфейси. Съчетаването на React с други инструменти и библиотеки допълнително разширява възможностите и гъвкавостта на разработката на модерни уеб приложения.

## **Регистриране и влизане в системата (Автентикация)**

ASP.NET Core има вградена такава, въз основа на която потребител може да се регистрира чрез своя имейл адрес и парола и след това да достъпва своя акаунт чрез „логване” с имейл адрес и парола. Веднага след регистрация данните се записват в релационна база данни, за която съм използвал MSSQL, както вече споменах. В момента, в който потребителят въведе своята парола, тя се хешира, а при влизане в системата се сравнява и съответно, ако се въведе грешна парола излиза съобщение на екрана.

Освен това, за удостоверяване на потребителите и осигуряване на безопасност на фронтенда използвам JWT (JSON Web Token). JWT е отворен стандарт за представяне на заявки между две страни чрез JSON обект в компактен и самоописателен формат. В React приложението на фронтенда, след успешно влизане в системата, се запазва JWT (в cookie, local storage или session storage на браузъра), който се включва във всички последващи заявки към API-то. Това позволява на API-то да идентифицира и удостоверява потребителя, базирайки се на валидността и съдържанието на JWT. Така се осигурява автентикация и защита на данните при комуникацията между клиента и сървъра.

Картина, която съдържа текст, екранна снимка, Шрифт, документ

Описанието е генерирано автоматично

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

## **Права и достъп**

В приложението **Book Store** има два типа потребители – администратори и клиенти. Съответно всеки от двата типа има достъп до различни неща. Това от страна на back-end частта е направено чрез създаване на собствени методи, които да проверяват в базата данни дали в таблицата, съдържаща данните за всички администратори или клиенти, има човек с такъв идентификационен номер. Методите, които съм направил, връщат резултат от тип boolean, тоест ако успеят да намерят такъв човек в базата, се връща резултат true, а в противен случай – false.

Картина, която съдържа текст, екранна снимка, Шрифт, линия

Описанието е генерирано автоматично

Във front-end частта също съм направил If-проверки, според които да се определя какво се визуализира пред един администратор и пред един клиент като бутони, таблици и т.н.

## **Начална страница**

Началната страница показва последните 10 добавени книги. Под тях има бутон **View all books**, който след натискането му отваря страница, на която потребителите могат да видят колекция от всички книги, както и да ги филтрират по заглавие, автор и жанр.

## **Промяна на въведените данни**

Направено е по много идентичен начин с първоначалното въвеждане на данни. Разликата е, че тук се използва методът FirstOrDefault, с който се намира идентификационния номер на потребителя, чиито данни ще бъдат променяни.

Картина, която съдържа текст, екранна снимка, Шрифт, линия

Описанието е генерирано автоматично

Както и след първоначалното въвеждане, така и тук се използва в края на метода друг метод, който ни е предоставен от C#, а именно SaveChanges(). Докато не се извика този метод, промените, които са направени, не се отразяват в базата данни.

В обобщение за книгите, авторите и поръчките използваме така наречените CRUD операции (C – Create, R- Read, U – Update, D – Delete).

## **Книги**

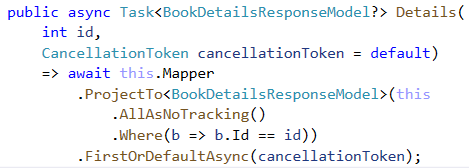
Тук е работено по същия начин като при предишната функционалност – основната част от логиката е изнесена в репозитори или услуга, която отговаря само за книгите. Всичките CRUD операции са използвани – имаме създаване на книга (create), преглеждане (read), промяна на данните (update) и изтриване (delete). В метода за извличане на списък с всички книги е добавено и кеширане на информацията, с което да се намали натоварването към сървъра. С изключение на преглеждането, всички други неща могат да се извършват единствено от потребители, които са администратори.

За пълноценната работа на CRUD операциите използваме и LINQ (Language-Integrated Query). LINQ представлява мощен набор от технологии, базирани на интегрирането на възможности за заявка директно в езика С#. Той е унифициран синтаксис на заявка за извличане на данни от различни източници и формати. Например за да извлечем детайли за някоя книга от базата, първо си създаваме предназначен модел (отделен клас), в който добавяме всички пропъртита на книгите, които искаме да покажем (заглавие, автор, жанр и т.н). След това използваме предоставените ни от LINQ методи .Where(), .Select(), .FirstOrDefault() и други според зависи какво искаме да постигнем.

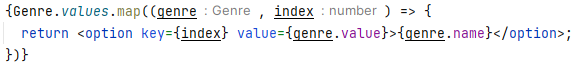
За автоматично мапване на пропъртитата вместо използването на **.Select()**, можем да използваме библиотеката AutoMapper. AutoMapper е инструмент за обектно-релационно мапване (ORM), който може да помогне при преобразуването на данни между два класа. Ето как можем да го използваме за мапване на пропъртитата на книгите:

1. Най-напред трябва да инсталираме AutoMapper чрез NuGet пакетния мениджър в Visual Studio или чрез командна конзола с командата **Install-Package AutoMapper**.
2. След като имаме инсталиран AutoMapper, трябва да дефинираме мапинг конфигурация между класовете, които искаме да мапнем. Нека предположим, че имаме класовете **Book** и **BookDetailsResponseModel**, където **BookDetailsResponseModel** е моделът, който искаме да върнем като резултат от LINQ заявката.

В случая за детайлите е направено с един .Where() метод, който намира по ID книгата, чиито данни искаме да видим, и след това със **.ProjectTo<BookDetailsResponseModel>** (предоставен от AutoMapper) избираме данните, които искаме да видим.



За изкарване на списък с всички книги заявката, която се използва е много подобна с разликата, че не използваме .Where() и след .Select() или .ProjectTo<>() добавяме .Order или .OrderByDescending в зависимост от това как искаме да бъдат подредени (например първо да излизат най-новите превозни средства) и след това пишем .ToList(), за да ни върне резултатите под формата на списък.

Да разгледаме и изгледите – за добавяне и промяна на данни на книга са използвани HTML, CSS и React, като е използван map-цикъл за изкарването на всички жанрове на книга под формата на падащ списък.

След правилно въведени данни натискаме бутона **Create**, който е закачен за метода **handleSubmit**:

Картина, която съдържа текст, екранна снимка, софтуер, дисплей

Описанието е генерирано автоматично

Той се грижи да изпрати **POST** заявка към back-end за създаване на книга. След успешен response от back-end се връща id-то на създадената книга и препращаме потребителя към новосъздадената книга. Съответно при неуспешен response се визуализира грешка.   
  
 Нека разгледаме какво се случва, когато бъде изпратена **POST** заявка за създаване на книга на сървъра.

Във **BooksController** имаме следния Action:

Картина, която съдържа текст, Шрифт, екранна снимка, линия

Описанието е генерирано автоматично

Endpoint-ът е достъпен само за потребители с роля администратор. В случай, че потребителят не е администратор се връща статус код **401.**

Когато командата бъде извикана, тя ще бъде препратена към **MediatR** пакета, който ще намери съответния хендълър:

Картина, която съдържа текст, екранна снимка, Шрифт, документ

Описанието е генерирано автоматично

**MediatR** е библиотека за посредничество (mediation), която предоставя поддръжка за изпращане на команди, извличане на заявки и обработка на събития във вашия код. Тя е създадена от Jimmy Bogard и предоставя лесен и прост начин за разделяне на задачите в приложението ви и изолиране на бизнес логиката от контролерите или другите слоеве.

В контекста на ASP.NET Core Web API, MediatR може да се използва за имплементация на патерна "CQRS" (Command Query Responsibility Segregation), който се препоръчва за съвременни архитектури на приложенията. С помощта на **MediatR** можете да изпращате команди към контролерите си, които след това ще бъдат обработени от отделни хендлъри (handlers), специфични за всяка команда.

Използването на **MediatR** позволява по-добро разделение на отговорностите и улеснява тестването на вашето приложение. Това също така ви дава гъвкавост да добавите нови команди и хендълъри без промяна на съществуващия код. **MediatR** улеснява взаимодействието между различните компоненти на вашия код и помага за постигане на по-добра модуларност и разширяемост на приложението.

## **Поръчки**

Тук отново са използвани всички CRUD операции с помощта на LINQ заявки.

Във front-end частта промяната на данни са направени отново идентично. Списъкът с всички поръчки е направен под формата на HTML таблица.



## **Тестване на приложението**

За тестване на приложението е избран подходът Unit Testing. Той представлява тестове, разписани от разработчика, които проверяват дали малка част от кода работи правилно. За целта са разписани тестове чрез Xunit.

Картина, която съдържа текст, екранна снимка, документ, номер

Описанието е генерирано автоматично

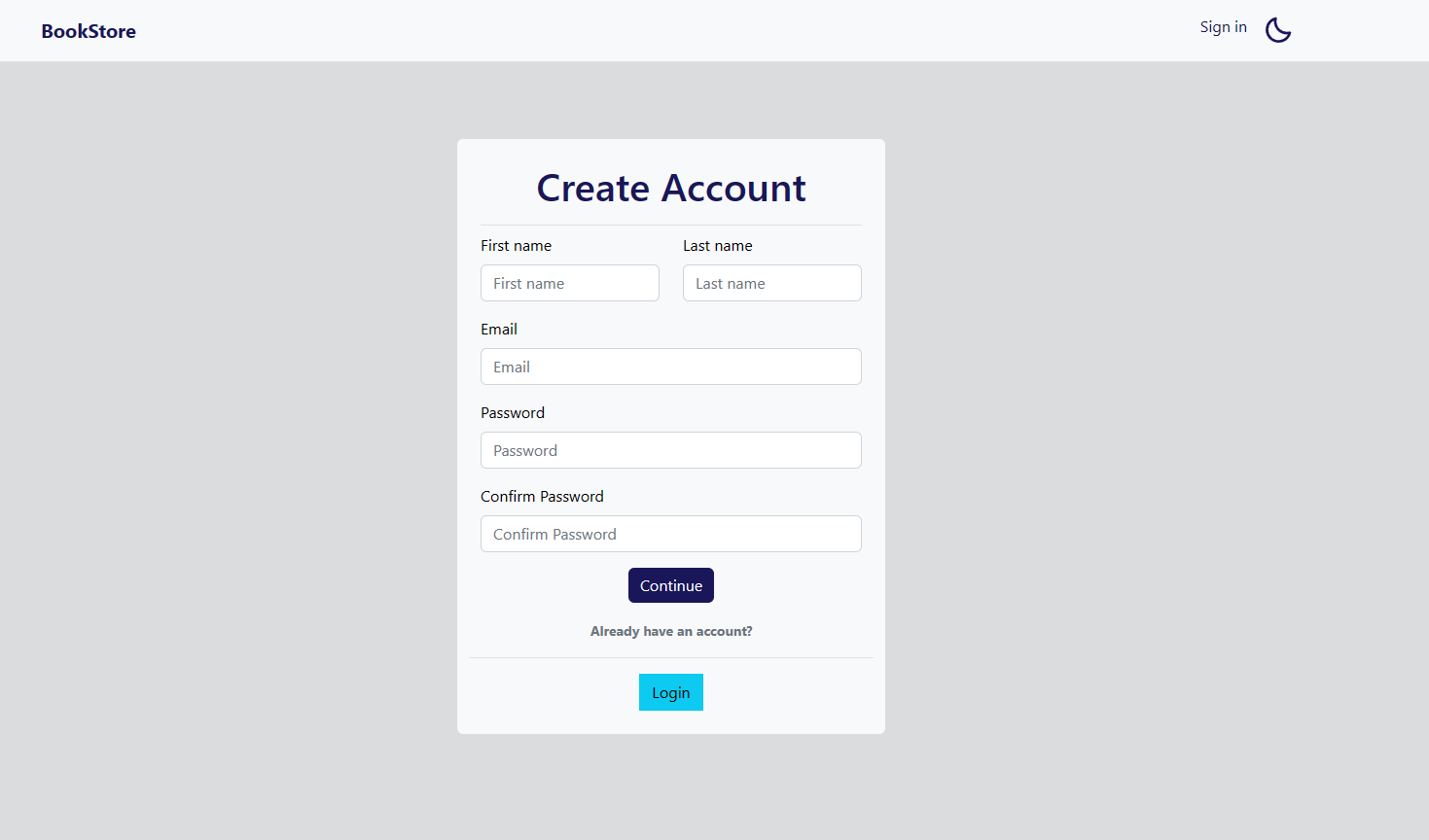
# **Ръководство на потребителя**

Когато се влезе в сайта, излиза следната начална страница:

Картина, която съдържа текст, екранна снимка, Онлайн реклама, Уебсайт

Описанието е генерирано автоматично

Първото нещо, което клиент трябва да направи е да се регистрира. Това става чрез натискане на бутона **Sign in** в горния десен ъгъл, който зарежда следната страница:



Тук клиентът попълва своите данни. Имейлът е от голямо значение, понеже след това се използва при всяко влизане в профила.

След попълване на данните клиентът трябва да натисне синия бутон **Continue** и ще бъде вписан в системата. При следващо влизане ще трябва само да въведе своят имейл и парола.

След регистрация клиентът бива препращан към началната страница. Както вече споменахме началната страница показва последните 10 добавени книги. Под тях има бутон **View all books**, който след натискането му отваря страница, на която клиентът може да види колекция от всички книги, както и да ги филтрира по заглавие, автор и жанр:

Картина, която съдържа текст, екранна снимка, Уебсайт, софтуер

Описанието е генерирано автоматично

След натискане на някоя от книги се показват детайлите на избраната книга:

Картина, която съдържа текст, екранна снимка, Уебсайт, Онлайн реклама

Описанието е генерирано автоматично

На тази страница може да се види заглавието, автора, жанра, цената и описанието на книгата, както и нейната корица. От тук също така може да се види повече информация за автора. След натискане на името на автора, клиентът бива препращан към следната страница:

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

Тук виждаме име и описание на дадения автор. След като натиснем бутона **More by “Author Name”** можем да видим колекция от книги писани от съответния автор:

Картина, която съдържа текст, екранна снимка, софтуер, Уеб страница

Описанието е генерирано автоматично

Да се върнем на детайлите на дадена книга. След натискане на бутона **Add to cart** клиентът има опция за добавянето ѝ в кошницата си.

От бутона **Cart**, който стои в горния десен ъгъл, клиентът може да види колекция от своите добавени книги:

Картина, която съдържа текст, екранна снимка, софтуер, Компютърна икона

Описанието е генерирано автоматично

От тук може да се редактира количеството, както и да се премахне дадена книга от кошницата.

От бутона **Order**, който се намира долу в десния ъгъл, клиентът създава поръчка с избираните от него книги. След натискането му той бива препращан към следната страница:

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

От тук може да се принтира фактурата, както и да видим детайлите на поръчката:

Картина, която съдържа текст, екранна снимка, софтуер, Компютърна икона

Описанието е генерирано автоматично

След натискане на бутона **My Account**, който се намира горе в десни ъгъл, клиентът отваря детайли за своя акаунт. От тук той може да прегледа своите поръчки, детайли, адрес, както и да излез от профила си:

Картина, която съдържа текст, екранна снимка, софтуер, номер

Описанието е генерирано автоматично

Нека да разгледаме функционалностите, които един администратор може да прави.

След влизане в системата, се появява бутон в навигацията **Administration:**

Картина, която съдържа текст, екранна снимка, Онлайн реклама, Марка

Описанието е генерирано автоматично

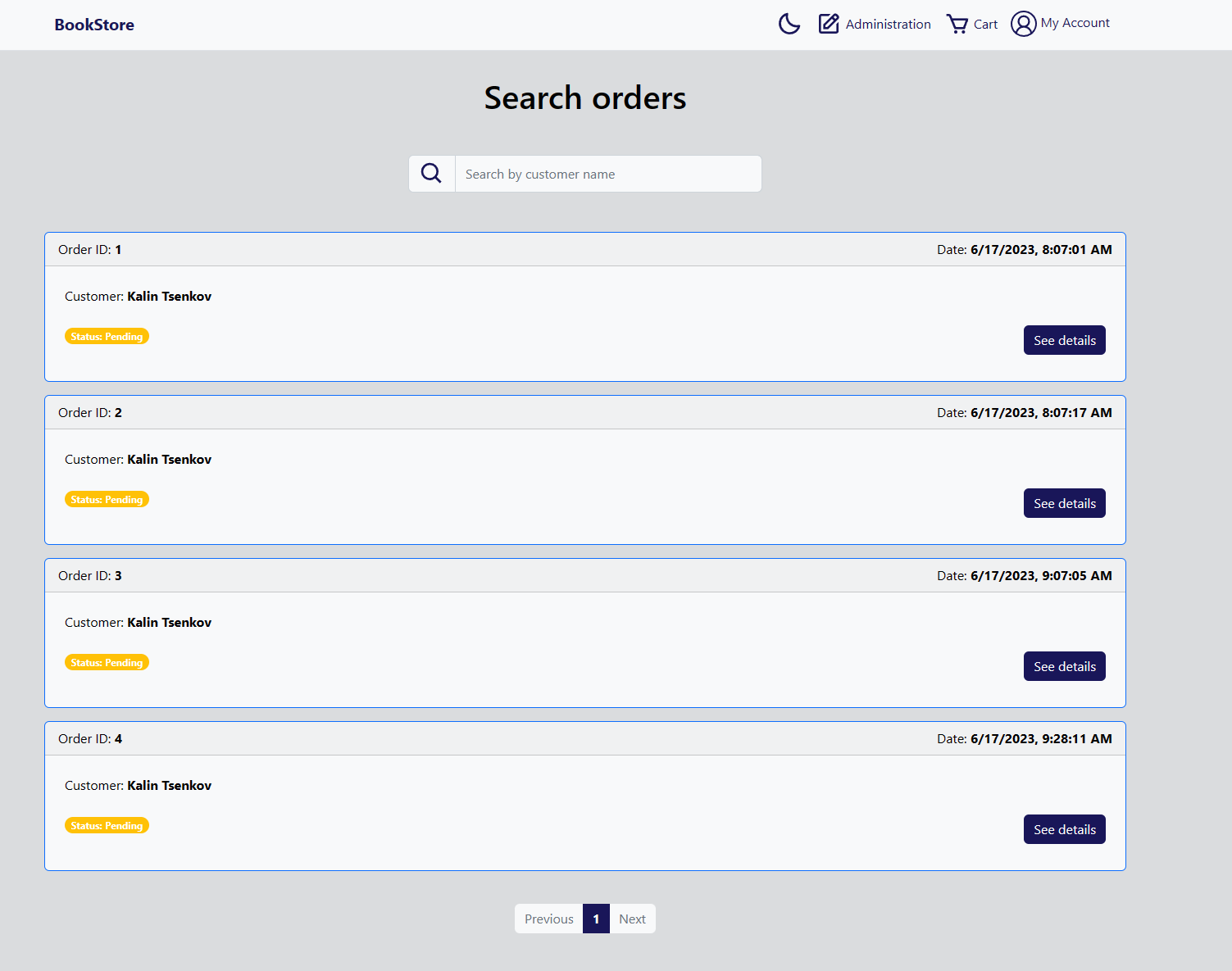
След натискането му администраторът бива препращан към следната страница:

Картина, която съдържа текст, екранна снимка, Шрифт, линия

Описанието е генерирано автоматично

От тук администраторите могат да:

* Проследяват и редактират поръчки:



Картина, която съдържа текст, екранна снимка, софтуер, Шрифт

Описанието е генерирано автоматично

* Търсят, редактират и изтриват автори:

Картина, която съдържа текст, екранна снимка, Шрифт, софтуер

Описанието е генерирано автоматично

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

Картина, която съдържа текст, екранна снимка, Шрифт, софтуер

Описанието е генерирано автоматично

* Създават нови автори:

Картина, която съдържа текст, екранна снимка, Шрифт, дизайн

Описанието е генерирано автоматично

* Създават нови книги:

Картина, която съдържа текст, екранна снимка, Шрифт, номер

Описанието е генерирано автоматично

* Редактират и изтриват книги:

Картина, която съдържа текст, екранна снимка, Уебсайт

Описанието е генерирано автоматично

Картина, която съдържа текст, екранна снимка, софтуер, Уеб страница

Описанието е генерирано автоматично

Картина, която съдържа текст, екранна снимка, софтуер, Уеб страница

Описанието е генерирано автоматичноКартина, която съдържа текст, екранна снимка, софтуер, графичен дизайн

Описанието е генерирано автоматичноВсички потребители могат да променят темата на приложението от **иконката на луничка/слънце**, която се намира горе в десни ъгъл:

# **Заключение**

В заключение, онлайн книжарницата **Book Store** е изградена с нови и надеждни технологии, с помощта на които е създаден интуитивен и лесен за употреба интерфейс. Включва основните функционалности, необходими за удобната покупка на книги онлайн. Потребителите могат лесно да търсят книги и да ги добавят в своята количка за пазаруване.

**Book Store** предлага широк избор от книги в различни категории и жанрове. Потребителите могат лесно да преглеждат каталога, да използват филтри и търсачка, за да намерят точно това, което търсят.

Процесът на добавяне на книги в количката и създаване на поръчки е опростен и интуитивен. Бутоните и функциите, необходими за тези действия, са поставени на видимо място, което улеснява потребителите да завършат своята покупка. Допълнителни стъпки като създаване на акаунт могат да бъдат опростени, например без нужда от писане на тайни въпроси и отговори, допълнителни потвърждения и други.

**Book Store** се стреми да осигури удобство и бързина при покупката на книги, като предоставя по-удобен и по-бърз начин за достъп до литературни произведения в сравнение с традиционните физически книжарници.

# **Използвана литература:**

1. [**https://en.wikipedia.org/wiki/Information\_system**](https://en.wikipedia.org/wiki/Information_system)

**Уикипедия, 31.05.2022г., Информационна система**

1. [**https://en.wikipedia.org/wiki/Computer\_hardware**](https://en.wikipedia.org/wiki/Computer_hardware)

**Уикипедия, 30.05.2022г., Хардуер**

1. [**https://en.wikipedia.org/wiki/Software**](https://en.wikipedia.org/wiki/Software)

**Уикипедия, 08.06.2022г., Софтуер**

1. [**https://bg.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D0%B8**](https://bg.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D0%B8)

**Уикипедия, 09.04.2022г., Данни**

1. [**https://bg.wiktionary.org/wiki/%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0**](https://bg.wiktionary.org/wiki/%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0)

**Wiktionary, 13.01.2021г., Процедури**

1. [**https://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0\_%D0%B2%D1%80%D1%8A%D0%B7%D0%BA%D0%B0**](https://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0_%D0%B2%D1%80%D1%8A%D0%B7%D0%BA%D0%B0)

**Уикипедия, 08.04.2021г., Обратна връзка**

1. [**https://www.investopedia.com/terms/d/decision-support-system.asp**](https://www.investopedia.com/terms/d/decision-support-system.asp)

**Investopedia, 03.01.2022г., Система за подкрепа на вземане на решения**

1. [**https://www.researchgate.net/publication/220268912\_Social\_Information\_Systems\_Review\_Framework\_and\_Research\_Agenda**](https://www.researchgate.net/publication/220268912_Social_Information_Systems_Review_Framework_and_Research_Agenda)

**Researchgate, 01.2011г., Социални информационни системи**

1. [**https://www.thebalancesmb.com/process-control-systems-pcs-2221184**](https://www.thebalancesmb.com/process-control-systems-pcs-2221184)

**The balance small business, 01.08.2019г., Системи за управление на процеса**

1. [**https://en.wikipedia.org/wiki/Management\_information\_system**](https://en.wikipedia.org/wiki/Management_information_system)

**Уикипедия, 29.04.2022г., Информационна система за управление**

1. [**https://www.oracle.com/database/what-is-a-data-warehouse/**](https://www.oracle.com/database/what-is-a-data-warehouse/)

**Oracle, Склад за данни**

1. [**https://www.oracle.com/erp/what-is-erp/**](https://www.oracle.com/erp/what-is-erp/)

**Oracle, Планиране на корпоративни ресурси**

1. [**https://en.wikipedia.org/wiki/Expert\_system**](https://en.wikipedia.org/wiki/Expert_system)

**Уикипедия, 12.06.2022г., Експертна система**

1. [**https://artsandculture.google.com/entity/search-engine/m02wvbrv**](https://artsandculture.google.com/entity/search-engine/m02wvbrv)

**Google, 10.06.2022г., Търсачка**

1. [**https://education.nationalgeographic.org/resource/geographic-information-system-gis**](https://education.nationalgeographic.org/resource/geographic-information-system-gis)

**National Geographic, 02.06.2022г., Географска информационна система**

1. [**https://en.wikipedia.org/wiki/Global\_information\_system**](https://en.wikipedia.org/wiki/Global_information_system)

**Уикипедия, 29.12.2021г., Глобална информационна система**

1. [**https://academic.oup.com/comjnl/article/34/3/225/442458**](https://academic.oup.com/comjnl/article/34/3/225/442458)

**Oxford Academic, 01.01.1991г., Мултимедийна информационна система**

1. [**https://www.solvexia.com/blog/what-is-office-automation-your-complete-guide**](https://www.solvexia.com/blog/what-is-office-automation-your-complete-guide)

**Solvexia, 30.03.2022г., Офисна автоматизация**

1. [**https://profio.co.id/en/6-stages-in-information-systems-development/**](https://profio.co.id/en/6-stages-in-information-systems-development/)

**Profio, 15.06.2020г., Разработване на информационна система**

1. [**https://en.wikipedia.org/wiki/Create,\_read,\_update\_and\_delete**](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete)

**Уикипедия, 13.06.2022г., CRUD операции**

1. [**https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core**](https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core)

**Microsoft, 2016г., ASP.NET Core**

1. [**https://react.dev/learn**](https://react.dev/learn)

**React, 2023г., Quick Start**

1. [**https://docs.microsoft.com/en-us/ef/core/**](https://docs.microsoft.com/en-us/ef/core/)

**Microsoft, 25.05.2021г., Entity Framework Core**

1. [**https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/**](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/)

**Microsoft, 18.02.2022г., LINQ**

1. [**https://github.com/kalintsenkov/BookStore**](https://github.com/kalintsenkov/BookStore)

**GitHub линк към проекта**