

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

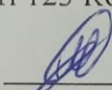
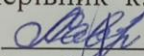

Кафедра обчислювальної техніки

## Пояснювальна записка

до магістерської кваліфікаційної роботи  
магістра

на тему «Веб-орієнтований програмний засіб для інтелектуального аналізу  
потреб споживача інтернет-магазину»

Виконав: студент 2 курсу, групи 2КІ-21м  
спеціальності 123 Комп'ютерна інженерія

  
Старий М.П.  
Керівник к.т.н., доц. каф. ОТ  
  
Савицька Л. А.  
Опонець д.т.н. проф., каф. ПЗ  
  
Романюк О.Н.

Допущено до захисту  
Завідувач кафедри ОТ  
д.т.н., проф. Азаров О.Д.

«23» 12 2022 р.

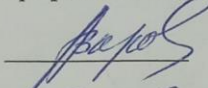
Вінниця 2022

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Освітньо-кваліфікаційний рівень магістр  
Спеціальність 123 Комп'ютерна інженерія

**ЗАТВЕРДЖУЮ**

Завідувач кафедри  
обчислювальної техніки  
проф., д.т.н. О. Д. Азаров

  
«15» 09 2022 р.

## **З А В Д А Н Н Я**

### **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Старому Максиму Петровичу

1 Тема роботи «Веб-орієнтований програмний засіб для інтелектуального аналізу потреб споживача інтернет-магазину», керівник роботи Савицька Людмила Анатоліївна к. т. н., доцент кафедри ОТ, затвержені наказом вищого навчального закладу від 15.09.2022 року № 205-А.

2 Строк подання студентом роботи 15.09.2022 р.

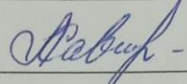
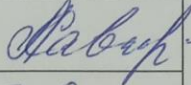
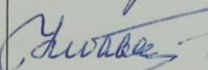
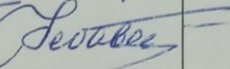
3 Вихідні дані до роботи: середовища розробки Visual Studio 2019 та, мови розробки C# з використанням .NET.CORE, операційна система — Windows 10.

4 Зміст розрахунково-пояснювальної записки: вступ; автоматизація аналізу та збору даних; аналіз даних; розробка структури додатку; розробка алгоритмів програмного додатку; розробка програмного модуля для аналізу статистичних даних, розробка програмного додатку; тестування додатку; висновки; перелік посилань; додатки.

5 Перелік графічного матеріалу: блок-схеми алгоритмів роботи додатку; графічний інтерфейс додатку; тестування додатку.

6 Консультантів розділів роботи представлено в табл. 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Савицька Л.А., к.т.н., доц. каф. ОТ		
4	Небава М.І., к.е.н., проф. каф. ЕП і ВМ		

7 Дата видачі завдання 15.09.2022 р.

8 Календарний план на ведено в табл. 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задач роботи	15.09.22	<i>вкл.</i>
2	Аналіз статистичних характеристик	15.09-16.09.22	<i>вкл.</i>
3	Аналіз переваг використання систем збору даних	16.09-24.09.22	<i>вкл.</i>
4	Огляд існуючих систем аналізу даних	24.09-04.10.22	<i>вкл.</i>
5	Розробка алгоритмів додатку	04.10-14.10.22	<i>вкл.</i>
6	Розробка інтерфейсу користувача	14.10-23.10.22	<i>вкл.</i>
7	Розробка програмних компонент	24.10-31.10.22	<i>вкл.</i>
8	Розробка тестового інтернет-магазину	01.11-16.11.22	<i>вкл.</i>
9	Розрахунок економічної частини роботи	17.11-30.11.22	<i>вкл.</i>
10	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12-06.12.22	<i>вкл.</i>
11	Аналіз виконання роботи, висновки, додатки	07.12-06.12.22	<i>вкл.</i>
12	Перевірка якості виконання магістерської роботи та усунення недоліків	15.12.21	<i>вкл.</i>

Студент

Старий М.П

Керівник роботи

Савицька Л.А.

## Анотація

УДК 004.93

Старий М.П Веб-орієнтований програмний засіб для інтелектуального аналізу потреб споживача інтернет-магазину. Магістерська кваліфікаційна робота зі спеціальності 123 – комп'ютерна інженерія. Вінниця ВНТУ 2022 127с. На укр. мові Бібліогр. 41 рис. 5 табл. 14

Розроблено модель алгоритмів аналізу даних потреб споживачів. Запропоновано принцип представлення аналітичних даних.

Розроблено алгоритми та програмний додаток для аналізу потреб споживачів інтернет-магазину.

Отримані в бакалаврській дипломній роботі результати можна використати для аналізу потреб споживачів з метою підвищення продаж.

Ключові слова: алгоритм, веб-додаток, аналіз, розробка, автоматизація, інтернет-магазин.

### Annotation

Stariy M.P Web-oriented software tool for intellectual analysis of the needs of the online store consumer. Master's thesis on the specialty 123 - computer engineering. Vinnytsia VNTU 2022 127 p. In Ukrainian language Bibliogr. 41 fig. 5 table. 14

In the master thesis, an application was developed to analyze the needs of consumers in order to increase sales. The analysis of methods and means of analysis of consumer data is carried out.

A model of algorithms for analyzing consumer needs data has been developed. The principle of presentation of analytical data is offered.

Algorithms and software application for analyzing the needs of consumers of the online store have been developed.

The results obtained in the bachelor's thesis can be used to analyze consumer needs in order to increase sales

Keywords: algorithm, web application, analysis, development, automation, online store.

The work has 127 pages including appendices and illustrations for the presentation.

## ЗМІСТ

ВСТУП .....	8
1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ .....	11
1.1 Аналіз систем для аналізу даних споживачів .....	11
1.2 Порівняльний аналіз аналогів .....	<b>Помилка! Закладку не визначено.</b>
1.3 Аналіз методів розв'язання поставленої задачі .....	19
1.4 Постановка задачі для розробки програмного додатку для аналізу потреб споживача з метою підвищення продаж .....	20
1.5 Висновки .....	21
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ .....	22
2.1 Аналіз вхідних даних .....	22
2.2 Обґрунтування вибору інтерфейсу .....	28
2.3 Розробка графічного інтерфейсу .....	30
2.4 Розробка алгоритмів роботи додатку .....	33
2.5 Висновки .....	37
3 РОЗРОБКА ДОДАТКУ ДЛЯ АНАЛІЗУ ПОТРЕБ СПОЖИВАЧІВ ІНТЕРНЕТ МАГАЗИНУ ДЛЯ ПІДВИЩЕННЯ РІВНЯ ПРОДАЖ.....	38
3.1 Варіативний аналіз та обґрунтування виборів засобів для реалізації додатку .....	38
3.2 Вибір архітектурного рішення.....	51
3.3 Розробка компонента View .....	52
3.3 Розробка компонентів .....	53
3.5 Висновки .....	54
4. ТЕСТУВАННЯ ПРОГРАМИ.....	55

					08-23.МКР.028.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Старий М.П.			Веб-орієнтований програмний засіб для інтелектуального аналізу потреб споживача інтернет-магазину	Лім.	Арк.	Аркушів
Керівник		.Савицька Л.А.					6	
Рецензент						<b>ВНТУ, гр. 2КІ-21м</b>		
Н. Контроль		Швець С.І.						
Затверджую		Азаров О.Д.						

4.1 Методи тестування.....	55
4.2 Тестування веб-додатку .....	56
4.3 Розробка інструкції користувача .....	64
4.4 Висновки .....	65
5. ЕКОНОМІЧНА ЧАСТИНА.....	66
5.1 Комерційний та технологічний аудит науково технічної розробки.....	66
5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи .....	69
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	75
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ... <b>Помилка! Закладку не визначено.</b>	
Додаток А Технічне завдання .....	<b>Помилка! Закладку не визначено.</b>
Додаток Б Лістинг програми.....	88
Додаток В Ілюстративний матеріал до захисту бакалаврської дипломної роботи.....	116

										Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	08-23.МКР.028.00.000 ПЗ					7

## ВСТУП

Актуальність розробки даного програмного забезпечення. З кожним роком кількість покупок зроблених через інтернет-магазин зростає. У 2018 році кількість інтернет-покупців становила приблизно сім мільйонів чоловік, а кількість витрачених грошей становила близько 78 млрд. гривень. При чому майже всі кошти були потрачені не просто як оплата через інтернет, а саме покупка товарів через інтернет, ця цифра сягає 65 млрд. гривень. А вже у 2019 році кількість коштів, потрачених через інтернет виросла на 25% і сягнула позначки 97 млрд. гривень, з яких 81 млрд гривень витрачено на покупку товарів та послуг саме в інтернет – магазинах.

Тенденція зростання продаж та покупок через інтернет помітна давно, адже набагато зручніше купувати з дому, маючи весь каталог товару перед очима, не потрібно очікувати черги в магазинах і т. д. Але найбільший «бум» покупок в інтернеті спричинив карантин. За даними маркетплейсу «Хорошоп» кількість онлайн покупок під час карантину виросло до більшого числа ніж під час новорічних свят до карантину, а кількість нових інтернет магазинів збільшилось приблизно в два рази, оскільки багато підприємців прийняли рішення перейти до онлайн продаж.

Таким чином продажі онлайн є набагато вигіднішою бізнес моделлю, яку може використовувати підприємець, але не все так просто як це може показатись на перший погляд. Інтернет магазин-це зовсім інша модель бізнесу. На відмінну від звичайного магазину, де є вітрина з товарами, продавець консультант, який вирішує конфліктні питання, де покупець може завжди спробувати товар, доторкнутися до нього, легко повернути товар назад до магазину. Всі ці питання потребують іншого підходу, якщо покупка відбувається в інтернет магазині.

Найцікавішим в бізнес моделі є система аналізу та обробки даних про відвідувачів інтернет-магазину. Такі системи аналізують дані про



користувачів для подальшого використання в рекламі, персоналізації, рекомендаціях тощо. Вони збирають великі масиви даних, обробляють, та видають метрики, які, менеджери та веб-аналітики використовують для покращення рівня продаж послуг, товарів, реклами тощо. Великі інтернет-магазини наймають для цього цілі команди аналітиків, щоб якнайкраще оцінити цільову аудиторію та підвищити рівень продаж. Кожне відвідування інтернет-магазину, який використовує подібні системи, користувачем записується та обробляються дії, зроблені на сторінці, також обробляються покупки. Також системи для аналізу даних про споживачів стараються оцінювати наскільки товар чи послуга були цікавими для споживача, таким чином веб-аналітики отримують метрики, по яким потім вирішують майбутнє магазину.

Кількість покупок в інтернет-магазинах вже через декілька років будуть перевищувати кількість покупок в звичних магазинах, що приводить до збільшення об'ємів даних про користувачів і товари. Аналіз потреб споживачів з подальшою оптимізацією товару чи послуг інтернет-магазин є одним і головних факторів повноцінного конкурування, що доводить актуальність розробки системи для аналізу потреб споживачів.

**Метою** роботи є розробка програмного додатку аналізу потреб споживача інтернет-магазину для підвищення ефективності обробки даних запитів користувачів і корегування стратегії розвитку інтернет-магазинів.

Для досягнення поставленої мети потрібно розв'язати такі задачі:

- провести аналіз основних систем аналізу потреб споживачів інтернет-магазину;
- запропонувати новий метод аналізу потреб споживачів інтернет-магазину;
- розробити програмні компоненти обробки даних;
- розробити додаток аналізу запитів користувачів;
- провести експериментальні дослідження розроблених засобів.

**Об'єкт дослідження** — процес аналізу потреб споживача інтернет-магазину для підвищення ефективності обробки даних.

**Предмет дослідження** — методи та засоби аналізу потреб споживача інтернет-магазину для підвищення ефективності обробки даних.

**Методи дослідження.** У процесі досліджень використовувались: теорія чисел та чисельних методів, лінійна алгебра, методи матричних обчислень для побудови відповідності між товарами та користувачами, методи статистичної обробки даним та математичного аналізу

**Наукова новизна дослідження.**

Полягає в тому що, подальшого розвитку набув метод аналізу статистичних даних запитів користувачів інтернет-магазину.

**Практична цінність отриманих результатів.**

Полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень розроблено алгоритми та програмні засоби аналізу потреб споживачів в інтернет-магазині

**Особистий внесок здобувача.** Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто.

## 1 АВТОМАТИЗАЦІЯ АНАЛІЗУ ТА ЗБОРУ ДАНИХ

### 1.1 Систем для аналізу персональних даних споживачів

Аналіз даних споживачів використовується для визначення аудиторії, яка відвідала сторінку. Кожен користувач, який користується веб-сервісом, є користувачем того чи іншого сервісу, і якщо він чи вона чимось користується, він чи вона має за це якимось чином заплатити. Оплата готівкою не обов'язкова. Наприклад, соціальні мережі безкоштовні. Користувач «платить» за це своїми даними, і на основі цих даних отримує рекламу. Це явище, відоме як «цільова реклама», по суті є рекламою «персоналізованих» продуктів і послуг для кожного користувача. Такий результат можна отримати лише шляхом аналізу наборів даних для кожного користувача окремо. Великі соціальні мережі, такі як «Facebook», «YouTube», «Instagram», завдяки нейронним мережам обробляють великі обсяги даних про своїх користувачів і на основі цих даних продають персоналізовану рекламу.

Персоналізація в сучасному світі є найважливішим рушієм продажів, тому що якщо ритейлер зможе якимось чином запропонувати свої товари користувачам, які в них вже зацікавлені, то вартість реклами буде дешевшою і при цьому працюватиме краще. Це не єдиний спосіб використання особистої інформації користувачів, яким користуються сучасні компанії. Іншим способом використання персональних даних, які використовуються інтернет-магазинами, є аналіз потреб споживачів. Знаючи потреби та обставини вашої цільової аудиторії, ви можете більш ефективно рекламувати та продавати свою продукцію.

Аналізуючи споживчу інформацію, інтернет-магазини отримують «картинку» споживача. У свою чергу, такі системи використовуються для вдосконалення самого продукту чи послуги, якщо покупець відомий, його

можна оптимізувати для зменшення збитків або збільшення прибутку. Системи аналізу даних зазвичай мають форму інструменту, який використовується менеджерами або власниками інтернет-магазинів, де зібрані дані можуть бути проаналізовані та оброблені. Інше використання аналітичних даних можна назвати розбиттям на сторінки. Отримавши «моментальний знімок» споживача, вони можуть приблизно передбачити, що саме їх зацікавило і чому вони відвідали сторінку, і на основі цієї інформації пропонувати персоналізовані знижки тощо.

Персоналізація сайту — це метод, який дозволяє персоналізувати контент у реальному часі під конкретного користувача. Найважливішим у цьому плані є те, що такі налаштування працюють у режимі реального часу, а це означає, що користувачеві не потрібно вже авторизуватися чи відвідувати сторінку. Кожного разу, коли користувач відвідує сторінку конфіденційності, він/вона отримує сповіщення про те, як будуть оброблятися його/її дані та файли cookie. Пересічний користувач зазвичай не звертає уваги на спливаючі вікна, коли вони вперше відвідують сторінку, але це спливаюче вікно є суттю цих систем.

Виникає питання, навіщо підлаштовувати сторінку під кожного користувача, навіщо винаходити п'яте колесо, якщо можна просто створити красиву, стилізовану сторінку, яка запам'ятається відвідувачу. Так, можна, але можна створити таку ж цікаву сторінку, але з контентом, для кожного споживача окремо, для чоловіка одна інформація, а для жінки показувати зовсім іншу. Сьогодні об'єкти без персональних атрибутів, а тим більше інтернет-магазини, навряд чи є конкурентоспроможними. Персоналізовані веб-сайти для споживачів мають майже вдвічі більший коефіцієнт конверсії, ніж сторінки без таких систем. Це пов'язано з тим, що якщо користувач помітить щось, що може негайно зацікавити, він відкрис цікавий вміст, посилання тощо. Такий підхід є безпрограшним як для користувача, так і для власника програми, тому що користувач отримує цікавий контент, а власник отримує продаж або постійного відвідувача свого додатка.

Сьогодні на ринку існує багато систем веб-аналітики, вони різні, вони задовольняють різні ринки та використовують різні робочі процеси, вони мають різні функціональні можливості та можливості, але основним недоліком кожної є необхідність наймати спеціалістів для їх використання, оскільки впровадження не є найпростішим і вимагає певних знань для повного використання можливостей системи, було вирішено розробити структуру для аналізу інформації, пов'язаної з потребами споживачів інтернет-магазину для збільшення продажів.

Сьогодні на ринку існує багато систем для аналітики веб-трафіку, вони різні, вони задовольняють різні ринки та використовують різні робочі процеси, вони мають різні функціональні можливості та можливості, але основним недоліком кожної є необхідність наймати спеціалістів для їх використання, оскільки впровадження не є найпростішим і вимагає певних знань для повного використання можливостей системи, було вирішено розробити структуру для аналізу інформації, пов'язаної з потребами споживачів інтернет-магазину для збільшення продажів.

## 1.2 Технічне порівняння аналогів

На ринку є багато варіантів продуктів для аналізу даних потреб споживачів. Все це працює за принципом SaaS через підключення API. «Інтерфейс прикладного програмування» — це система, в якій користувач взаємодіє з віддаленим сервером, у якому користувач робить запити на сервер, який, у свою чергу, надсилає необхідну інформацію.[1] Тобто, щоб процес споживчої аналітики працював на сторінці, її потрібно підключити до бази даних через API. Таким чином, користуватися цими системами дуже просто, потрібно лише правильно все налаштувати, і ви можете негайно почати користуватися платними послугами.[2]

Кожна система аналізу споживчих даних пропонує різні можливості та функціональні можливості, але єдине, що об'єднує таке програмне забезпечення, полягає в тому, що всі вони відстежують, звідки надходить

трафік, а саме те, як користувач потрапив на сторінку та час, витрачений на неї. Крім того, функціональність змінюється залежно від потреб покупця, деякі обробляють і зберігають дані користувача, деякі оцінюють інтерес користувача, який відвідує програму, на основі їх статистики та виводять відповідні показники, деякі також можуть переглядати списки конкурентів, що допомагає краще оцінити ринок тощо d.

Кожна платформа аналітики даних користувача пропонує різні можливості. Деякі повністю налаштовують вміст сторінки під користувачів, деякі зосереджуються лише на інтернет-магазинах, інші можуть лише відправляти розсилку та рекламувати свій додаток у зручний спосіб. Давайте розглянемо загальні протоколи, які використовуються для налаштування веб-додатків.[3]

Google Analytics (рис. 1.1), система аналізу аналітики даних про користувачів веб-додатків, розроблена корпорацією Google, є найпопулярнішим API-сервісом серед таких компаній. За оцінками 2020 року, Google Analytics встановлено на 29 мільйонах веб-сайтів. Причиною його популярності є інтеграція з Google Ads.

Google Ads – це інструмент для збирання оголошень Google в Інтернеті. Реклама показується від самої компанії в залежності від користувача, який відвідав додаток. Більше того, сама ця реклама жодним чином не залежить від змісту та категорії відвіданої роботи. Google рекламує продукти, які дуже привабливі для користувача, на основі наявних у нього даних. За відсутності інформації про відвідувача реклама буде відображатися відповідно до програми та її специфікацій.[4]

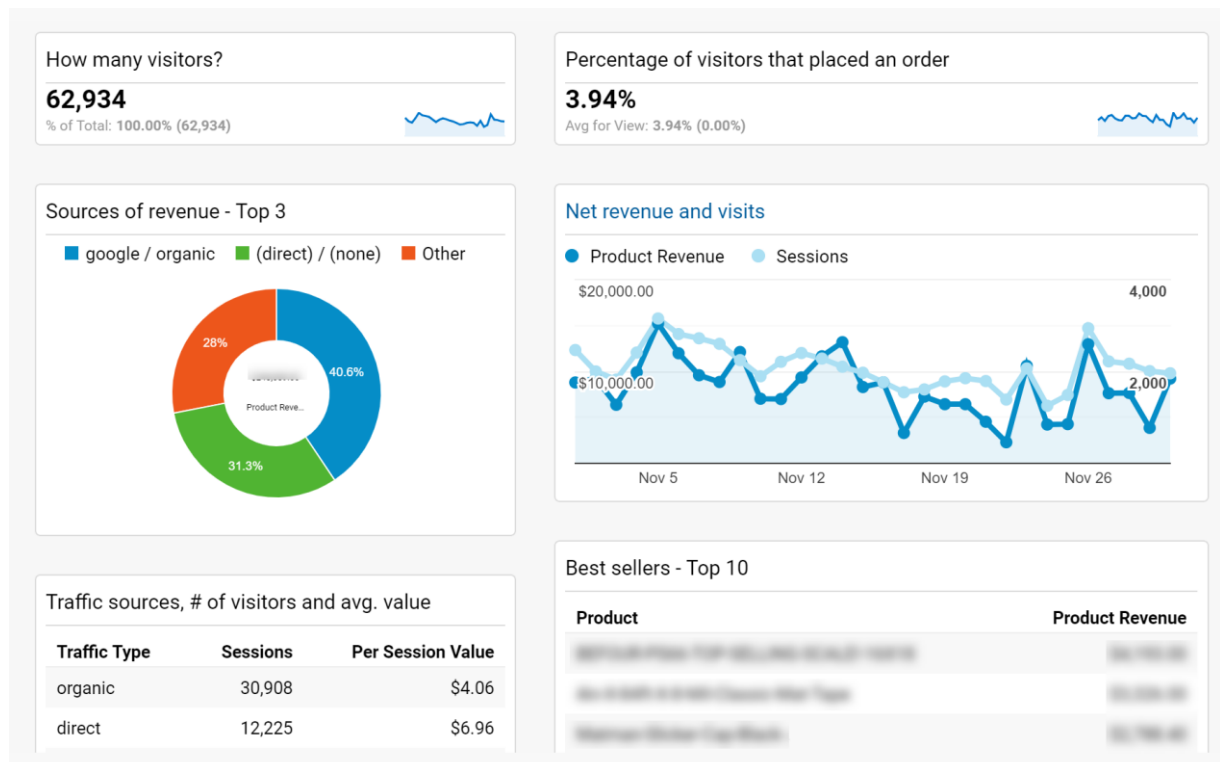


Рисунок 1.1 – Google Analytics

Яндекс.Метрика (рис. 1.2) – безкоштовний сервіс для аналізу та обробки споживчих даних, розроблений корпорацією Яндекс, виконує практично ті ж функції, що і Google Analytics, але більше адаптований для російських сервісів, також використовує інтеграцію з її пошуковими сервісами. Для кращої аналітики та збільшення продажів його найкраще використовувати разом із сервісом персоналізації реклами Яндекс.Директ, який працює на Яндекс.Метриці.

По суті своїй даний аналог майже нічим не відрізняється по функціоналу від Google Analytics. Головною відмінністю між ними являє собою принципи роботи та рекомендацій. Також в один з головних недоліків російського ресурсу є його безпекова складова. Дуже часто відбувались витоки даних користувачів з даної корпорації.[6]

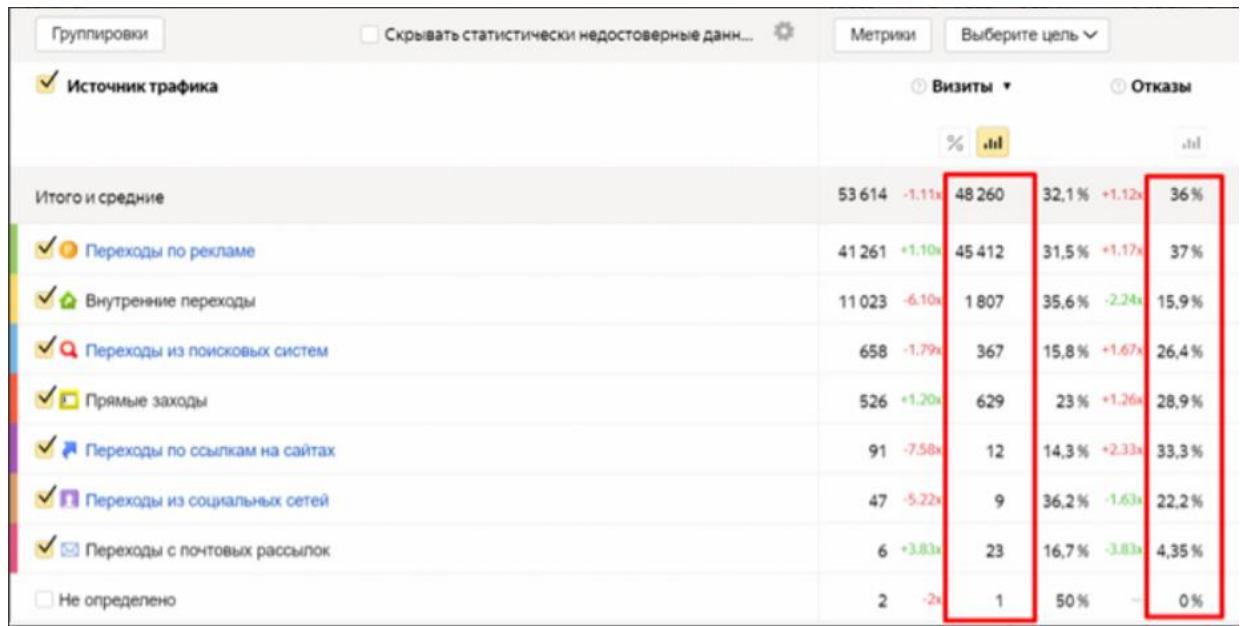


Рисунок 1.2 – Яндекс.Метрика

Finteza (рис. 1.3) – сервіс, що спеціалізується на аналізові і обробці даних споживачів, основною перевагою сервісу є розміщення на різних серверах, це ніяким чином не сповільнює роботу ресурсу, що його використовує сервер. Також однією з переваг даного ресурсу можна вважати захищеність та зберігання даних про користувачів не в на власних серверах, а на серверах клієнтів, це унеможлиблює використання даних «третіми» особами.

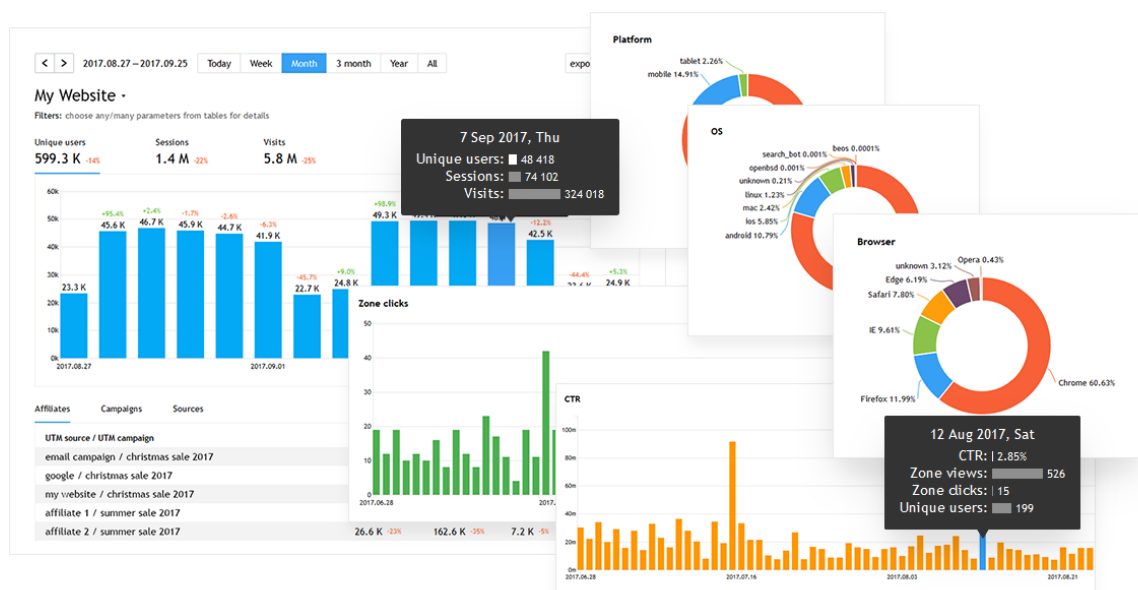


Рисунок 1.3 – Finteza



HubSpot (рис.1.4) хмарне ПЗ, яке в першу чергу позиціонує себе як повноцінний «БекЕнд» інтернет-магазину. Включає як і програмне забезпечення для підвищення продаж так і системи для внутрішнього функціонування проектів.

Працює по принципу збору даних про користувачів, а потім в власній CRM системі дозволяє інтегрувати ці дані в конкретні графіки та інформацію, яку вже буде обробляти працівник магазину.

Недоліком даної системи являє собою необхідність працівників знань не тільки маркетинга а також і CRM систем або ж утримання двох робітників для роботи з даною системою.

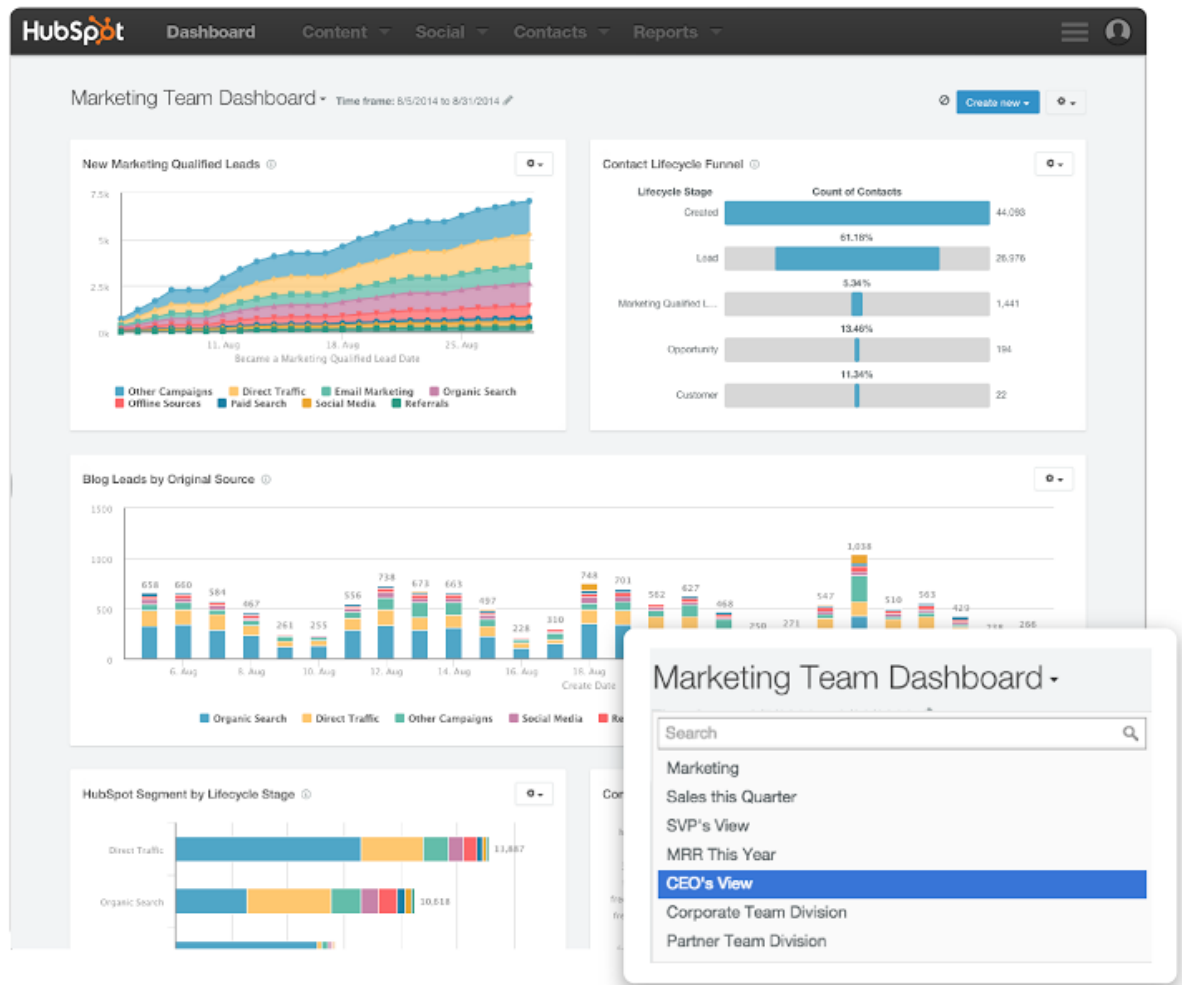


Рисунок 1.4 – HubSpot

В результаті аналізу найпоширеніших на ринку аналогів систем аналізу споживчих даних інтернет-магазину було виявлено їх сильні та слабкі сторони, які були враховані при розробці власного програмного забезпечення що називається Piedava. (див. табл.1.1).

Таблиця 1.1. – Порівняльні характеристики систем персоналізації

Критерій	Google Analytics	Яндекс.Метрика	Finteza	HubSpot	Piedava
Аналіз джерела трафіку	1	1	1	1	1

Аналіз часу проведеного на ресурсі	1	1	1	1	1
Аналіз «підозрілих» відвідувачів	0	0	1	1	0
Визначення ефективності рекламних кампаній	1	1	0	0	1
Аналіз КРІ сайту	1в	1	0	0	1
Рекомендації по зміні дизайну сайту	0	0	1	1	1
Сумарне значення	4	4	4	4	5

Порівнявши наразі представлені на ринку системи, для аналізу даних та персоналізації було зроблено висновки про доцільність розробки даного продукту, оскільки система аналізу даних під назвою Piedava врахувала основні недоліки існуючих рішень, та вирішила їх.

### 1.3 Аналіз методів розв'язання поставленої задачі

Фреймворк для аналізу даних аналізу потреб споживачів інтернет-магазину можна розробити за допомогою мікросервісу або монолітної архітектури. Під час розробки єдиної архітектури програмне забезпечення керується та використовується в середині одного модуля, що унеможливило використання будь-якої частини продукту окремо від інших. Розробка програмного забезпечення таким чином проста з точки зору архітектури мікросервера, але вона несе певні ризики, такі як: додавання нових програм із

високим споживанням ресурсів, є ризик припинення роботи цілого модуля, навіть при незначних помилках.[7]

Мікросерверна архітектура - це принцип створення програмного продукту, в якому кожен модуль є окремим і незалежним від іншого, обмінюючись лише необхідною для роботи інформацією. Таким чином, програмний пакет, заснований на архітектурі мікросервера, може бути частково профінансований, легше додавати нові сервіси до таких сервісів, і він більш стійкий до збоїв.[8]

Щоб розробити інструмент для аналізу споживчих потреб інтернет-магазину, була використана архітектура мікросервера, оскільки вона дозволяє додавати нові послуги за відносно короткий проміжок часу та працювати з додатком різними способами в різних секторах. Власник інтернет-магазину необхідний у часовий проміжок, з можливістю легко інтегрувати нові пакети послуг.

#### 1.4 Постановка задачі для розробки програмного додатку для аналізу потреб споживача з метою підвищення продаж

Після аналізу питання розробки додатку для дослідження потреб споживачів з метою підвищення рівня продаж інтернет-магазину, було визначено такі завдання:

- визначення найбільш ефективного підходу для розробки веб-орієнтованого програмного засобу для аналізу потреб споживачів з метою підвищення рівня продаж інтернет-магазину
- розробка графічного інтерфейсу, для показу результатів
- розробка інтернет-магазину з метою тестування розробленого засобу
- розробка веб-орієнтованого програмного засобу для аналізу потреб споживачів з метою підвищення рівня продаж інтернет-магазину
- проведення тестування веб-орієнтованого додатку.

## 1.5 Висновки

У першому розділі було описано аналіз стану питання, веб-орієнтованого програмного засобу для аналізу потреб споживача інтернет магазину з метою підвищення продаж.

Найбільш поширені аналоги були проаналізовані та порівняні з програмним забезпеченням, що розробляється, і доведено, що це можливо.

Було проведено опитування щодо існуючих методологій розробки для вирішення даного проекту, в результаті опитування було вирішено використовувати мікросервісну архітектуру проекту.

Визначено завдання, які необхідно виконати для розробки програми оцінки потреб споживачів для збільшення обсягу продажів інтернет-магазину.

## 2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

### 2.1 Аналіз вхідних даних

Коли ви відвідуєте веб-сайт, який використовує систему для сканування веб-сторінок, користувач не може побачити результат або фактичні функції цієї системи, розробка проводиться для самого ресурсу, а не користувача. Такі програми використовують сервера ресурсів, на яких вони розташовані і збирають дані про відвідувачів ресурсу для подальшої обробки. Обробка цієї інформації означає аналіз поведінки користувачів або відвідувачів. Програмний модуль аналізує час перебування на сайті, звідки відвідувач відкрив сторінку, покупки, кошик тощо. Чим більше даних збирає система, тим кращим і точнішим буде аналіз [9].

Веб застосунок Piedava використовує наступні особисті дані::

- джерело відвідувача,
- час, проведений на ресурсі,
- користувацькі купівлі,
- продукти, збережені в кошику,
- переглянута продукція,
- пошукові запити,
- попередні замовлення клієнтів.

Всі ці параметри опиняються в базі даних не прив'язуючись до конкретного відвідувача. Далі відбувається аналіз отриманих даних. По-перше, він обробляє статистику про джерела відвідувань, тобто визначає ефективність тієї чи іншої реклами (рис. 2.1.). Отримання статистики щодо джерел конверсій є однією з ключових функцій Piedava, оскільки це допомагає краще розподіляти бюджет на рекламу.

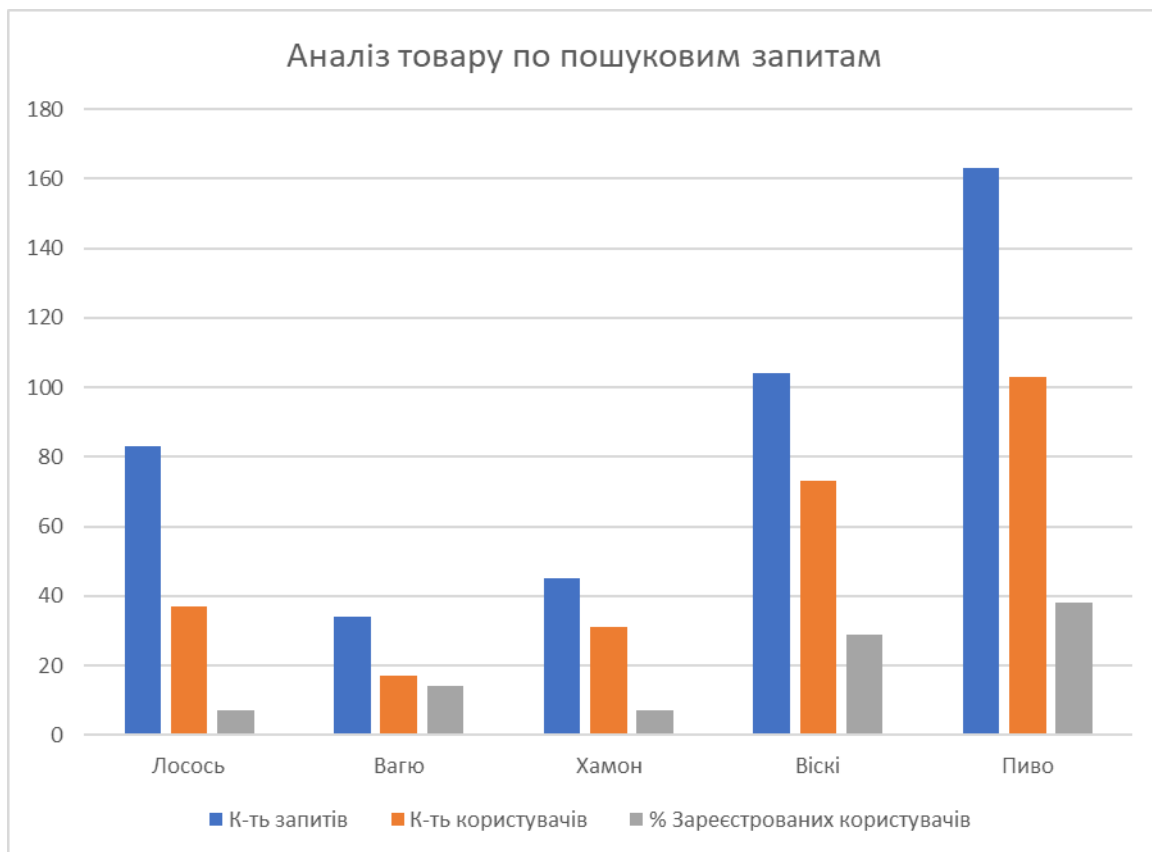


Рисунок 2.1 – Аналіз товару по запитам

Друге – обробка даних постійних покупців. Тобто обробляти дані користувачів, які вже зареєструвалися і часто використовують веб -ресурс, де встановлена дана система. Ця програма аналізує найбільш продавані категорії продуктів. Ця функція дозволяє власникам пропонувати знижки та переваги, щоб утримувати постійного клієнта в магазині[10].

Програмне забезпечення Piedava аналізує, скільки часу відвідувачі проводять на вашому сайті та що вони роблять під час відвідування. Для надійних результатів користувачі, які відвідують сторінку менше двох секунд, такі відвідувачі вважаються фальшивими. Користувачі, які відвідують сайт від десяти до тридцяти секунд, вважаються незацікавленими в даному ресурсі. Користувачі, які відвідали сайт від хвилини до трьох хвилин протягом періоду опитування, вважалися незадоволеними ціною або самим товаром.

Відвідувачі з декількома пошуковими запитами входять до групи користувачів, які не знайшли те, що шукали, і в цьому випадку пошукові запити, які не дали результатів для цього користувача, будуть перелічені в базі даних. Крім того, можуть бути обставини, коли в магазині немає продукту, який цікавить користувача, але, на жаль, ми не можемо отримати інформацію про те, чого немає в магазині. В таких випадках працівник зобов'язаний провести аналітику, та вибрати нерелевантні запити, які дали не результати через відсутність продуктів у каталозі та відповідали отриманим даним[11].

Відвідувачі, які шукають 10 або більше термінів, вважаються зацікавленими та поділяються на декілька категорій. Ці категорії розбиті за типом продукту, і користувачі можуть сортувати різні категорії на основі результатів тесту, щоб глибше вивчити інтереси, спеціальні пропозиції та знижки.

На основі замовлень і звернень постійних клієнтів розробіть стратегію підвищення кваліфікації магазину. Продукти, придбані споживачем більше одного разу, асоціюються з особистістю споживача, і відповідно до цієї статистики ми можемо пропонувати спеціальні пропозиції та знижки.

Оцінка візків для покупок також важлива, оскільки споживачі можуть вибирати з широкого діапазону варіантів. Це необхідно для того, щоб можна було обговорити пропозиції та знижки разом з іншими речами, наприклад, якщо ви купуєте мобільний телефон, ви також можете купити чохол для мобільного телефону разом.

Аналіз продукту виконується програмним забезпеченням Piedava за допомогою пошукових запитів і базується на співвідношенні пошукових запитів до кількості користувачів, які зробили запити. Дивлячись на гістограму відгуків про товар за запитами (рис. 2.1.), бачимо, що співвідношення пошукових запитів до кількості користувачів, які їх створили, різниться в залежності від категорії товару. Аналізуючи цю гістограму, ви можете визначити найбільш і найменш бажані продукти серед



споживачів, щоб ви могли розробити стратегії продажу цих продуктів і максимізувати страхування та прибутки.

Ми використовуємо інші методи аналітики для оцінки зареєстрованих користувачів, які відвідують наш інтернет-магазин, використовуючи статистичні дані. Коли зареєстровані користувачі переглядають продукти, програма записує те, що вони бачать, і використовує цю інформацію для оцінки потреб та інтересів (рис. 2.2).

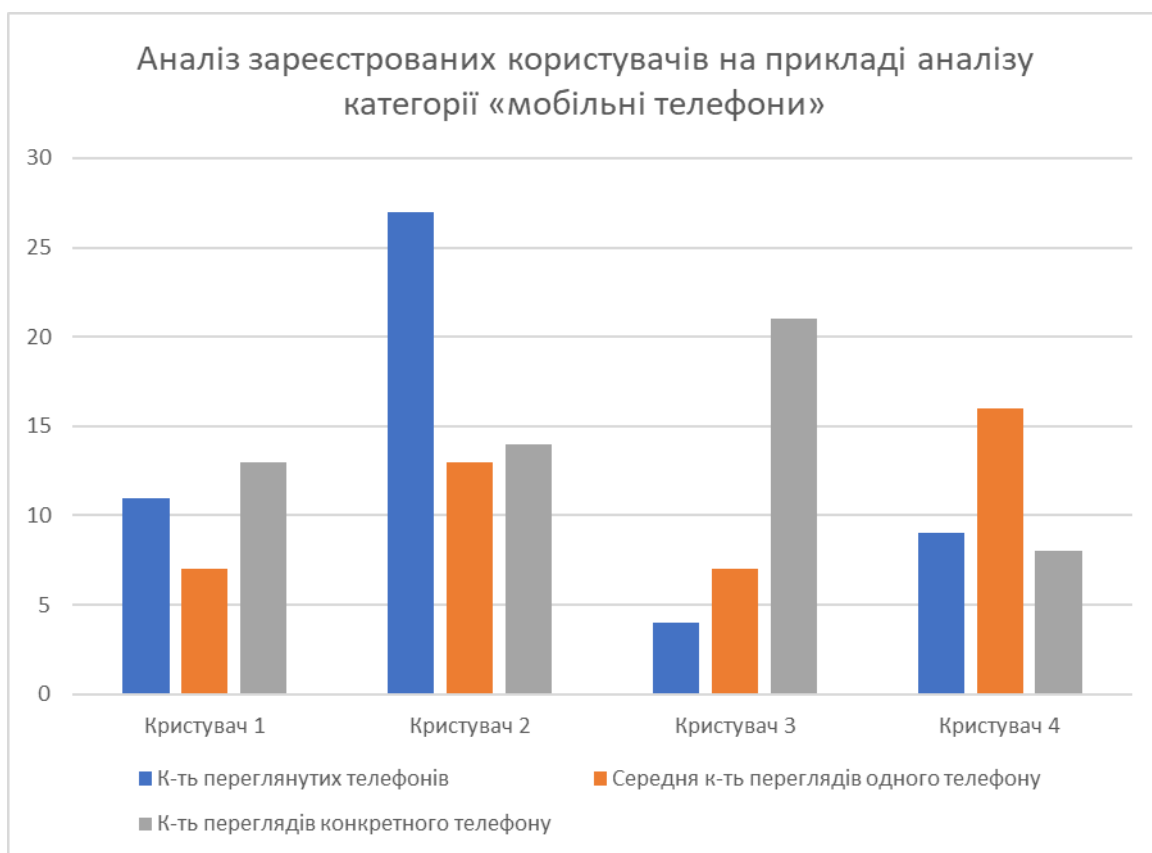


Рисунок 2.2 – Аналіз зареєстрованих користувачів на прикладі аналізу категорії «мобільні телефони»

Зареєстровані клієнти регулярно піддаються аналізу на відношення перегляду конкретного товару до середньої кількості переглядів даного товару. Якщо переглядів значно більше, ніж середній для даного продукту, то цей споживач, ймовірно, зацікавлений у цьому товарі. Дане відношення буде коливатись в залежності від магазину та товару, оскільки ціна в даному

випадку має вирішальне значення. Дорогі покупки зазвичай є більш зваженим рішенням для покупця і відповідно переглядів буде більше, ніж в дешевих продуктах. [12] .

Дані, які показано на (рис 2.2) показують описану вище залежність, які визначають потреби користувачів щодо певного продукту, оскільки кількість показів певного продукту значно перевищує середню кількість показів інших товарів. Ця програма надсилає даного користувача до набору користувачів, які вже знайшли для себе продукт, але з якихось причин ще не готові його купити.

Співвідношення середньої кількості переглядів до кількості переглядів «Користувача 2» про конкретний продукт є приблизно рівним, що вказує на те, що він не зацікавлений у продуктах, вибраних для аналізу. Також кількість показів показує, що другий користувач зараз щось вибирає для себе і хоче щось купити. Додаток розглядає цих користувачів як потенційних покупців.

«Користувач 3» переглянув продукт приблизно втричі більше, ніж у середньому, тож він, імовірно, уже вибрав продукт і готовий його купити. Крім того, оскільки кількість показів низька, можна побачити, що цей користувач досліджує сторінку онлайн-магазину з певним розумінням того, що він або вона хоче купити.

Щоб визначити, скільки товарів користується попитом, менеджер інтернет-магазину визначає значення параметрів окремо для кожної товарної категорії на основі цільової демографічної групи товару, середньої ціни товарів у категорії тощо.

«Користувач 4» не зацікавлений у перегляді, оскільки за даними (рис. 2.2) кількість переглядів окремого товару, значно менша за середню кількість переглядів даного товару. Отримані дані опитування допоможуть нам удосконалити ресурс, на який встановлено дану систему, дозволяючи нам пропонувати додаткові персоналізовані продукти та знижки.

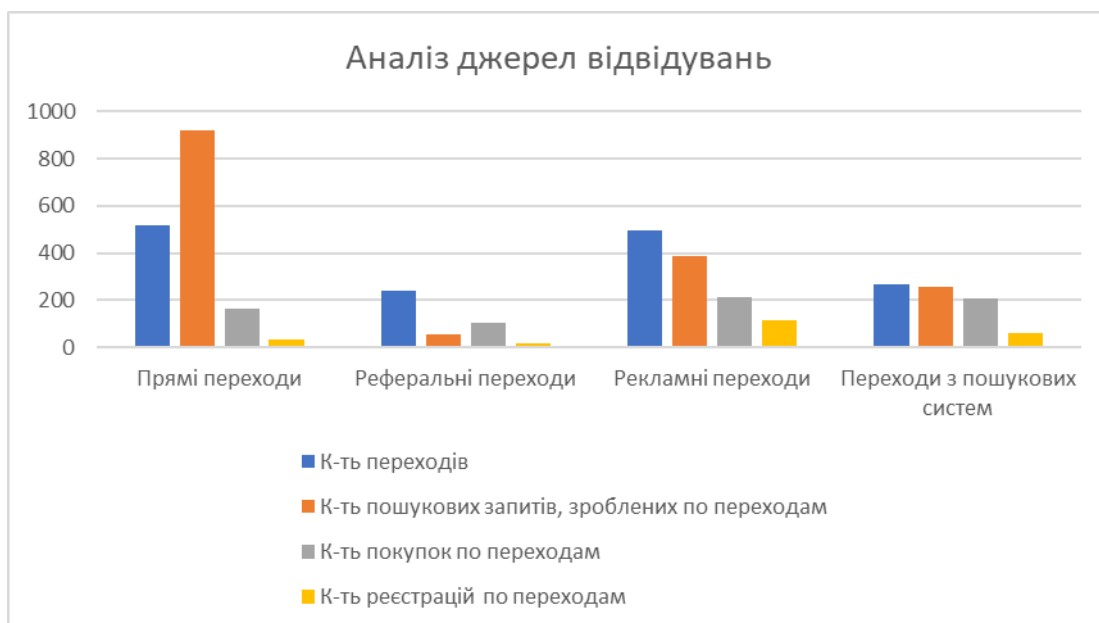


Рисунок 2.3. – Аналіз джерел відвідувань

Розподіл витрат клієнтів на рекламу здійснюється на основі даних, наведених на (рис. 2.3). Статистика прямих конверсій показує відносно низьку кількість реєстрацій і показів нових користувачів, але це пов'язано з тим, що у випадку прямої конверсії користувачі реєструються в магазині та є постійними клієнтами. Ці оцінки підтверджуються у стовпцях із прямими змінними.

Прямі посилання користуються найбільшою видимістю та попитом, оскільки пряме посилання рекламує певний продукт, а не весь магазин, тому користувачі, які заходять у ваш інтернет-магазин за прямим посиланням, одразу купуватимуть цей продукт.

Дані про переходи по рекламним посиланням показують, що в результаті переходом, по рекламним посиланням реклами більше покупок, ніж прямих при прямих переходах на сайт.

Співвідношення покупок до реєстрації становить приблизно 1:2 (рис. 2.3), і хоча співвідношення покупок низьке, можна судити про те, що такі користувачі частіше купують додаткові товари в інтернет-магазині.

Щоб зрозуміти, як правильно аналізувати статистику кліків із пошукових систем, потрібно зрозуміти, що користувачі шукають не сам

магазин. Гістограма (рис. 2.3.) показує, що кількість відвідувань магазину дуже близька до кількості покупок одного товару, при великих розбіжностях реальних показників з гістограмою, валснику слід задуматись над зміною рекламної кампанії. Співвідношення пошукових запитів до куплених товарів має бути точно таким же. Щоб підтримувати цей зв'язок, ваш веб-сайт має бути оптимізовано для високого рейтингу в пошукових системах.

Аналіз джерел конверсії необхідний для аналізу результатів рекламних кампаній, і на рисунку 2.3 показані оптимальні результати реклами. Результати змінюватимуться залежно від ціни продукту та цільової демографічної групи цього продукту, але загалом підтримувати форму, подібну до гістограми, необхідно для стабільної реклами та продажів для зростання магазину. Якщо дані не збігаються з гістограмою, власник магазину повинен подумати про зміну реклами. [13]

Тому для аналізу потреб споживачів в інтернет-магазинах використовується система під назвою «Piedava». Результати розрахунків на основі результатів Piedava записуються у файл Excel, щоб персонал міг легко їх використовувати та змінювати.

## 2.2 Обґрунтування вибору інтерфейсу

Інтерфейс користувача складається з програмних і апаратних додатків, які використовуються для обробки та відображення інформації користувачеві.

Для розробки програмного забезпечення Piedava було вирішено використання веб-орієнтованого застосунку. Перевага веб-додатку над графічним інтерфейсом користувача полягає в тому, що запуск додатку через Інтернет не потребує встановлення додаткового програмного забезпечення та можливості використовувати різні типи браузерів для роботи додатка.

Класифікацію веб-сайтів приведено на рис 2.4.

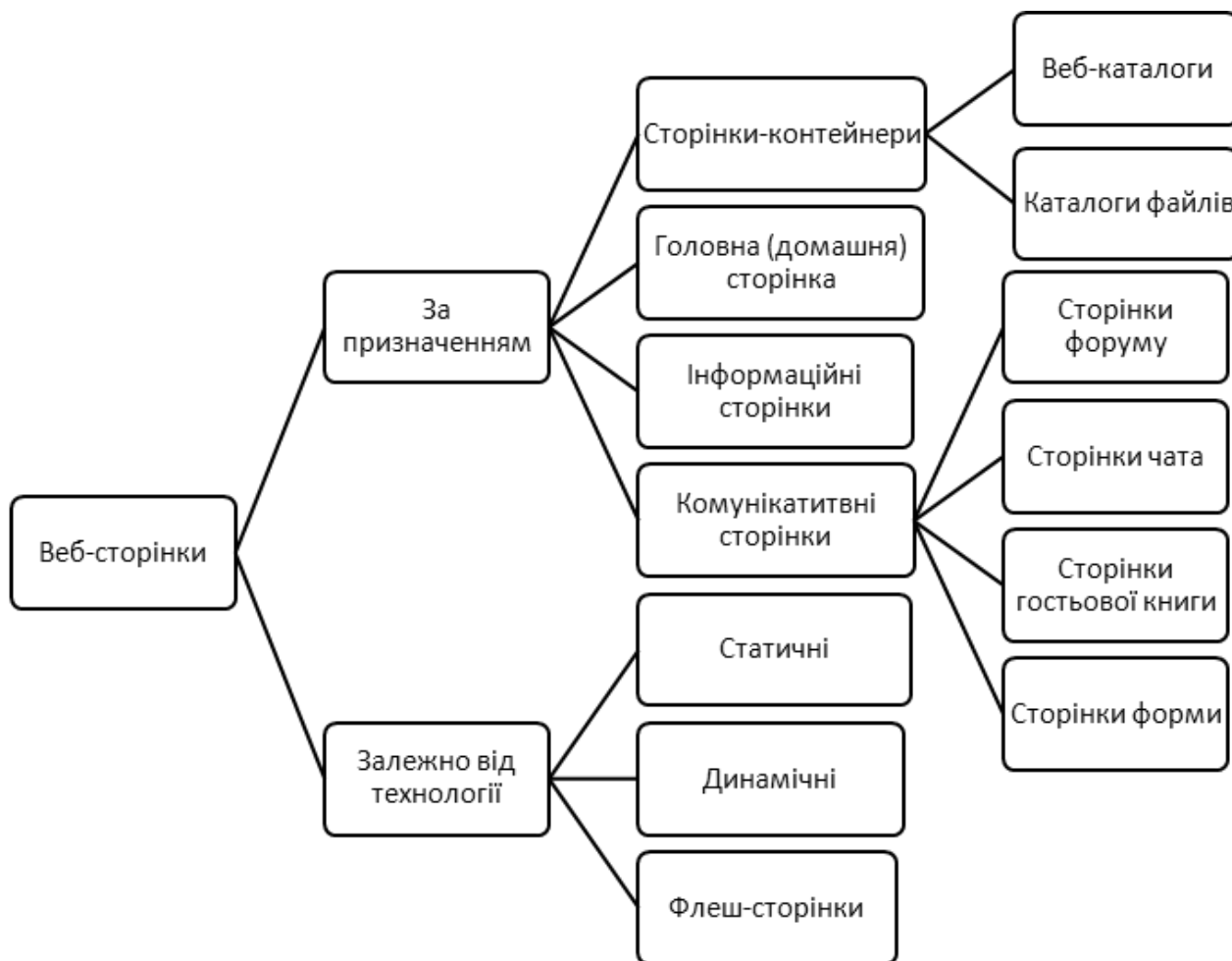


Рисунок 2.4 - Класифікація веб-сайтів

Статичні сторінки часто створюються за допомогою HTML, мови гіпертекстової розмітки. Для всіх клієнтів, які відвідують цей сайт, дані сторінки залишаються незмінними. Такі сторінки не вимагають повторних змін даних чи структури чи спеціального програмного забезпечення для зберігання даних. На цій сторінці розміщені навчальні матеріали, історичні відомості, описи музейних експонатів тощо. [14].

Динамічні сторінки створюються за допомогою мов програмування, таких як PHP (Інструменти персональної домашньої сторінки), ASP (Активні серверні сторінки) і PERL (Практична мова вилучення та звітів). Ця сторінка призначена для відображення інформації, яка часто оновлюється. Він може містити сторінку новин, каталоги інтернет-магазинів, добірку популярних

відеокліпів і багато іншого. Користувачі можуть переміщатися по цих сторінках, щоб отримати доступ до різноманітної інформації[16].

Флеш-сторінки (flash) створюються за допомогою програми візуалізації Adobe Flash, в результаті чого виходять яскраві, привабливі сторінки з великою кількістю графічних і звукових ефектів. Відтворення флеш-анімації змінює вигляд веб-сайту залежно від того, де знаходиться курсор. Ось чому веб-сайти для дизайнерів, художників, фотографів тощо часто містять флеш-сторінки, які роблять їхні роботи більш вражаючими в динамічному стилі. У той же час, флеш-сторінки важко редагувати, тому розміщувати публікації й подібні матеріали здебільшого не є проблемою. Flash-сторінки повільні через розмір файлу. Щоб отримати до нього доступ, потрібно встановити плагін флеш-анімації.

Проаналізувавши, як користувачі взаємодіють із сайтом, і зрозумівши характеристики кожного, ми створили сайт Piedava, використовуючи статичні сайти.

### 2.3 Розробка графічного інтерфейсу

Було використано Asp.NET Core для інтерфейсу зображення. Інтерфейс графіка має кнопки для керування статистикою й аналізом, експортування результатів у файли Excel і перетворення в категорії аналізу. Інтерфейс реалізований у форматі бази даних, і ви можете завантажити необхідні файли у форматі .XLSX. Профіль користувача призначений для власників магазинів, які використовують цю систему. Завантажте Piedava, щоб виконати аналіз даних і відобразити результати статистичного аналізу на сторінці.

На титульній сторінці відображаються категорії, для яких доступна статистика. Натиснувши на вкладку з потрібною категорією, ви повернетесь до сторінки, яка підсумовує огляд і аналіз цієї категорії. Нижче наведено інформацію про виробника та сертифікацію. Скріншот головного вікна веб-додатку показано на (рис 2.5.)

Головна сторінка сайту виглядає так: Графічна схема головної сторінки веб-додатку представлено на (рис. 2.6).



Рисунок 2.5 Зображення головного вікна Piedava

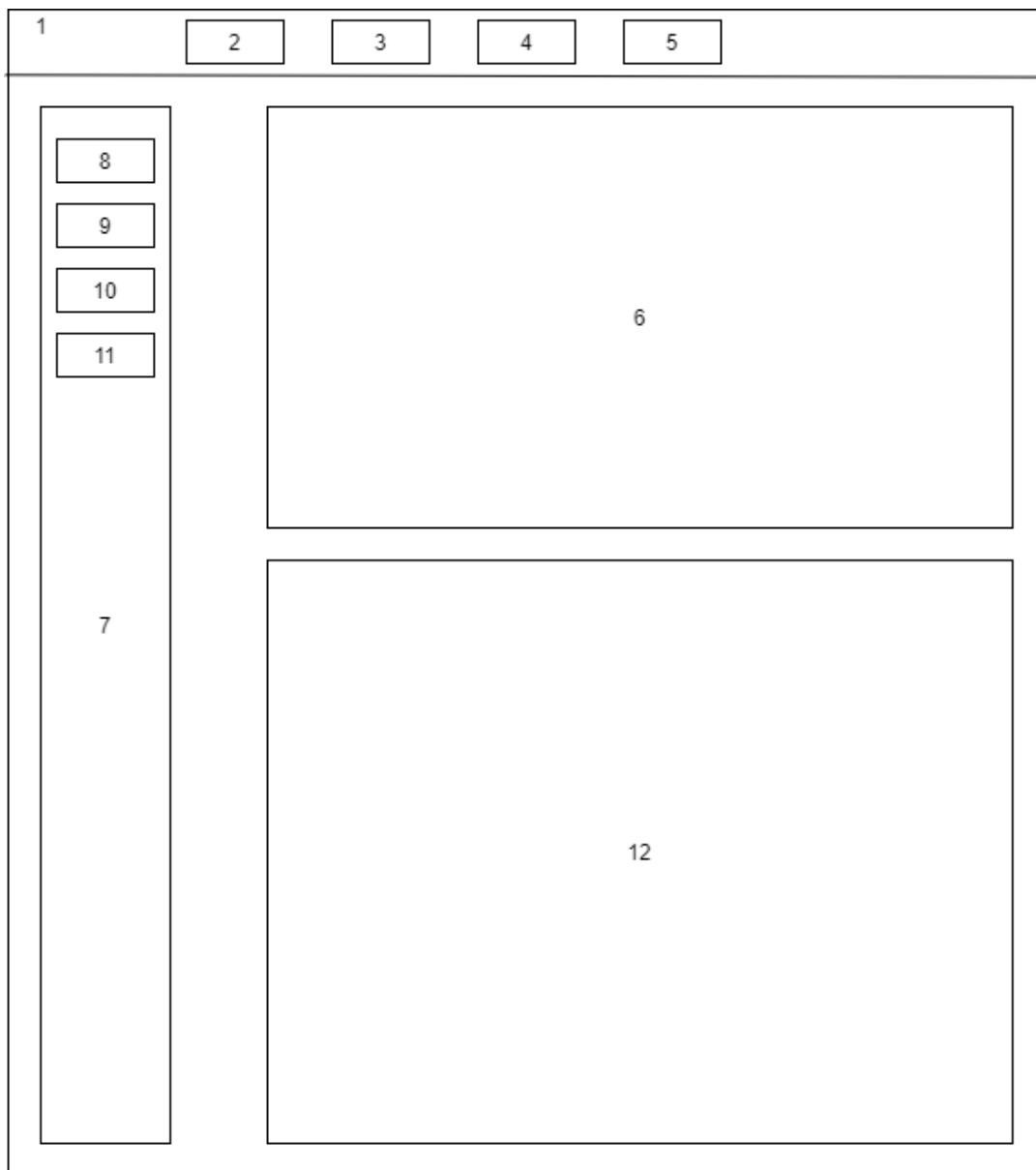


Рисунок 2.6 – Графічна схема головної сторінки веб-додатку

Основні елементи інтерфейсу головної сторінки веб-додатку:

1. Заголовок.
  - Посилання на головну сторінку.
  - Посилання на статистики відвідувань.
  - Посилання на статистики товарів.
  - Посилання на статистику клієнтів.
  - Загальна інформація про додаток.
  - Бокова панель навігації.



- Перехід на головну сторінку.
- Перехід до аналізу відвідувань.
- Перехід до аналізу товарів.
- Перехід до аналізу клієнтів.
- Логотип додатку.

## 2.4 Розробка алгоритмів роботи додатку

Загальний алгоритм роботи Веб-орієнтований програмний засіб для інтелектуального аналізу потреб споживача інтернет-магазину з метою підвищення продаж на (рис 2.7.)

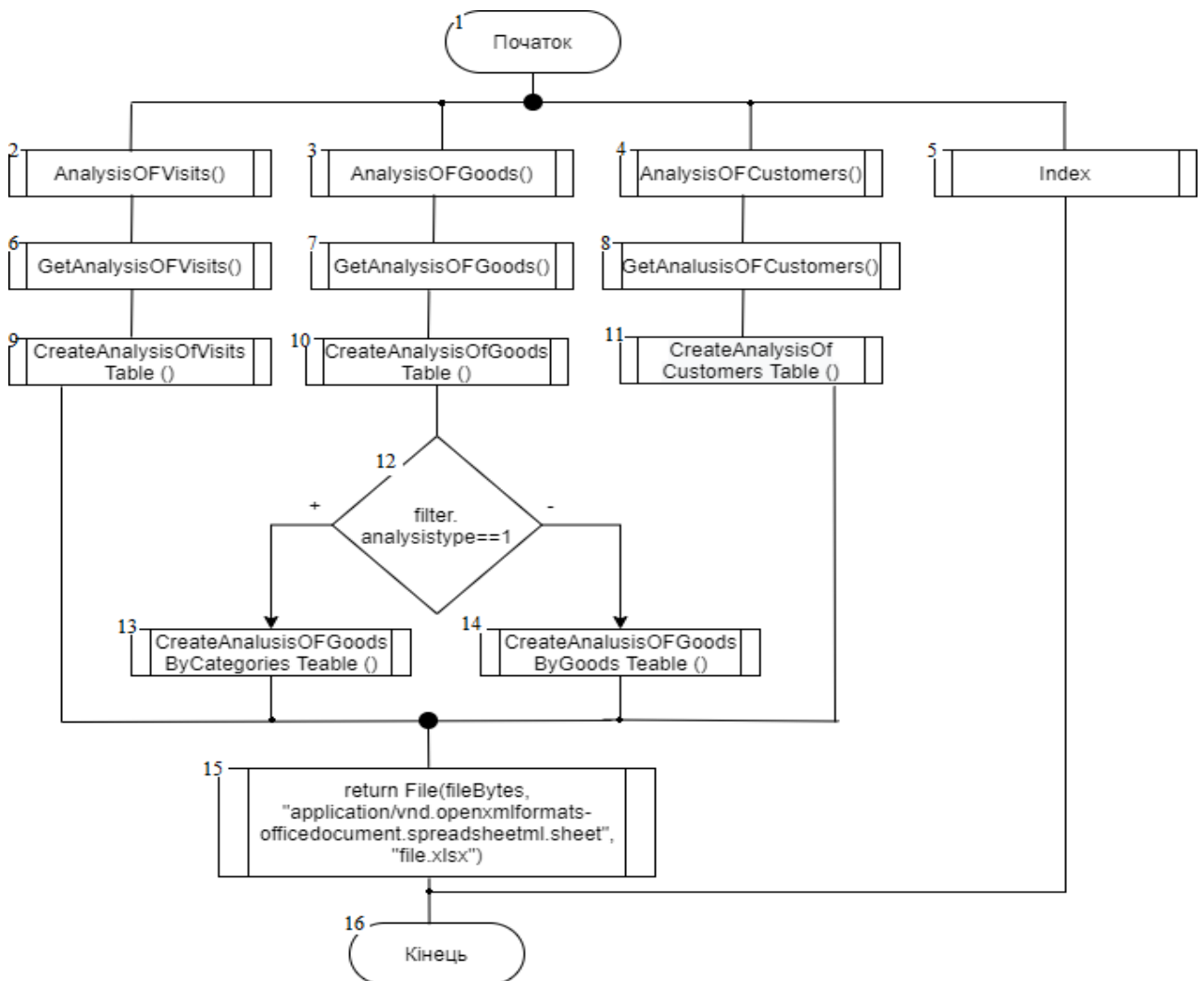


Рисунок 2.7 Загальний алгоритм роботи додатку

Робота загального алгоритму додатку полягає у виконанні таких кроків:

- вхідні дані можуть бути отримані з трьох джерел: аналіз відвідувань сайту, аналіз товарів або аналіз клієнтів;
- вибір параметрів в головному меню;
- отримання інформації від віддаленого серверу по запити;
- при потребі користувача завантажити аналіз по продуктах, то в нього на вибір декілька варіантів завантажити аналіз по категорії товарів, чи по конкретному товару;
- витяг з бази даних, результату аналізу;
- створення таблиці в форматі XLSX (алгоритм створення таблиці описана на рисунку 2.8);
- відправка до клієнтського модуля сформованої таблиці за заданими користувацькими параметрами.

Алгоритм запису даних у таблицю Excel проілюстровано на рисунку 2.7. На момент формування таблиці, аналіз потреб відвідувачів інтернет-магазину проведено, а таблиця є місцем зберігання результатів обробки даних.

Робота алгоритму передачі даних в XLSX-файл складається з таких кроків:

- зчитування інформації з БД, розташування яких знаходиться на серверах інтернет-магазину;
- створення першого набору з бази даних;
- цикл для фільтрації даних;
- запис даних до таблиці;
- перевірка, на завершення циклу;
- створення таблиці.

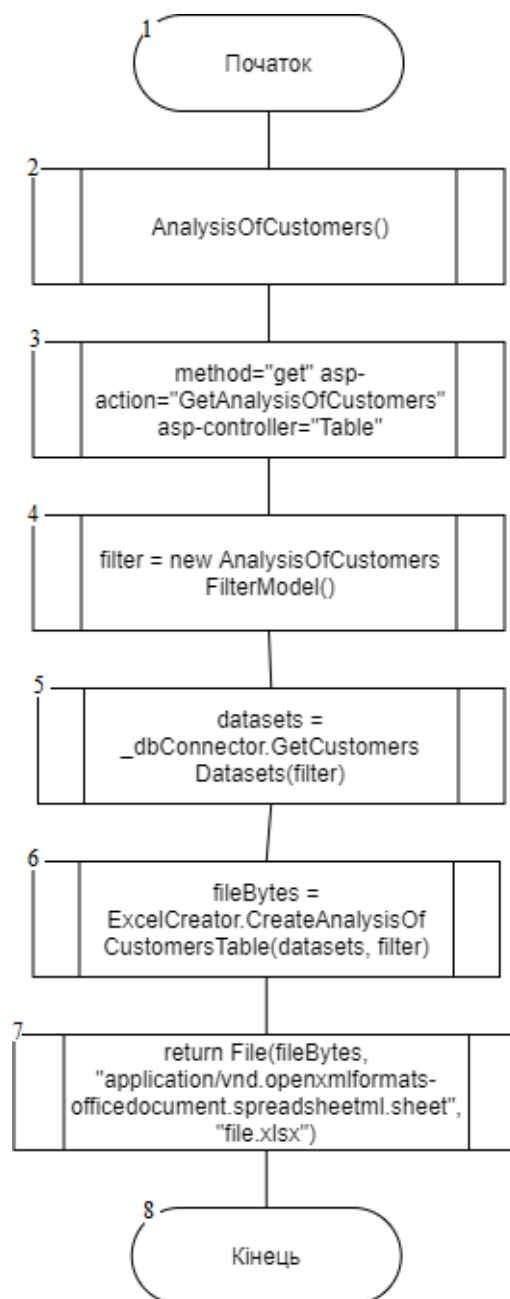


Рисунок 2.8 – Блок схема запису даних в XLSX-файл

Алгоритм аналізу відвідувачів опрацьовує дані про перегляди зареєстрованих користувачів (рис. 2.9.). Робота алгоритму аналізу користувачів складається з таких кроків.

- застосунок перевіряє який чи є товар, обраний для перевірки останнім;
- при відсутності позитивного результату обирається продукт для перевірки

кількості переглядів цього продукту обраним користувачем;

— відношення переглядів обраного продукту, до медіани переглядів товарів в інтернет магазині – саме таким чином і відбувається перевірка;

— для визначення інтересу відвідувача до обраного товару використовується відношення кількості переглядів обраного товару до середньої кількості переглядів товару відвідувачем (відношення є динамічним, для окремо взятого продукту ця пропорція відрізняється, або для категорії продуктів, граничне значення пропорції визначається працівниками інтернет-магазину або власником);

— визначення зацікавленості відвідувача до обраного товару;

— ситуація при якій кількість переглядів конкретного продукту перевищує середню кількість переглядів в  $G$  разів, такий користувач потрапляє до категорії користувачів, які вже готові придбати продукт, йому варто рекомендувати знижки на обраний продукт, з метою підвищення вірогідності на придбання саме в даній крамниці.

— якщо кількість переглядів обраного продукту приблизно по пропорції до медіани переглядів рівна  $G$ , в такому випадку користувач відноситься до категорії користувачів, які зацікавлені даним продуктом, але ще не обрали категорично, такий користувач не вважається зацікавленим в даному продукті, але перевірка вже не продовжується, оскільки є лише три параметри, і проводити додаткову перевірку немає потреби.

— перевірка чи був продукт крайнім, якщо так, то додаткових перевірок не проводиться.

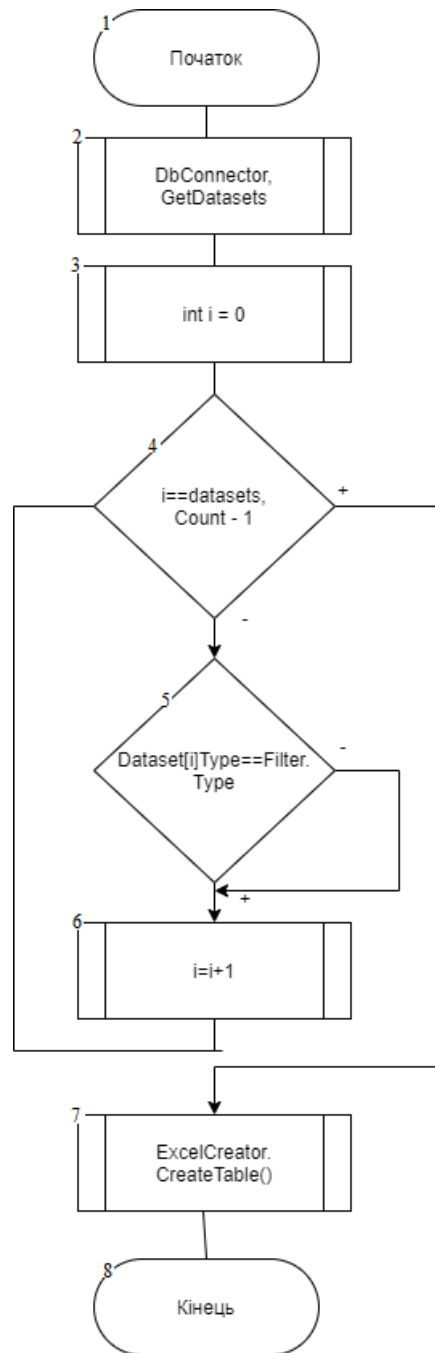


Рисунок 2.9 – Алгоритм аналізу зареєстрованих користувачів.

## 2.5 Висновки

В даному розділі був зроблений аналіз даних, також розроблені гістограми, алгоритми для веб-орієнтованого застосування для аналізу даних користувачів інтернет магазину з метою підвищення продаж.

### 3 РОЗРОБКА ВЕБ-ОРІЄНТОВАНОГО ПРОГРАМНОГО ЗАСОБУ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ПОТРЕБ СПОЖИВАЧА ІНТЕРНЕТ МАГАЗИНУ

3.1 Варіативний аналіз та обґрунтування виборів засобів для реалізації додатку

Вибір інструментів є важливою частиною будь-якого проекту програмного забезпечення, оскільки правильні методи виробництва та інструменти спрощують процес розробки.

Обговорюється ООП-сумісна методологія розробки для розробки інструменту для аналізу потреб користувачів онлайн-магазину для покращення продажів.

Найбільш популярними мовами програмування, які підтримують цю модель, є C++, C#, Java і Python. Розглянемо переваги та недоліки кожного.

C++ (рис. 3.1) — це мова програмування високого рівня, яка підтримує об'єктно-орієнтоване програмування, яке може використовувати об'єктно орієнтована модель програмування. [16].

Переваги:

- можливість написання програми, використовуючи як і ООП так і процедурний принцип програмування;
- завдяки статичній типізації, можна писати простий машинний код, що виключає багато помилок, під час компіляції;
- можливість використання простого середовища для програмування;
- стандартизація мови за міжнародним стандартом, поточна версія якого датована 2011 роком.

Недоліки:

— неймовірна повільна компіляція коду, під час використання великої кількості підключених модулів. Все це відбувається через препоцедурну вставку (`#include`), уникнути цього можна лише використовуючи різні середовища

— необхідна висока майстерність програміста, через свою відносно низькорівневість та відсутність збірок «сміття», щоб звільнити пам'ять.

— мова `c++` є складною для вивчення та компіляції.

```
#include "stdafx.h"
#include<iostream>
using namespace std;
int max(int x, int y)
{
    if (x > y)
        return x;
    else return y;
    if (x == y)
    {
        int sum = x + y;
        return sum; }
}

int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "Ukr" );
    int a, b, c;
    cout << "Ведіть a = " << endl;
    cin >> a;
    cout << "Ведіть b = " << endl;
    cin >> b;
    c = max(a, b);
    if(c == (a + b))
        cout << "Числа рівні"<< endl;
    else cout << "Максимальне число = " << c << endl;
    system("pause");
    return 0;
}
```

Рисунок 3.1 – Мова програмування C++

Python (рис 3.2) – Скриптова мова програмування, яка потребує віртуального середовища для своєї роботи. Також має власний інтерпретатор,

з відкритим вихідним кодом, та підтримкою об'єктно орієнтованого програмування. Завдяки інтерпретатору з відкритим кодом, написаним на мові програмування C, завдяки чому можна створювати нові модулі самостійно.

```
def add(request, slug):
    product_add = Products.objects.get(slug=slug)
    request.session.set_expiry(30)
    try:
        active = request.session['cart']
    except:
        request.session['cart'] = 'Empty'

    if request.session['cart'] != 'Empty':
        cart = request.session['cart']
        update_cart = Cart.objects.get(id=cart)
        update_cart.products.add(product_add)
        update_cart.save()
        request.session['total_items'] = len(update_cart.products.all())
    else:
        new_cart = Cart()
        new_cart.save()
        new_cart.products.add(product_add)
        request.session['cart'] = new_cart.id
        request.session['total_items'] = len(new_cart.products.all())

    return HttpResponseRedirect('/products/%s' % (slug))
```

Рисунок 3.2 – Мова програмування Python

Переваги:

- динамічна типізація дозволяє одній і тій самій змінній використовуватись по різному в різних частинах коду в залежності від ситуації;
- можливість написання власних модулів, написані під конкретне програмне забезпечення на мові C або C++;
- мова програмування Python, завдяки створенню віртуального середовища він є кросплатформним, що дозволяє використання на всіх існуючих операційних системах;
- завдяки можливості підключення модулів для мови програмування



Python, написання коду стає доволі простим та швидким, через використання вже готових рішень;

Недоліки:

— явна нестача продуктивності, через інтерпретатор та віртуальне середовище;

— динамічна типізація являє собою і основним недоліком даної мови програмування. Використання динамічної типізації може призвести до великої кількості багів, при найменшій помилці програміста та зменшується продуктивність компілятора.

Java (рис.3.3) – Мова програмування з підтримкою об'єктно орієнтованого програмування. В основному використовується для написання застосунків на Android, але в той же час є повністю кросплатформеною. Гаслом розробників під час розробки було WORA (Write once, run anywhere), тобто написав раз, запустив будь – де. Одна з C – подібних мов, яка перетворює код в байт-код, що і є головною специфікою компілятора даної мови програмування.

Java virtual machine – компілятор мови програмування Java, який і зумовлю можливість кросплатформеності. Завдяки цьому мова набрала популярності, оскільки немає необхідності писати програму для різних ОС.

Переваги:

— кросплатформеність – як і в мові програмування Python, Java дозволяє використання коду на різних ОС, що сильно зменшує витрати на написання коду. Досягнуто це завдяки Java virtual machine – особливому компілятору мови програмування Java;

— простота в освоєнні – Робота з мовою програмування Java ніяким чином не вимагає детального вивчення та досвіду програмування. Для роботи необхідно знати синтаксис та базове розуміння об'єктно

орієнтованого програмування. Також, оскільки С-подібний синтаксис, то перехід з іншої мови програмування буде безболісним та відносно простим;

— продуктивність – хоча це і є мова програмування з власним компілятором, але серед подібних мов, Java виділяється швидкістю, за рахунок оптимізації.

Недоліки:

— продуктивність – хоч вище і було описано про продуктивність як перевагу, але вона продуктивна відносно компіляторних мов, але все одно використовує віртуальне середовище, що сильно сповільнює її по відношенню до тієї самої С або С++

— відсутність нативного дизайну – сама по собі мова програмування не надає ніяких можливостей в написанні та створенні графічних інтерфейсів для розробника. Для створення користувацьких інтерфейсів необхідно використовувати сторонні шаблони або ж створювати. Даний недолік накладає певні обмеження на проекти, розроблені на даній мові програмування, такі як додаткові витрати та необхідність додаткових знань для розробника

```

import java.util.*;
import javax.swing.*;

public class Main {

    public static void main(String[] args) {

        int chislo= (int)(Math.random()*20)+1; //задаємо випадкове число
        //System.out.println(chislo);
        int N =-1;
        int lic=0;
        String S="Введіть число";
        do {
            String in = JOptionPane.showInputDialog(null, S);
            N = Integer.parseInt(in);
            if (N>chislo)S="Число менше, спробуй ще раз. Введіть число";
            if (N<chislo)S="Число Більше, спробуй ще раз. Введіть число";
            lic++;
        } while (chislo!=N);
        System.out.println("Ви вгадали число "+N+" за "+lic+" раз" );
    }
}

```

Рисунок 3.3 – Мова програмування Java

C# (рис. 3.4) – Одна з найпопулярніших сучасних мов програмування. Об'єктно орієнтована мова програмування, зі строгою типізацією даних. Дана мова програмування також має власну віртуальну машину .NET, яке викликає загально мовне середовище виконання (CLR). Також дана мова програмування пропонує багато різних мовних конструкцій та бібліотек. Що наймовірно спрощує написання різних компонентів застосунку, які між собою взаємодіють. Також, оскільки мова програмування одна з найпопулярніших, то дуже багато створених бібліотек з великою кількістю інформації в інтернеті.

Переваги:

—об'єктно орієнтоване програмування – мова програмування C# спеціально була написана для розробки с використанням об'єктно орієнтованого програмування, тому основні функції та можливості можна завдяки розумінню патернів та принципів.;

—готові конструкції та бібліотеки – основною перевагою C# є можливість встановлювати чисельну кількість бібліотек, або ж писати їх власноруч. Бібліотеки дозволяють зменшити кількість коду та час, на його написання, оскільки для використання більшості функції необхідне просто їх розуміння, вони вже написані;

—ЛТ – проміжна мова програмування (ЛТ). Написаний код перетворюється на машинний проміжною мовою програмування, що сильно підвищує продуктивність та швидкість роботи коду та компіляції;

Недоліки:

—C# - має певні проблеми з безпекою, оскільки код, написаний мовою C# дуже легко дизамблюється, тобто відносно легко отримати вихідний код програми зловмисниками. Звісно існує багато засобів протидії та усуненню даного недоліку, але це вимагає додаткових заходів безпеки та повністю виключити дану вразливість майже неможливо;

—C# - є повністю Windows – орієнтована мова програмування, оскільки випускається під логотипом однойменної корпорації. Код написаний даною мовою програмування майже неможливо імпортувати на іншу операційну систему. Хоча є фреймворк .NET.CORE, що частково вирішує дану проблему, але лиш частково, імпорт все одно залишається доволі ресурсозатратною справою та необхідність мати спеціальні навички.

```

1 using System;
2
3 namespace Циклы
4 {
5     class MainClass
6     {
7         public static void Main (string[] args)
8         {
9             Console.WriteLine ("Даров");
10
11
12             string str;
13             do {
14                 Console.WriteLine ("Введите 'Выход' чтобы выйти из программы");
15                 str = Console.ReadLine ();
16             } while (str != "Выход") ;
17         }
18     }
19 }
20

```

Рисунок 3.4 – Мова програмування С#

Оскільки С# є дуже популярною мовою програмування, а для ОС Windows являється найпопулярнішою, то існує неймовірна велика кількість середовищ розробки для даної мови програмування.

Visual Studio – середовище, розроблене корпорацією Microsoft спеціально для даної мови програмування. Для порівняння з даним середовищем розробки було обрано таких конкурентів Project Rider, Eclipse та Code::Blocks

Visual Studio (рис. 3.5) – Середовище для розробки, розроблене корпорацією Microsoft. Підтримує наступні технології:

- Visual Basic .NET, а до його появи — Visual Basic
- Visual C++
- Visual C#
- Visual F# (входить до складу Visual Studio 2010);
- Visual Studio Debugger

### Переваги:

— Веб-сервер – для обслуговування веб-додатків та сторінок, є можливість створення локального веб-серверу для тестування продукту без розгортання на віддалений сервер. При створенні проекту через ASP.NET, Visual Studio тестувати сайт через локально.

— Мови програмування – хоч дане середовище було спеціально розроблене для мови програмування C#, але використання інших мов програмування також підтримується. Дана функція в першу дозволяє створювати великі проекти різними командами з одним і тим самим інтерфейсом та ділитися проектами.

— Форматування – код, написаний в даному середовищі форматується по мірі його введення, що робить дане середовище дуже зручне для розробника.

### Недоліки:

— Перехід з пробної версії на повноцінну може призвести до несумісності деяких компонентів застосунку.

— Відладчик Visual Studio неможливість працювати в режимі кода ядра

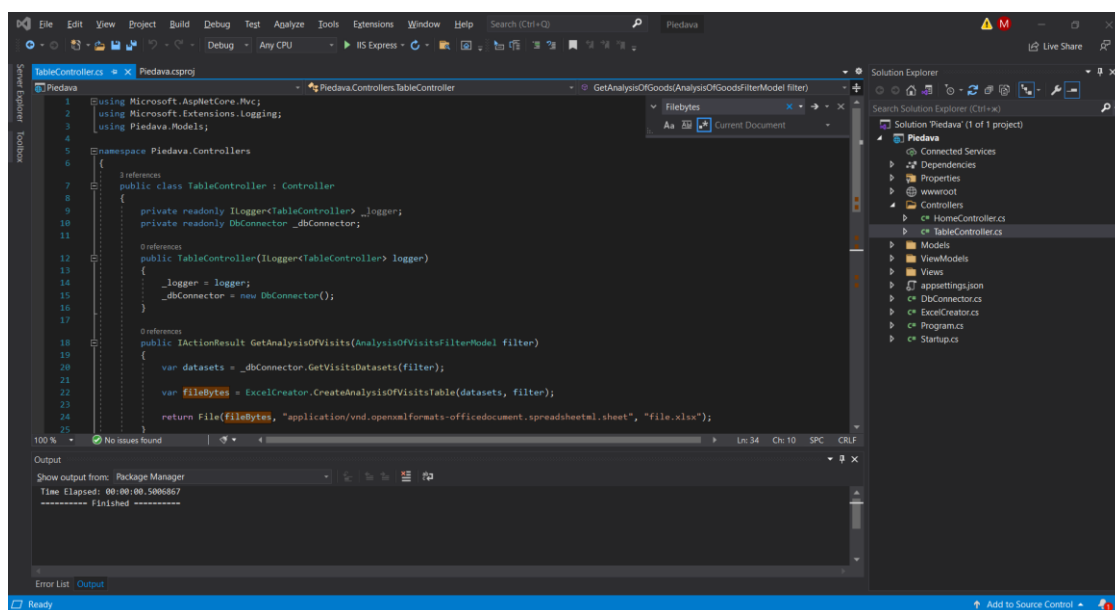


Рисунок 3.5 Інтерфейс Visual Studio

Project Rider (рис 3.6) – середовище розробки, основною перевагою якого являється кросплатформеність, яка була розроблена корпорацією JetBrains, яка була побудована IntelliJ. Перевагами даної IDE можна назвати:

- Зручна віртуальна файлова система
- Модуль користувацького інтерфейсу
- Текстовий редактор
- Підтримку синтаксиса мови програмування
- Компоненти для роботи з кодом
- Функцію контролю версій
- Тестувальник з відладчиком

Переваги:

- Кросплатформеність – на даний момент дане середовище має головну перевагу на Visual Studio це кросплатформеність
- Інтеграція з Visual Studio та Unity
- Неймовірна зручна навігація та можливість дуже просто рефакторинга
- Підтримка найпопулярніших мов програмування, таких як: C#, VB, XAML, HTML, JavaScript, TS

Недоліки:

- Оскільки на даний момент, даний продукт знаходиться в бета версії, то можливі непередбачувані баги та помилки
- Доволі дороге програмне забезпечення, але є знижки для студентів та учбових закладів

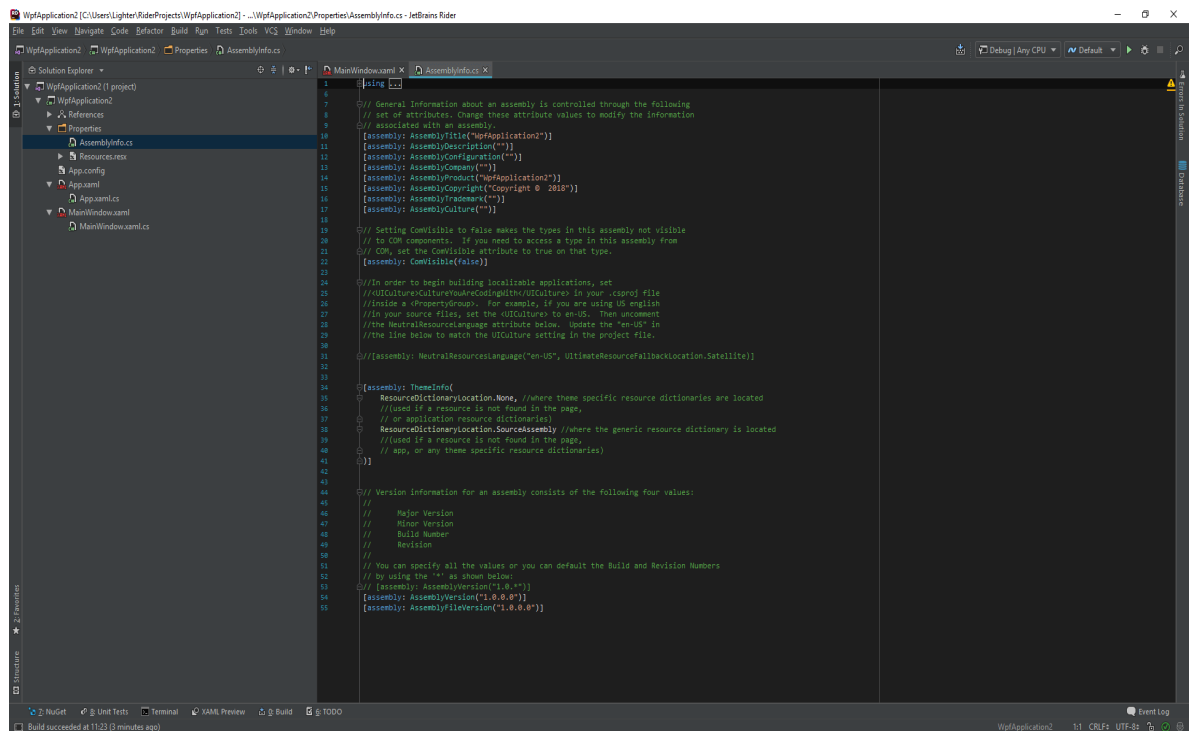


Рисунок 3.6 – Інтерфейс Project Rider

Eclipse IDE(рис. 3.7) – Середовище програмування з відкритим кодом, яке було написано на мові програмування Java. Підтримується та розробляється компанією Eclipse Foundation.

#### Переваги:

— Оскільки дане програмне забезпечення являється безплатним та доволі популярним з відкритим вихідним кодом, то є велика кількість користувацьких модулів так і від самих розробників.

— Один з найшвидший компіляторів, добилися такого результату за рахунок неймовірної простоти

— Один з найкращих відладчиків.

— Повна кастомізація

#### Недоліки:

— Оскільки продукт безплатний, то окремі баги в системі можуть виправлятися відносно довго, що може призводити до цікавих неочікуваних багів та помилок.



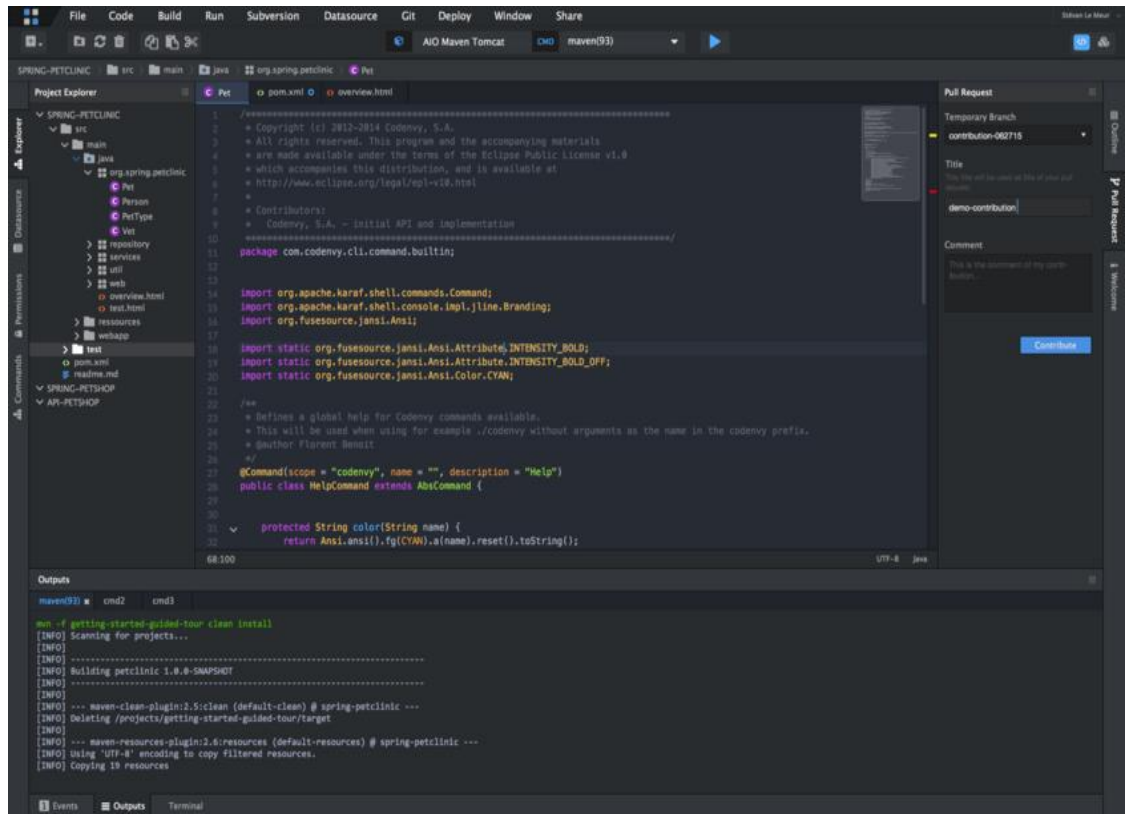


Рисунок 3.7 Інтерфейс середовища розробки Eclipse

Code::Blocks (рис 3.8) – Ще одне середовище розробки написане на мові програмування C++. Дуже схоже по принципах роботи з Eclipse, оскільки розробники обрали для себе схожий шлях з відкритим кодом та можливістю написання користувацьких модулів. Також є кросплатформним середовищем.

Переваги:

- Code::Blocks – являє собою дуже просту концептуально та підходить для навчання, оскільки немає лишнього навантаження в інтерфейсі
- кросплатформність
- модулі – Так же і Eclipse, є можливість написання модулів та використання інших модулів написаних користувачами. Також є можливість покупки модулів від розробників.

Недоліки:

— для розробки великих проектів Code::Blocks не має достатнього функціоналу.

— де має можливості великі проекти розробляти, використовуючи дане середовище розробки, оскільки функціонал доволі обмежений, навіть з використанням самописних модулів.

— Code::Blocks – дуже нестабільне середовище, використання якого може погано повпливати на проект, через неочікувані баги та помилки.

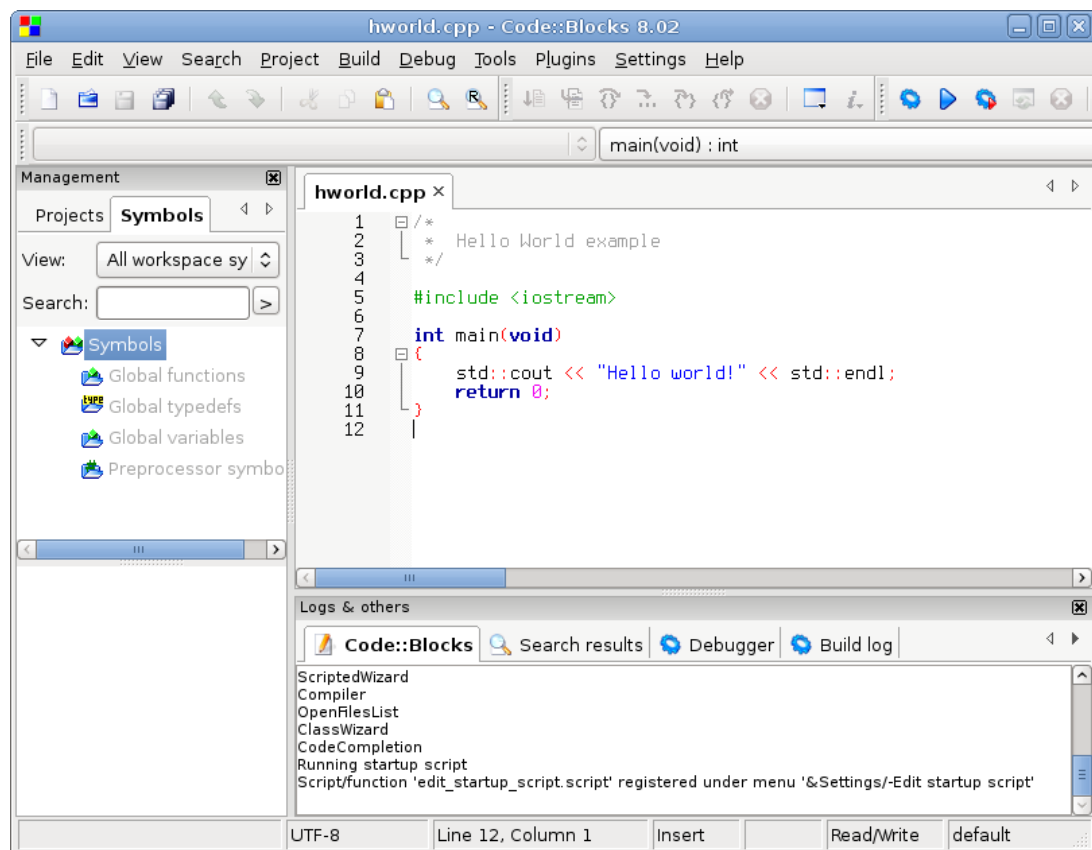


Рисунок 3.8 –Інтерфейсу Code::Blocks

Після аналізу всіх переваг та недоліків описаних вище IDE. Було прийнято рішення розробляти веб-орієнтований програмний застосунок для аналізу потреб споживача інтернет магазину з метою підвищення продаж в середовищі Visual Studio.

### 3.2 Вибір архітектурного рішення

Додаток Piedava було реалізовано використовуючи архітектурний патерн Model-View-Controller (рис 3.9) .

Model-View-Controller – архітектурний патерн, принцип роботи якого заключається в розробці трьох модулів, які і слугують інтерфейсом користувача та виконуючою частиною системи. Основною перевагою даного патерну є те, що зміни в інтерфейсі або в виконавчій частині ніяким не впливають один на одного через, те що керуються вони третім модулем.[17]

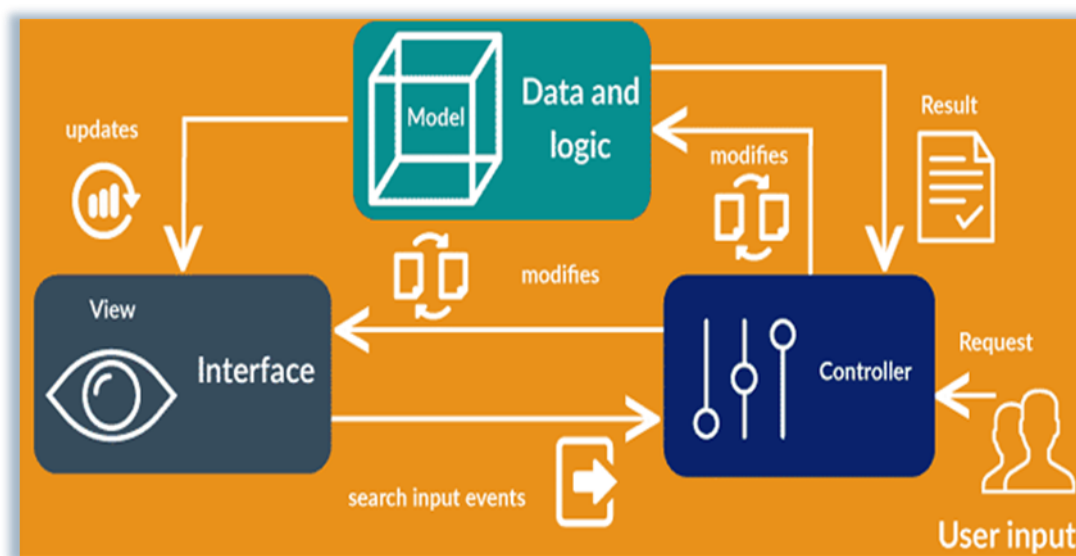


Рисунок 3.9 –Патерн Model-View-Controller

Ілюстрація зображена на (рис. 3.9) показує взаємодію між модулями. Користувач звертається до контролера, той в свою чергу отримує дані з моделі, і вже модель звертається знову до контролера, щоб він віддав дані назад користувачеві.

IDE Visual Studio надає можливість створювати проект використовуючи даний патерн з запропонованих шаблонів, що і було використано при розробці даного застосунку.

### 3.3 Розробка компонента View

Модуль View, отримує інформацію одразу від користувача, в даному випадку користувачем виступає інтернет магазин, який хоче отримати запит. Саме під час розробки даного модуля створюється графічний інтерфейс, файлі даного модуля мають формат гіпертекстової розмітки. (рис. 3.10). Під час розробки даного модуля був використаний шаблон середовища розробки Visual Studio

```

10 <p>Створення статистики покупок та переглядів товарів групами користувачів.</p>
11
12 <hr />
13 </div>
14
15 <div class="border border-info rounded p-3 mb-3" style="background-color: #f0fffe;">
16 <h3>Налаштування</h3>
17
18 <div>
19 <h5>Оберіть період (включно):</h5>
20
21 <div class="form-inline">
22 <span class="m-2">Від:</span>
23 <select asp-items="ViewBag.Years" asp-for="FromYear" class="form-control m-2"></select>
24 <span class="m-2">До:</span>
25 <select asp-items="ViewBag.Years" asp-for="ToYear" class="form-control m-2"></select>
26
27 </div>
28
29 <div class="mt-3">
30 <h5 class="mb-3">Оберіть категорію:</h5>
31
32 <select asp-items="ViewBag.Categories" asp-for="CategoryId" class="form-control m-2"></select>
33
34 <hr />
35 </div>
36
37 </div>
38 </div>
39
40 <div>
41 <input type="submit" class="btn btn-info btn-lg" value="Завантажити звіт (Файл .XLSX)" />
42 </div>

```

Рисунок 3.10 –Компонента View AnalysisOFCustomers

Роботою даного модуля є сторінка з аналізом клієнтських даних. При натисканні же клавіші «Завантажити звіт» в контролер передається введені користувачем параметри класом `class="btn btn-info btn-lg"` `value="Завантажити звіт (Файл .XLSX)"`

### 3.3 Розробка компонентів

Controller (рис.3.11) отримує дані від View після чого виконує функцію `dbConnector.GetGoodsDatasets(filter)`, вона же фільтрує дані з БД по обраним раніше фільтрам. Потім виконується функція `ExcelCreator.CreateAnalysisOfCustomersTable(datasets, filter)`. Функція `fileBytes`. Функція `fileBytes` вже викликає компонент Model.

```

0 references
public IActionResult GetAnalysisOfVisits(AnalysisOfVisitsFilterModel filter)
{
    var datasets = _dbConnector.GetVisitsDatasets(filter);

    var fileBytes = ExcelCreator.CreateAnalysisOfVisitsTable(datasets, filter);

    return File(fileBytes, "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "file.xlsx");
}

0 references
public IActionResult GetAnalysisOfGoods(AnalysisOfGoodsFilterModel filter)
{
    var datasets = _dbConnector.GetGoodsDatasets(filter);

    var fileBytes = ExcelCreator.CreateAnalysisOfGoodsTable(datasets, filter);

    return File(fileBytes, "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "file.xlsx");
}

0 references
public IActionResult GetAnalysisOfCustomers(AnalysisOfCustomersFilterModel filter)
{
    var datasets = _dbConnector.GetCustomersDatasets(filter);

    var fileBytes = ExcelCreator.CreateAnalysisOfCustomersTable(datasets, filter);

    return File(fileBytes, "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "file.xlsx");
}

```

Рисунок 3.11 – Зображення компонента Controller TableControllers

Компонент Model в свою чергу викликає клас під назвою `DbConnector` (рис. 3.12), який отримавши фільтри від Controller витягує з бази даних потрібні аналітичні дані. Таким чином і відбувається безпосередньо витяг з даних користувачем, для подальшого використання.

```
1  using Piedava.Models;
2  using System.Collections.Generic;
3  using System.Linq;
4
5  namespace Piedava
6  {
7      public class DbConnector
8      {
9          public List<IdNameModel> GetCategories()
10         {
11             return _categories;
12         }
13
14         public List<IdNameModel> GetYears()
15         {
16             return _years;
17         }
18
19         public List<IdNameModel> GetMonths()
20         {
21             return _months;
22         }
23     }
24 }
```

Рисунок 3.12 – Зображення класу DbConnector

### 3.5 Висновки

В даному розділі був описаний принцип вибору мови програмування, середовища розробки та патерна розробки. Всі ці параметри було між собою порівняно та на основі порівняння було обрано мову програмування середовище розробки та архітектурний патерн для розробки веб-орієнтованого програмного застосунку для аналізу даних користувачів інтернет магазину з метою підвищення продаж.

## 4. ТЕСТУВАННЯ ПРОГРАМИ

### 4.1 Методи тестування

Існує три основні підходи до тестування програмного забезпечення: тестування білого ящика, тестування чорного ящика та тестування сірого ящика.

Тестування білої скриньки – це метод тестування, який виконується після того, як особа, відповідальна за тестування, дізнається про внутрішній стан системи, конструкцію, роботу тощо. Тестери вибирають вхідні значення на основі свого знання коду та розуміння внутрішніх процесів, що відбуваються під час тестування. Отже, експериментатор також знає результати досліду. Основа тестування білого ящика полягає в тому, що тестувальник знає всі характеристики функції, що тестується. Таким чином, тестування білого ящика проникає глибоко всередину системи [18].

Тестування чорної скриньки визначається як метод тестування, який перевіряє продуктивність перевіреної програми без урахування внутрішньої структури коду, даних реалізації або знання внутрішніх шляхів програмного забезпечення. Цей тип тестування повністю залежить від вимог і специфікацій програмного забезпечення. Тестування чорної скриньки ігнорує внутрішні знання програми та зосереджується лише на входах і виходах програмного забезпечення. Тестування чорного ящика — це активне і пасивне тестування без доступу до внутрішньої структури компонентів системи. Методи тестування чорної скриньки — це процедури отримання та відбору тестових випадків на основі аналізу зразків (функціональних чи нефункціональних).

Тестування Graybox — це метод тестування частини програмного забезпечення або програми з частковим знанням її внутрішньої структури. Тестування сірого ящика не вимагає від тестувальника перегляду вихідного

коду. Тести пишуться на основі знань про алгоритми, архітектуру, внутрішні параметри та інші складні описи поведінки програми. Завдяки цьому методу тестувальники мають доступ до документації методу та можуть підготувати та виконати більш точні та повні тести та тестові випадки. Переваги використання «білого ящика» найбільш очевидні, коли йдеться про веб-додатки, веб-сервери, безпеку, графічний інтерфейс користувача та тестування продуктивності.

#### 4.2 Тестування веб-додатку

Тестування веб-застосунку — це тестування набору програм і перевірка їх функціональності. Тестування проводиться для оцінки продуктивності веб-додатків і виявлення помилок, які пізніше виправляються. Це пов'язано з тим, що якщо помилки в програмі не будуть виявлені вчасно, розробник втратить багато грошей. Зараз на тестування виділяється близько 60% бюджету розробки [19].

Для підтвердження роботи веб-додатку Piedava було розроблено такі тестові випадки(табл.4.1).

Таблиця 4.1. Методика тестування додатку

Ідентифікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-1	Загальна перевірка роботи	1. Відкрити застосунок	1 Відкриється головне вікно Piedava з описом додатком та логотипом	Виконано



Продовження таблиці 4.1

Ідентифікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-2	Перевірка посилання на головну сторінку	1 Перейти на любую іншу сторінку веб-додатку 2 Натиснути на назву додатку	Користувача переадресує на головну сторінку	Виконано
ТВ-3	Перевірка адаптивності веб-додатку	1.Змінити масштаб браузера	1 Веб-додаток повинен змінювати масштаб разом із браузером	Виконано
ТВ-4	Перевірка завантаження XLSX	1 Вибрати в заголовку потрібний аналіз 2 Вибрати параметри завантаження 3 Натиснути клавішу «Завантажити XLSX-файл»	1 XLSX – файл повинен завантажитись до стандартної папки завантажень	Виконано

Продовження таблиці 4.1

Ідентифікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-5	Перевірка можливості зміни даних для запису	1 Вибрати в заголовку потрібний аналіз 2 Вибрати потрібні параметри 3 Натиснути клавішу завантажити файл	1 Дати повинні співпадати	Виконано
ТВ-6	Перевірка правильності запису даних в таблицю	1 Вибрати в заголовку потрібний аналіз 2 Вибрати потрібні параметри 3 Натиснути клавішу завантажити файл	1 Таблиця повинна бути заповнена коректно без пропусків 2 Кожен часовий проміжок повинен відображати дані	Виконано

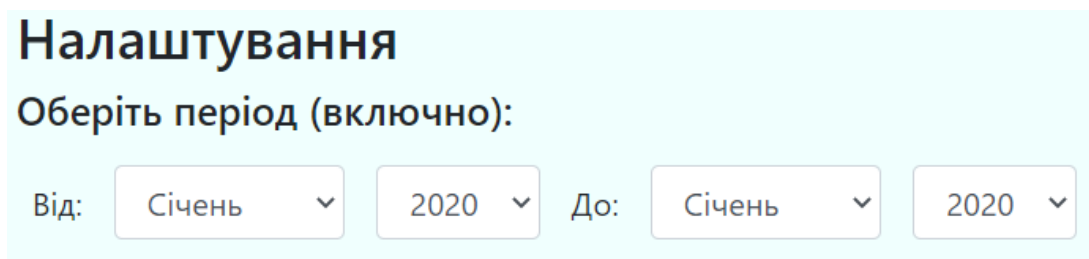
Продовження таблиці 4.1

Ідентифікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-7	Перевірка запису даних в таблицю	1 Вибрати в заголовку потрібний аналіз 2 Вибрати потрібні параметри 3 Натиснути клавішу завантажити файл	1 Файл повинен завантажитись на пристрій 2 Файл не повинен бути пустий	Виконано
ТВ – 8	Перевірка Аналізу товарів	1 Вибрати в заголовку аналіз товарів 2 Вибрати потрібні параметри 3 Натиснути клавішу завантажити файл	1 Файл XLSX повинен завантажитись 2 Файл повинен містити аналітичні дані товарів в вибраний період	Виконано

Продовження таблиці 4.1

Ідентифікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-9	Перевірка Аналізу відвідувань	1 Вибрати в заголовку аналіз відвідувань 2 Вибрати потрібні параметри 3 Натиснути клавішу завантажити файл	1 Файл XLSX повинен завантажитись 2 Файл повинен містити аналітичні дані відвідувань сторінки в вибраній період	Виконано
ТВ-10	Перевірка Аналізу клієнтів	1 Вибрати в заголовку аналіз клієнтів 2 Вибрати потрібні параметри 3 Натиснути клавішу завантажити файл	1 Файл XLSX повинен завантажитись 2 Файл повинен містити аналітичні дані клієнтів сторінки в вибраній період	Виконано

Веб-додаток Piedava надає можливість працівникам інтернет магазину завантажувати аналітичні дані у різні періоди, які вибирає сам працівник. Вибір часового періоду зображено на рисунку 4.1



**Налаштування**  
Оберіть період (включно):

Від:   До:

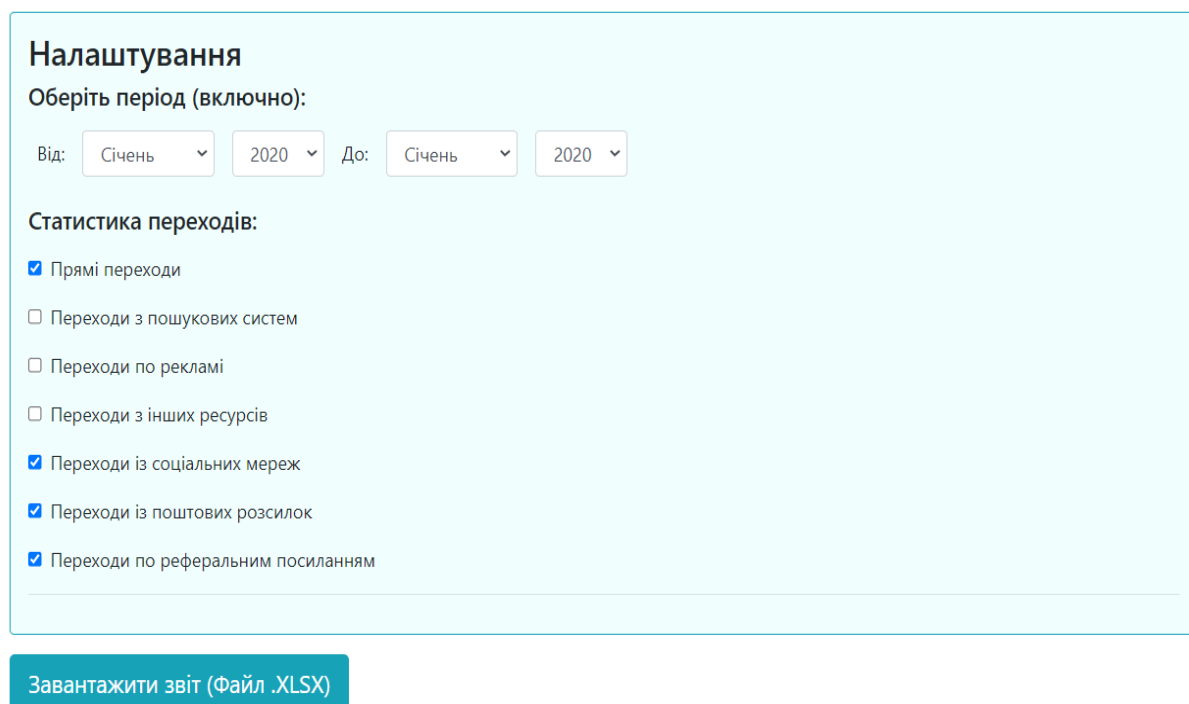
Рисунок 4.1 – Вибір часового періоду, за який буде завантаження

При завантаженні аналізу відвідувань сайту працівник магазину, може обрати, які саме джерела його цікавлять(рис.4.2).

## Аналіз відвідувань сайту

Створення статистики відвідувань та переходів на інтернет-ресурс

За допомогою налаштувань оберіть період та пункти, за якими буде отримана статистика.



**Налаштування**  
Оберіть період (включно):

Від:   До:

**Статистика переходів:**

- Прямі переходи
- Переходи з пошукових систем
- Переходи по рекламі
- Переходи з інших ресурсів
- Переходи із соціальних мереж
- Переходи із поштових розсилок
- Переходи по реферальним посиланням

[Завантажити звіт \(Файл .XLSX\)](#)

Рисунок 4.2 - Вибір джерел відвідувань для завантаження

Результатом роботи натискання клавіші «Завантажити звіт (Файл.XLSX) з заданими параметрами зображено на рисунку 4.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Вид переходів	Січень 2020											
2	Прямі переходи	5241											
3	Переходи із соціальних мереж	1121											
4	Переходи із поштових розсилок	15											
5	Переходи по реферальним посиланням	582											
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													

Рисунок 4.3 – Створена таблиця Excel

Додаток Piedava надає аналіз по товарам конкретної категорії, або аналіз по всім категоріям товару представленим в магазині(рис.4.4).

## Аналіз товарів

Створення статистики переглядів та покупок категорій та конкретних одиниць товарів.

### Налаштування

Оберіть період (включно):

Від:   До:

Створення статистики на основі:

Категорій товарів

Товарів категорії:

Завантажити звіт (Файл .XLSX)

Рисунок 4.4 – Вибір які аналітичні дані записувати в таблицю  
 На рисунку 4.5 зображено аналітичні дані по категоріям товарів

The screenshot shows an Excel spreadsheet with the following data:

Категорія	Січень 2020	Лютий 2020	Березень 2020	Квітень 2020	Травень 2020	Червень 2020	Липень 2020
Ноутбуки та комп'ютери	1275	1531	1356	1406	1359	1198	1644
Смартфони, ТВ та електроніка	2414	2462	2560	2558	2301	2399	2444
Товари для геймерів	355	314	315	334	303	296	336

Рисунок 4.5 – Аналітичні дані по категоріям товарів

На рисунку 4.6 зображено аналітичні дані по конкретній категорії

The screenshot shows an Excel spreadsheet with the following data:

Apple iPhone SE 2020	233	239	292	243	201	286	252
OnePlus Nord	242	269	298	294	205	206	205
Samsung Galaxy M51	268	298	213	253	206	247	298
Realme 7 Pro	204	217	244	276	243	233	220
Apple iPhone 12 mini	289	223	209	263	204	249	215
Samsung Galaxy Note20 Ultra	264	242	279	296	225	221	300
Sony KDL-43WF804	222	204	228	217	220	223	243
LG 43LM5700	230	244	221	242	267	229	260

Рисунок 4.6 - Аналітичні дані по конкретній категорії

### 4.3 Розробка інструкції користувача

Першим етапом розробки інструкції користувача є визначення технічних вимог web додатку. У таблицях 4.2 та 4.3 наведено мінімальну та рекомендовану конфігурацію персонального комп'ютера для запуску програми[20].

Таблиця 4.2 – Мінімальна конфігурація.

Тип процесора	32-розрядний (x86) або 64-розрядний (x64) процесор з тактовою частотою 2 ГГц
Об'єм оперативної пам'яті	4 ГБ для 32-розрядної системи і 8 ГБ для 64-розрядної системи
Місце на жорсткому диску	Оскільки додаток Piedava є веб-ресурсом, пам'ять для використання не потрібна
Графічний пристрій	Інтегрований графічний пристрій
Операційна система	Windows 10

Таблиця 4.3 – Рекомендована конфігурація.

Тип процесора	32-розрядний (x86) або 64-розрядний (x64) процесор з тактовою частотою 3 ГГц
Об'єм оперативної пам'яті	4 ГБ для 32-розрядної системи і 16 ГБ для 64-розрядної системи
Місце на жорсткому диску	Оскільки додаток Piedava є веб-ресурсом, пам'ять для використання не потрібна
Графічний пристрій	Інтегрований графічний пристрій



Операційна система	Windows 10
--------------------	------------

#### 4.4 Висновки

У четвертому розділі було проведено тестування додатку, у результаті якого було доведено його повну працездатність та відповідність поставленому технічному завданню. Розроблено інструкцію користувача по встановленню web додатку.

## 5. ЕКОНОМІЧНА ЧАСТИНА

### 5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку веб-орієнтованого програмного засобу для інтелектуального аналізу потреб споживача інтернет-магазину – Piedava. Основною ціллю розробки алгоритму для аналізу потреб споживачів є покращення досвіду покупців інтернет ресурсів в виборі товару та збільшення обсягу продажів для самих інтернет ресурсів, покращенням рекомендацій для цих самих покупців. Розробка є ітеративна та інкрементна. Особливістю розробки є проста система аналізу даних про споживачів, що не вимагає найму спеціалізованих працівників для обробки отриманих даних. Аналогом може бути Finteza - від \$4 за 100000 відвідувань, за рік не менше 5 млн. відвідувань, т.б. близько 8000 грн/рік або HubSpot від 21600 грн/рік.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 5.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	2	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	2
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	3	3	4
Сума	41	42	42
Середньоарифметична сума балів	$(41+42+42) / 3 = 41,67$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що ціллю веб-орієнтованого програмного засобу для інтелектуального аналізу потреб споживача інтернет-магазину Piedava є покращення досвіду покупців інтернет ресурсів в виборі товару та збільшення обсягу продажів для самих інтернет ресурсів, покращенням рекомендацій для цих самих покупців. Розробка є ітеративна та інкрементна. Особливістю розробки є проста система аналізу даних про споживачів, що не вимагає найму спеціалізованих працівників для обробки отриманих даних

## 5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де  $M$  – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_p$  – число робочих днів за місяць, 21 днів;

$t$  – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	28000	1333,33	31	41333,333
Програміст	23000	1095,24	31	33952,381
Всього				75285,71

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 13,3 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 13,3 \% / 100 \% \quad (4.2)$$

$$Z_d = (75285,71 \cdot 13,3 \% / 100 \% ) = 10013,00 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_3 = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (4.3)$$

$$H_3 = (75285,71 + 10013,00) \cdot 22 \% / 100 \% = 18765,72 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_v} \cdot \frac{t_{вик}}{12} \text{ [грн.]}. \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$  – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 25000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,48 міс.

$$A_{обл} = \frac{25000}{2} \times \frac{1,48}{12} = 1537,7 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн. (OS Windows Home, віртуальний сервер), то даний нематеріальний актив

не амортизується, а його вартість включається у вартість розробки повністю,  
 $V_{нем.ак.} = 997$  грн.

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (Asus Book X513FL Black)	25000	2	1,48	1537,698
Офісне обладнання (меблі)	20500	4	1,48	630,456
Приміщення	680000	20	1,48	4182,540
Всього				6350,69

5.2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (4.5)$$

де  $V$  – вартість 1 кВт-години електроенергії для 1 класу підприємства,  
 $V = 6,2$  грн./кВт;

$P$  – встановлена потужність обладнання, кВт.  $P = 0,35$  кВт;

$\Phi$  – фактична кількість годин роботи обладнання, годин.

$K_{\Pi}$  – коефіцієнт використання потужності,  $K_{\Pi} = 0,9$ .



$$V_e = 0,9 \cdot 0,35 \cdot 8 \cdot 31 \cdot 6,2 = 484,344 \text{ (грн.)}$$

### 5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.6)$$

де  $H_{ib}$  – норма нарахування за статтею «Інші витрати».

$$I_e = 75285,71 \cdot 78\% / 100\% = 58722,86 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.7)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 75285,71 * 118 \% / 100 \% = 88837 \text{ (грн.)}$$

### 5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 75285,71 + 10013,00 + 18765,72 + 6350,69 + 997 + 484,34 + 58722,86 + 88837 = 259456,47 \text{ грн.}$$

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (4.8)$$

де  $\eta$  – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то  $\eta=0,1$ ; технічного проектування, то  $\eta=0,2$ ; розробки конструкторської документації, то  $\eta=0,3$ ; розробки технологій, то  $\eta=0,4$ ; розробки дослідного зразка, то  $\eta=0,5$ ; розробки промислового зразка, то  $\eta=0,7$ ; впровадження, то  $\eta=0,9$ . Оберемо  $\eta = 0,5$ , так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 259456,47 / 0,5 = 518913 \text{ грн.}$$

### **5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором**

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

*а)* вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

*б)* зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

*в)* кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

*г)* визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності

інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.9)$$

де  $\pm\Delta\Pi_o$  – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$\Pi_o$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,  $\Pi_o = \Pi_o \pm \Delta\Pi_o$ ;

$\Pi_b$  – вартість програмного продукту у році до впровадження результатів розробки;

$\Delta N$  – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ .

$p$  – коефіцієнт, який враховує рентабельність продукту;

$\vartheta$  – ставка податку на прибуток, у 2022 році  $\vartheta = 18\%$ .

Припустимо, що при прогнозованій ціні 2500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 550 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 3000 шт., протягом другого року – на 2000 шт., протягом третього року на 1000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*550 + (2500 + 550) * 3000) * 0,8333 * 0,39 * (1 - 0,18) = 1998749,920 \text{ грн.}$$

$$\Delta\Pi_2 = (0*550 + (2500 + 550) * (3000 + 2000)) * 0,8333 * 0,39 * (1 - 0,18) = 4064124,837 \text{ грн.}$$

$$\Delta\Pi_3 = (0*550 + (2500 + 550) * (3000 + 2000 + 1000)) * 0,8333 * 0,39 * (1 - 0,18) = 4876949,805 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 10939824,56 грн.

### 5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків  $\Pi\Pi$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.10)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

$T$  – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;

$t$  – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\text{ПП} = (1998749,920/(1+0,1)^1) + (4064124,837/(1+0,1)^2) + (4876949,805/$$

$$/(1+0,1)^3) = 1817045,38 + 3358780,857 + 3664124,572 = 8839950,811$$

грн.

Далі розраховують величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (4.11)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{инв} = 2 \dots 5$ , але може бути і більшим;

$ZB$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 518913 = 1037825,88 \text{ грн.}$$

Тоді абсолютний економічний ефект  $E_{abc}$  або чистий приведений дохід ( $NPV$ , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (4.12)$$

$$E_{abc} = 8839950,811 - 1037825,88 = 7802124,93 \text{ грн.}$$

Оскільки  $E_{abc} > 0$  то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності ( $IRR$ , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_e$ . Для цього використаємо формулу:

$$E_e = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.13)$$

$T_{жс}$  – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 7802124,93/1037825,88) - 1} = 1,042$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.14)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d = (0,09...0,14)$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,5)$ .

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як  $E_b > \tau_{\min}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (4.15)$$

$$T_{ок} = 1 / 1,042 = 0,96 \text{ р.}$$

Оскільки  $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,96 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 518913 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,96 роки.



## ВИСНОВКИ

У магістерській роботі розроблено додаток для аналізу потреб споживачів інтернет-магазину з метою підвищення продажів.

В результаті виконання роботи розв'язано такі задачі: проведено аналіз існуючих додатків для аналізу потреб споживачів, для покращення алгоритму розподілу користувачів по категоріям. Запропоновано новий спосіб представлення аналітичних даних та вперше використано мікросервісну архітектуру для розробки подібного додатку, розроблено програмні компоненти та проведено експериментальні дослідження розробленого додатку для аналізу потреб споживачів інтернет-магазину.

Проведено інваріантний аналіз та обґрунтування вибору програмних засобів при розробці програми, обрано мову програмування C# з використанням фреймворка .NET.CORE та ASP.Net. Для розробки користувацького інтерфейсу було використано синтаксичний код RAZOR. Розроблено схеми алгоритмів занесення даних в файл.XLSX, структуру додатку та модулі додатку.

Було розглянуто види та методики тестування, обрано методику тестування «чорної скриньки», складено набір тест-випадків для тестування програми, які довели повну працездатність даного додатку та відповідність поставленому технічному завданню. Розроблено інструкцію користувача та визначено вимоги до персонального комп'ютера користувача.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Google Analytics [Електронний ресурс] – Режим доступу до ресурсу <https://analytics.google.com/analytics/web/provision/#/provision>
2. Яндекс.Метрика [Електронний ресурс] - Режим доступу до ресурсу <https://passport.yandex.ru/auth?retpath=https%3A%2F%2Fmetrika.yandex.ru%2Flist&origin=metrika>
3. Турум-Бурум [Електронний ресурс] Режим доступу до ресурсу <https://turumburum.ua/blog/kak-pravilno-sostavit-portret-pokupatelya-internet-magazina-i-povysit-konversiyu-/>
4. .NET Documentation [Електронний ресурс] Режим доступу до ресурсу <https://docs.microsoft.com/ru-ru/dotnet/>
5. Convert Monster [Електронний ресурс] Режим доступу до ресурсу <https://convertmonster.ru/blog/analitica-blog/kak-analizirovat-dannye-onlajn-prodazh/>
6. Andrew Troelsen, Philip Japikse. C# 6. 0 and the . NET 4. 6 Framework. Joseph Albahari C# 6.0 in a Nutshell: The Definitive Reference/ Joseph
7. Albahari, Ben Albahari – O'Reilly Media, Inc., 2015. – 1136 ст.
8. Документація ASP.Net Core [Електронний ресурс] – Режим доступу до ресурсу: <https://www.aspnetcore.com/>.
9. «Buyology Truth and Lies about Why We Buy» Lindstrom Martin ;1 edition 2012, - 212ст.
10. Seth Godin This is Marketing. You Can't Be Seen Until You Learn To See 1 edition 2018. – 288 ст
11. Eyal, N. Hooked: How to Build Habit-Forming Products. 2014 – 256 ст.

12. Интернет-маркетинг с нуля [Электронный ресурс] Режим доступа до ресурсу <https://tilda.education/courses/marketing/internet-marketing-beginning/>
13. Документація RAZOR від Microsoft [Электронный ресурс] <https://docs.microsoft.com/ru-ru/aspnet/core/mvc/views/razor?view=aspnetcore-5.0>
14. .Документація ASP.Net Core[Электронный ресурс] – Режим доступа до ресурсу: <https://www.aspsnippets.com/>.
15. Wikipedia [Электронный ресурс] Режим доступа до ресурсу Wikipedia.org
16. Optimizely [Электронный ресурс] Режим доступа до ресурсу <https://www.optimizely.com/optimization-glossary/online-marketing/>
17. Upload and Download files from SQL Server Database in ASP.Net[Электронный ресурс] – Режим доступа до ресурсу: <https://www.aspsnippets.com/Articles/Upload-and-Download-files-fromSQL-Server-Database-in-ASPNet.aspx>.
18. SearchEngineJournal - [Электронный ресурс] – Режим доступа до ресурсу: <https://www.searchenginejournal.com/internet-marketing/230047/>
19. Investopedia - [Электронный ресурс] – Режим доступа до ресурсу <https://www.investopedia.com/terms/d/digital-marketing.asp>
20. NeilPatel - [Электронный ресурс] – Режим доступа до ресурсу <https://neilpatel.com/what-is-online-marketing/>

**ДОДАТОК А**

Міністерство освіти та науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ОТ

\_\_\_\_\_ проф., д.т.н. О. Д. Азаров

«\_\_\_» \_\_\_\_\_ 2022 р.

**ТЕХНІЧНЕ ЗАВДАННЯ**

на виконання магістерської кваліфікаційної роботи  
«Веб-орієнтований програмний засіб для інтелектуального аналізу потреб  
споживача інтернет-магазину»  
08-23.МКР.028.00.000 ПЗ

Науковий керівник: к. т. н., доц. каф ОТ

\_\_\_\_\_ Савицька Л.А.

Магістрант групи 2КІ-21м

\_\_\_\_\_ Старий М.П.

Опонент к.т.н. доц. каф. МБІС

\_\_\_\_\_ Карпінєць В.В.

## 1. Підстава виконання магістерської кваліфікаційної роботи

1.1 Актуальність розробки даного програмного забезпечення. З кожним роком кількість покупок зроблених через інтернет-магазин зростає. У 2018 році кількість інтернет-покупців становила приблизно сім мільйонів чоловік, а кількість витрачених грошей становила близько 78 млрд. гривень. При чому майже всі кошти були потрачені не просто як оплата через інтернет, а саме покупка товарів через інтернет, ця цифра сягає 65 млрд. гривень. А вже у 2019 році кількість коштів, потрачених через інтернет виросла на 25% і сягнула позначки 97 млрд. гривень, з яких 81 млрд гривень витрачено на покупку товарів та послуг саме в інтернет – магазинах.

Саме тому розробко програмного продукту для детального аналізу даних потреб споживача того чи іншого ресурсу на сьогоднішній день є доволі перспективною задачею.

1.2 Наказ про затвердження теми магістерської кваліфікаційної роботи.

## 2 Метою і призначенням МКР

2.1 Метою роботи є розробка програмного додатку аналізу потреб споживача інтернет-магазину для підвищення ефективності обробки даних запитів користувачів і корегування стратегії розвитку інтернет-магазинів.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

## 3. Вихідні дані для виконання МКР

Створити додаток для інтелектуального аналізу потреб споживачів інтернет магазину.з метою підвищення продаж.

## 4. Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

- проведення аналізу відвідувачів інтернет магазину;
- створення звіту по результатам аналізу;
- аналіз по користувачам;
- аналіз по товарам;
- аналіз по категоріям;

## 5. Етапи МКР та очікувані результати

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задач роботи	15.09.22	
2	Аналіз статистичних характеристик	15.09-16.09.22	
3	Аналіз переваг використання систем збору даних	16.09-24.09.22	
4	Огляд існуючих систем аналізу даних	24.09-04.10.22	
5	Розробка алгоритмів додатку	04.10-14.10.22	
6	Розробка інтерфейсу користувача	14.10-23.10.22	
7	Розробка програмних компонент	24.10-31.10.22	
8	Розробка тестового інтернет-магазину	01.11-16.11.22	
9	Розрахунок економічної частини роботи	17.11-30.11.22	
10	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12-06.12.22	
11	Аналіз виконання роботи, висновки, додатки	07.12-06.12.22	
12	Перевірка якості виконання магістерської роботи та усунення недоліків	15.12.21	

## 6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив рецензента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

## 7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

## 8 Вимоги до оформлення МКР

Вимоги викладені в «Положенні про кваліфікаційну роботу у Вінницькому національному технічному університеті» з урахуванням змін,

що подані у бюлетені ВАК України № 9-10, 2011р., а також в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008-2015, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

9 Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання отримав \_\_\_\_\_ Старий М.П.

## Додаток Б – Лістинг програми

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.Extensions.Logging;
using Piedava.Models;
using System.Diagnostics;

namespace Piedava.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;
        private readonly DbConnector _dbConnector;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
            _dbConnector = new DbConnector();
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult AnalysisOfVisits()
        {
            ViewBag.Months = new SelectList(_dbConnector.GetMonths(),
            "Id", "Name");
        }
    }
}
```



```

        ViewBag.Years = new SelectList(_dbConnector.GetYears(), "Id",
"Name");

        return View();
    }

    public IActionResult AnalysisOfGoods()
    {
        ViewBag.Months = new SelectList(_dbConnector.GetMonths(),
"Id", "Name");
        ViewBag.Years = new SelectList(_dbConnector.GetYears(), "Id",
"Name");
        ViewBag.Categories = new
SelectList(_dbConnector.GetCategories(), "Id", "Name");

        return View();
    }

    public IActionResult AnalysisOfCustomers()
    {
        ViewBag.Months = new SelectList(_dbConnector.GetMonths(),
"Id", "Name");
        ViewBag.Years = new SelectList(_dbConnector.GetYears(), "Id",
"Name");
        ViewBag.Categories = new
SelectList(_dbConnector.GetCategories(), "Id", "Name");

        return View();
    }
}

```

```

        [ResponseCache(Duration = 0, Location =
ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId =
Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}

```

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using Piedava.Models;

```

```

namespace Piedava.Controllers

```

```

{
    public class TableController : Controller
    {
        private readonly ILogger<TableController> _logger;
        private readonly DbConnector _dbConnector;

        public TableController(ILogger<TableController> logger)
        {
            _logger = logger;
            _dbConnector = new DbConnector();
        }
    }
}

```

```

        public IActionResult
GetAnalysisOfVisits(AnalysisOfVisitsFilterModel filter)
    {

```

```

var datasets = _dbConnector.GetVisitsDatasets(filter);

var fileBytes = ExcelCreator.CreateAnalysisOfVisitsTable(datasets,
filter);

return File(fileBytes, "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet", "file.xlsx");
}

public IActionResult
GetAnalysisOfGoods(AnalysisOfGoodsFilterModel filter)
{
var datasets = _dbConnector.GetGoodsDatasets(filter);

var fileBytes = ExcelCreator.CreateAnalysisOfGoodsTable(datasets,
filter);

return File(fileBytes, "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet", "file.xlsx");
}

public IActionResult
GetAnalysisOfCustomers(AnalysisOfCustomersFilterModel filter)
{
var datasets = _dbConnector.GetCustomersDatasets(filter);

var fileBytes =
ExcelCreator.CreateAnalysisOfCustomersTable(datasets, filter);

```

```

        return File(fileBytes, "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet", "file.xlsx");
    }
}
}

```

```
namespace Piedava.Models
```

```

{
    public class AnalysisOfCustomersFilterModel
    {
        public int FromYear { get; set; }
        public int ToYear { get; set; }
        public int CategoryId { get; set; }
    }
}

```

```
namespace Piedava.Models
```

```

{
    public class AnalysisOfGoodsFilterModel
    {
        public int FromMonth { get; set; }
        public int FromYear { get; set; }
        public int ToMonth { get; set; }
        public int ToYear { get; set; }

        public int AnalysisType { get; set; }
        public int CategoryId { get; set; }
    }
}

```

```

namespace Piedava.Models
{
    public class AnalysisOfVisitsFilterModel
    {
        public int FromMonth { get; set; }
        public int FromYear { get; set; }
        public int ToMonth { get; set; }
        public int ToYear { get; set; }

        public bool Direct { get; set; }
        public bool SearchSystem { get; set; }
        public bool Advertising { get; set; }
        public bool AnotherResouce { get; set; }
        public bool SocialNetwork { get; set; }
        public bool Mailing { get; set; }
        public bool Referral { get; set; }
    }
}

using Piedava.Models;
using System.Collections.Generic;
using System.Linq;

namespace Piedava
{
    public class DbConnector
    {
        public List<IdNameModel> GetCategories()
        {
            return _categories;
        }
    }
}

```

```

    }

    public List<IdNameModel> GetYears()
    {
        return _years;
    }

    public List<IdNameModel> GetMonths()
    {
        return _months;
    }

    public List<VisitsDataset>
    GetVisitsDatasets(AnalysisOfVisitsFilterModel filter)
    {
        var result = _visitsDatasets.Where(d => {

            if (d.Type == VisitsDatasetType.Advertising &&
!filter.Advertising) return false;

            if (d.Type == VisitsDatasetType.AnotherResouce &&
!filter.AnotherResouce) return false;

            if (d.Type == VisitsDatasetType.Direct && !filter.Direct) return
false;

            if (d.Type == VisitsDatasetType.Mailing && !filter.Mailing)
return false;

            if (d.Type == VisitsDatasetType.Referal && !filter.Referal) return
false;

            if (d.Type == VisitsDatasetType.SearchSystem &&
!filter.SearchSystem) return false;

```

```

        if (d.Type == VisitsDatasetType.SocialNetwork &&
!filter.SocialNetwork) return false;

```

```

        if (d.Year > filter.FromYear && d.Year < filter.ToYear)
        {
            return true;
        }
        else if ((d.Year == filter.FromYear && d.MonthNumber >=
filter.FromMonth) && (d.Year == filter.ToYear && d.MonthNumber <=
filter.ToMonth))
        {
            return true;
        }
        return false;
    }).ToList();

    return result;
}

```

```

    public List<GoodsDataset>
    GetGoodsDatasets(AnalysisOfGoodsFilterModel filter)
    {
        var result = _goodsDatasets.Where(d => {

            if (filter.AnalysisType == 2)
            {
                switch (filter.CategoryId)
                {
                    case 1:
                        if (d.Category != "Ноутбуки та комп'ютери") return false;

```

```

        break;
    case 2:
        if (d.Category != "Смартфони, ТВ та електроніка")
return false;

        break;
    case 3:
        if (d.Category != "Товари для геймерів") return false;
        break;
    default:
        break;
    }
}

if (d.Year > filter.FromYear && d.Year < filter.ToYear)
{
    return true;
}
else if ((d.Year == filter.FromYear && d.MonthNumber >=
filter.FromMonth) && (d.Year == filter.ToYear && d.MonthNumber <=
filter.ToMonth))
{
    return true;
}
return false;

}).ToList();

return result;
}

```



```

public List<CustomersDataset>
GetCustomersDatasets(AnalysisOfCustomersFilterModel filter)
{
    var result = _customersDatasets.Where(d => {

        if (filter.CategoryId != _categories.Where(c => c.Name ==
d.Category).First().Id) return false;

        if (d.Year <= filter.ToYear && d.Year >= filter.FromYear) return
true;

        return false;

    }).ToList();

    return result;
}

private List<IdNameModel> _years = new List<IdNameModel>
{
    /*new IdNameModel{Id = 2005, Name = "2005"},
    new IdNameModel{Id = 2006, Name = "2006"},
    new IdNameModel{Id = 2007, Name = "2007"},
    new IdNameModel{Id = 2008, Name = "2008"},
    new IdNameModel{Id = 2009, Name = "2009"},
    new IdNameModel{Id = 2010, Name = "2010"},
    new IdNameModel{Id = 2011, Name = "2011"},
    new IdNameModel{Id = 2012, Name = "2012"},
    new IdNameModel{Id = 2013, Name = "2013"},
    new IdNameModel{Id = 2014, Name = "2014"},

```

```

new IdNameModel{Id = 2015, Name = "2015"},
new IdNameModel{Id = 2017, Name = "2017"},
new IdNameModel{Id = 2018, Name = "2018"},
new IdNameModel{Id = 2019, Name = "2019"},*/
new IdNameModel{Id = 2020, Name = "2020"}
//new IdNameModel{Id = 2021, Name = "2021"}
};

private List<IdNameModel> _months = new List<IdNameModel>
{
    new IdNameModel{Id = 1, Name = "Січень"},
    new IdNameModel{Id = 2, Name = "Лютий"},
    new IdNameModel{Id = 3, Name = "Березень"},
    new IdNameModel{Id = 4, Name = "Квітень"},
    new IdNameModel{Id = 5, Name = "Травень"},
    new IdNameModel{Id = 6, Name = "Червень"},
    new IdNameModel{Id = 7, Name = "Липень"},
    new IdNameModel{Id = 8, Name = "Серпень"},
    new IdNameModel{Id = 9, Name = "Вересень"},
    new IdNameModel{Id = 10, Name = "Жовтень"},
    new IdNameModel{Id = 11, Name = "Листопад"},
    new IdNameModel{Id = 12, Name = "Грудень"}
};

private List<IdNameModel> _categories = new List<IdNameModel>
{
    new IdNameModel{Id = 1, Name = "Ноутбуки та комп'ютери"},
    new IdNameModel{Id = 2, Name = "Смартфони, ТВ та
електроніка"},
    new IdNameModel { Id = 3, Name = "Товари для геймерів" }
};

using OfficeOpenXml;

```

```

using Piedava.Models;
using System.Collections.Generic;
using System.Linq;

namespace Piedava
{
    public class ExcelCreator
    {
        private static string[] _months = { "Січень", "Лютий", "Березень",
"Квітень", "Травень", "Червень", "Липень", "Серпень", "Вересень",
"Жовтень", "Листопад", "Грудень" };

        public static byte[] CreateAnalysisOfVisitsTable(List<VisitsDataset>
datasets, AnalysisOfVisitsFilterModel filter)
        {
            ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

            using (ExcelPackage excelPackage = new ExcelPackage())
            {
                var workSheet =
excelPackage.Workbook.Worksheets.Add("Sheet1");

                workSheet.Cells[1, 1].Value = "Вид переходів";
                workSheet.Column(1).AutoFit();

                int minMonth = datasets.Select(d => d.MonthNumber).Min();
                int maxMonth = datasets.Select(d => d.MonthNumber).Max();
                int minYear = datasets.Select(d => d.Year).Min();
                int maxYear = datasets.Select(d => d.Year).Max();
            }
        }
    }
}

```

```

for (int m = minMonth, y = minYear, c = 2; y <= maxYear; c++)
{
    workSheet.Cells[1, c].Value = $"{_months[m - 1]} {y}";
    workSheet.Column(c).AutoFit();

    if (m == 12 || (y == maxYear && m > maxMonth))
    {
        m = 1;
        y++;
    }
    else
    {
        m++;
    }
}

int rowIndex = 2;

if (filter.Direct)
{
    workSheet.Cells[rowIndex, 1].Value = "Прямі переходи";
    workSheet.Column(1).AutoFit();

    var records = datasets.Where(d => d.Type ==
VisitsDatasetType.Direct).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
    int columnIndex = 2;

    foreach (var record in records)
    {

```

```

        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;

        workSheet.Column(columnIndex).AutoFit();
        columnIndex++;
    }

    rowIndex++;
}

if (filter.SearchSystem)
{
    workSheet.Cells[rowIndex, 1].Value = "Переходи з
ПОШУКОВИХ СИСТЕМ";
    workSheet.Column(1).AutoFit();

    var records = datasets.Where(d => d.Type ==
VisitsDatasetType.SearchSystem).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
    int columnIndex = 2;

    foreach (var record in records)
    {
        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;

        workSheet.Column(columnIndex).AutoFit();
        columnIndex++;
    }

    rowIndex++;
}

```

```

if (filter.Advertising)
{
    workSheet.Cells[rowIndex, 1].Value = "Переходи по рекламі";
    workSheet.Column(1).AutoFit();

    var records = datasets.Where(d => d.Type ==
VisitsDatasetType.Advertising).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
    int columnIndex = 2;

    foreach (var record in records)
    {
        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
        workSheet.Column(columnIndex).AutoFit();
        columnIndex++;
    }

    rowIndex++;
}

if (filter.AnotherResouce)
{
    workSheet.Cells[rowIndex, 1].Value = "Переходи з інших
ресурсів";
    workSheet.Column(1).AutoFit();
}

```

```

        var records = datasets.Where(d => d.Type ==
VisitsDatasetType.AnotherResouce).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
        int columnIndex = 2;

        foreach (var record in records)
        {
            workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;

            workSheet.Column(columnIndex).AutoFit();
            columnIndex++;
        }

        rowIndex++;
    }

    if (filter.SocialNetwork)
    {
        workSheet.Cells[rowIndex, 1].Value = "Переходи із
соціальних мереж";
        workSheet.Column(1).AutoFit();

        var records = datasets.Where(d => d.Type ==
VisitsDatasetType.SocialNetwork).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
        int columnIndex = 2;

        foreach (var record in records)
        {

```

```

        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;

        workSheet.Column(columnIndex).AutoFit();
        columnIndex++;
    }

    rowIndex++;
}

if (filter.Mailing)
{
    workSheet.Cells[rowIndex, 1].Value = "Переходи із поштових
розсилок";

    workSheet.Column(1).AutoFit();

    var records = datasets.Where(d => d.Type ==
VisitsDatasetType.Mailing).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
    int columnIndex = 2;

    foreach (var record in records)
    {
        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;

        workSheet.Column(columnIndex).AutoFit();
        columnIndex++;
    }

    rowIndex++;
}

```



```

        if (filter.Referal)
        {
            workSheet.Cells[rowIndex, 1].Value = "Переходи по
реферальным посещениям";
            workSheet.Column(1).AutoFit();

            var records = datasets.Where(d => d.Type ==
VisitsDatasetType.Referal).OrderBy(d => d.Year).ThenBy(d =>
d.MonthNumber).ToList();
            int columnIndex = 2;

            foreach (var record in records)
            {
                workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
                workSheet.Column(columnIndex).AutoFit();
                columnIndex++;
            }

            rowIndex++;
        }

        return excelPackage.GetAsByteArray();
    }
}

public static byte[] CreateAnalysisOfGoodsTable(List<GoodsDataset>
datasets, AnalysisOfGoodsFilterModel filter)
{

```

```

if (filter.AnalysisType == 1)
{
    return CreateAnalysisOfGoodsByCategoriseTable(datasets, filter);
}
else
{
    return CreateAnalysisOfGoodsByGoodsTable(datasets, filter);
}
}

public static byte[]
CreateAnalysisOfCustomersTable(List<CustomersDataset> datasets,
AnalysisOfCustomersFilterModel filter)
{
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

    using (ExcelPackage excelPackage = new ExcelPackage())
    {
        var workSheet =
excelPackage.Workbook.Worksheets.Add("Sheet1");

        workSheet.Cells[1, 1].Value = "Категорія користувачів";

        int minQuarter = 1;
        int maxQuarter = 4;
        int minYear = datasets.Select(d => d.Year).Min();
        int maxYear = datasets.Select(d => d.Year).Max();

        for (int q = minQuarter, y = minYear, c = 2; y <= maxYear; c++)
        {

```

```

workSheet.Cells[1, c].Value = $"{q} квартал, {y}р.";
workSheet.Column(c).AutoFit();

if (q == 4 || (y == maxYear && q > maxQuarter))
{
    q = 1;
    y++;
}
else
{
    q++;
}
}

int rowIndex = 2;

var records = datasets.Where(d => d.Age <= 25 &&
d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
workSheet.Cells[rowIndex, 1].Value = "До 25 років, чоловіки";

int columnIndex = 2;
foreach (var record in records)
{
    workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
    columnIndex++;
}
rowIndex++;

```

```

        records = datasets.Where(d => d.Age <= 25 &&
!d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
        workSheet.Cells[rowIndex, 1].Value = "До 25 років, жінки";

        columnIndex = 2;
        foreach (var record in records)
        {
            workSheet.Cells[rowIndex,      columnIndex].Value      =
record.Number;
            columnIndex++;
        }
        rowIndex++;

        records = datasets.Where(d => d.Age <= 35 && d.Age > 25 &&
d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
        workSheet.Cells[rowIndex, 1].Value = "26 - 35 років, чоловіки";

        columnIndex = 2;
        foreach (var record in records)
        {
            workSheet.Cells[rowIndex,      columnIndex].Value      =
record.Number;
            columnIndex++;
        }
        rowIndex++;

        records = datasets.Where(d => d.Age <= 35 && d.Age > 25 &&
!d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
        workSheet.Cells[rowIndex, 1].Value = "26 - 35 років, жінки";

```

```

columnIndex = 2;
foreach (var record in records)
{
    workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
    columnIndex++;
}
rowIndex++;

```

```

records = datasets.Where(d => d.Age <= 45 && d.Age > 35 &&
d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
workSheet.Cells[rowIndex, 1].Value = "36 - 45 років, чоловіки";

```

```

columnIndex = 2;
foreach (var record in records)
{
    workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
    columnIndex++;
}
rowIndex++;

```

```

records = datasets.Where(d => d.Age <= 45 && d.Age > 35 &&
!d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
workSheet.Cells[rowIndex, 1].Value = "36 - 45 років, жінки";

```

```

columnIndex = 2;
foreach (var record in records)
{

```

```

        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
        columnIndex++;
    }
    rowIndex++;

    records = datasets.Where(d => d.Age > 45 &&
d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
    workSheet.Cells[rowIndex, 1].Value = "Від 46 років, чоловіки";

    columnIndex = 2;
    foreach (var record in records)
    {
        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
        columnIndex++;
    }
    rowIndex++;

    records = datasets.Where(d => d.Age > 45 &&
!d.IsMale).OrderBy(d => d.Year).ThenBy(d => d.Quarter).ToList();
    workSheet.Cells[rowIndex, 1].Value = "Від 46 років, жінки";
    workSheet.Column(1).AutoFit();

    columnIndex = 2;
    foreach (var record in records)
    {
        workSheet.Cells[rowIndex, columnIndex].Value =
record.Number;
        workSheet.Column(columnIndex).AutoFit();

```

```

        columnIndex++;
    }
    rowIndex++;

    return excelPackage.GetAsByteArray();
}
}

public static byte[]
CreateAnalysisOfGoodsByCategoriseTable(List<GoodsDataset> datasets,
AnalysisOfGoodsFilterModel filter)
{
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

    using (ExcelPackage excelPackage = new ExcelPackage())
    {
        var workSheet =
excelPackage.Workbook.Worksheets.Add("Sheet1");

        workSheet.Cells[1, 1].Value = "Категорія";

        int rowIndex = 2;
        var categories = datasets.Select(d => d.Category).Distinct();

        foreach (var category in categories)
        {
            workSheet.Cells[rowIndex++, 1].Value = category;
        }
        workSheet.Column(1).AutoFit();
    }
}

```

```

int minMonth = datasets.Select(d => d.MonthNumber).Min();
int maxMonth = datasets.Select(d => d.MonthNumber).Max();
int minYear = datasets.Select(d => d.Year).Min();
int maxYear = datasets.Select(d => d.Year).Max();
int columnIndex = 2;

for (int m = minMonth, y = minYear, c = 2; y <= maxYear; c++)
{
    workSheet.Cells[1, c].Value = $" {_months[m - 1]} {y}";
    workSheet.Column(c).AutoFit();

    rowIndex = 2;

    foreach (var category in categories)
    {
        var sum = datasets.Where(d => d.Category == category &&
d.Year == y && d.MonthNumber == m).Select(d => d.Number).Sum();

        workSheet.Cells[rowIndex, columnIndex].Value = sum;
        workSheet.Column(columnIndex).AutoFit();

        rowIndex++;
    }

    columnIndex++;

    if (m == 12 || (y == maxYear && m > maxMonth))
    {
        m = 1;
        y++;
    }
}

```



```

    }
    else
    {
        m++;
    }
}

return excelPackage.GetAsByteArray();
}
}

public static byte[]
CreateAnalysisOfGoodsByGoodsTable(List<GoodsDataset> datasets,
AnalysisOfGoodsFilterModel filter)
{
    ExcelPackage.LicenseContext = LicenseContext.NonCommercial;

    using (ExcelPackage excelPackage = new ExcelPackage())
    {
        var workSheet =
excelPackage.Workbook.Worksheets.Add("Sheet1");

        workSheet.Cells[1, 1].Value = "Товар";
        workSheet.Column(1).AutoFit();

        int minMonth = datasets.Select(d => d.MonthNumber).Min();
        int maxMonth = datasets.Select(d => d.MonthNumber).Max();
        int minYear = datasets.Select(d => d.Year).Min();
        int maxYear = datasets.Select(d => d.Year).Max();
    }
}

```

```

for (int m = minMonth, y = minYear, c = 2; y <= maxYear; c++)
{
    workSheet.Cells[1, c].Value = $"{_months[m - 1]} {y}";
    workSheet.Column(c).AutoFit();

    if (m == 12 || (y == maxYear && m > maxMonth))
    {
        m = 1;
        y++;
    }
    else
    {
        m++;
    }
}

int rowIndex = 2;

var articles = datasets.Select(d => d.Article).Distinct();

foreach (var article in articles)
{
    workSheet.Cells[rowIndex, 1].Value = article;
    workSheet.Column(1).AutoFit();

    var records = datasets.Where(d => d.Article ==
article).OrderBy(d => d.Year).ThenBy(d => d.MonthNumber).ToList();
    int columnIndex = 2;

    foreach (var record in records)

```

```
        {
            workSheet.Cells[rowIndex,    columnIndex].Value    =
record.Number;
            workSheet.Column(columnIndex).AutoFit();
            columnIndex++;
        }

        rowIndex++;
    }

    return excelPackage.GetAsByteArray();
}
}
}
}
```

Додаток В – Ілюстративний матеріал до захисту бакалаврської дипломної роботи

**ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХИСТУ БАКАЛАВРСЬКОЇ  
ДИПЛОМНОЇ РОБОТИ**

Завідувач кафедри ПЗ, д. т. н., професор \_\_\_\_\_ О. Н. Романюк

Науковий керівник, к.т.н., доц. кафедри ПЗ \_\_\_\_\_ О. М. Рейда

Рецензент, д.т.н., проф. кафедри МБІС \_\_\_\_\_ Ю. Є. Яремчук

Нормоконтроль, к.т.н., доц. кафедри ПЗ \_\_\_\_\_ О. М. Рейда

Виконавець, студент групи ЗПІ-176 \_\_\_\_\_ М.П. Старий



Рисунок В.1 – Назва роботи



Рисунок В.2 – Мета, завдання та об'єкт дослідження

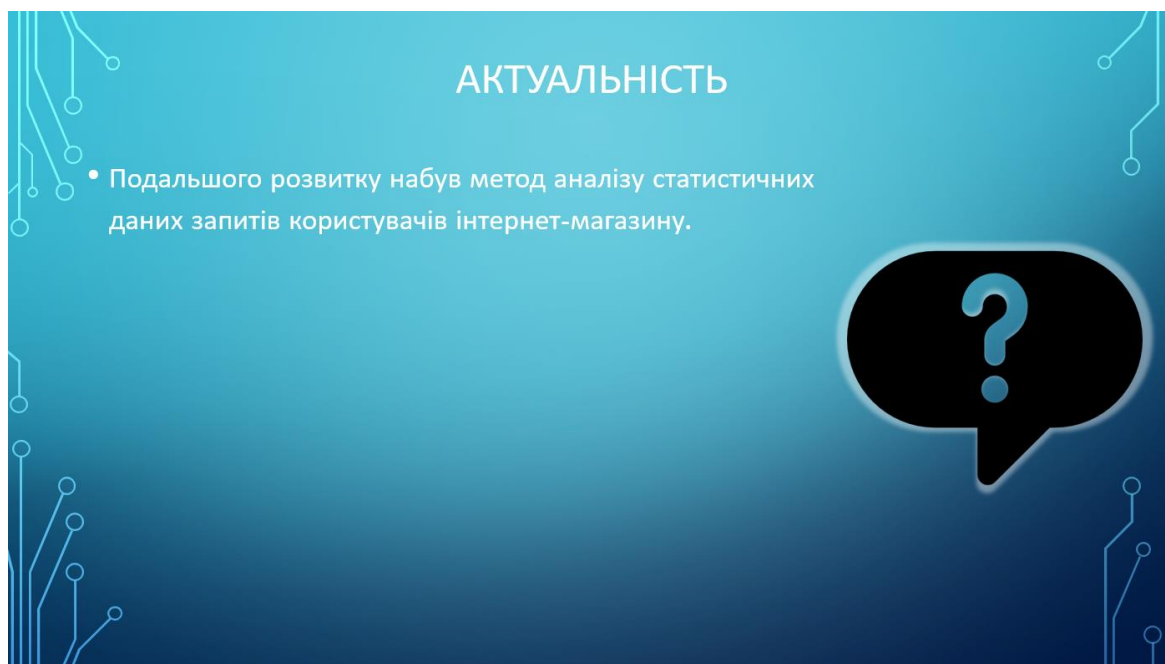


Рисунок В.3 – Наукова новизна та практична цінність



Рисунок В.4 – Аналіз предметної області



Рисунок В.5 – Опис аналізу



Рисунок В.6 – Розробка програмного додатку



Рисунок В.7 – Інтерфейс до доступу до сервісу



Рисунок В.8 - Фреймворк ASP.NET Core



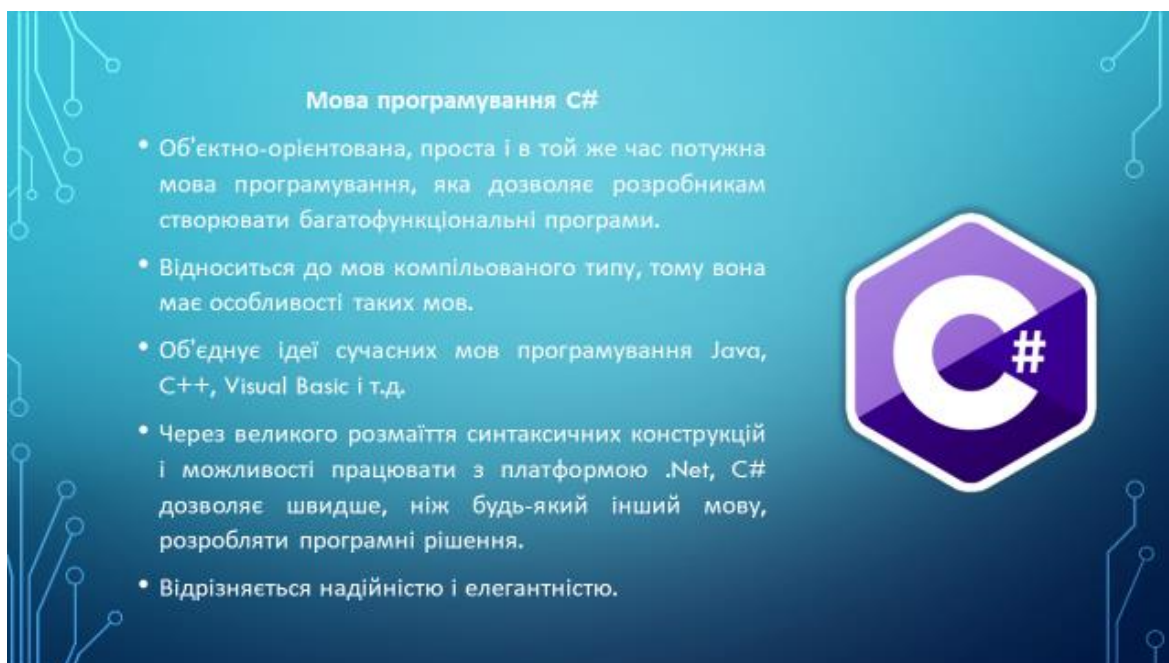


Рисунок В.9 - Мова програмування C#

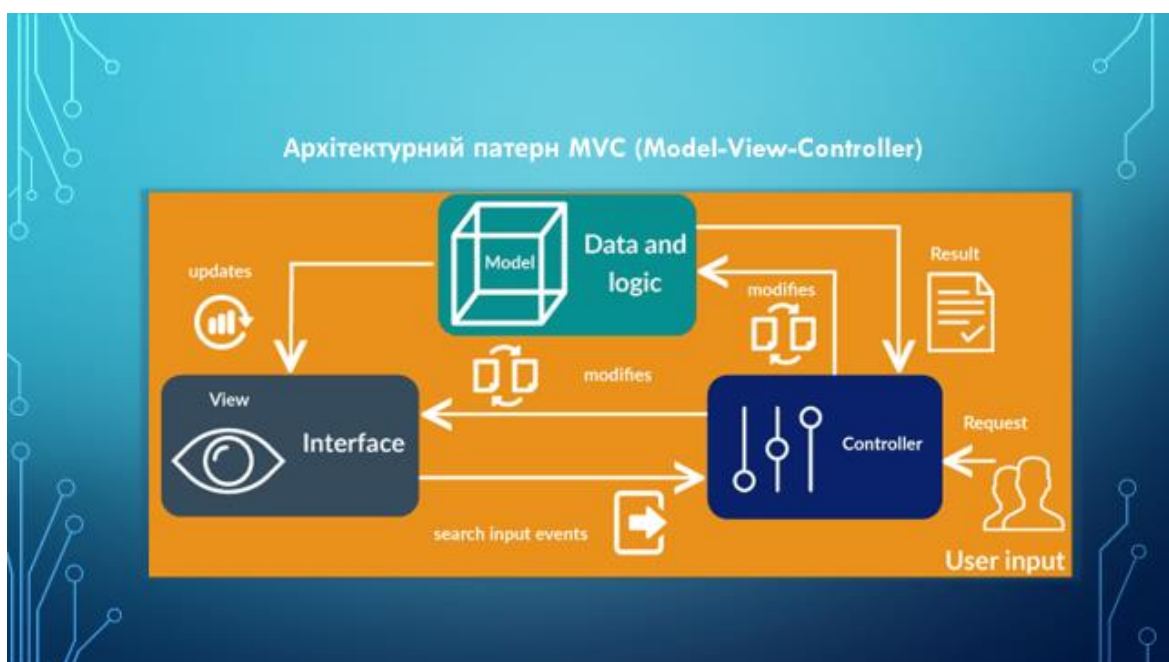


Рисунок В.10 - Архітектурний патерн MVC (Model-View-Controller)

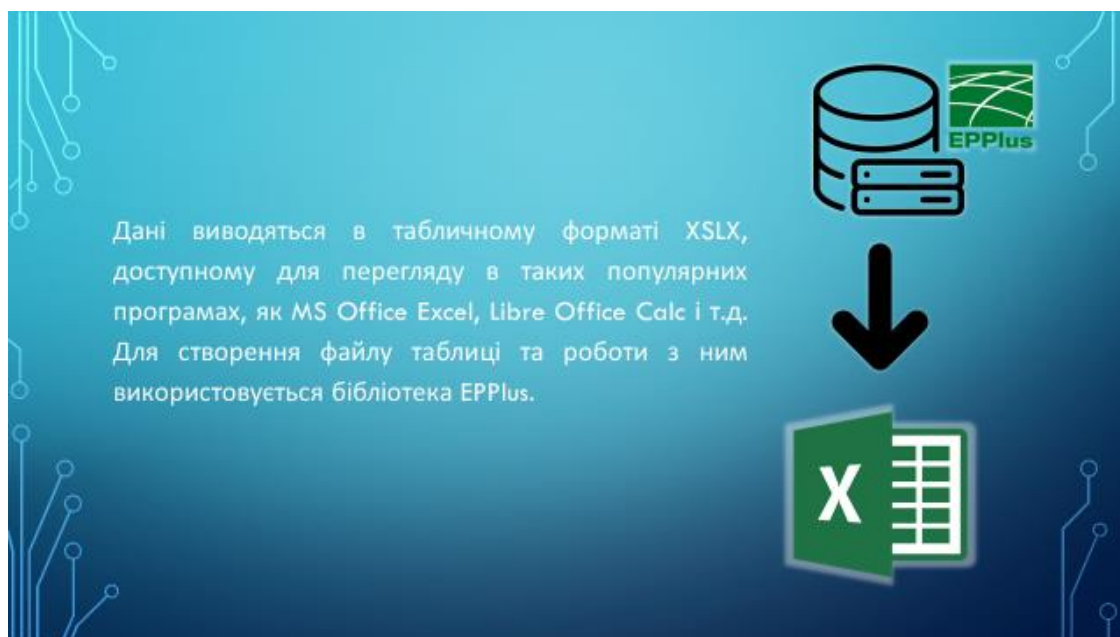


Рисунок В.11 – Робота з БД

Приклад коду контролера генерації таблиць:

```
public class TableController : Controller {
    private readonly ILogger<TableController> _logger;
    private readonly DbConnector _dbConnector;
    public TableController(ILogger<TableController> logger) {
        _logger = logger;
        _dbConnector = new DbConnector();
    }
    public IActionResult GetAnalysisOfCustomers(AnalysisOfCustomersFilterModel filter) {
        var datasets = _dbConnector.GetCustomersDatasets(filter);
        var fileBytes = ExcelCreator.CreateAnalysisOfCustomersTable(datasets, filter);
        return File(fileBytes,
            "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",
            "file.xlsx");
    }
}
```

Рисунок В.12 - Приклад коду контролера генерації таблиць:

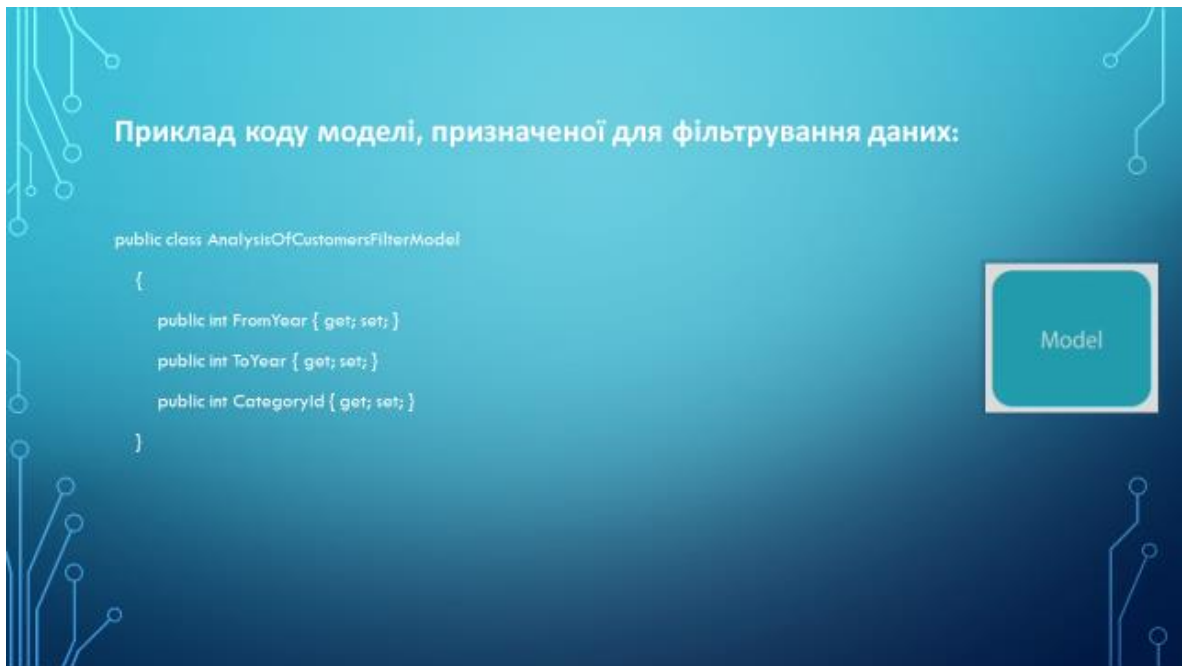


Рисунок В.13 - Приклад коду моделі, призначеної для фільтрування даних

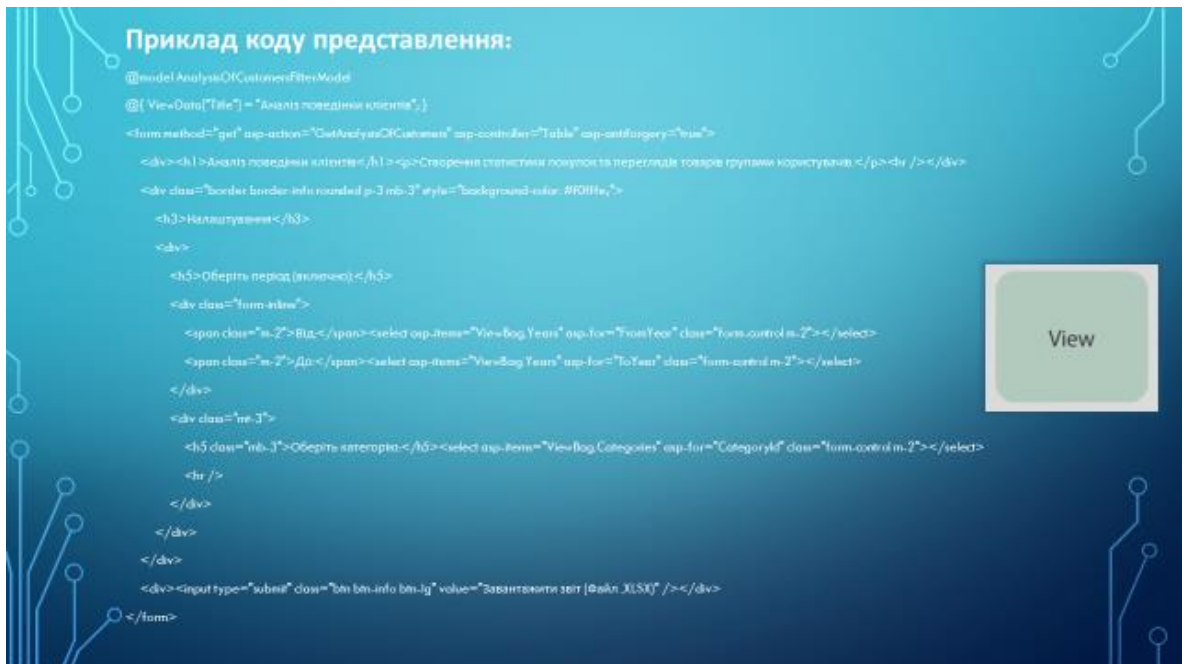


Рисунок В.14 - Приклад коду представлення



Рисунок В.15 - Інтерфейс Web-сторінки



Рисунок В.16 - Вихідні дані в форматі таблиць



Рисунок В.17 – Висновки

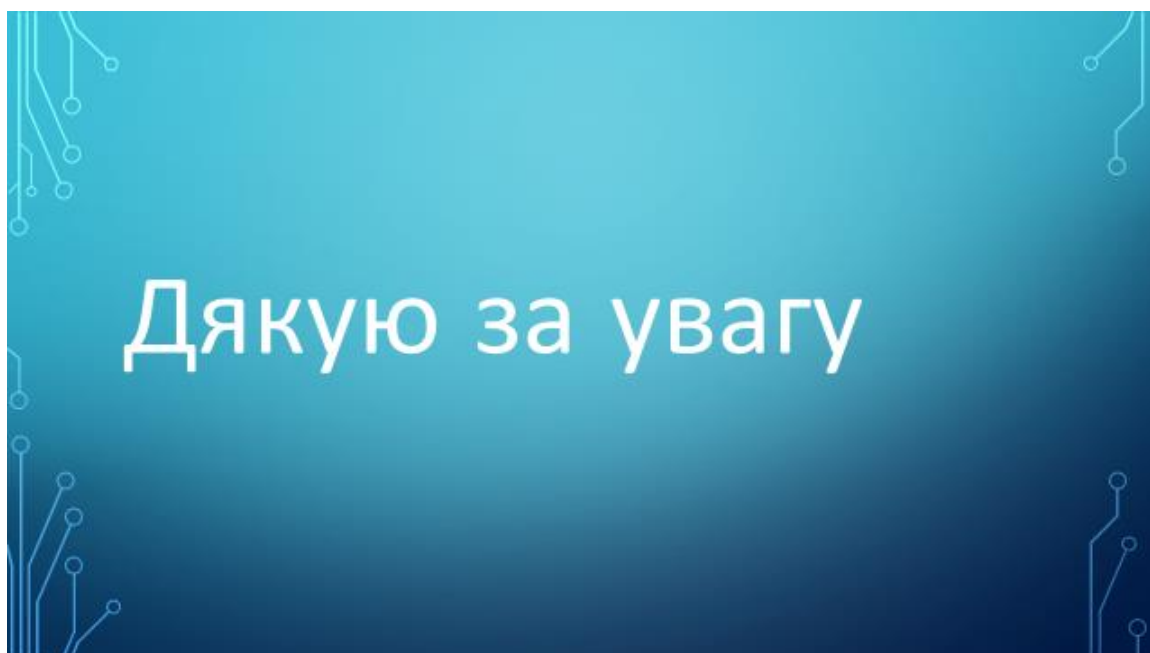


Рисунок В.18 – Подяка

**ПРОТОКОЛ  
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ  
РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ  
ЗАПОЗИЧЕНЬ**

Назва роботи: \_\_\_\_\_

Тип роботи: \_\_\_\_\_ магістерська кваліфікаційна робота  
(БДР, МКР)

Підрозділ \_\_\_\_\_ кафедра обчислювальної техніки  
(кафедра, факультет)

**Показники звіту подібності  
Unicheck**

Оригінальність \_\_\_\_\_ 80% \_\_\_\_\_ Схожість \_\_\_\_\_ 20% \_\_\_\_\_

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_ Захарченко С.М.  
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи \_\_\_\_\_  
(підпис) (прізвище, ініціали)

Керівник роботи \_\_\_\_\_  
(підпис) (прізвище, ініціали)