

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2022 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНТЕРНЕТ-МАГАЗИН З ПРОДАЖУ КОМП'ЮТЕРНИХ
ІГОР

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.21810225

Виконав студент 4-го курсу, групи 402

_____ *В.О. Степанюк*

«21» червня 2022 р.

Керівник: ст. викладач

_____ *В.В. Кошовий*

«21» червня 2022 р.

Миколаїв – 2022

цифрової дистрибуції, розробки веб застосунків, проектування баз даних та бекенду.

5. Перелік графічних матеріалів презентація.

6. Завдання до спеціальної частини опис основних питань охорони праці пов'язаних з професійною діяльністю та використання обрахунків для роботи в офісі.

7. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	ст. викладач Е.В. Макарова	

Керівник роботи ст. викладач Кошовий В.В.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Степанюк В.О.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 23 » листопада 2021 р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студента групи 402 ЧНУ ім.

Петра Могили

Степанюка Владислава Олександровича

Тема: «інтернет-магазин з продажу комп'ютерних ігор»

Об'єкт роботи – технології та алгоритми для досягнення поставленого завдання, а саме розробки інтернет-магазину. Система продажу ігор для клієнтів.

Предмет роботи – особливості проектування та створення інтернет-магазинів. Архітектура програми-клієнта, алгоритми і методи шифрування та захист даних користувачів в рамках цих систем.

Метою бакалаврської кваліфікаційної роботи є створення інтернет-магазину для онлайн дистрибуції ігор користувачам ресурсу.

Робота складається з фахового розділу і спеціальної частини з охорони праці. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі дипломної роботи представлені основні положення та проаналізовано методи створення інтернет-магазинів. Аналізуються існуючі представлення інтернет-магазинів та порівнюються між собою.

У другому розділі дипломної роботи представлена послідовність розробки інтернет-магазину. Описується стек технологій, який був обраний для виконання завдань, архітектура додатку та база даних, обрана для цього проекту.

У третьому розділі дипломної роботи описано процес розробки інтернет-магазину та бази даних до нього.

У четвертому розділі описується аналіз умов праці, безпеки праці, гігієни праці та виробничої гігієни. Також описані вимоги пожежної безпеки та освітлення і мікроклімату.

За результатами роботи зроблено висновки та пропозиції щодо створення інтернет-магазину.

Бакалаврська кваліфікаційна робота містить 74 сторінки, 45 рисунок, 14 таблиць, 17 використаних джерел та 0 додатків.

Ключові слова: Веб, Bootstrap, EFCore, MVC, SQLite, ASP.NET, C#, інтернет-магазин, дизайн, сервер, клієнт.

ABSTRACT

**bachelor's degree work of a student of group 401 at Petro Mohyla Black Sea
National University**

Stepaniuk Vladyslav Oleksandrovich

on topic: «Internet-shop for game distribution»

The object of work - technologies and algorithms to achieve the task, namely the development of an internet-shop. System for game distribution between clients.

Subject of work - Features of designing and creating an internet-shops. Client program architecture, algorithms and methods of encryption and protection of user data within these systems.

The purpose of the bachelor's thesis is to create an internet-shop to distribute games between clients.

The work consists of a professional section and a special part on labor protection. The explanatory note consists of an introduction, four sections, conclusions and appendices.

The first section of the thesis presents the main provisions and analyzes the methods of creating internet-shops. Existing views of online stores are analyzed and compared with each other.

The second section of the thesis presents the sequence of internet-shop development. Describes the technology stack selected for the tasks, the application architecture, and the database selected for this project.

The third section of the thesis describes the process of developing a internet-shop and database for it.

The fourth section describes the analysis of working conditions, occupational safety, occupational health and industrial hygiene. The requirements of fire safety and lighting and microclimate are also described.

Based on the results of the work, conclusions and suggestions for creating an internet-shop were made.

The bachelor's thesis contains 74 pages, 45 figures, 14 tables, 17 sources used and 0 appendices.

Key words: Web, Bootstrap, EFCore, MVC, SQLite, ASP.NET, C#, internet-shop, design, server, client.

ЗМІСТ

ЗМІСТ	2
ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ	7
1.1 Останні дослідження та тенденції розвитку напряму інтернет-магазинів... 7	
1.2 Огляд та аналіз наявних аналогів для виявлення вдалих рішень	10
1.3 Постановка задачі	13
Висновки до розділу 1.....	14
2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ СТВОРЕННЯ ВЕБ-ДОДАТКУ.....	15
2.1 Вибір та аналіз мови програмування.....	15
2.2 ASP.NET Core	17
2.3 Entity Framework Core	18
2.4 ASP.NET MVC Framework	20
2.5 LINQ to Entities	22
2.6 Bootstrap.....	23
2.7 Опис баз даних.....	24
Висновки до розділу 2.....	31
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ..	32
3.1 Архитектура проекту	32
3.2 Принцип роботи сайту	34
3.3 Принцип роботи відображення сайту	40
3.4 Аналітична частина сайту	45
Висновки до розділу 3.....	47
4 ОХОРОНА ПРАЦІ	50
Висновки до розділу.....	60
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних

СУБД – система управління базами даних

ASP – Active Server Pages

EFC – Entity Framework Core

HTTP – Hypertext Transfer Protocol

MVC – Model View Controller

SQL – Structured Query Language

ВСТУП

В епоху інтернету технології мають тенденцію розвиватись в напрямку спрощення повсякденної рутини. Ті справи що займали у нас, ще години в недалекому минулому, перетворились в декілька кліків мишки. На сьогоднішній день в тебе немає потреби витратити дорогоцінний час на подорож до магазину, який може навіть знаходитись в іншому місті, так як є інтернет-магазини. З їх появою зменшилась взаємодія з системою а також збільшилась різноманітність товарів та розширились опис та їх характеристика. Інші клієнти можуть залишити свої відгуки що-до товару або роботи магазину, що в свою чергу допомагає іншим клієнтам вирішити який саме товар або послуга їм підійдуть.

Створення інтернет-магазину шлях для покращення та спрощення продажу товарів. Можливість створення такого сайту не є великої проблемою, так як інтернет наповнений різноманітними інструментами які спрощують розробку та розміщення його в мережі. З появою великої різноманітності інтернет-магазинів та їх розвитком було створено узагальнені вигляд та функціональність, які очікують користувачі від сайту. Зважаючи на це незвичний дизайн та ускладнений інтерфейс все це може відлякує користувачів.

В інтернеті можна знайти велику кількість інтернет-магазинів на різні теми. Більшість таких ресурсів, окрім велетнів індустрії, мають доволі архаїчний вигляд. Тому важливо балансувати між загальними трендами та не допускати базових помилок.

Очевидно що за час створення кваліфікаційної роботи не є можливим створення високофункціонального продукту але розробка міцного фундаменту з подальшою можливістю масштабування є основною цілю роботи. Під час створення інтернет-магазину було вирішено зосередити увагу на інтуїтивно зрозумілому інтерфейсі, імперативної функціональності та надійного рівня безпеки.

Для реалізації даного проекту було сформовано наступні завдання:

- проаналізувати наявні аналоги для виявлення вдалих рішень;

- проаналізувати та виявити особливості проектування та створення інтернет-магазинів;
- розробити унікальний інтерфейс магазину;
- провести тести магазину.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

1.1 Останні дослідження та тенденції розвитку напряму інтернет-магазинів

Однією з головних тенденцій розвитку веб-застосунків і технологій є розробка мобільних додатків. Сьогодні люди все частіше використовують мобільні пристрої замість стаціонарних комп'ютерів. У цьому напрямку є багато проблем і нюансів, які вимагають певного способу їх вирішення, наприклад:

- інтернет, яким користується клієнт, зазвичай не безлімітний, і можуть виникати тривалі затримки в передачі даних між сервером і «клієнтом» (наприклад, смартфоном користувача). Це може призвести до того, що програма або сайт не зможуть своєчасно відповідати на запити користувачів, що може призвести до поганої роботи програми та втрати клієнтів. Крім того, відсутність необмежених з'єднань накладає певний ліміт на кількість даних, якими сервер може періодично обмінюватися, що призводить до необхідності виконання певних розрахунків щодо ємності пристрою користувача;
- З попереднього тезису частина проблеми полягає в наступному: необхідно враховувати потужність більшості її користувачів, особливо з точки зору оперативної пам'яті, процесора та акумулятора. Не всі комп'ютери можна розмістити на пристрої користувача. На сьогоднішній день найкращим рішенням є рівномірний розподіл процесів між сервером і пристроями користувача.
- Основною проблемою мобільних пристроїв можна вважати їх сильні сторони: їх розмір і спосіб управління ними. Пам'ятайте, що екрани мобільних пристроїв набагато менші за екрани комп'ютерів, тому вам потрібно буде масштабувати інтерфейс і зробити його адаптивним.

Ці проблеми з мобільними пристроями частково можна пояснити проблемами на комп'ютерах, але на комп'ютерах вони працюють набагато гірше.

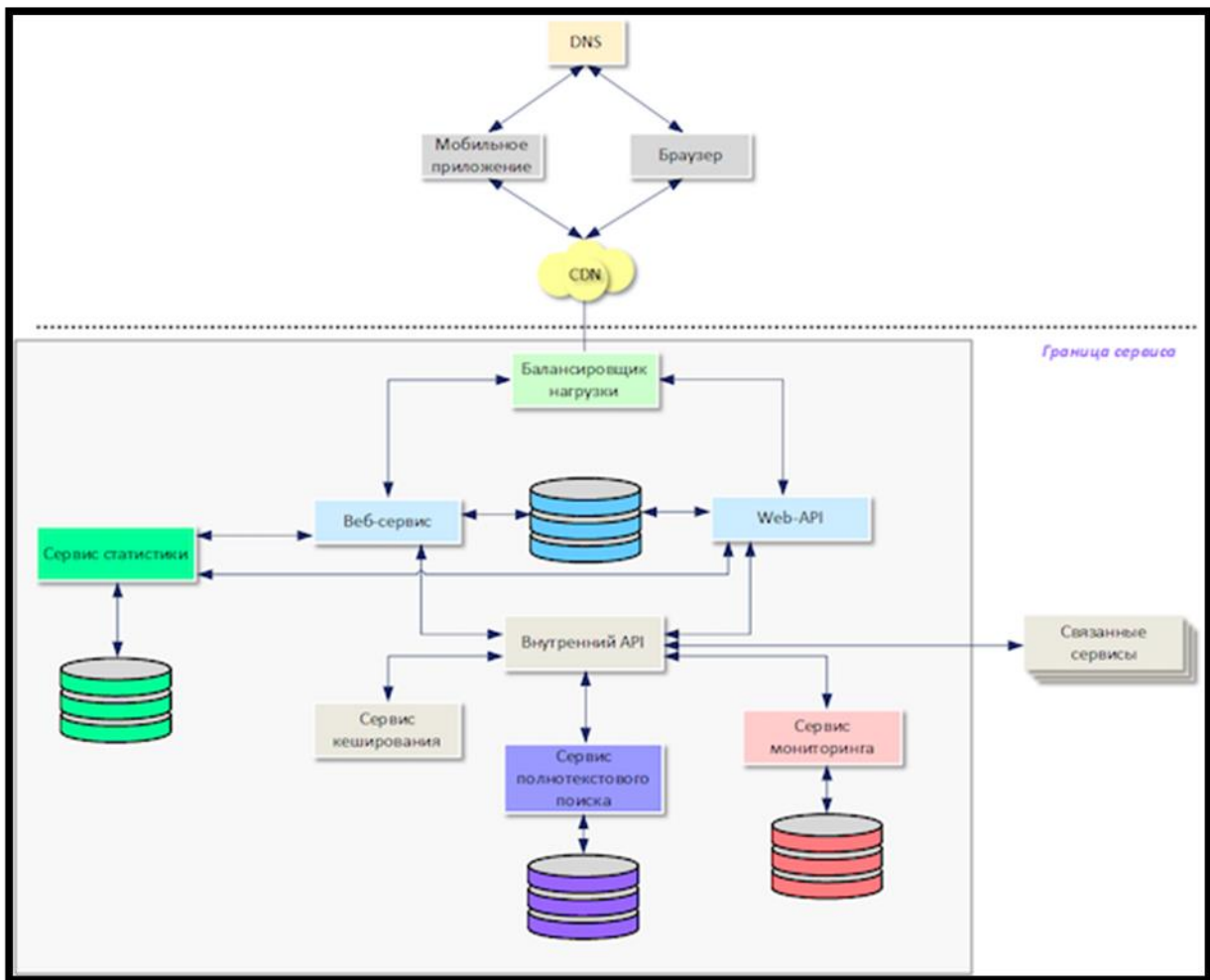


Рисунок 1.1 – Приклад сучасної архітектури веб-застосунку

Основна тенденція розвитку технологій, які мають використовувати веб-сервіси для клієнтів і спрощувати розробку для програмістів, — хмарні технології, тобто потужні центри обробки даних з великою обчислювальною потужністю для нормальної роботи, недостатня якість у багатьох частинах нашого світу Інтернет. Microsoft, Amazon та Google є одними з найважливіших компаній, що розвиваються в цій галузі.

Загалом принцип мікросервісної архітектури [1] для веб-додатків активно розвивається, замінюючи монолітну архітектуру (див. рис. 1.1).

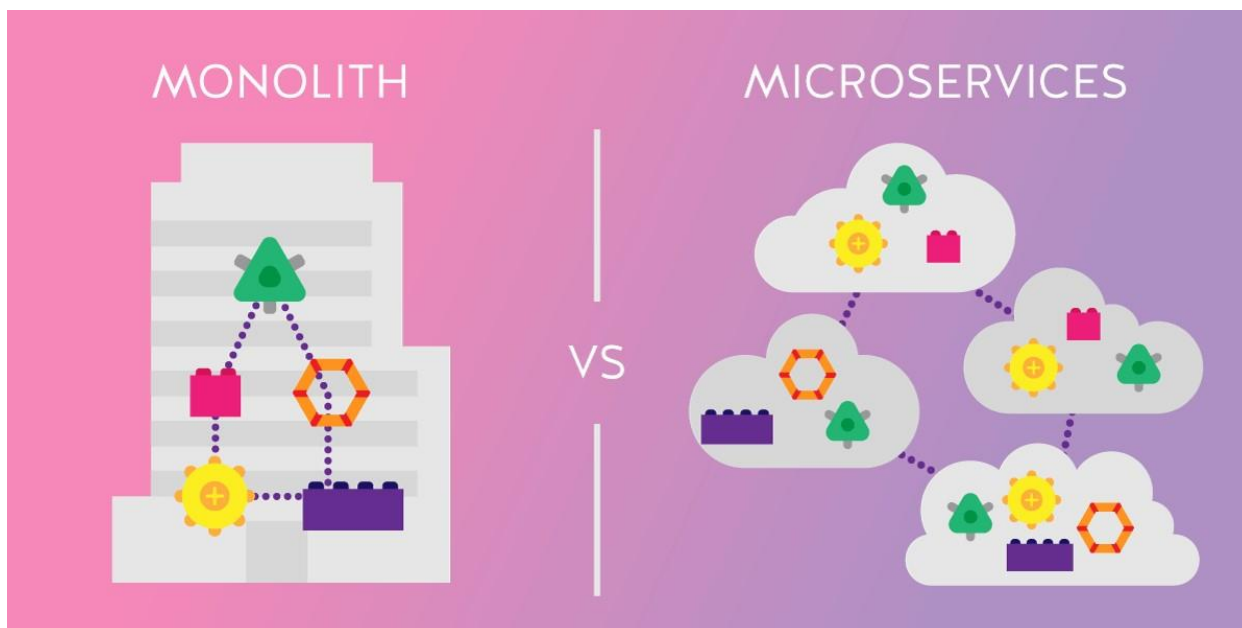


Рисунок 1.2 – Порівняння монолітної та мікросервісної архітектур веб-застосунків

Величезною перевагою архітектури мікросервісів є можливість динамічно змінювати програми без необхідності мати справу з переробкою усього продукту з нуля. Наприклад, якщо веб-сайт, побудований на монолітній архітектурі, потребує додавання нових функцій або внесення змін до існуючих систем, йому, швидше за все, потрібно буде врахувати це в усьому коді а, у гіршому випадку переробити весь проект.

Мікросервісна архітектура (див. рис. 1.2) дозволяє додавати нові модулі як у кошик, і різні модулі будуть взаємодіяти один з одним за певними правилами. Але ці модулі повністю ізольовані один від одного і не можуть вносити зміни безпосередньо один в одного. Крім того, ця архітектура дозволяє створювати мережеві дистрибуції та центри обробки даних, де можна навіть створювати резервні копії (екстрене копіювання) даних.

Крім того, однією з найважливіших розробок у сфері веб-додатків є різноманітність програм, інструментів для захисту користувачів ваших послуг і загальної інформації вашого сервісу. Сьогодні з розвитком нових способів розробки програмного забезпечення та збільшенням потужності комп'ютерної

техніки з'являється також багато нових способів створення хакерських атак, крадіжки персональних даних користувачів та «розміщення» веб-сайтів.

Це призводить до необхідності у такому ж швидкому темпі розвивати нові способи боротьби із цими негараздами (див. рис. 1.3).

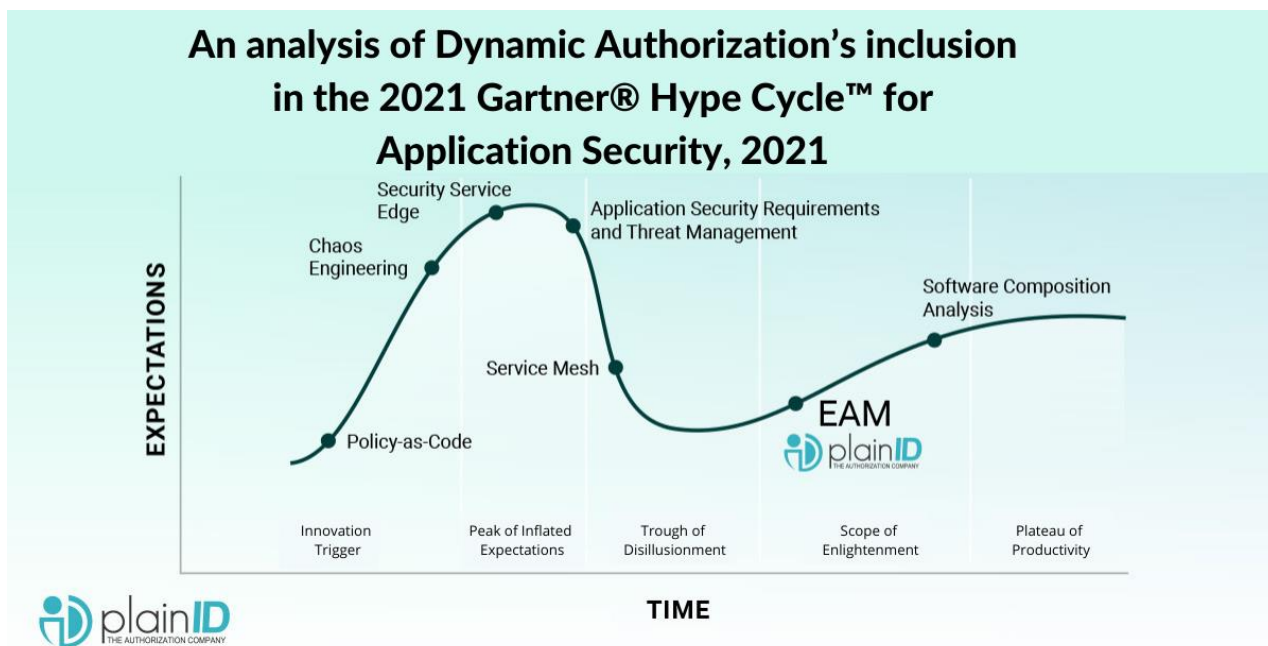


Рисунок 1.3 – Графік розвитку засобів захисту веб-застосунків

Сучасними засобами для захисту веб-сайтів можна назвати фаєрволи, використання HTTPS замість HTTP, різні додатки та застосунки для захисту від DOS-атак, тощо.

1.2 Огляд та аналіз наявних аналогів для виявлення вдалих рішень

Основні ринки, як Amazon та eBay, мають обмежувальні правила щодо продавців. Ці місця применшують брендинг торговців, щоб зберегти власний однорідний імідж.

Продавці з власними інтернет-магазинами грають в іншу гру. Вони повністю контролюють досвід своїх клієнтів. Продавці можуть використовувати свій веб-сайт, щоб безпосередньо зв'язуватися зі своїми клієнтами та заохочувати до покупок.

Згідно з дослідженням Kissmetrics, 93% покупців вважають зовнішній вигляд найважливішим фактором при покупці.

Елементи дизайну повинні захоплювати відвідувачів, наприклад, використовувати зображення на всю ширину та уникати відволікання за допомогою мінімального тексту. Ось приклад дизайну, який Reebok вибрав для свого інтернет-магазину (див. рис. 1.4):

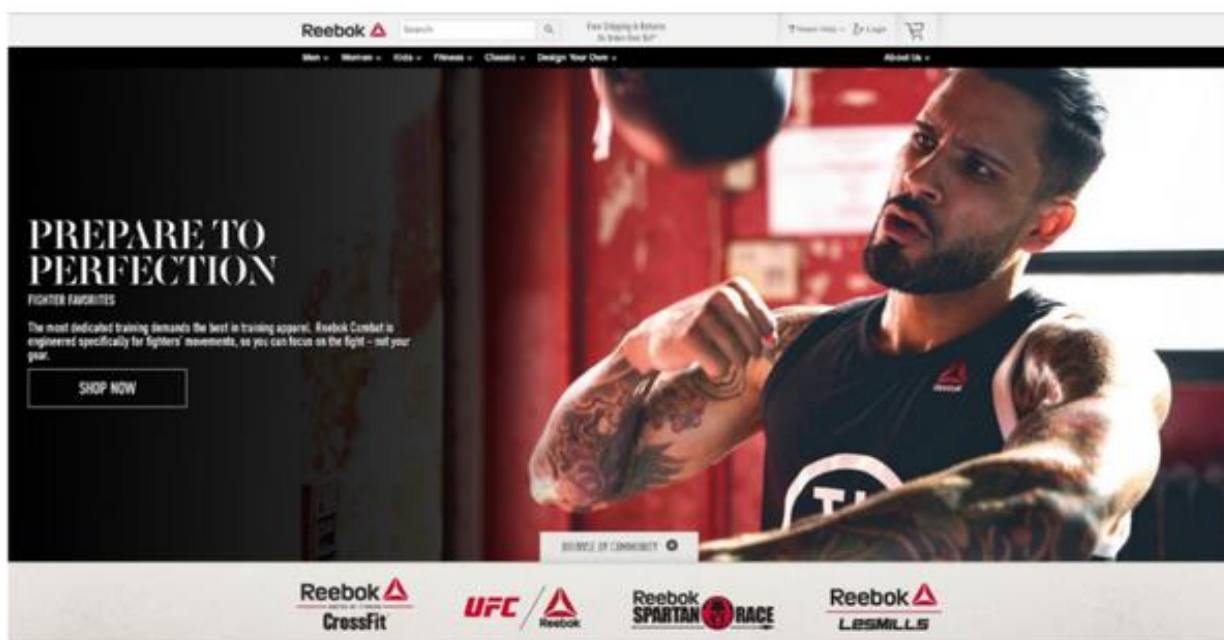
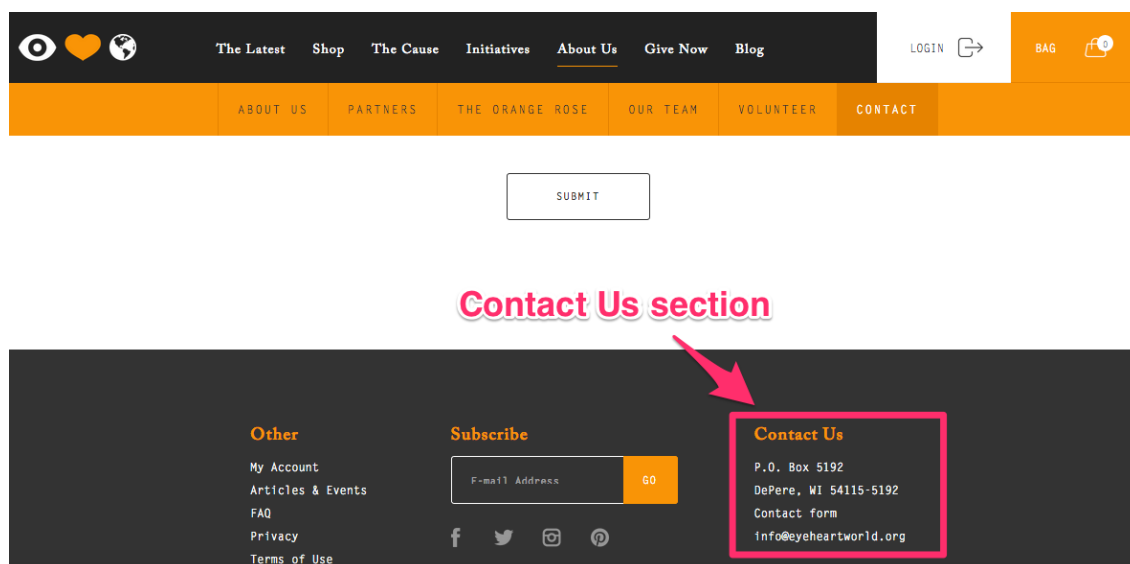


Рисунок 1.4 – Приклад дизайну сайту Reebok

Якщо відвідувач заходить сторінку «Про нас», це означає, що він уже зацікавився і дає можливість довести свою зацікавленість продуктом. Основна задача – чітко розповісти про себе, свою діяльність, і, головне, що пропонується клієнтам. Усі відгуки, досягнення та будь-які соціальні докази тут вітаються.

Наповнення сторінки «Про нас» занадто великою кількістю інформації може бути надзвичайним для відвідувачів сайту. Деякі інтернет-магазини, як Tens у наведеному нижче прикладі (див. рис. 1.5), вибирають мінімалістичний дизайн, щоб інформація була чіткою та легко зрозумілою.



Риссунок 1.5 – Приклад дизайну сайту Tens

Жоден інтернет-магазин не може існувати без платежів. Потрібен широкий вибір популярних способів оплати, щоб клієнти завжди могли легко зробити покупку.

Потрібно дозволити клієнтам створювати обліковий запис у інтернет-магазині та зберігати свою платіжну інформацію, щоб зробити майбутні покупки більш зручними.

Інтернет-магазин повинен отримувати трафік, щоб здійснити покупки. Ключем до створення трафіку є пошукова оптимізація. Включаючи ключові слова SEO в назви та описи продуктів, сайт з більшою ймовірністю буде займати вищі місця в результатах пошуку. Підвищена видимість є ключовим компонентом для більшої кількості відвідувань сайту та збільшення кількості конверсій.

У зв'язку з зростаючим використанням телефонів для покупок в Інтернеті, потрібно переконатися, що дизайн сайту буде зручним для мобільних пристроїв, щоб покупці могли робити покупки, незалежно від того, який пристрій вони використовують.

Інтернет-магазини надає повну автономію щодо досвіду клієнтів. Замість того, щоб дотримуватися політики ринку, можна створити власний інтернет-

магазин відповідно до своїх умов, щоб формувати взаємодію з продуктом своїх покупців.

З такою свободою можна заохочувати більше конверсій, якщо знати, як створити кожен елемент магазину. Дотримуючись певних стратегій можна краще зрозуміти, як клієнти сприймають кожен елемент магазину, і застосувати ці знання для покращення магазину. Організація інтернет-магазину відповідно до побажань і мотивації ключових покупців дає перевагу перед менш уважними конкурентами.

1.3 Постановка задачі

Тема дипломної роботи являє собою створення веб-сайту який складається із клієнта та сервера, який дозволить робити покупки ігор в інтернеті та робити пропозиції клієнтам на основі минулих транзакцій.

Завданням цієї роботи є створення веб-сайту який складається із клієнта та сервера, який дозволить робити покупки ігор в інтернеті та робити пропозиції клієнтам на основі минулих транзакцій.

Об'єктом дослідження – технології та алгоритми для досягнення поставленого завдання, а саме розробки веб-сайту.

Предметом досліджень – інформаційні процеси, моделі створення веб-сайтів. Особливості проектування та створення веб-сайтів. Архітектура веб-сайтів, реалізація клієнт-серверного зв'язку.

Етапи для вирішення поставленої задачі:

- розробка структури проекту, для створення міцного фундаменту подальшої масштабованості;
- створення серверної частини, щоб обробляти записи клієнтів для подальшої передачі, зберігання та обробки інформації;
- створення клієнтської частини, для відображення отриманої інформації від сервера, та використання предоставлених функцій додатку, за допомогою веб інтерфейсу;

- створення архітектури БД, для зручного збереження великих об'ємів даних та їх подальшого використання;
- налагодження зв'язків між клієнтом, сервером та БД, та тестування всієї системи;
- підготовка та розгортання додатку в інтернеті, використовуючи серверні можливості хостингів.

Виконавши головні етапи створення веб-додатку, протрібно провести повне тестування всієї системи, для оцінки якості, швидкодії та навантажоздатності.

Висновки до розділу 1

Обрана тема ДР присвячена розробці додатків для організації взаємодії між користувачами на прикладі веб-сайту. В першому розділі розглянута актуальність застосунку та його аналоги.

За мету було обрано створити інтернет-магазин задовольняючий потреби клієна, використовуючи сучасні інструменти та технології для розробки веб-сайтів.

2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ СТВОРЕННЯ ВЕБ-ДОДАТКУ

2.1 Вибір та аналіз мови програмування

C# сьогодні є однією з найпопулярніших і затребуваних мов програмування [2]. Це пов'язано з рядом суттєвих переваг, які дозволяють використовувати мову як у простих настільних додатках, так і в мобільних додатках, а також у розробці веб-додатків.

Ця мова програмування була розроблена в 2000 році компанією Microsoft (див. рис. 2.1) (що є значною перевагою під час реалізації дипломного проекту, але про це пізніше) і завдяки плідній роботі розробників C# вона взяла на озброєння позитивні сторони мов програмування, таких як C++ і Java, але також став покращеною версією. Це пояснює різноманітність використання мови.



Рисунок 2.1 – Логотип C#

C# підтримує парадигми об'єктно-орієнтованого програмування (ООП), що дозволяє використовувати абстрактні типи даних, успадковувати та створювати роботу програми, взаємодіючи з окремими класами та їх екземплярами. Клас Object, який успадковується від усіх інших класів, дозволяє використовувати стандартний набір команд, але функції кожного класу за потреби розширюють можливості об'єктів.

Розглянемо основні переваги мови C#:

- багатоплатформенний;
- хоча мова була створена Microsoft, завдяки .NET, мова C# може використовуватися на різних платформах;
- бібліотеки;

- багато бібліотек C# створено для різних цілей. Один із них буде використаний при реалізації програмного забезпечення цієї дипломної роботи;
- платформа .NET;
- ця платформа досить потужна і містить багато компонентів різного призначення (робота з базами даних, веб-додатками, графічним інтерфейсом тощо). Все це реалізовано таким чином, що під час розробки дуже легко поєднувати різні технології в одному додатку;
- середовище розвитку.

Потужне середовище розробки Visual Studio було створено для написання програм на C# (та інших мовах Microsoft). Він підходить як для новачків, так і для професіоналів.

C# був розроблений для CLR, що розширює можливості самої мови програмування. Наприклад, C# не реалізує збір сміття, але двигун CLR виконує це завдання.

Що стосується самого CLR, то він є компонентом .NET Framework, середовища компіляції коду. Загалом, платформа .NET – це масштабний проект від Microsoft, що містить багато корисних компонентів, що дозволяють працювати з веб-додатками, базами даних тощо. (ASP.NET, ADO.NET відповідно).

Якщо підсумувати велику кількість можливостей, які пропонує платформа та її функціональні можливості, ми повинні виділити наступне:

- .NET компілює програмні додатки в проміжний код MSIL (інша назва: IL, проміжна мова), який перетворюється на загальний код у CLR;
- платформа має велику кількість бібліотек, що дозволяє використовувати в одному проекті різні технології, які завдяки спільній платформі можна легко комбінувати та будувати їх взаємодію;
- середовище розробки Visual Studio має широкий спектр інструментів для роботи з технологіями платформи та постійно оновлюється та вдосконалюється;

- спектр можливостей, які пропонує платформа, спрямований на найнеобхідніші напрямки програмування.

2.2 ASP.NET Core



Рисунок 2.2 - ASP.NET Core

ASP.NET Core — це безкоштовна веб-структура з відкритим вихідним кодом, створена Microsoft та спільнотою [3]. Він продуктивніший, ніж ASP.NET 4. Відокремлений Framework працює як із .NET Framework, у Windows і .NET Core між етапами. У будь-якому випадку наведена нижче доставка ASP.NET Core 3 працюватиме лише з довідкою про видалення .NET Core для .NET3 Framework.

Структура являє собою завершену переробку, яка об'єднує ASP.NET MVC і ASP.NET Web API в єдину модель програмування.

Незважаючи на те, що це інша структура, з огляду на інший веб-стек, вона має серйозний рівень подібності з ASP.NET MVC. Додатки ASP.NET Core підтримують поруч один з одним форми, де різні програми, що працюють на схожому ПК, можуть орієнтуватися на різні варіанти ASP.NET Core. Це виходить за межі можливостей минулих адаптацій ASP.NET.

Фреймворк під кодовою назвою ASP.NET vNext мав називатися ASP.NET 5, коли він був доставлений. У будь-якому випадку, щоб показати, що це все,

крім простого оновлення поточної ASP.NET Framework, Microsoft змінила назву на ASP.NET Core в годину її публікації.

2.3 Entity Framework Core

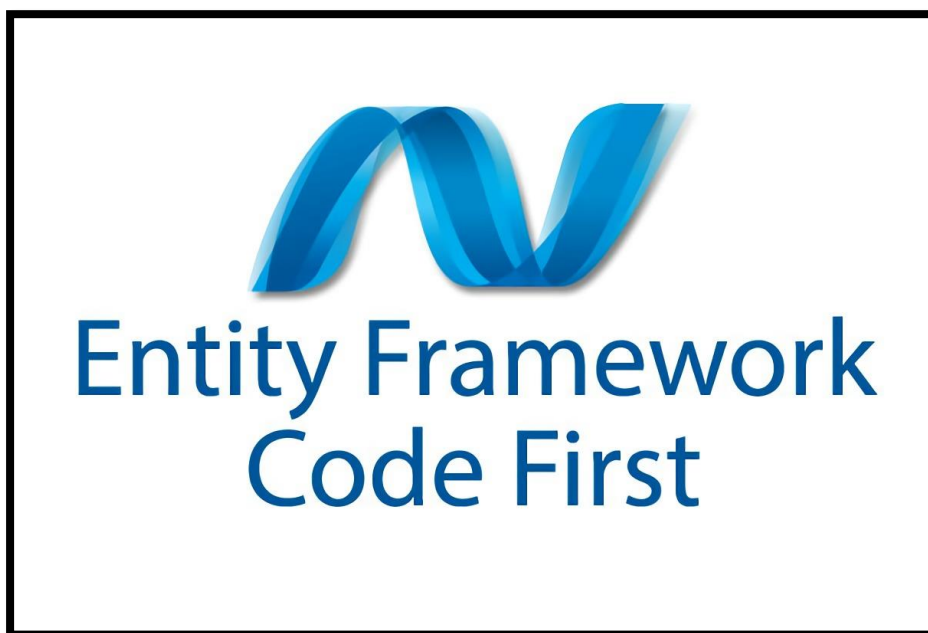


Рисунок 2.3 – Entity Framework

Це набір API доступу до даних для Microsoft .NET Framework [4], націлений на версію ADO.NET, яка постачається з .NET Framework 3.5. Він був випущений як окреме оновлення разом з пакетом оновлень 1 для .NET Framework після випуску .NET Framework 3.5 і Visual Studio 2008. Нова версія Entity Framework (версії 4.0) буде випущена разом із Visual Studio 2010. та .NET Framework 4.0.

Сутність Entity Framework — це об'єкт, який має ключ, що представляє первинний ключ логічної сутності сховища даних. Концептуальна модель даних сутності (модель сутності-відношення) зіставляється з моделлю схеми сховища даних. Використовуючи модель даних Entity, Framework дозволяє розглядати дані як сутності незалежно від їх базового представлення сховища даних.

Entity SQL — це мова, подібна до SQL, для запитів моделі даних Entity (а не базового сховища даних). Аналогічно, розширення Linq, Linq-to-Entities, надають запити, введені в моделі даних Entity. Запити Entity SQL і Linq-to-Entities внутрішньо перетворюються на канонічне дерево запитів, яке потім перетворюється на зрозумілий запит до базового сховища даних (наприклад, у SQL у випадку реляційної бази даних). Суб'єкти можуть використовувати свої зв'язки, а їх зміни повертаються в сховище даних.

Перша версія Entity Framework (EFv1) була включена в пакет оновлень 1 для .NET Framework 3.5 і пакет оновлень 1 Visual Studio 2008, випущені 11 серпня 2008 року. Ця версія зазнала широкої критики.

Друга версія Entity Framework, названа Entity Framework 4.0 (EFv4), була випущена як частина .NET 4.0 12 квітня 2010 року, і багато критики, висунутої версією 1, були розглянуті.

Третя версія Entity Framework, версія 4.1, була випущена 12 квітня 2011 року за підтримки Code First.

Оновлення до версії 4.1 під назвою Entity Framework 4.1 Update 1 було випущено 25 липня 2011 р. Воно включає виправлення помилок та нові підтримувані типи.

Версія 4.3.1 була випущена 29 лютого 2012 року. Є кілька оновлень, наприклад підтримка міграції.

Версія 5.0.0 має такі характеристики:

- нові типи даних (перераховані типи (Enum), просторові типи даних, доступні через типи DBGeography і DbGeometry);
- можливість роботи з функціями, які повертають табличні дані. Результат подібний до виклику збереженої процедури, за винятком того, що ми можемо використовувати ці функції в запитах LINQ;
- усі запити LINQ автоматично компілюються та кешуються. Це забезпечує кращу продуктивність при повторюваних викликах запитів;

- підтримка кількох діаграм для однієї моделі даних. Частина всієї моделі можна відобразити на менших діаграмах, покращуючи читабельність і вивчення моделі;
- загальні покращення продуктивності EF порівняно з попередніми версіями;
- можливість використовувати збережені процедури, які повертають декілька наборів результатів. Кожен набір результатів процедури буде відображено на власний складний тип.

Наразі існують дві новіші версії Entity Framework: EF 6 і EF Core.

2.4 ASP.NET MVC Framework



Рисунок 2.4 – ASP.NET MVC Framework

ASP.NET MVC Framework — це фреймворк для створення веб-додатків, що реалізує макет контролера для перегляду моделі [5].

У квітні 2009 року вихідний код ASP.NET MVC був випущений під публічною ліцензією Microsoft (MS-PL). 27 березня 2012 року ліцензію було змінено на Apache License 2.0.

ASP.NET MVC 6 (див. рис. 2.4.) наразі будується як функція ASP.NET Core; 27 червня 2016 року була доставлена адаптація 1.0.0.

Етап ASP.NET MVC залежить від комбінації трьох сегментів: Controller, Model і Render. контролер отримує вимоги, вимірює внески клієнта, співпрацює з моделлю та презентацією та повертає результати підготовки запиту клієнта.

Модель відноситься до рівня, який відображає обґрунтування об'єднання інформації в додатку. Подання отримує інформацію від контролера і створює компоненти інтерфейсу для відображення даних.

Механізм відображення використовується для керування розміткою та вбудованим кодом у поданні. Раніше MVC 5 використовував два механізми: Web Forms і Razor.

Починаючи з MVC 5, єдиним двигуном, який працював природно, був Razor. Механізм WebForms використовує реєстри .aspx, а Razor використовує документи .cshtml і .vbhtml для зберігання коду розташування. Передумовою структури мови Razor є знак @, після чого він перемикається на кодування C#/VB.NET. Ви також можете використовувати сторонні механізми. Журнали показу не є стандартними статичними сторінками з HTML-кодом, і протягом часу, коли контролер витрачає на створення відповіді за допомогою представлень, вони розміщуються в класах, з яких будується сторінка HTML.

Під час підготовки вимог структура ASP.NET MVC залежить від системи керування, яка показує всі майбутні запити на курси, характеризується структурою, яка вказує, який контролер і модель повинні обробляти цей запит. Явний курс, звичайно, надає структуру з трьома інтерфейсами:

- контролер;
- модель;
- презентація.

ASP.NET MVC і ASP.NET Web Forms — це дві пов'язані інновації, які залежать від цього етапу ASP.NET. Але тоді ASP.NET MVC має кілька переваг перед веб-формами ASP.NET:

- розподіл обов'язків (окреме вдосконалення різних частин - регулятора, моделей, зображень);
- покращений процес тестування;
- покращена адаптивність і корисні налаштування для ваших потреб.

2.5 LINQ to Entities



Рисунок 2.5 – LINQ

Language-Integrated Query (LINQ) — це назва набору технологій, що базуються на інтеграції функцій запитів безпосередньо в мову C#. Традиційно під час компіляції або підтримки IntelliSense запити даних представлялися у вигляді простих рядків без перевірки типу. Крім того, вам потрібно вивчити різну мову запитів для кожного типу джерела даних: баз даних SQL, документів XML, різних веб-сервісів тощо. У LINQ запит є першокласною мовною конструкцією, такою самою, як класи, методи та події. Запити пишуться для колекцій строго типізованих об'єктів за допомогою ключових слів мови та знайомих операторів. Сімейство технологій LINQ забезпечує узгодженість запитів для об'єктів (LINQ to Objects), реляційних баз даних (LINQ to SQL) та XML (LINQ to XML).

Для розробника, який пише запити, найбільш помітною «мовною» частиною LINQ є вираз запиту. Висловлювання запиту записуються у декларативному синтаксисі запитів. Використовуючи синтаксис запиту, ви

можете виконувати операції фільтрації, сортування та групування в джерелах даних із мінімальним кодом. Він використовує ті ж самі базові шаблони виразів запитів для запиту та перетворення даних у базах даних SQL, наборах даних ADO.NET, XML-документах та потоках, а також колекціях .NET.

Ви можете писати запити LINQ на C# для баз даних SQL Server, XML-документів, наборів даних ADO.NET та будь-якої колекції об'єктів, що підтримують IEnumerable або універсальний інтерфейс IEnumerable<T>. Підтримка LINQ також надається третіми сторонами для багатьох веб-служб та інших реалізацій баз даних.

2.6 Bootstrap

Bootstrap — це веб-фреймворк для розробки веб-сайтів та веб-додатків (див. рис. 2.6).

Bootstrap містить шаблони дизайну на основі HTML і CSS для розробки інтерфейсу, включаючи такі функції, як типографіка, форми, кнопки, навігація та інші компоненти інтерфейсу [7].



Рисунок 2.6 – Bootstrap

Вони можуть включати настроювані спадні меню, панелі навігації, різні типи кнопок. Ідея полягає в тому, що ці елементи можна легко об'єднати, щоб створити веб-сайт, готовий до мобільних пристроїв.

Bootstrap є відкритим вихідним кодом і розміщується, розробляється та підтримується на GitHub.

Марк Отто та Джейкоб Торнтон створили Bootstrap у Twitter для роботи над узгодженістю пристроїв, що використовуються на сайті, та зменшенням обслуговування. Продукт раніше був відомий як Twitter Blueprint, а іноді його називають Twitter Bootstrap.

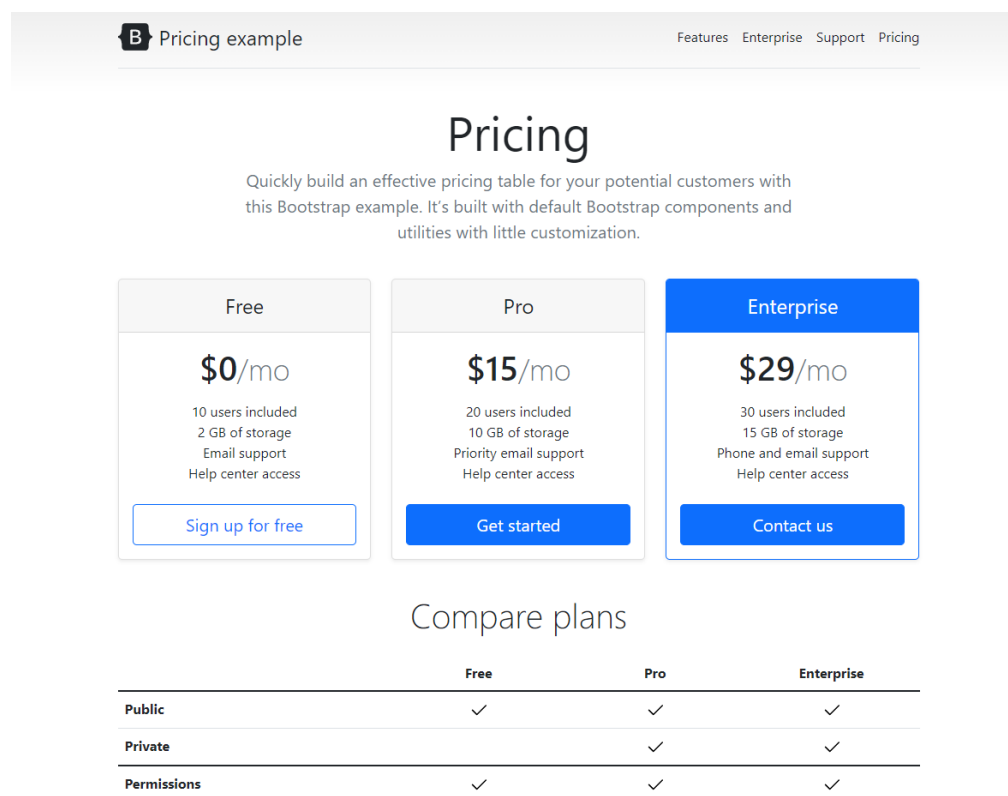


Рисунок 2.7 - Приклад сайту з використанням Bootstrap.

2.7 Опис баз даних

База даних — це набір даних, організований таким чином, щоб пізніше можна було легко отримати доступ до інформації. Дані будуть точнішими, надійнішими та простими у використанні, якщо є база даних.

За визначенням, дані в базі даних мають бути впорядковані відповідно до послідовного, логічного набору основних принципів. Термін модель даних

описує логічну структуру бази даних, яка визначає правила організації та керування інформацією.

Існує багато різних типів БД, які зазвичай класифікуються відповідно до базових моделей даних. Наприклад, найпопулярніша модель даних, реляційна модель даних, організовує інформацію у пов'язані двовимірні таблиці, тоді як модель даних графіка організовує інформацію у вузли (точки) і ребра (лінії, що з'єднують ці точки)(див. рис. 2.8).

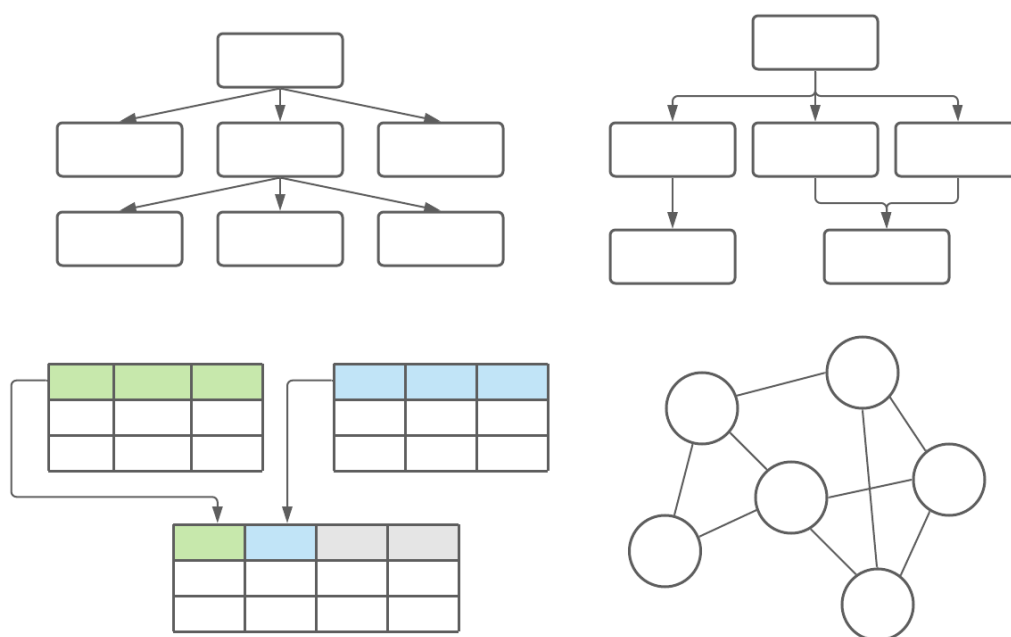


Рисунок 2.8 - Абстрактні представлення моделей даних

Різні моделі даних мають різні переваги та недоліки залежно від того, що треба досягти. Якщо потрібно створити базу даних з високим ступенем цілісності даних, яка може гнучко масштабуватися, реляційна БД, ймовірно, працюватиме найкраще; якщо потрібна база даних, яка може обробляти велику кількість даних, графічна база даних може бути хорошим варіантом [8].

Незалежно від того, з якими типами даних потрібно працювати, вони повинні бути організовані відповідно до схеми БД. Можна розглядати схему як план бази даних, що описує деталі реалізації, наприклад, необхідні типи даних або інші обмеження.

Схема БД (див. рис. 2.9) – це те, що відрізняє її від списку чи електронної таблиці: зі схемою можна бути впевненим, що дані у базі даних будуть організовані відповідно до певного набору правил, тому що так і повинно бути.

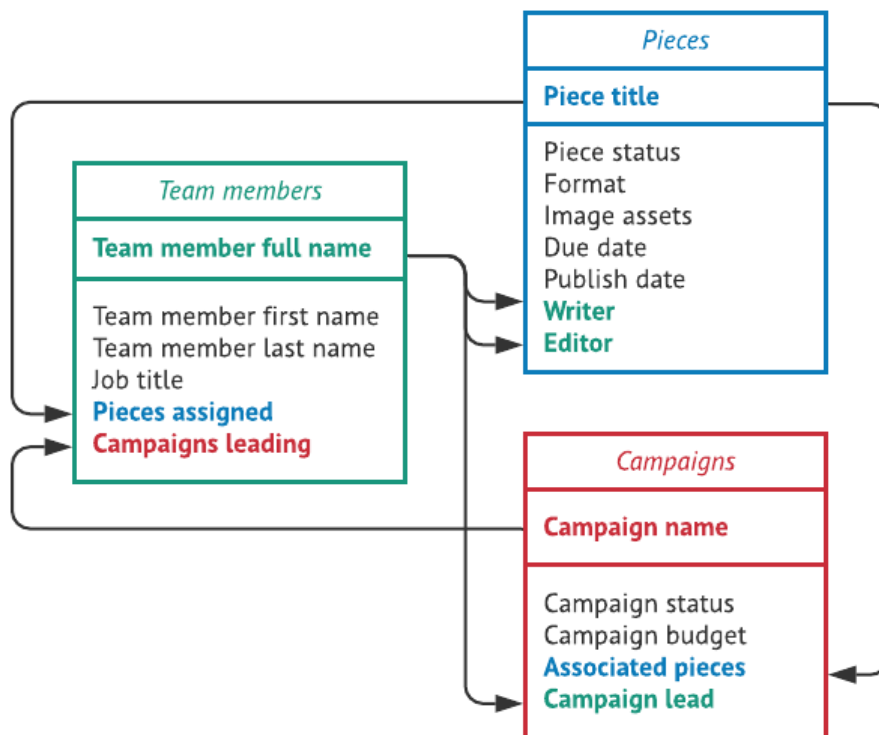


Рисунок 2.9 – Приклад схеми БД

Після організації потрібен спосіб взаємодії з БД для виконання бажаних дій. СУБД — це програмне забезпечення, яке дає можливість кінцевим користувачам створювати, змінювати та керувати базами даних, а також визначати, зберігати, маніпулювати та отримувати дані всередині цих баз даних. Деякі приклади СУБД включають MySQL, MongoDB, Oracle, Microsoft Azure та SQLite.

MySQL — це система управління реляційною базою даних (RDBMS), названа на честь дочки засновника Майкла Віденіуса як «Му», а SQL означає «Structured Query Language». Перша версія цієї бази даних була випущена в 1995 році, а 2010 року її придбала Oracle [11].



Рисунок 2.10 – MySQL

Переваги MySQL:

- проста у використанні;
- безкоштовна;
- захищена;
- краща продуктивність;
- масштабованість;
- доступність для різних платформ;
- дружній інтерфейс.

MongoDB — це документно-орієнтована база даних NoSQL, яка використовується для зберігання великих обсягів даних. MongoDB — це БД яка з'явилася приблизно в середині 2000-х років. Вона відноситься до категорії баз даних NoSQL.



Рисунок 2.11 - MongoDB

Цей тип СУБД використовує динамічні схеми, які означають, можна створювати записи без попереднього визначення структури, наприклад полів або типів та їх значень.

MongoDB дозволяє змінювати структуру записів, називають документами, додаючи нові поля або видаляючи наявні.

Переваги MongoDB:

- дуже гнучка;
- проста система СУБД;
- висока швидкість зберігання робочих тимчасових наборів даних;
- підтримує реплікацію БД;
- можна використовувати як систему зберігання файлів;
- в основному використовує об'єкти JavaScript замість процедури;
- використовує динамічну схему бази даних JSON;
- може працювати на кількох серверах.

Oracle Database — це конвергентна база даних, оскільки вона має єдиний інтегрований сервер і підтримує кілька моделей даних. Вона може розміщуватися на приватній, загальнодоступній або гібридній платформі хмарних обчислень як хмарна база даних, яку клієнти можуть розміщувати, не купуючи спеціальне обладнання [12].



Рисунок 2.12 – Oracle database

База даних Oracle забезпечує повну підтримку для запуску в ній процедурної мови PL/SQL. Крім того, вона поставляється з APEX, платформою розробки з низьким кодом, тому вам може навіть не знадобитися встановлювати та керувати окремим сервером додатків для виконання бізнес-логіки та аналізу даних.

Переваги Oracle database:

- висока продуктивність;
- підтримка керування кількома БД на одному сервері;
- різноманітність ПЗ для різних потреб;
- використання кластерів;
- диспетчер відновлення після збою;
- підтримує розширення PL/SQL.

SQL Azure — це платформа хмарних обчислень та онлайн-портал, що дозволяє отримувати доступ до хмарних служб і ресурсів, наданих Microsoft, і керувати ними. Ці послуги та ресурси включають зберігання даних та їх перетворення залежно від потреб. Щоб отримати доступ до цих ресурсів і служб, все, що потрібно, це активне інтернет-з'єднання та можливість під'єднання до порталу Azure.



Рисунок 2.13 – SQL Azure

Однією з головних причин використання бази даних Azure SQL є те, що вона забезпечує дуже високу доступність. Однією з головних переваг цього рішення полягає в масштабованості, можна легко збільшити або зменшити БД відповідно до потреб бізнесу. Завдяки порталу Azure інструмент можна легко контролювати та створювати звіти.

Переваги SQL Azure:

- інфраструктура як послуга (IaaS);
- потужна підтримка для аналітики;
- розширяє наявну IT-підтримку;

- унікальна система зберігання;
- покращена масштабованість;
- підвищена гнучкість.

SQLite — програмне забезпечення, яке дозволяє користувачам взаємодіяти з реляційною базою даних. У SQLite база даних зберігається в одному файлі — особливість, яка відрізняє її від інших механізмів баз даних. Цей факт забезпечує велику доступність: копіювання бази даних не складніше, ніж копіювання файлу, в якому зберігаються дані, спільний доступ до бази даних може означати надсилання вкладення електронної пошти.



Рисунок 2.14 – SQLite

Переносність підпису SQLite, на жаль, робить його поганим вибором, коли багато різних користувачів оновлюють таблицю одночасно (щоб підтримувати цілісність даних, лише один користувач може одночасно записувати у файл). Це також може вимагати додаткової роботи для забезпечення безпеки приватних даних завдяки тим самим функціям, які роблять SQLite доступним.

Переваги SQLite:

- відкритий вихідним кодом;
- не має сервера;
- полегшує роботу з кількома базами даних в одному сеансі одночасно;
- це кросплатформна СУБД;
- не вимагає ніякої конфігурації.

Однак, не зважаючи на переваги і недоліки вище описаних БД, в ДР було використано SQLite, враховуючи простоту використання та можливість змінити у подальшому на іншу БД за допомогою EFC.

Висновки до розділу 2

Цей розділ описує функціональність програмного забезпечення для підготовки веб-додатку та детально аналізує кожну функцію, оскільки вона стосується завдання.

Для реалізації сайту було обрано написати серверну частину на мові C# використовуючи ASP.Net Core, за допомогою середовища розробки Visual Studio 2022.

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Архитектура проекту

Даний проект складеться із 3х частин:

- сервер (API сервіс);
- клієнт;
- БД.

Додаток спроектований за MVC шаблоном. MVC — це шаблон проектування, який використовується для відокремлення інтерфейсу користувача (представлення), даних (модель) і логіки програми (контролер). Цей шаблон допомагає досягти розділення системи.

Використовуючи шаблон MVC для веб-сайтів, запити направляються до контролера, який відповідає за роботу з моделлю для виконання дій та/або отримання даних. Контролер вибирає вид для відображення та надає йому модель. Подання відтворює кінцеву сторінку на основі даних у моделі.

Контролер приймає усі запити від клієнтів, обробляє їх, та працює з БД, яка приймає та відправляє SQL запити від або до сервера використовуючи модель [9].

Данну залежність можна побачити на зображенні (див. рис. 3.1).



Рисунок 3.1 – Приклад шаблону MVC.

Проект поділений на такі частини:

- контролери;
- моделі;
- представлення;
- представлення моделей;
- доменні об'єкти;
- класи для роботи з даними;
- файли конфігурацій.

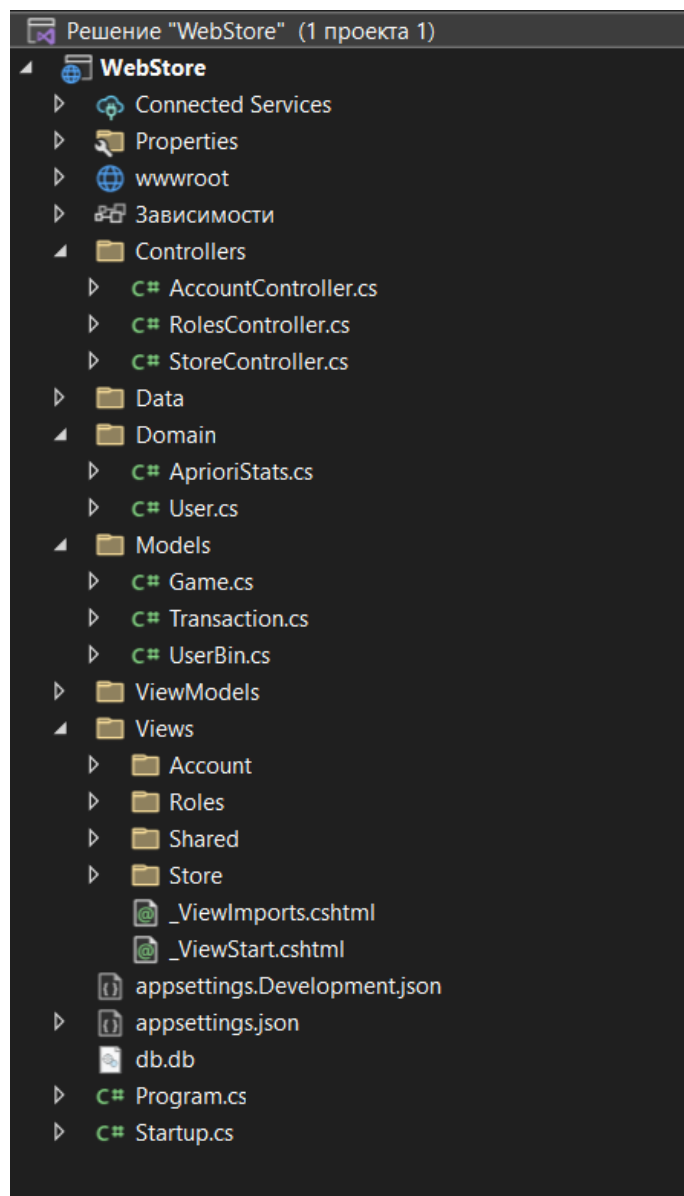


Рисунок 3.2 – Приклад структури проекту.

3.2 Принцип роботи сайту

Для роботи сайту потрібно підключення до БД. Принцип підключення виглядає наступним чином (рис 3.3).

```
1  {  
2  "ConnectionStrings": {  
3    "DefaultConnection": "Data Source=.\db.db",  
4    "StoreConnection": "Data Source=.\db.db"  
5  },  
}
```

Рисунок 3.3 – Рядок підключення до локальної БД SQLite.

Після налаштування підключення до БД треба створити декілька контекстів. Перший контекст буде оперувати даними користувачів (див. рис. 3.4).

```
Ссылка: 10  
public sealed class ApplicationContext : IdentityDbContext<User>  
{  
    Ссылка: 0  
    public ApplicationContext(DbContextOptions<ApplicationContext> options)  
        : base(options)  
    {  
        Database.EnsureCreated();  
    }  
}
```

Рисунок 3.4 – Контекст для роботи з даними користувачів.

Другий контекст Оперує даними інтернет-магазину, такими як: ігри та транзакції (див. рис. 3.5).

```
Ссылк: 7
public sealed class StoreContextDb : DbContext
{
    Ссылк: 9
    public DbSet<Game> Games { get; set; }
    Ссылк: 5
    public DbSet<Transaction> Transactions { get; set; }

    Ссылк: 0
    public StoreContextDb(DbContextOptions<StoreContextDb> options)
        : base(options)
    {
        Database.EnsureCreated();
    }
}
```

Рисунок 3.5 – Контекст для роботи з даними інтернет-магазину.

Важною частиною коректної роботи БД є створення міграції, щоб наповнити її таблицями. Це можна зробити за допомогою інструменту EntityFrameworkCore.Design. Використовуючи командний рядок, вказавши усі необхідні дані, після виконання команди з'являться такі файли (див. рис. 3.6).

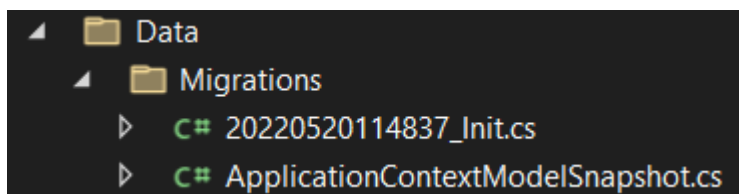


Рисунок 3.6 – Налаштування серверу при початку роботи.

Для заповнення таблиць можна створити клас з даними, який спрацює лише коли таблиці пусті (див. рис. 3.7). Цей спосіб можна використовувати під час розробки сайту, щоб були дані з якими можна проводити маніпуляції.

```
Ссылка 1
public static class Seed
{
    Ссылка 1
    public static void Initialize(StoreContextDb context)
    {
        if (context.Games.Any()) return;

        context.AddRange(
            new Game
            {
                Name = "Resident Evil 3",
                Publisher = "CAPCOM Co., Ltd.",
                Developer = "CAPCOM Co., Ltd.",
                Description = "Головна героїня, Джил, знову зустрітяться в бою з ворогами.",
                Price = 620,
                Genre = "Екшени",
                Image = "\\Images\\Games\\resident-evil-3.jpg"
            },
            new Game{...},
            new Game{...},
            new Game{...},
            new Game{...}
        );
        context.SaveChanges();
    }
}
```

Рисунок 3.7 – Клас для наповнення тимчасовими даними.

Створивши міграцію, та запустивши проект, EFC відправить запит до БД на створення нових таблиць. Що можна побачити у SQLite Studio.

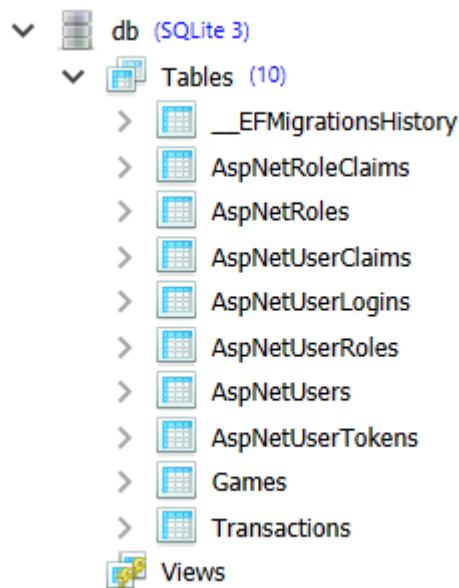


Рисунок 3.8 – Створені таблиці у БД.

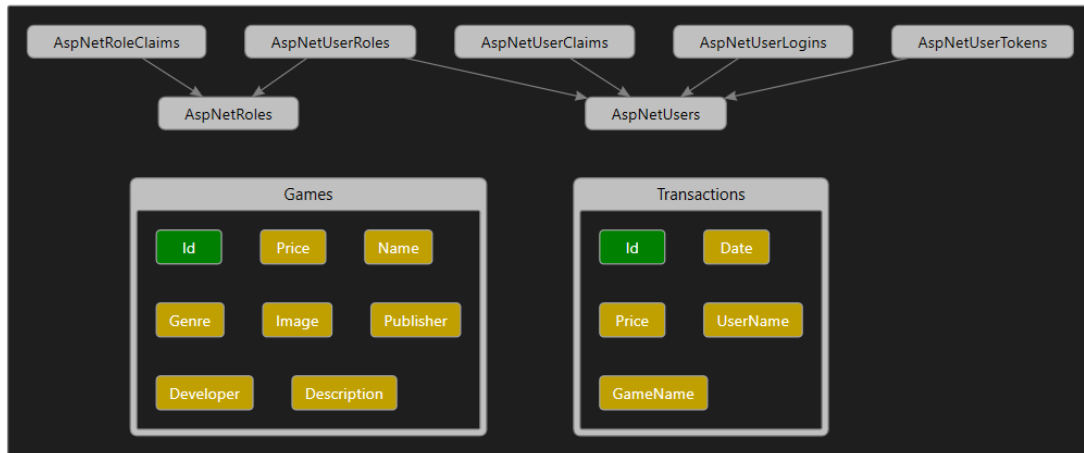


Рисунок 3.9 – Діаграма таблиць Games та Transactions

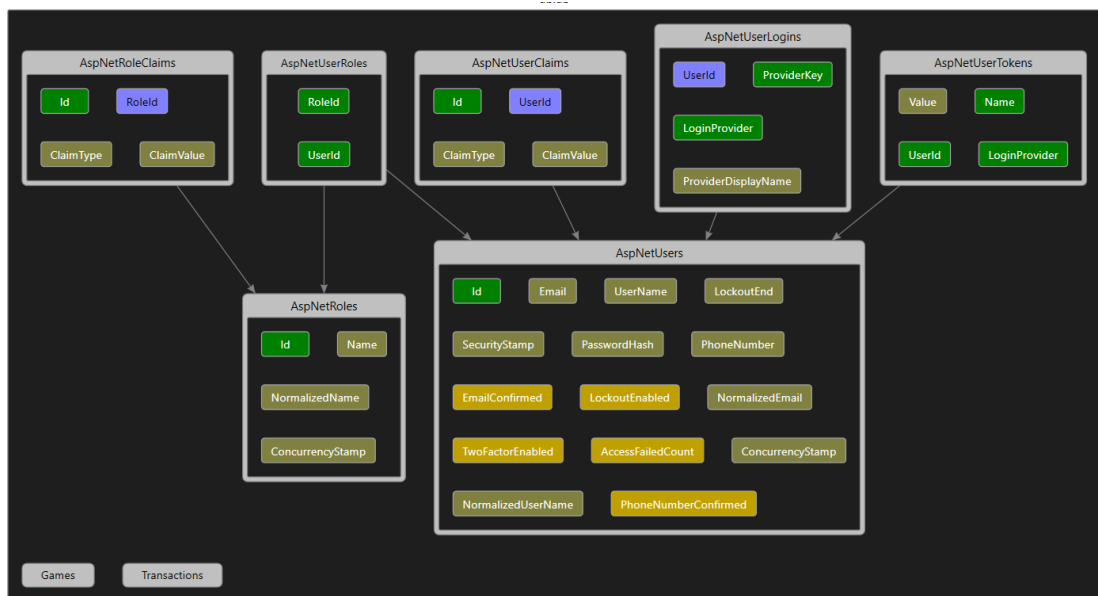


Рисунок 3.10 – Діаграма таблиць IdentityUser

Після створення, налаштування та наповнення БД потрібно створити контролери, які будуть приймати запити від клієнту та оброблювати їх. Для наших потреб можна створити 3 контролери, а саме:

- контролер профілів;
- контролер ролей;
- контролер інтернет-магазину.

Контролер профілів (див. рис. 3.11) буде працювати з такими процесами

ЯК:

2022 р.

Степанюк В.О.

122 – БКР – 402.21810225

- логін старих клієнтів;
- реєстрація нових клієнтів;
- вихід із профілю;
- зміна даних профілю;
- зміна пароллю;
- показ даних профілю.

```
Ссылка: 1
public interface IAccountController
{
    Ссылка: 1
    Task<IActionResult> ChangePassword();
    Ссылка: 1
    Task<IActionResult> ChangePassword(ChangePasswordViewModel model);
    Ссылка: 1
    Task<IActionResult> EditAccount();
    Ссылка: 1
    Task<IActionResult> EditAccount(EditUserViewModel model);
    Ссылка: 1
    IActionResult Login(string returnUrl = null);
    Ссылка: 1
    Task<IActionResult> Login(LoginViewModel model);
    Ссылка: 2
    Task<IActionResult> Logout();
    Ссылка: 1
    IActionResult Profile();
    Ссылка: 1
    IActionResult Register();
    Ссылка: 1
    Task<IActionResult> Register(RegisterViewModel model);
}
```

Рисунок 3.11 – Інтерфейс контролеру профілів.

Контролер ролей буде оброблювати запити адміністратора для створення, зміни, або додавання ролей користувачів. Так як деякі функції сайту будуть доступні лише при певній ролі (див. рис. 3.12).

```
Ссылка: 1
public interface IRolesController
{
    Ссылка: 1
    IActionResult Create();
    Ссылка: 1
    Task<IActionResult> Create(string name);
    Ссылка: 1
    Task<IActionResult> Delete(string id);
    Ссылка: 1
    Task<IActionResult> Edit(string userId);
    Ссылка: 1
    Task<IActionResult> Edit(string userId, List<string> roles);
    Ссылка: 1
    IActionResult Index();
    Ссылка: 1
    IActionResult UserList();
}
```

Рисунок 3.12 – Інтерфейс контролеру ролей.

На останок Контролер інтернет-магазину буде працювати з даними товарів, та транзакцій. Він надає сайту список товарів та можливість додавати товари до кошика користувача, видаляти їх або проводити операцію купівлі. Однією з функцій є показ транзакцій купівлі користувачу.

```
Ссылка: 1
public interface IStoreController
{
    Ссылка: 1
    Task<IActionResult> AddDevice(Game model);
    Ссылка: 1
    IActionResult AddToBin(int id, bool redirect = true);
    Ссылка: 1
    IActionResult Buy();
    Ссылка: 1
    IActionResult DeleteFromBin(int id);
    Ссылка: 1
    IActionResult Error();
    Ссылка: 1
    IActionResult Game(int id);
    Ссылка: 1
    IActionResult Index();
    Ссылка: 1
    IActionResult Privacy();
    Ссылка: 1
    IActionResult SearchItem();
    Ссылка: 1
    IActionResult SearchItem(string searchQuery);
    Ссылка: 1
    IActionResult ShowTransactions();
    Ссылка: 1
    IActionResult UserBin();
}
```

Рисунок 3.13 – Інтерфейс контролеру інтернет-магазину.

3.3 Принцип роботи відображення сайту

Після створення БД та усіх контролерів, треба створити відображення. Відображення це html код з використанням C#. На даних сторінках можна написати структуру сторінки з використанням класів Bootstrap та C# коду, який потім скомпілюється в html код.

Для кожного контролеру треба створити свою папку з потрібними нам відображеннями (див. рис. 3.14).

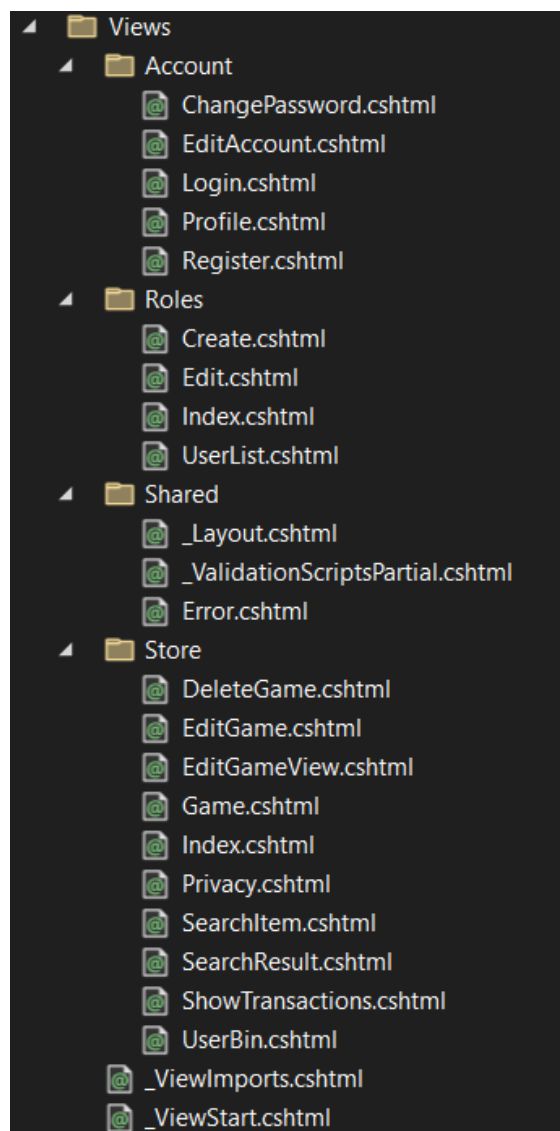


Рисунок 3.14 – Структура відображень проекту.

Нам потрібна також папка Shared для коду, який має файл Layout.cshtml. Це буде нашим основним файлом представлення, який матиме

незмінні частину сайту, такі як: шапка та підвал. Інші відображення будуть змінюватися у тілі сайту відповідно до посилання сторінки. Такий ефект можна отримати за допомогою методу `@RenderBody()` (див. рис. 3.15).

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Не Розетка</title>
5      <meta charset="utf-8">
6      <meta name="keywords" content="">
7      <meta name="description" content="">
8      <link rel="stylesheet" href="~/css/index.css">
9      <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
10     <link rel="stylesheet" href="~/css/site.css" />
11 </head>
12 <body>
13     <header>
14         ...
15     </header>
16
17     <main style="margin-bottom: 80px;">
18         @RenderBody()
19     </main>
20
21     <footer class="border-top footer text-muted bg-dark">
22         ...
23     </footer>
24
25     <script src="~/lib/jquery/dist/jquery.min.js"></script>
26     <script src="~/lib/bootstrap/dist/js/bootstrap.min.js"></script>
27     <script src="~/js/site.js" asp-append-version="true"></script>
28     @RenderSection("Scripts", required: false)
29 </body>
30 </html>

```

Рисунок 3.15 – Код незмінного відображення сайту.

Щоб написати відображення інших сторінок сайту нам вже не потрібно писати всю структуру html, а лише ті частини які потім будуть вставленні в головному відображенні замість методу `@RenderBody()`. Так для написання домашньої сторінки сайту, на якій будуть відображатися усі ігри для покупки, треба написати лише таку частину коду (рис 3.16). Як можна побачити дане відображення включає в себе html та C# код. А також специфічні атрибути для ASP.NET MVC Framework.


```

1  @model List<WebStore.Models.Game>;
2
3  <div class="container w-50">
4      @if
5      {
6          foreach (var game in Model)
7          {
8              var description = 300 < game.Description.Length
9              ? game.Description.Substring(0, 300) + "..."
10             : game.Description;
11
12             <div class="card my-4">
13                 <div class="row no-gutters">
14                     <div class="col-md-4">
15                         
16                     </div>
17                     <div class="col-md-8">
18                         <div class="card-body">
19                             <h5 class="card-title">@game.Name</h5>
20                             <p class="card-text">@description</p>
21                             <p class="card-text"><small class="text-muted">@game.Genre</small></p>
22                             <a asp-area="" asp-controller="Store" asp-action="Game" asp-route-id="@game.Id"
23                                 class="btn btn-outline-secondary">
24                                 Дивитись повністю
25                             </a>
26                             <form method="post" style="display: inline;" asp-controller="Store" asp-action="AddToBin"
27                                 asp-route-id="@game.Id">
28                                 <input type="submit" class="btn btn-success" value="До кошика @game.Price €"/>
29                             </form>
30                         </div>
31                     </div>
32                 </div>
33             </div>
34         }
35     </div>
    
```

Рисунок 3.16 – Код відображення домашньої сторінки сайту.

Зробивши ці етапи ми отримаємо такий результат (див. рис. 3.17).

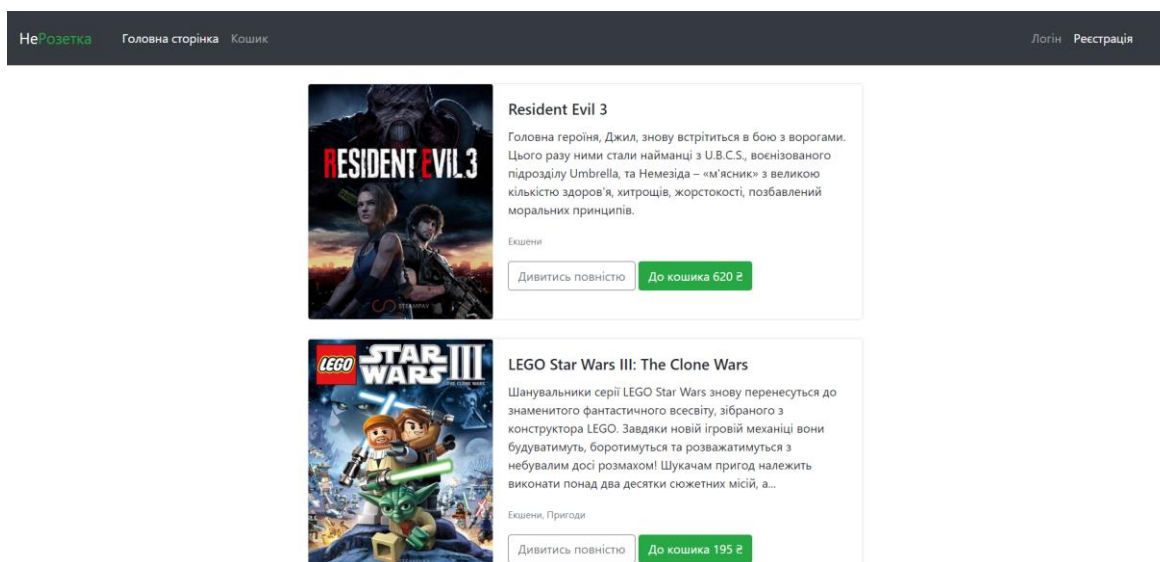


Рисунок 3.17 – Вигляд домашньої сторінки.

Для купівлі ігор клієнт може натиснути на кнопку «Додати до кошика», або переглянути повну інформацію про гру натиснувши на «Дивитись повністю» (див. рис. 3.18).

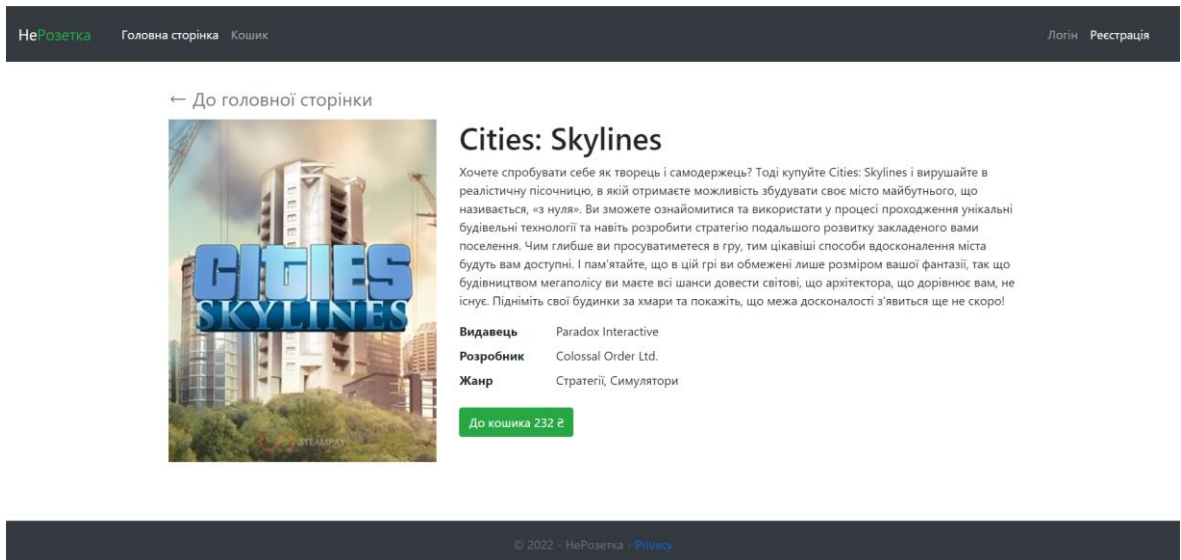


Рисунок 3.18 – Вигляд сторінки з грою.

Коли клієнт вибрав усі ігри які йому потрібні, він може перейти до вкладки з кошиком (див. рис. 3.19). На даній сторінці можна додати запропоновану сайтом гру, яку беруть з обраними іграми, або сплатити всі товари.

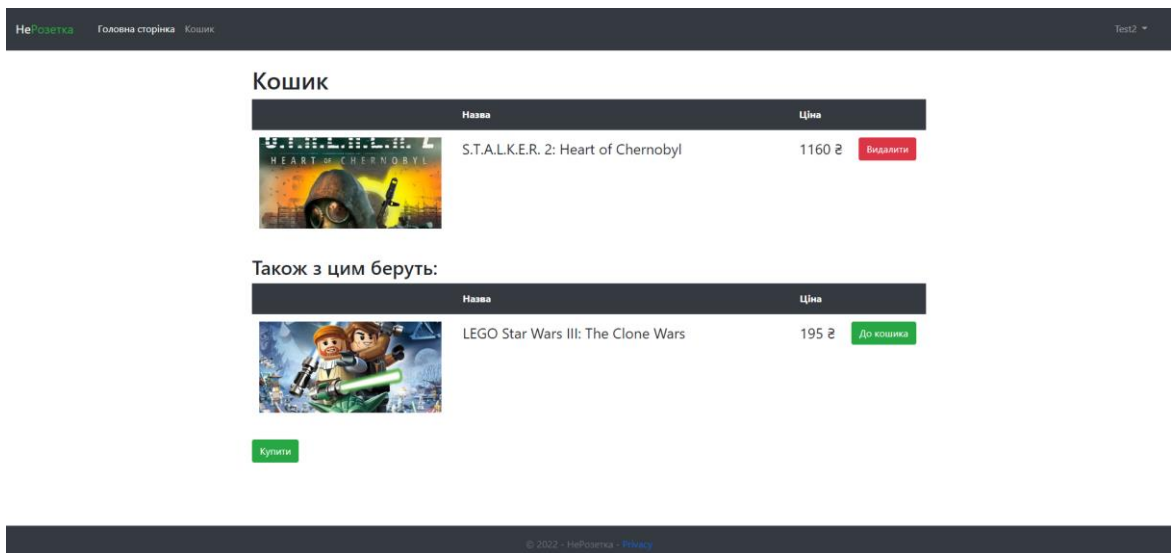


Рисунок 3.19 – Вигляд сторінки з кошиком.

Якщо клієнт не зайшов до свого профілю, або ще його не створив, сайт перекине на сторінку для входу (див. рис. 3.20).

HeРозетка Головна сторінка Кошик Логін Реєстрація

Вхід

Логін

Restorator

Пароль **Обов'язкове поле**

Введіть пароль

Запам'ятати мене

Увійти

© 2022 - HeРозетка - Privacy

Рисунок 3.20 – Сторінка входу до сайту.

При вході до сайту спрацьовує валідація даних та відображення помилок валідації. Якщо клієнт не має свого профілю, то його можна створити на сторінці реєстрації (див. рис. 3.21).

HeРозетка Головна сторінка Кошик Логін Реєстрація

Реєстрація

Почта *

Введіть вашу пошту

Логін *

Введіть логін

Пароль *

Введіть пароль

Підтвердження пароль *

Підтвердіть пароль

Зареєструватися

© 2022 - HeРозетка - Privacy

Рисунок 3.21 – Відображення сторінки реєстрації.

Коли клієнт зробив замовлення, він може передивитися свої транзакції натиснувши на свій логін у шапці сайту, та обравши з випадаючого списку вкладку «Замовлення». На сторінці замовлень відображаються всі транзакції з описом ігор та цін за них (рис 3.22).

Транзакція #1	
20.05.2022 14:57:12	
Resident Evil 3	620€
Doom	373€
Вся ціна	993€

Транзакція #2	
28.05.2022 19:43:13	
Doom	373€
LEGO Star Wars III: The Clone Wars	195€
S.T.A.L.K.E.R. 2: Heart of Chernobyl	1160€
Вся ціна	1728€

Рисунок 3.22 – Вигляд сторінки замовлень.

Усі елементи працюють і на мобільних телефонах (див. рис. 3.23).

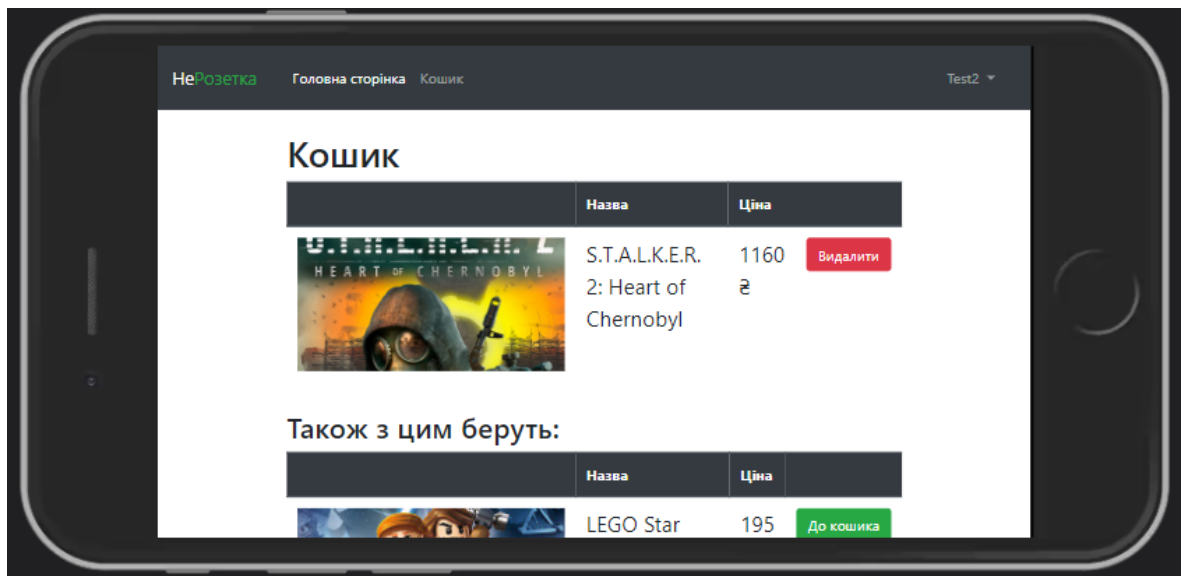


Рисунок 3.23 – Приклад використання сайту на телефоні.

3.4 Аналітична частина сайту

У аналізі даних Apriori є класичним алгоритмом для вивчення правил асоціації. Apriori розроблено для роботи з базами даних, що містять транзакції (наприклад, колекції товарів, куплених клієнтами, або відомості про відвідування веб-сайту) [10].

Інші алгоритми призначені для пошуку правил асоціації в даних, які не мають транзакцій (Winer і Miner) або не мають часових позначок.

Вся суть алгоритму (і аналізу даних загалом) полягає в тому, щоб витягти корисну інформацію з великих обсягів даних. Наприклад, інформація про те, що клієнт, який купує клавіатуру, одночасно купує і мишу, отримується з наведеного нижче правила асоціації:

Підтримка - відсоток релевантних задачі транзакцій даних, для яких шаблон вірний.

$$\text{Підтримка (Клавіатура -> Миша)} = \frac{\text{кількість транзакцій з клавіатурою та мишею}}{\text{кількість транзакцій}}$$

Довіра - міра впевненості або надійності, пов'язана з кожною виявленою закономірністю.

$$\text{Довіра (клавіатура -> Миша)} = \frac{\text{кількість транзакцій з клавіатурою та мишею}}{\text{кількість транзакцій з клавіатурою}}$$

Алгоритм має на меті знайти правила, які задовольняють як мінімальний поріг підтримки, так і мінімальний поріг довіри (суворі правила).

- товар: артикул в кошику;
- набір предметів: група предметів, придбаних разом за одну транзакцію.

Як працює Apriori:

1. знайти всі часті набори елементів:
 - a. отримайте часті предмети;
 - i. елементи, зустріч яких у базі даних більше або дорівнює мініальному порогу підтримки;
 - b. отримайте часті набори предметів:
 - i. генеруйте кандидатів з частих елементів.
 - ii. обріжте результати, щоб знайти найпоширеніші набори елементів.

2. створити жорсткі правила асоціації з частих наборів елементів:

- а. правила, які задовольняють мінімальний поріг підтримки та мінімальний поріг довіри.

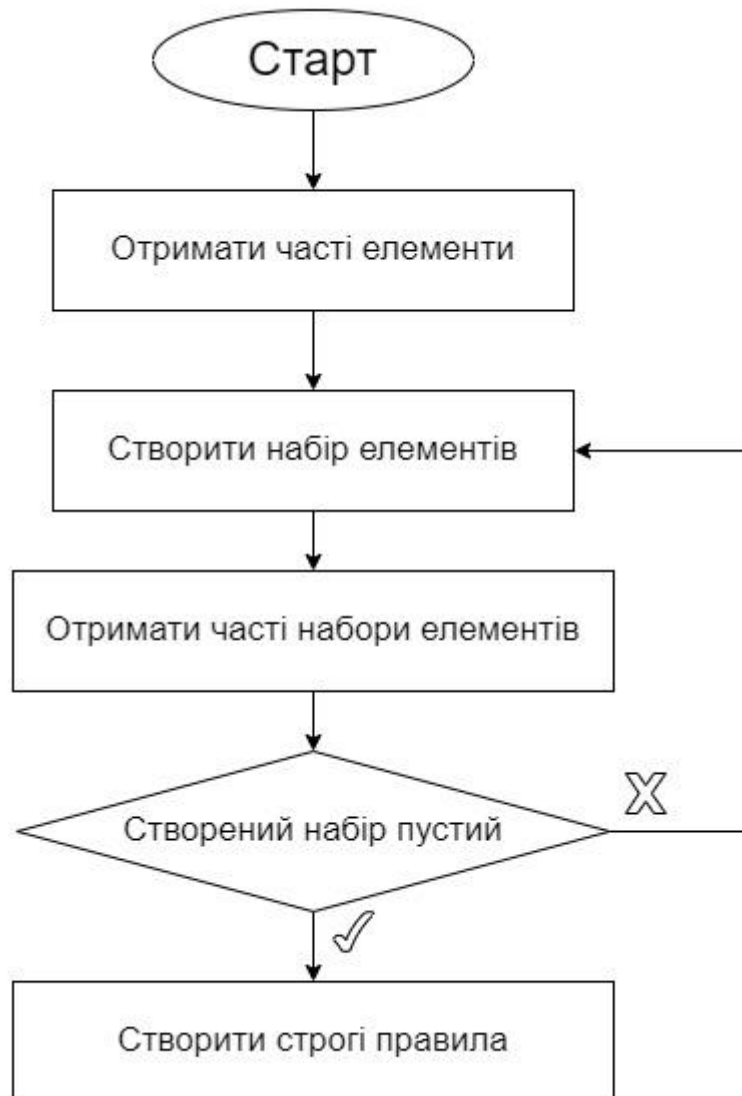


Рисунок 3.24 - Приклад роботи алгоритму Apriori.

Висновки до розділу 3

В даному розділі було надано архітектуру проекту, його реалізацію та описано використання обраних технологій. При написанні сайту були використані БД, система користувачів та аналітичний алгоритм Apriori.

При тестуванні сайту не було виявлено критичних помилок, які б впливали на працездатність та зручність його користуванням.

При розробці інтернет-магазину було виконано всі поставлені задачі і створено фундамент для його вдосконалення для подальших перспектив:

- додавання більш розширеної інформації для товарів;
- долучення маркетингової системи для покращення прибутку;
- оптимізація запитів для швидкодії сайту;
- впровадження нових технологій для тестування;
- додавання нових елементів інтерфейсу за запитами користувачів.

Спеціальний розділ

ОХОРОНА ПРАЦІ

до бакалаврської роботи

на тему:

**«ІНТЕРНЕТ-МАГАЗИН З ПРОДАЖУ КОМП'ЮТЕРНИХ
ІГОР»**

Спеціальність 122 – Комп'ютерні науки

122-БКР-402.21810225

Виконав студент 4 курсу, групи 402

_____ ***В.О. Степанюк***

«21» червня 2022 р

Консультант: ст. викладач

_____ ***Е.В. Макарова***

«21» червня 2022 р

Миколаїв – 2022

4 ОХОРОНА ПРАЦІ

ВСТУП

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини під час трудової діяльності. Чинна система соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, які забезпечують збереження здоров'я і працездатність людини під час праці.

Законодавство про працю містить норми і вимоги з техніки безпеки і виробничої санітарії, норми, що регулюють робочий час і час відпочинку, звільнення та переведення на іншу роботу норми праці щодо жінок, молоді, гігієнічні норми і правила тощо.

Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний – на професійні спілки. Контроль за безпекою праці здійснюють також, державні й відомчі спеціалізовані інспекції.

Охорона здоров'я – один з пріоритетних напрямів державної діяльності та національної політики. Держава формує політику охорони здоров'я та забезпечує її реалізацію, використовуючи світовий досвід роботи з поліпшення умов і безпеки праці.

Метою роботи є створення безпечних і здорових умов праці на робочих місцях, в робочих зонах, у виробничих приміщеннях та забезпечення безпеки людини у надзвичайних ситуаціях.

Відповідно до мети виділені наступні **завдання**:

1. Встановити загальні умови до приміщень з використанням комп'ютерної техніки.
2. Описати вимоги щодо організації робочих місць.
3. Виконати опис комп'ютерної лабораторії та виробничого обладнання.
4. Створити план поведінки в надзвичайних ситуаціях.

4.1 Оцінка потенційних небезпек і шкідливих факторів

У дипломній роботі розглянуті питання розробки інтернет магазину, тому у цьому розділі подано аналіз шкідливих факторів, потенційних небезпек у комп'ютерному класі. Визначено технологічні, організаційні заходи індивідуального захисту для усунення шкідливих факторів [13]. План переміщення комп'ютерної аудиторії (див. рис. 4.1).

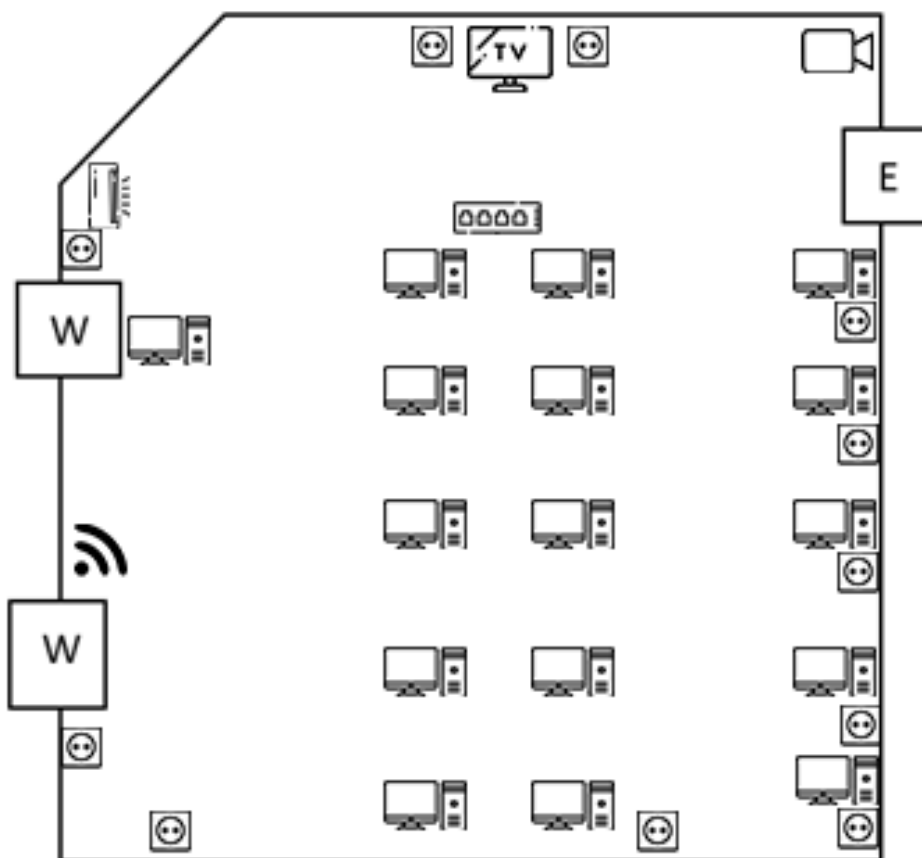


Рисунок 4.1 – план приміщення комп'ютерної аудиторії

Робочі місця офісних працівників, обладнані персональними комп'ютерами, повинні відповідати вимогам «Правил охорони праці під час експлуатації електронно-обчислювальних машин», затверджених Наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 року № 65 (Правила), та «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-

обчислювальних машин», затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 (ДСАНПІН 3.3.2-007-98).

Таблиця 4.1 – Розташування обладнання у комп'ютерному класі

№	Назва обладнання	Марка обладнання	Кількість	Габарити екземпляру	Положення на схемі
1	монітор	View sonic	16	500*600	1
2	стіл	-	16	1700*800	2
3	стілець	-	16	450*450	3
4	клавіатура	x-tech	16	500*200	4
5	телевізор	-	1	1500*2000	5

Визначено небезпечні і шкідливі фактори у комп'ютерному класі [14], що дозволяє визначити небезпеки та розробити заходи по покращенню умов праці.

Таблиця 4.2 – небезпечні шкідливі фактори при роботі в комп'ютерному класі

Вид роботи	Небезпечні шкідливі фактори		
	Фізичні	Хімічні	Психофізичні
Робота за комп'ютером	1. Рівень шуму 2. Порушення мікроклімату 3. Електробезпека 4. Підвищений рівень випромінювання(електромагнітні) 5. Підвищений рівень випромінювання(УФ)	відсутні	1. Недостатня рухливість 2. Пасивність 3. Сповільнене мислення

4.2 Параметри мікроклімату

Під мікрокліматом виробничих приміщень розуміють клімат внутрішнього середовища виробничого приміщення, який визначається поєднаними діями на організм людини, температури, вологості, швидкості руху повітря та теплових випромінювань.

Для того, щоб фізіологічні процеси в організмі людини відбувалися нормально, тепло, що виділяється організмом, повинне повністю відводитися у навколишнє середовище. Порушення теплового балансу може призвести до перегрівання або до переохолодження організму людини і, зрештою, до втрати працездатності, втрати свідомості та до теплової смерті. Величина тепловиділення організмом людини залежить від ступеня фізичного напруження за певних кліматичних умов і складає від 85 (у стані спокою) до 500 Дж/с (важка робота). Нормальне теплове самопочуття має місце, коли тепловиділення ($Q_{\text{тв}}$) організму людини повністю сприймаються навколишнім середовищем ($Q_{\text{тн}}$), тобто коли має місце тепловий баланс ($Q_{\text{тв}} = Q_{\text{тн}}$). У цьому випадку температура внутрішніх органів залишається постійною на рівні 36,6°C.

На сьогодні основним нормативним документом, що визначає параметри мікроклімату в комп'ютерному класі є санітарні норми ДСН 3.3.6.042-99. (таб.4.3).

Таблиця 4.3 - джерела впливу на мікроклімат у комп'ютерному класі

№	Джерело зміни показників мікроклімату	Наслідок
1	Висока температура зовні	Підвищення температури в приміщенні
2	Низька температура зовні	Зниження температури в приміщенні
3	протяги	Захворювання студентів

У таблиці 4.4 розроблено заходи для нормалізації мікроклімату в комп'ютерному класі.

Таблиця 4.4 – заходи нормалізації мікроклімату в класі

№	Заходи		Реалізація
1	технічні	Т.П.Р.	Провітрювання, прибирання
		Х.П.Р.	опалення
2	організаційні		Вимірювання показників мікроклімату
3	ЗІЗ		відсутні

4.3 Природне і штучне освітлення

Природне освітлення в будівлях мусить відповідати будівельним нормам і правилам (ДБН В2.5-28-2006 «Природне і штучне освітлення») [14]. У комп'ютерному класі є природного і штучного освітлення. Засоби небезпек через природне і штучне освітлення відображені в таблиці 4.5

Таблиця 4.5 – джерела небезпеки пов'язані від освітлення

№	Джерело небезпеки	Наслідок
1	Недостатнє освітлення	Перенапруження і порушення зору студентів, зниження працездатності
2	Підвищена яскравість світла	
3	Аварійна робота джерел штучного світла	

У таблиці 4.6 наведено заходи для нормалізації загального штучного освітлення

Таблиця 4.6 – заходи попередження порушення освітлення

№	Заходи	Реалізація
1	Технічні при недостатній освітленості	Додаткові джерела світла
2	Технічні при перевищенні освітленості	Лампи ЛДС зниженої потужності
3	організаційні	Вимірювання яскравості світла
4	ЗІЗ	–

4.4 Рівень шуму

Рівень шуму регулюється санітарними нормами виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99 . Шум є одним з факторів, що впливає на роботу студентів. Засоби регулювання шуму відображені в таблиці 4.7

Таблиця 4.7 – джерела шуму

№	Джерело шуму	Наслідок
1	Кулер комп'ютера	Емоційна напруженість, знижена працездатність
2	Лампи ЛДС	

В таблиці 4.8 описано заходи для зниження небезпек пов'язаних з шумом.

Таблиця 4.8 – заходи для зниження небезпек пов'язаних з шумом

№	Заходи	Реалізація
1	Технічні	Вчасна заміна ламп ЛДС, обстеження комп'ютера
2	організаційні	Раціональне планування роботи
3	ЗІЗ	Навушники AGK

4.5 Електромагнітне випромінювання

В Україні норми електромагнітної безпеки регламентуються «Державними санітарними нормами і правилами захисту населення від впливу електромагнітного випромінювання», які затверджені МОЗ України (№ 239 від 01.08.1996) [17]. Наведено джерела небезпечного випромінювання і наслідки для працівника при тривалому впливі (таб.4.9)

Таблиця 4.9 – джерела небезпечного випромінювання і наслідки

№	Джерело небезпеки	Наслідок
1	Електромагнітне випромінювання комп'ютера	Головні болі, підвищення пульсу
2	УФ-випромінювання монітор комп'ютера	Негативний вплив на сітківку ока

Способи уникнення небезпек електромагнітного випромінювання наведено в таблиці 4.10

Таблиця 4.10 – способи уникнення небезпек електромагнітного

№	Заходи	Реалізація
1	технічні	Екранування, знаходження на відстані від джерела випромінювання
2	організаційні	Обмежена робота з джерелами ЕМВ і УФ
3	ЗІЗ	Захисні окуляри Plexiglas ax

Роботодавці зобов'язані організувати проведення вимірювань потужності ЕМВ на робочих місцях. Загальні вимоги до проведення вимірювань і гігієнічної оцінки результатів досліджень передбачені ДСанПіНЗ.3.6.096-2002.

Електромагнітні поля частотою 1 кГц — 300 МГц нормуються за інтенсивністю та енергетичними навантаженнями електричних і магнітних полів, зважаючи на час впливу. Одиницею напруженості електричного поля є В/м, магнітного поля — А/м; енергетичне навантаження — це добуток квадрата

потужності ЕМП і часу його впливу; виражається: $(В/м) \cdot год$ — для електричного поля та $(А/м) \cdot год$ — для магнітного поля.

Електромагнітне поле в діапазоні частот 300 МГц — 300 ГГц нормується за інтенсивністю та енергетичним навантаженням щільності потоку енергії (далі — ЩПЕ). Одиницею вимірювання ЩПЕ є $Вт/м^2$ ($мВт/см^2$, $мкВт/см^2$). Енергетичне навантаження — це добуток ЩПЕ випромінювання, що падає, і часу його впливу протягом робочої зміни в годинах (год); виражається у $Вт \cdot год/м^2$ ($мВт \cdot год/см^2$, $мкВт \cdot год/см^2$).

Рівні постійних магнітних полів протягом робочого дня не повинні перевищувати 8 кА/м. Пунктом 4.5.3 ДСанПіН 3.3.6.096-2002 установлені значення ГДР щільності потоку енергії залежно від тривалості дії ЕМВ [16].

Вимірювання ЕМВ здійснює акредитована лабораторія, яка пройшла акредитацію у відповідній галузі та має засоби вимірювальної техніки. Такі засоби повинні мати свідоцтва про калібрування від ДП «Укрметртестстандарт». Методи вимірювання ЕМВ наведені у спеціалізованій технічній документації. Інструментальні виміри ЕМВ здійснюються на висоті 0,5, 1,0 і 1,7 м від поверхні землі, підлоги приміщення, де розташоване обладнання. До того ж, вимірювання параметрів електричного і магнітного поля проводять на відстані 0,5 м від обладнання, що є джерелом ЕМВ. Як вимірювач напруженості ЕМВ використовують високоточну техніку ненаправленого приймання, оснащену ізотропними (трикоординатними) датчиками з допустимою відносною похибкою $\pm 20\%$.

Для вимірювання параметрів електричного й магнітного полів застосовують прилад «В&Е – метр» типу А002м (див. рис. 4.2).

Прилад призначений для проведення експрес – вимірювань середньоквадратичного значення осциляції електричної й магнітної складової електромагнітного поля в житлових і робочих приміщеннях. Прилад може бути використаним для проведення санітарно – гігієнічного обстеження приміщень з електрообладнанням (комп'ютери, ігрові автомати), а також для

аналізу електромагнітного фону джерел інтенсивного електромагнітного випромінювання.



Рисунок 4.2 – Загальний вигляд приладу “В&Е – метр”

4.6 Небезпека враження людини електрострумом

Для безпечної роботи з електрообладнанням розроблені спеціальні норми. Норми викладені в «Правилах експлуатації електроустановок споживачів» і «Правилах безпечної експлуатації електроустановок споживачів» [14].

Джерела електронебезпеки комп'ютерного класу відображено в таблиці 4.11

Таблиця 4.11 – способи уникнення небезпек електромагнітного випромінювання

№	Джерело електронебезпеки	Наслідки
1	Відсутність ізоляції	Ураження струмом, пошкодження шкіри, внутрішніх органів, електротравми
2	Несправність електрообладнання	
3	Відсутність у обладнання захисних стінок	

Засоби уникнення враження людини електрострумом представлені в таблиці

Таблиця 4.12

Таблиця 4.12 – Засоби уникнення враження людини електрострумом

№	Заходи	Реалізація
1	Технічні	Автомати, УЗО, своєчасна ліквідація пошкоджень електрообладнання
2	Організаційні	Інструктажі з правил безпеки , заземлення, занулення, профілактика нещасних випадків
3	ЗІЗ	Електрозахисті рукавиці і взуття

4.7 Пожежна безпека

Забезпечення пожежної безпеки є складовою виробничої та іншої діяльності посадових осіб і працівників підприємства, отже вся відповідальність за забезпечення пожежної безпеки на підприємстві покладається на його власника та керівника (п. 3 статті 55 Кодексу цивільного захисту України).

Керівник підприємства, розробляючи організаційні заходи щодо забезпечення пожежної безпеки відповідно до Правил пожежної безпеки в Україні, призначає відповідальних осіб за пожежну безпеку окремих територій, ділянок, зон, об'єктів, споруд, будівель та їх частин, технологічного та інженерного устаткування, а також за утримання й експлуатацію засобів протипожежного захисту.

Джерела пожежної безпеки комп'ютерного класу відображено в таблиці 4.13

Таблиця 4.13 – Джерела пожежної безпеки

№	Джерело безпеки	Небезпечний фактор	Наслідок
1	Несправність електропроводки	Коротке замикання	Виникнення пожежі,

2	Перевантаження проводки	Оплавлення ізоляції	травмування людей, погіршення або втрата здоров'я, знищення цінного майна
3	Недотримання заходів пожежної безпеки	загорання	
4	Легкозаймисті речовини	загорання	

В таблиці 4.14 перелічено засоби попередження пожежі

Таблиця 4.14 – засоби попередження пожежі

Засоби регулювання пожежної безпеки	
Технічні засоби	суворе дотримання правил і норм, визначених чинними нормативними документами для приміщень, будівель та об'єктів, технічному оснащенні виробництва, експлуатації чи можливому переобладнанні електромереж, опалення, вентиляції, освітлення
Організаційні засоби	організацію пожежної охорони на об'єкті, проведення навчань з питань пожежної безпеки (включаючи інструктажі та пожежно-технічні мінімуми), застосування наочних засобів протипожежної пропаганди агітації, організацією ДПД та ПТК, проведення перевірок, оглядів стану пожежної безпеки приміщень
Засоби індивідуального захисту	Порошковий вогнегасник

Висновки до розділу

В цьому розділі розглянуто важливість забезпечення вимог охорони праці в приміщенні комп'ютерної лабораторії вищого навчального закладу.

Перевірено забезпечення вимог охорони праці, а саме санітарні нормативи освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступеня і сили вібрації, звукового шуму і вогнестійкості приміщення, а також характеристики електромагнітного випромінювання.

Отже охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини під час трудової діяльності.

Виявлено, що оцінка умов праці на робочому місці відноситься до IV категорії, коли спостерігається робота у несприятливих умовах праці.

Також було виявлено наслідки якщо знехтувати всіма правилами наприклад якщо знехтувати правилами освітлення в кімнаті або в нашому випадку в аудиторії буде порушення зору. Якщо знехтувати правилами пожежної безпеки, в наслідок може виникнути пожежа, може бути травмування людей, погіршення або втрата здоров'я, а також може бути знищення цінного майна.

Наприклад дорогі сервери. Най головніше дотримуватися всіма правилами охорони праці та охорони здоров'я щоб в майбутньому не було проблем

Охорона здоров'я - один з пріоритетних напрямів державної діяльності та національної політики. Держава формує політику охорони здоров'я та забезпечує її реалізацію, використовуючи світовий досвід роботи з поліпшення умов і безпеки праці.

Під час виконання спеціальної частини з охорони праці було виконано усі завдання та мету роботи.

ВИСНОВКИ

Під час написання наукової роботи, враховуючи специфіку дизайну, за допомогою потужних і сучасних інструментів і методик було створено веб-сайт із використанням та аналізу бази даних. Підсумком дипломної роботи є дослідження, присвячене розробці веб-сайту, який може містити корисну інформацію. Вивчено та проаналізовано основні принципи створення інтернет-магазинів Досліджено, від яких авторських принципів залежить успіх сайту (це тематичний, структурований і динамічний)

Підсумовуючи у першому розділі була розглянута актуальність створення інтернет-магазину та огляд його аналогів.

Крім теоретичної частини, в одному з розділів надається повний функціонал програмного забезпечення для підготовки інтернет-магазину, з детальним розбором кожної функції для виконання поставленої задачі.

Щоб реалізувати інтернет-магазин було обрано написати серверну частину на мові C# використовуючи ASP.Net Core, а клієнтську частину за допомогою ASP.Net Core MVC. Для розробки сайту та роботи з БД було обрано Visual Studio 2019 та SQLite Studio.

Працюючи над розділом третім дипломної роботи, враховуючи сучасні тенденції розвитку технологій та функціонування веб-сайтів було віднесено наступні критерії:

- виконання деяких функцій на стороні користувача;
- захист веб-сайту від копіювання та розповсюдження контенту;
- написання веб-сайту за правилами SEO, для кращого результату в пошукових системах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мікросервісна архітектура. Medium: вебсайт. URL: <https://medium.com/@IvanZmerzlyi/microservices-architecture-461687045b3d> (дата звернення: 10.06.2022).
2. C# Programming Language. Geekforgeeks: вебсайт. URL: <https://www.geeksforgeeks.org/csharp-programming-language/> (дата звернення: 10.06.2022).
3. ASP.NET documentation. Microsoft: вебсайт. URL: <https://docs.microsoft.com/en-us/aspnet/overview> (дата звернення: 10.06.2022).
4. Entity Framework Core. Microsoft: вебсайт. URL: <https://docs.microsoft.com/ru-ru/ef/core/> (дата звернення: 10.06.2022).
5. Архитектура современных веб-сервисов и способы их защиты. Anti-malware: вебсайт. URL: https://www.anti-malware.ru/analytics/Technology_Analysis/Architecture-of-modern-web-services (дата звернення: 12.06.2022).
6. C# LINQ. Medium: вебсайт. URL: <https://medium.com/@jaethedeveloper/c-linq-969adff88509> (дата звернення: 10.06.2022).
7. Build fast, responsive sites with Bootstrap. Bootstrap: вебсайт. URL: <https://getbootstrap.com/> (дата звернення: 12.06.2022).
8. Защита веб-приложений. Anti-malware: вебсайт. URL: https://www.anti-malware.ru/analytics/Technology_Analysis/web-security-myths-and-reality (дата звернення: 12.06.2022).
9. Client-Server Architecture. Sciencedirect: вебсайт. URL: <https://www.sciencedirect.com/topics/computer-science/client-server-architecture> (дата звернення: 12.06.2022).

10. Apriori Algorithm in Machine Learning [Електронний ресурс] // Javatpoint – Режим доступу до ресурсу: <https://www.javatpoint.com/apriori-algorithm-in-machine-learning>.
11. Митин А. И. Работа с базами данных Microsoft SQL Server: сценарии практических занятий / А. И. Митин. – Москва|Берлин: Директ-Медиа, 2020. – 23 с.
12. Куликов С. Работа с MySQL, MS SQL Server и Oracle в примерах / Святослав Куликов., 2019. – 329 с.
13. Виробниче середовище та його вплив на людину. URL: <http://dspace.wunu.edu.ua/jspui/bitstream/316497/9184/1/опорний%20конспект%20лекцій.pdf> (дата звернення: 26.05.2022).
14. Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин (НПАОП 0.00-1.31-99). URL: [https://zakon.rada.gov.ua/laws/show/z047214#:~:text=4%20клас%20\(небезпечні%20умови%20праці,1.2](https://zakon.rada.gov.ua/laws/show/z047214#:~:text=4%20клас%20(небезпечні%20умови%20праці,1.2) (дата звернення: 28.06.2022).
15. Щербак Ю.Г., Макарова О.В. Методичні рекомендації до практичних занять із дисципліни «Основи охорони праці». – Миколаїв: Вид-во ЧДУ імені Петра Могили, 2014. – 68 с. – (Методична серія; Вип.215)
16. В. А. Батлук, Б. О. Білінський, В. В. Ковалишин, О. Л. Мірус . Основи охорони праці в підрозділах МНС України: навчальний посібник Вид-во Міністерство освіти і науки, молоді та спорту України 2011. – 503 с.
17. Електромагнітне випромінювання Охорона праці і пожежна безпека: вебсайт. URL: <https://oppb.com.ua/news/elektromagnitne-vuprominyuvannya-chastyna-3>. (дата звернення: 15.06.2022).