

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

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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
АНАЛІЗ СТАТИСТИЧНОГО ЯДРА E-COMMERCE
ПЛАТФОРМИ НА ОСНОВІ SHORIFY

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.21610209

Студент _____ Кондратюк І. В.

«___» лютого 2022 р.

Консультант _____ Воробйова А. І.

канд. техн. наук, доцент

«___» лютого 2022 р.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	6
1 АНАЛІЗ ОБРАНОЇ ПРЕДМЕТНОЇ СФЕРИ.....	8
1.1 Поняття e-commerce.....	8
1.2 Огляд e-commerce платформи Shopify.....	9
1.3 Огляд Google analytics.....	11
1.4 Огляд конкурентів Google Analytics.....	14
1.4.1 Сервіс «Matomo».....	14
1.4.2 Сервіс «Wopra».....	15
1.4.3 Сервіс «GoSquared».....	16
1.4.4 Сервіс «FoxMetrics».....	17
1.4.5 Сервіс «Mixpanel».....	19
Висновки до розділу 1.....	20
2 ФУНКЦІОНАЛ СИСТЕМИ АНАЛІТИКИ GOOGLE ANALYTICS ТА МЕНЕДЖЕРА ТЕГІВ GOOGLE TAG MANAGER.....	21
2.1 Налаштування Google Tag Manager (GTM).....	21
2.2 Налаштування Google Analytics.....	23
2.3 Налаштування тегу аналітики за допомогою Google Tag Manger.....	25
2.4 Налаштування цілей Google Analytics.....	30
2.5 Підключення до Google Search Console.....	33
Висновки до розділу 2.....	37
3 РОЗРОБКА WEB-ДОДАТКУ ДЛЯ АНАЛІТИКИ ІНТЕРНЕТ МАГАЗИНУ.....	38
3.1 Інструментальні засоби для створення web-додатку. Мови програмування та розмітки тексту.....	38

3.1.1 PHP.....	39
3.1.2 JavaScript.....	40
3.1.3 HTML.....	42
3.1.3 CSS.....	44
3.1.4 SQL.....	45
3.2 Створення додатку аналітики для інтернет-магазину на платформі Shopify.	46
3.3 Розробка головної сторінки додатку.....	48
3.3.1 Група налаштувань «Common settings».....	50
3.3.2 Група налаштувань «Chart settings».....	53
3.3.3 Група налаштувань «Table settings».....	59
3.4 Розробка сторінки з тарифними планами.....	61
3.5 Розробка сторінки з загальною інформацією про додаток.....	62
3.6 Розробка сторінки для зворотного зв'язку.....	63
3.7 База даних для веб-додатку.....	64
Висновки до розділу 3.....	67
4 МЕТОДИЧНА ЧАСТИНА.....	69
Вступ.....	69
Практична робота №1 Перевірка статистичних гіпотез у середовищі Maple.....	71
5 СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ ТА БЕЗПЕКИ У НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	87
ВИСНОВКИ.....	98
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	100
ДОДАТОК А.....	106
ДОДАТОК Б.....	107

ПЕРЕЛІК СКОРОЧЕНЬ

IT	– Інформаційні технології
HTML	– HyperText Markup Language
CSS	– Cascading Style Sheets
E-commerce	– Електронна торгівля
JS	– Javascript
PHP	– Hypertext Preprocessor
GTM	– Google Tag Manager
UA	– Univeral Analytics
GA	– Google Analytics
UI	– User Interface
UX	– User Experience
SQL	– Structured Query Language
SSL	– Secure Sockets Layer
PCI	– Payment Card Industry
SEO	– Search Engine Optimization
API	– Application Programming Interface
AMP	– Accelerated Mobile Pages
URL	– Uniform Resource Locator

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«АНАЛІЗ СТАТИСТИЧНОГО ЯДРА E-COMMERCE ПЛАТФОРМИ НА ОСНОВІ SHOPIFY»

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.21610209

Студент _____ Кондратюк І. В.

«___» лютого 2022 р.

Консультант _____ Воробйова А. І.

канд. техн. наук, доцент

«___» лютого 2022 р.

ВСТУП

Актуальність роботи. Світові обсяги продажів в e-commerce щорічно збільшуються. В 2020 році 20% від усіх покупок в світі були зроблені через мережу інтернет. За рік звичайні продажі зросли всього на 1%, за цей же час в інтернеті збільшилися на 24%. Популярність e-commerce збільшується по всій земній кулі [1].

Пандемія пришвидшила перехід від офлайн торгівлі до e-commerce приблизно на п'ять років, такий висновок зробили експерти компанії IBM. Магазины, які були тільки в офлайн, почали працювати за допомогою e-commerce, тим самим відкрили нові ніші і розширили аудиторію покупців. Останнім часом, піднялась частка продажів продуктів в інтернеті, хоча раніше це було в основному офлайн-заняття [1].

Активне наповнення e-commerce новими продавцями і покупцями дало результат – обсяг продажів в 2020 році в світі збільшився на 40%. При цьому знизився середній чек, тому що люди не ризикують робити великі покупки і обмежуються тільки найнеобхіднішим. Так що не на всіх напрямках пандемія допомогла e-commerce [2].

Світові обсяги продажів в e-commerce щорічно збільшуються. В 2020 році 20% від усіх покупок в світі були зроблені через мережу інтернет. За рік звичайні продажі зросли всього на 1%, за цей же час в інтернеті збільшилися на 24%. Популярність e-commerce збільшується по всій земній кулі [2].

Пандемія пришвидшила перехід від офлайн торгівлі до e-commerce приблизно на п'ять років, такий висновок зробили експерти компанії IBM. Магазины, які були тільки в офлайн, почали працювати за допомогою e-commerce, тим самим відкрили нові ніші і розширили аудиторію покупців. Останнім часом, піднялась частка продажів продуктів в інтернеті, хоча раніше це було в основному офлайн-заняття [3, 4, 5].

Активне наповнення e-commerce новими продавцями і покупцями дало

результат – обсяг продажів в 2020 році в світі збільшився на 40%. При цьому знизився середній чек, тому що люди не ризикують робити великі покупки і обмежуються тільки найнеобхіднішим. Так що не на всіх напрямках пандемія допомогла e-commerce [6].

Метою даної магістерської роботи є аналіз аналітичної системи Google Analytics, виявлення недоліків та розробка власного веб-додатку для аналітики налаштованого на e-commerce платформу Shopify.

Для досягнення мети магістерської роботи було поставлено декілька задач:

1. Аналіз функціоналу аналітичної системи Google Analytics.
2. Аналіз функціоналу подібних ресурсів та аналіз розбіжностей з функціоналом предмету дослідження.
3. Розробка власного веб-додатку аналітики для e-commerce платформи Shopify.

Об'єктом дослідження магістерської роботи є інформаційно-аналітична система збору та аналітики статистики інтернет магазину на e-commerce платформі Shopify.

Предметом дослідження є функціонал аналітичної системи Google Analytics, зокрема проектування та розробка власного веб-додатка для аналітики.

Практична значимість виконаних досліджень полягає в тому, щоб розробити веб-додаток аналітики, який буде допомагати користувачам легко аналізувати статистику продажів.

Новизна створюваного веб-додатку заключається в тому, що додаток буде створений спеціально для e-commerce платформи Shopify, враховуючи нюанси роботи даної платформи.

1.1 Поняття e-commerce

E-commerce – це скорочена версія словосполучення «electronic commerce» (електронна комерція), яка по суті описує будь-який тип обміну валюти на товари чи послуги в інтернеті [7].

E-commerce – це загальний термін, який охоплює все, що пов'язано з купівлею або продажом в інтернеті [7].

Оскільки визначення електронної комерції є таким відкритим (воно буквально включає будь-який тип купівлі чи продажу товарів чи послуг онлайн), існує так багато різних типів бізнесу електронної комерції. З погляду власників бізнесу, виробників та постачальників електронна комерція – це просування та надання своїх товарів чи послуг через мережу інтернет. А з погляду покупців (клієнтів) – перегляд, вибір, порівняння та замовлення.

Інтернет-комерція має кілька дуже вагомих переваг для кожної із сторін:

1. Переваги e-commerce для клієнта: можливість переглядати та купувати товари у будь-який час доби та день тижня, доставка за адресою або до найближчого відділення транспортної служби, детальна інформація про товар/послуги/сервіс. Можна легко порівнювати пропозиції від різних компаній, знаходити найвигідніші, економити час та гроші [7].

2. Плюси e-commerce для організацій: суттєве розширення географії комерційної діяльності, підвищення ефективності бізнес-процесів та реальна економія грошей. Наприклад, інтернет-магазину не потрібно орендувати велику площу для торгівлі, наймати продавців-консультантів тощо. Докладніше про це читайте у нашій статті «Бізнес-процеси інтернет-магазину» [7].

1.2 Огляд e-commerce платформи Shopify

Shopify — це e-commerce платформа, яке дозволяє розробити та запустити власний інтернет-магазин і підтримувати його успішну роботу за щомісячний платіж. Розроблена для користувачів без передових дизайнерських і технічних навичок, платформа має найпростішу функціональність і широкий спектр шаблонів, які допоможуть користувачу швидко та легко запустити будь-який веб-сайт [8].

Shopify був заснований в Канаді в 2006 році Скоттом Лейком, Деніелом Вейландом і Тобіасом Лутке. З моменту заснування компанії платформа швидко виросла, від невеликого проекту до одного з провідних сучасних рішень для онлайн-торгівлі [8].

За даними Shopify, наразі він обслуговує близько мільйона інтернет-магазинів [8].

Переваги Shopify:

1. Швидке налаштування, а також простота у використанні. Shopify пропонує простий спосіб швидкого запуску інтернет-магазину без клопоту та витрат на розробку, які можуть виникнути з такими платформами, як Magento. Інтерфейс особистого кабінету адміністратора дуже зручний і інтуїтивно зрозумілий, оскільки всі функції викладені логічно. На веб-сайті Shopify можна знайти багато статей, посібників і відеоуроків [8, 9].

2. Щоб керувати інтернет-магазином, не потрібні технічні знання. Все програмне забезпечення, хостинг та сертифікати SSL надаються Shopify. Хостинг Shopify набагато швидший і безпечніший, ніж конкуренти, і легко адаптується до стрибків трафіку. Shopify також подбає про всі оновлення програмного забезпечення. Таким чином, можна повністю зосередитися на продажу товару і не турбуватися про технічні проблеми [8, 9].

3. Безпека та надійність. Наявність інтернет-магазину означає, що користувач має справу з конфіденційною інформацією про клієнтів, включаючи дані кредитної картки. Це означає, що веб-сайт має бути швидким і безпечним. Shopify піклується про підтримку та оновлення сервера, щоб інтернет-магазин завжди був доступний для клієнтів, а також пропонує швидке завантаження сторінок [8, 9].

Сертифікати SSL інтегровані в інтернет-магазин Shopify для шифрування даних і передачі їх через безпечне з'єднання. Крім того, Shopify піклується про відповідність стандарту PCI (необхідно під час роботи з кредитними картками) [8, 9].

4. Цілодобова підтримка клієнтів. Shopify зосереджується на підтримці клієнтів. Команда підтримки доступна в цілодобовому режимі роботи і майже одразу відповідає на запити. Зв'язатися зі службою підтримки можна по телефону, електронній пошті або в онлайн-чаті. Крім того, є довідковий центр Shopify, а також безліч тематичних форумів по роботі з платформою [8, 9].

5. Адаптивний дизайн. Оскільки кількість переглядів сторінок з мобільних пристроїв зараз набагато більше, ніж з комп'ютерів, оптимізувати сторінку під різні роздільної здатності екрана не бажано, а необхідно. Усі теми Shopify адаптовані для мобільних пристроїв. Це означає, що інтернет-магазин повинен добре виглядати на всіх пристроях, а клієнти можуть робити покупки в будь-якому місці [8, 9].

Крім того, Shopify пропонує безкоштовні програми для iPhone та Android, які можна використовувати для керування магазином зі свого мобільного пристрою [8, 9].

6. Темы Shopify. Shopify Themes Store має понад 70 різних тем (безкоштовних і платних), усі адаптовані для мобільних пристроїв. Крім того, сотні тем доступні на таких сайтах, як ThemeForest та інших. Кожну тему можна налаштувати, просто відредагувавши код [8, 9].

7. Застосунки у Shopify. Shopify App Store – це скарбниця функцій, які можна додати до інтернет-магазину. Користувач може створювати різноманітні програми лояльності, списки улюблених продуктів, інтегрувати бухгалтерське програмне забезпечення, не кажучи вже про численні маркетингові додатки, щоб збільшити середній чек і підвищити конверсію веб-сайту. В «Shopify App Store» присутні понад 1500 додатків на різноманітну тематику і функціонал [8, 9].

8. SEO та маркетингові інструменти. Ще однією важливою особливістю Shopify є його потужні функції пошукової оптимізації (SEO), які допомагають веб-сайту займати вищі рейтинги в пошуковій видачі. Користувач має доступ до розширеної аналітики, яка показує, звідки приходять його клієнти, проаналізувавши дані він може змінити свою маркетингову стратегію за потреби [8, 9]

9. Відновлення покинутого кошика. Згідно зі статистикою, більше двох третин потенційних клієнтів не купують товари, які додали до кошика, тому Shopify пропонує безоплатну функцію відновлення кошика. Сервіс автоматично відстежує та надсилає потенційним клієнтам електронний лист про завершення покупки [8, 9]

10. Власна платіжна система. Shopify дозволяє інтегруватися з десятками різних платіжних систем, але також пропонує власну на основі Stripe. Якщо обрати Shopify Payments, клієнт позбувається від додаткових комісій за транзакції [8, 9].

1.3 Огляд Google analytics

Google Analytics – це служба веб-аналітики, яка надає статистику та основні аналітичні інструменти для пошукової оптимізації (SEO) і маркетингових цілей. Послуга є частиною Google Marketing Platform і доступна безкоштовно всім, хто має обліковий запис Google [10].

Google Analytics використовується для відстеження ефективності веб-сайту та збору інформації про відвідувачів. Це може допомогти організаціям визначити основні джерела трафіку користувачів, оцінювати успіх їх маркетингової діяльності та кампаній, відстежувати досягнення цілей (наприклад, покупки, додавання продуктів у кошик), виявляти закономірності та тенденції взаємодії користувачів та отримувати іншу інформацію про відвідувачів, наприклад демографічні дані. Веб-сайти малого та середнього розміру роздрібною торгівлі часто використовують Google Analytics для отримання й аналізу різноманітної аналітики поведінки клієнтів, яку можна використовувати для покращення маркетингових кампаній, збільшення трафіку веб-сайту та кращого утримання відвідувачів [10].

Google Analytics отримує дані користувача від кожного відвідувача веб-сайту за допомогою тегів сторінки. Тег сторінки JavaScript вставляється в код кожної сторінки. Цей тег працює у веб-переглядачі кожного відвідувача, збирає дані та надсилає їх на один із серверів збору даних Google. Потім Google Analytics може створювати настроювані звіти для відстеження та візуалізації таких даних, як кількість користувачів, показники відмов, середня тривалість сеансів, сеанси за каналами, перегляди сторінок, досягнення цілей тощо [10].

Тег сторінки функціонує як «веб-жучок» або «веб-маяк» для збору інформації про відвідувачів. Однак, оскільки вона покладається на файли cookie, система не може збирати дані для користувачів, які їх вимкнули [10].

Google Analytics містить функції, які можуть допомогти користувачам визначити тенденції та закономірності взаємодії відвідувачів із їхніми веб-сайтами. Функції забезпечують збір даних, аналіз, моніторинг, візуалізацію, звітність та інтеграцію з іншими програмами [10].

Ці функції включають:

- інструменти візуалізації та моніторингу даних, включаючи інформаційні панелі, системи показників і діаграми руху, які відображають зміни в даних з часом;
- фільтрація даних, маніпулювання та аналіз воронки;
- інтерфейси прикладних програм збору даних (API);
- прогнозна аналітика, розвідка та виявлення аномалій;
- сегментація для аналізу підмножин, таких як конверсії;
- спеціальні звіти для реклами, придбання, поведінки аудиторії та конверсії;
- обмін та спілкування на основі електронної пошти; і
- інтеграція з іншими продуктами, включаючи Google Ads, Google Data Studio, Salesforce Marketing Cloud, Google AdSense, Google Optimize 360, Google Search Ads 360, Google Display & Video 360, Google Ad Manager і Google Search Console.

На інформаційній панелі Google Analytics користувачі можуть зберігати профілі для кількох веб-сайтів і переглядати деталі категорій за замовчуванням або вибирати спеціальні показники для відображення для кожного сайту. Доступні категорії для відстеження включають огляд вмісту, ключові слова, сайти, які посилаються, огляд відвідувачів, накладення на карту та огляд джерел трафіку [10].

Метрика є еталоном кількісного вимірювання. Google Analytics дозволяє користувачам відстежувати до 200 різних показників для вимірювання ефективності їх веб-сайтів. Хоча деякі показники можуть бути більш цінними для певних компаній, ніж інші, це деякі з найпопулярніших показників [10]:

- **користувачі.** Користувач є унікальним або новим відвідувачем веб-сайту;

- **показник відмов.** Відсоток відвідувачів, які переглянули лише одну сторінку. Ці відвідувачі ініціювали лише один запит до сервера Google Analytics;
- **сесії.** Група взаємодій відвідувачів, які відбуваються протягом 30-хвилинної активності;
- **середня тривалість сеансу.** Скільки в середньому кожен відвідувач залишається на сайті;
- **відсоток нових сеансів.** Відсоток відвідувань веб-сайту, які відвідують його вперше;
- **сторінок за сеанс.** Середня кількість переглядів сторінки за кожен сеанс;
- **завершення цілі.** Кількість разів, коли відвідувачі виконують певну бажану дію. Це також відомо як конверсія;
- **перегляди сторінки.** Загальна кількість переглянутих сторінок.

1.4 Огляд конкурентів Google Analytics

Програмне забезпечення цифрової аналітики є широко використовуваною технологією, і багато людей шукають продуктивні, потужні програмні рішення з a/b-тестуванням, сегментацією та мобільними звітами. Інші важливі фактори, які слід враховувати під час дослідження альтернатив Google Analytics, включають маркетингові кампанії та дані користувачів. Далі буде розглянуто список рішень, які рецензенти визнали як найкращі загальні альтернативи та конкуренти Google Analytics.

1.4.1 Сервіс «Matomo»

Matomo (раніше Piwik) показаний на рис. 1.1. дозволяє імпортувати історичні дані Google Analytics і служить інструментом веб-аналітики з повним набором послуг [11, 12].

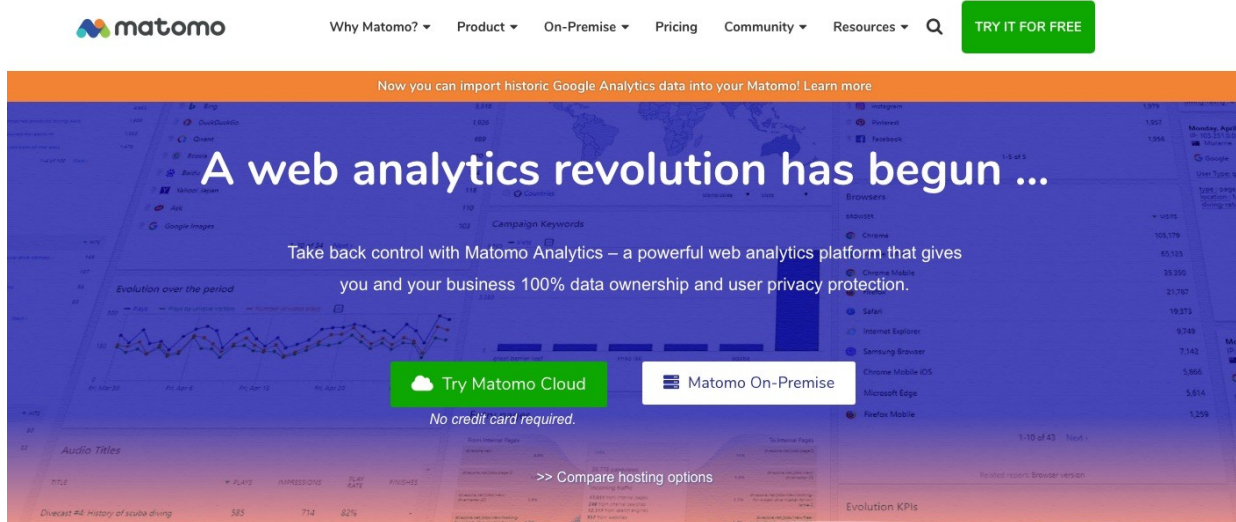


Рис 1.1. Сервіс «Matomo»

Сервіс надає теплові карти, А/В-тестування, послідовності переходів і цілі, а також багато інших функцій оптимізації конверсій [11, 12].

Існує можливість розмістити Matomo на власних серверах, якщо клієнт турбується про конфіденційні дані, але вони також пропонують хмарний хостинг на серверах, розташованих у Франції [11, 12].

Хоча на момент написання МКР сервіс ще не доступний, запланована інтеграція з Google Ads, Bing Ads і Facebook Ads. У Matomo також немає обмежень щодо даних [11, 12].

Існує безкоштовна пробна версія з цінами на продукт від 19 доларів США на місяць для спрощеного використання та користувацькими цінами для корпоративного використання [11, 12].

1.4.2 Сервіс «Woopra»

Woopra – це інструмент, який надає інформацію для продуктів, маркетингу, продажів і команд підтримки, оскільки сервіс відстежує користувачів протягом усього їхнього досвіду на сайті [13, 14]. Вигляд домашньої сторінки сервісу показаний на рис. 1.2.

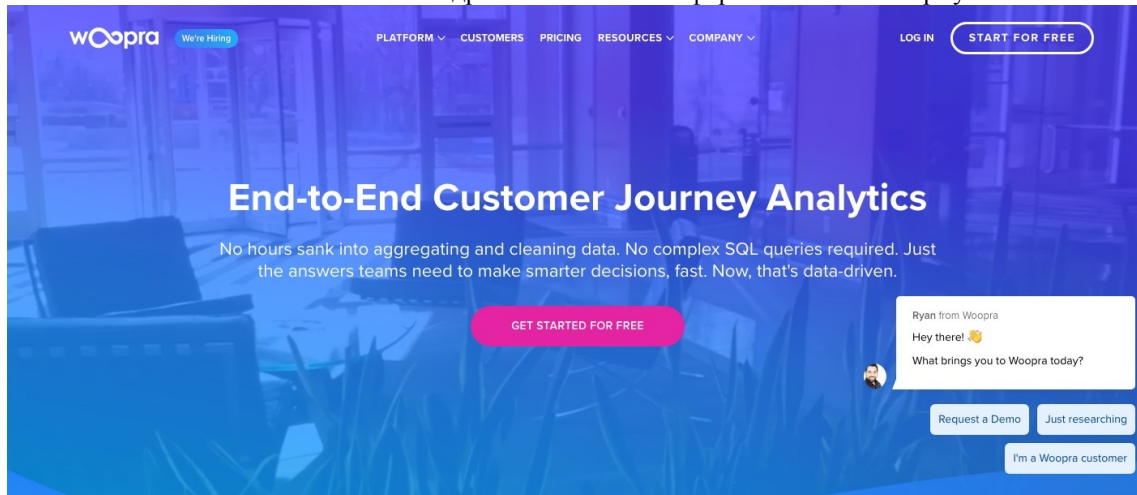


Рис. 1.2. Сервіс «Wopra»

Сервіс має спеціальні можливості відстеження та більше ніж 50 інтеграцій з такими продуктами, як Salesforce, Google Ads, Google Drive, Hubspot і Campaign Monitor [13, 14].

Сервіс також може виконувати дії в режимі реального часу «на льоту», як от надсилання повідомлення на канал продажів Slack (корпоративний «месенджер»), коли новий користувач реєструється [13, 14].

Існує безкоштовна пробна версія та безкоштовний план із більш надійними планами, починаючи з 999 доларів США на місяць [13, 14].

1.4.3 Сервіс «GoSquared»

GoSquared надає аналітику та живий чат, щоб підвищити продажі. Клієнт може перевірити активність перегляду, яка привела користувача на сайт, і, якщо користувач дасть свою адресу електронної пошти, можна побачити більше про його соціальну активність [15, 16]. Сервіс показаний на рис. 1.4.

Grow your subscription business with us.

Award-winning software to understand and engage with your customers.



Engage

Effortless personalised multi-channel messaging.

[Learn more →](#)



Live Chat for Sales

Convert website visitors into customers.

[Learn more →](#)



Web Analytics

Simple, privacy-focused website analytics.

[Learn more →](#)

Trusted by thousands of businesses.



Рис. 1.3. Сервіс «GoSquared»

Існує можливість придбати лише Analytics або Live Chat або об'єднати їх у пакет Suite із ціною від 79 доларів США на місяць за пакет [15, 16].

GoSquared також пропонує Центр даних клієнтів і автоматизацію для компаній SaaS. Для використання програмного забезпечення необхідно встановити код Javascript на своєму сайті [15, 16].

Сервіс має інтеграцію з Salesforce, Slack, Zapier, Twitter, Drip тощо. Модулі Analytics і Live Chat мають невеликий безкоштовний план [15, 16].

1.4.4 Сервіс «FoxMetrics»

FoxMetrics зберігає персональні вихідні дані в «сховищі даних», до якого клієнт має доступ, з можливістю створювати звіти за допомогою SQL [17, 18]. Сервіс показаний на рис. 1.4.

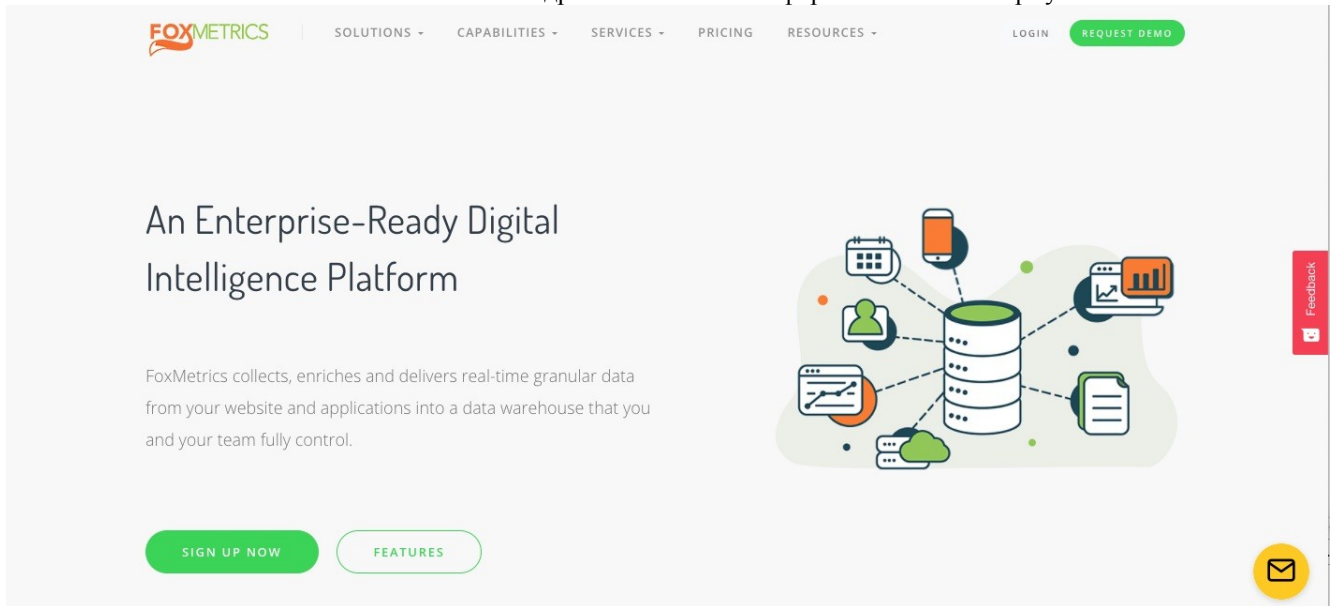


Рис. 1.4. Сервіс «FoxMetrics»

Завдяки можливості використовувати SQL для доступу до даних, клієнт може створювати конкретні запити, щоб отримати точну інформацію і не прив'язуватися до звичайної пропонованої звітності [17, 18].

Клієнт може використовувати бібліотеку Javascript від FoxMetrics або використовувати продукт під назвою Segment, щоб отримати потрібні дані [17, 18].

Цей продукт дозволяє компаніям електронної комерції виконувати все звичне (відстежувати й сегментувати клієнтів, аналізувати залишені кошики тощо), але він також допомагає оптимізувати спливаючі вікна та опитування на сайті клієнтів [17, 18].

Сервіс пропонує двотижневу безкоштовну пробну версію з щомісячним обслуговуванням від 450 доларів США [17, 18].

1.4.5 Сервіс «Mixpanel»

Mixpanel який показаний на рис. 1.5., дозволяє глибоко проаналізувати поведінку користувачів [19, 20].

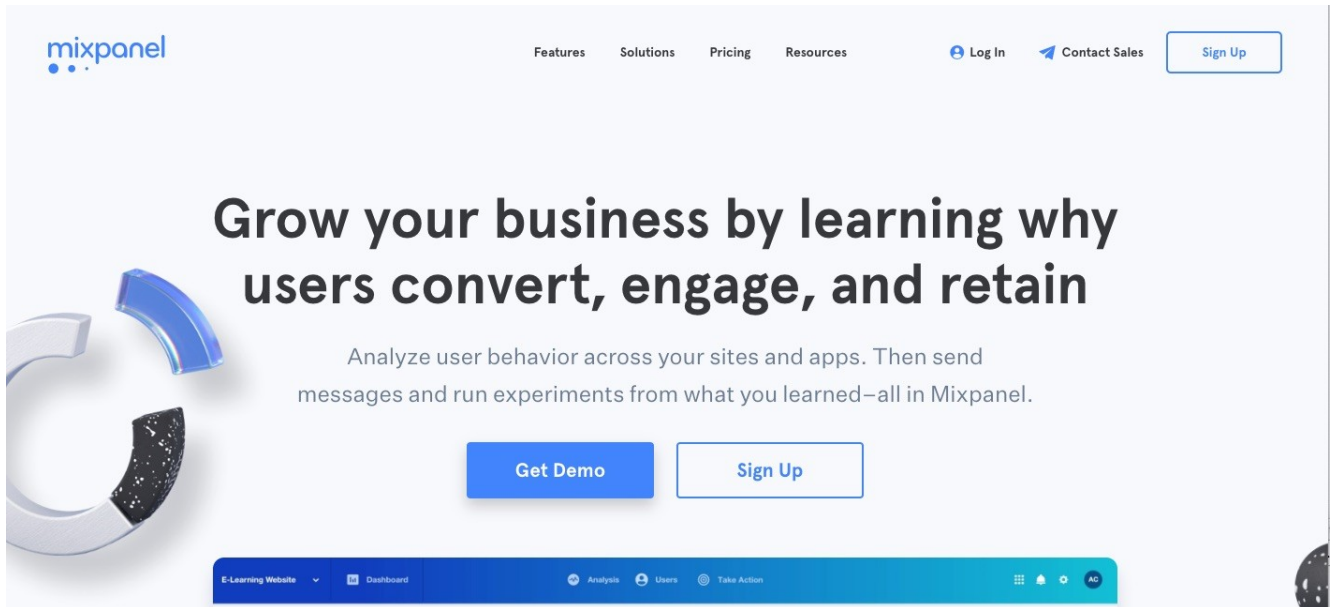


Рис. 1.5. Сервіс «MixPanel»

Сервіс дає можливості [19, 20]:

- «Підштовхнути» користувачів повідомленнями на основі заданої поведінки в реальному часі.
- Проаналізувати прогностні моделі даних, щоб побачити, які користувачі з більшою ймовірністю здійснили конверсію.
- Отримувати сповіщення, коли конкретні показники змінюються разом із інформацією про користувачів, які спричиняють ці зміни.

Існує безкоштовний пакет з пакетом наступного рівня, починаючи з 779 доларів США на рік [19, 20].

Висновки до розділу 1

У даний час представлено багато сервісів з аналітики «web-сайтів». Різні сервіси надають різноманітний функціонал для своїх користувачів. Однак, якщо глибше зануритись у сферу аналітики «web-сайтів», можна швидко зрозуміти, що Google Analytics має значні переваги перед конкурентами в тому, що:

- Google – це всесвітньо впізнаваний бренд;
- Google analytics має більший спектр можливостей, ніж конкуренти;
- Google analytics є безкоштовним;
- Google analytics вже має всі інтеграції із сервісами Google, що полегшує переміщення даних;
- В Google analytics досить легка інтеграція на сайт користувача.

Перевагою створюваного веб-додатку являється те, що додаток буде створений спеціально для e-commerce платформи Shopify, враховуючи нюанси роботи даної платформи.

1.

2 ФУНКЦІОНАЛ СИСТЕМИ АНАЛІТИКИ GOOGLE ANALYTICS ТА МЕНЕДЖЕРА ТЕГІВ GOOGLE TAG MANAGER

2.1 Налаштування Google Tag Manager (GTM)

Google Tag Manager – це безкоштовна система керування тегами від Google.

Принцип роботи простий: GTM бере всі дані на веб-сайті та надсилає їх на інші платформи, такі як Facebook Analytics і Google Analytics.

Це також дозволяє легко оновлювати та додавати теги до коду Google Analytics без необхідності вручну писати код на веб-сайті, заощаджуючи час, а також зникають можливі проблеми у майбутньому [21].

Наприклад, користувач хоче мати можливість відстежувати, скільки людей натиснуло посилання PDF, яке можна завантажити. Без GTM користувачу доведеться вручну змінити всі посилання для завантаження, щоб зробити це. Однак якщо користувач використовує GTM, він може просто додати новий тег до GTM, щоб відстежувати завантаження [21].

Спочатку потрібно створити обліковий запис на інформаційній панелі GTM.

Add a New Account

1 Setup Account

Account Name

e.g. My Company

Country

United States

Share data anonymously with Google and others ⓘ

Continue

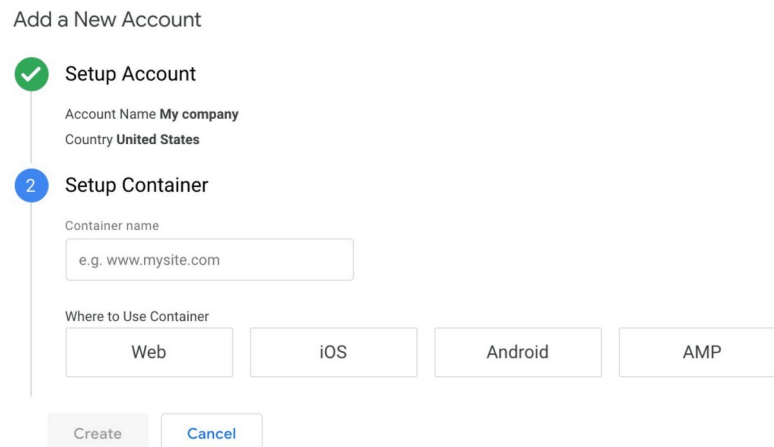
2 Setup Container

Create Cancel

Рис. 2.1 Створення аккаунту для інформаційної панелі GTM

На даному етапі потрібно ввести ім'я облікового запису та натиснути «Continue».

Далі необхідно налаштувати контейнер, який, по суті, є сховищем, що містить усі «макроси, правила та теги» для веб-сайту клієнта [22].



Add a New Account

✓ Setup Account
Account Name **My company**
Country **United States**

2 Setup Container
Container name
e.g. www.mysite.com

Where to Use Container

Web iOS Android AMP

Create Cancel

Рис. 2.2 Налаштування контейнера GTM

На рис. 2.2. видно, що потрібно дати контейнеру описову назву та вибрати тип вмісту, з яким він буде пов'язаний (веб, iOS, Android або AMP) [22].

По закінченню потрібно натиснути «Create», переглянути «Умови надання послуг» та прийняти їх [22].

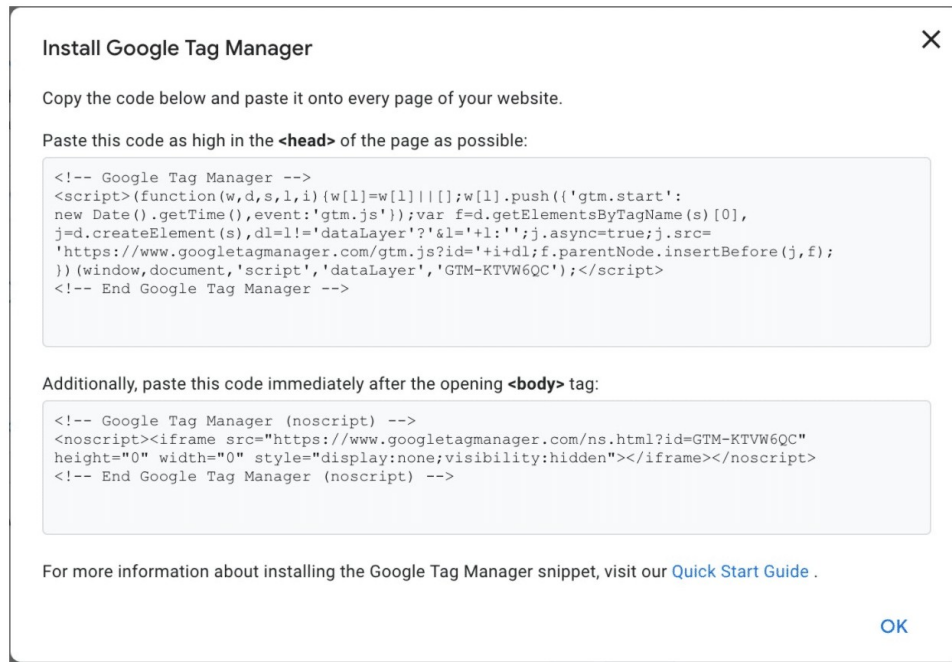


Рис. 2.3. Фрагмент коду встановлення GTM

На рис. 2.3. продемонстровано фрагмент коду, який користувачу потрібно додати в код свого веб-сайту, щоб керувати своїми тегами. Для цього необхідно скопіювати та вставити два фрагменти коду на кожну сторінку веб-сайту. Як сказано в інструкції, перший необхідно додати в тег `<head>`, а другий — після відкриття тегу `<body>` [22].

2.2 Налаштування Google Analytics

Як і Google Tag Manager, потрібно створити обліковий запис Google Analytics, зареєструвавшись на сервісі Google Analytics [23].

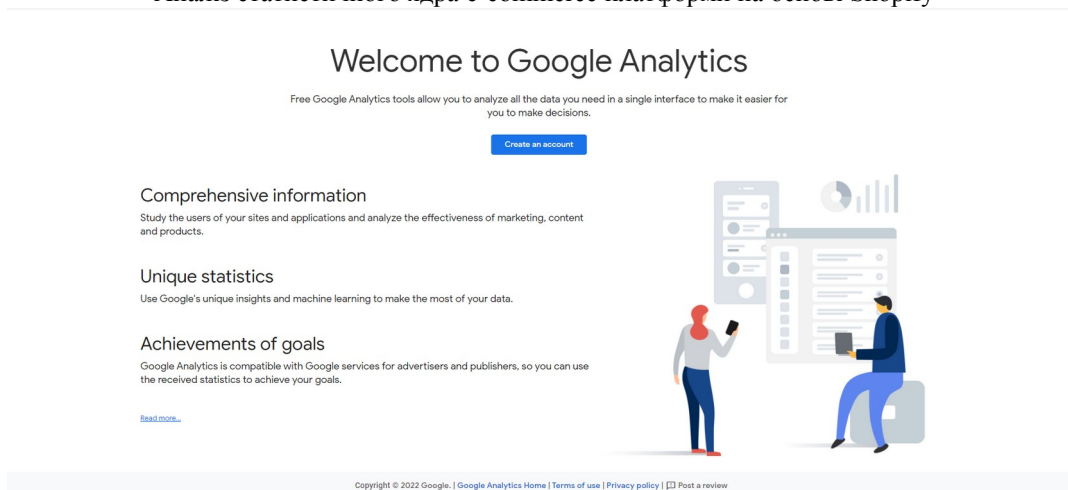


Рис. 2.4. Вітальне вікно перед реєстрацією на платформі Google Analytics

Як показано на рис. 2.5., необхідно ввести назву свого облікового запису та веб-сайту, а також URL-адресу веб-сайту. Важливо вказати галузеву категорію веб-сайту та часовий пояс, за яким будуть формуватися звіти [23].

Рис. 2.5. Реєстраційна форма Google Analytics

Зробивши вище наведене необхідно прийняти «Умови використання», щоб отримати ідентифікатор відстеження, приклад показаний на рис. 2.6.

Кафедра інтелектуальних інформаційних систем
Аналіз статистичного ядра е-commerce платформи на основі Shopify

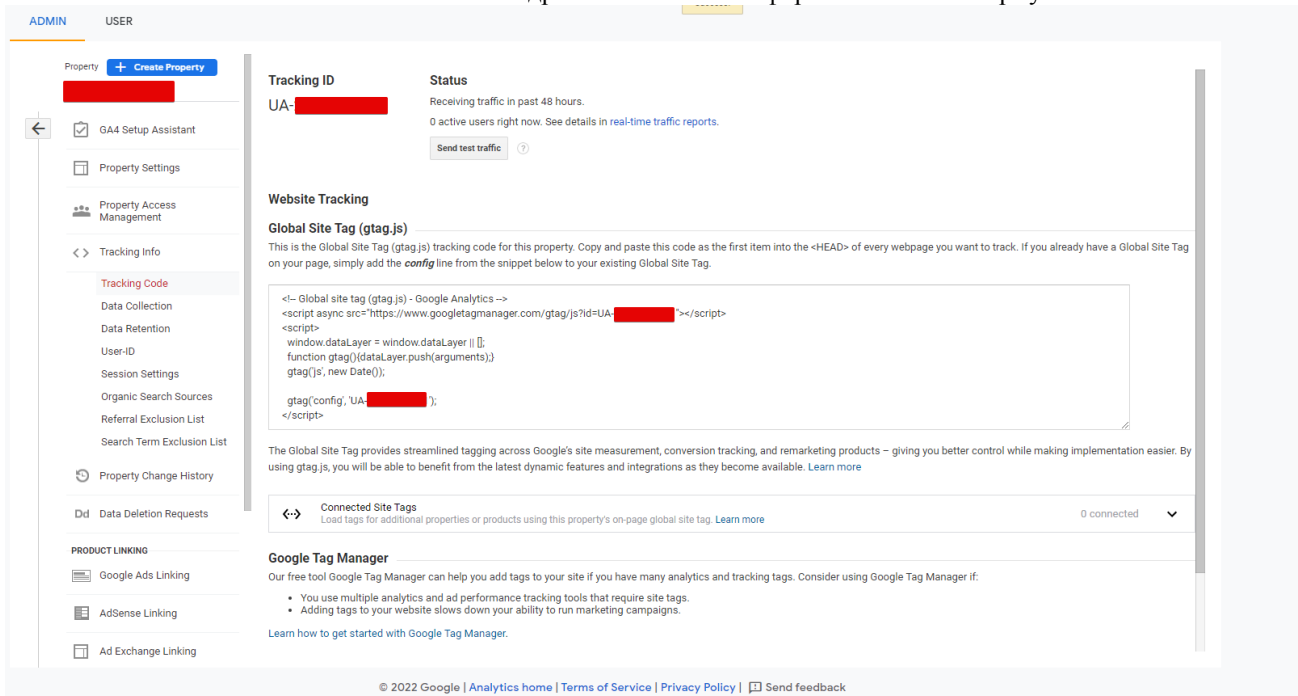


Рис. 2.6. Ідентифікатор та код відстеження веб-сайту

Ідентифікатор відстеження – це рядок чисел, який повідомляє Google Analytics що треба надіслати аналітичні дані. Це число, яке виглядає як UA-000000-1. UA розшифровується як Universal Analytics (універсальна аналітика). Перший набір чисел (000000) – це номер особистого рахунку, а другий набір (1) – номер власності, пов'язаний з обліковим записом, тобто Google Analytics знає, який аккаунт намагається запросити дані, власник, менеджер і т.д [23].

Це унікально для будь-якого веб-сайту та особистих даних користувачів, тому нікому не потрібно повідомляти ідентифікатор відстеження [23].

2.3 Налаштування тегу аналітики за допомогою Google Tag Manger

Для початку необхідно перейти на інформаційну панель Google Tag Manger і натиснути на кнопку «Add a new tag» як показано на рис. 2.7.

Кафедра інтелектуальних інформаційних систем
 Аналіз статистичного ядра e-commerce платформи на основі Shopify

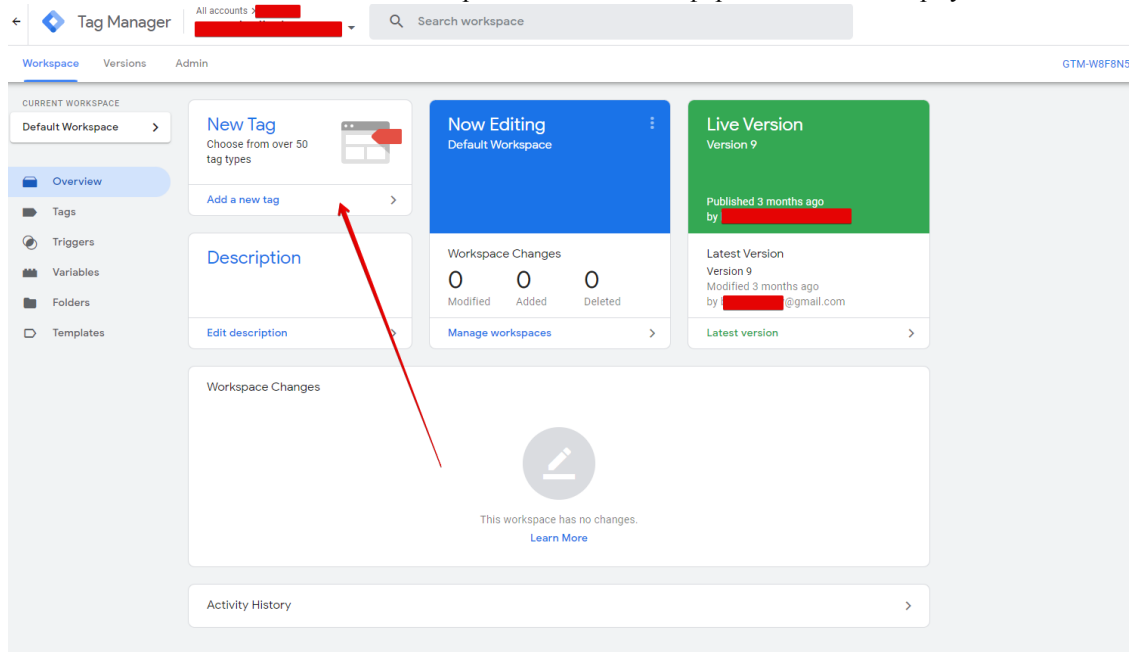


Рис. 2.7. Додавання нового тегу в Google Tag Manger

Після натискання, відкриється спливаюче вікно, де можна створити новий тег веб-сайту [24].

У вікні, що відкрилося, будуть наступні налаштування для тегу, що додається [24], які показані на рис 2.8:

- **Конфігурація.** Куди підуть дані, зібрані тегом.
- **Запуск.** Який тип даних буде відстежуватись.

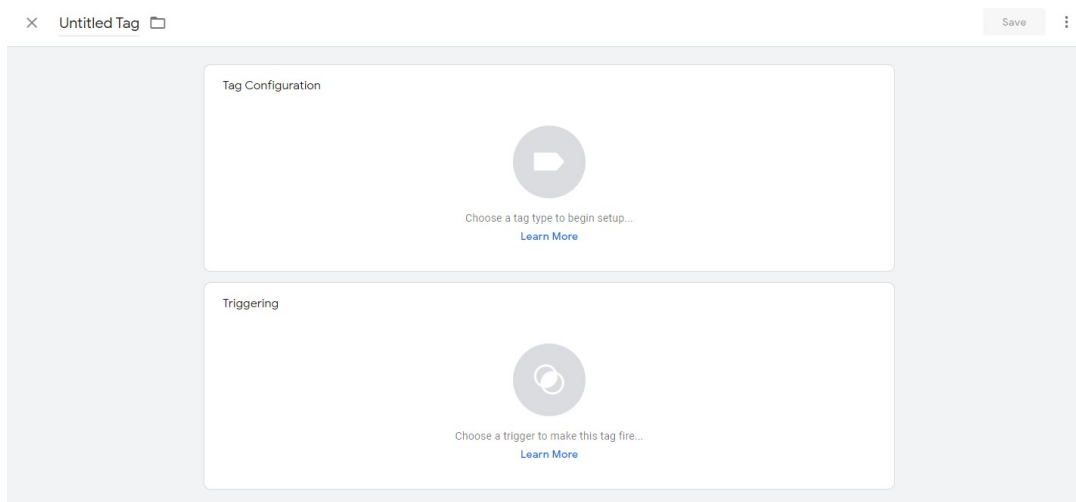


Рис. 2.8. Інформація про конфігурацію та активацію в Google Tag Manger

Для того, щоб вибрати тип тегу, який потрібно створити, необхідно натиснути на кнопку «Tag Configuration» [24]. Щоб створити тег Google Analytics, потрібно вибрати параметр «Universal Analytics» як показано на рис 2.9.

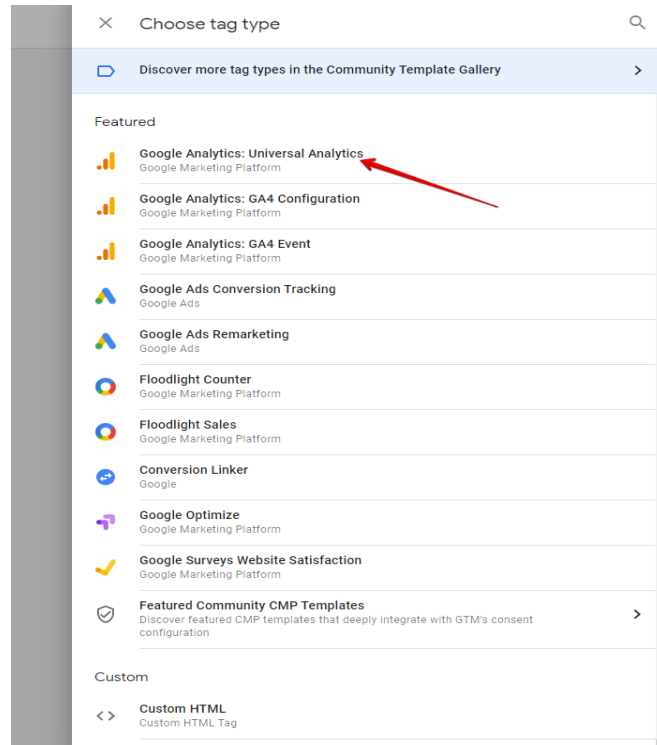


Рис. 2.9. Кнопка конфігурації тегу Universal Analytics

Після натискання на нього буде надана можливість вибрати тип даних, який потрібно відстежувати. Далі потрібно перейти до «Налаштування Google Analytics» і обрати «New variable...» зі спадного меню як показано на рис. 2.10. [24]

Кафедра інтелектуальних інформаційних систем
Аналіз статистичного ядра e-commerce платформи на основі Shopify

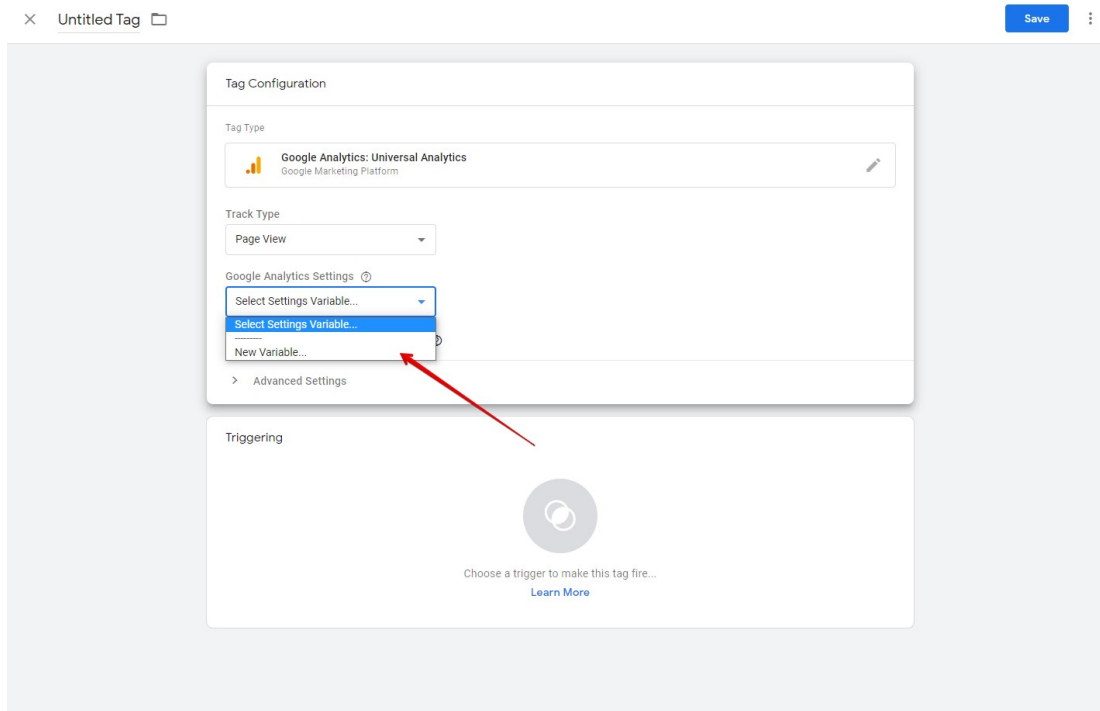


Рис. 2.10. Нова змінна налаштування Google Analytics

Як можна бачити на рис. 2.11., після натискання на кнопку «New variable...» відкриється нове вікно, де буде можливість ввести свій ідентифікатор відстеження Google Analytics. Це надішле дані веб-сайту прямо в Google Analytics, де можна їх побачити як тільки данні будуть зібрані [24].

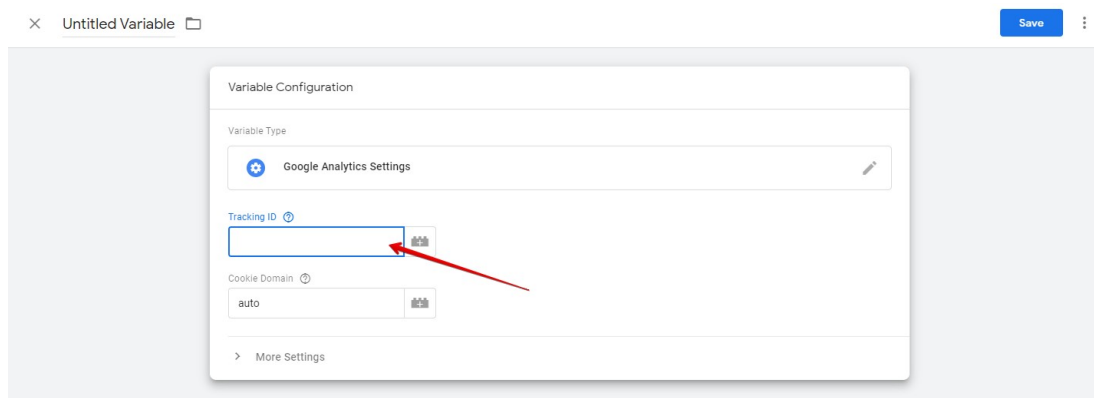
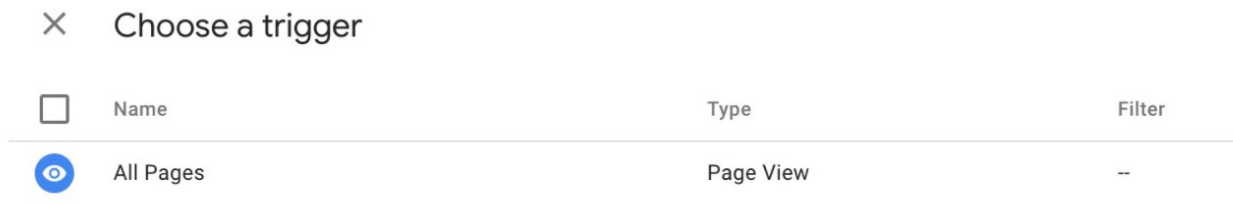


Рис. 2.11. Введення ідентифікатору відстеження Google Analytics

Після цього необхідно перейти до розділу «Triggering», щоб вибрати дані, які потрібно надіслати в Google Analytics.

Як і у випадку з «Configuration», необхідно натиснути кнопку «Triggering», щоб перейти на сторінку «Chose a trigger». Далі необхідно обрати «All pages», щоб тригер надсилав дані з усіх веб-сторінок [24].



Р

ис. 2.12. Вибір тригера, який буде відправляти обраний тип даних в Google Analytics

Кінцевий результат створення тегу має виглядати приблизно так як показано на рис 2.13:

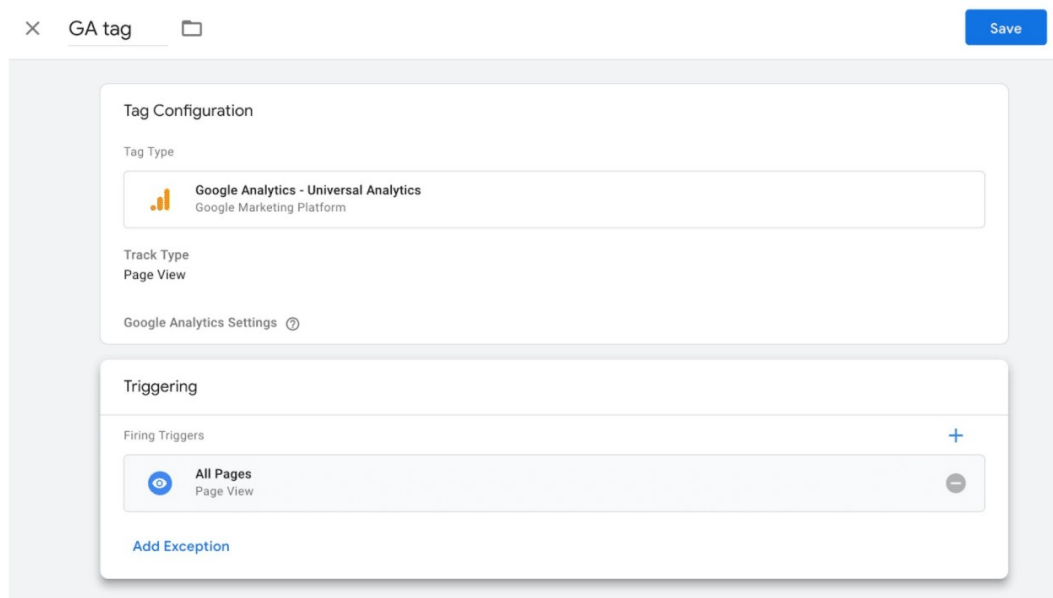


Рис. 2.13. Кінцева сторінка тегу Google Analytics

В кінцевому результаті необхідно просто натиснути на кнопку «Save» і новий тег для Google Tag Manager буде створено і додано на веб-сайт, який буде надсилати зібрані дані про користувачів веб-сайту до Google Analytics [24].

2.4 Налаштування цілей Google Analytics

Хоча клієнт, напевно, знає ключові показники ефективності для власного веб-сайту та бізнесу, Google Analytics ні, тому потрібно надати Google інформацію, що саме вважається успіхом для веб-сайту. Для цього необхідно встановити цілі на інформаційній панелі Google Analytics [25].

Для початку треба натиснути кнопку «Admin» у нижньому лівому куті як показано на рис. 2.14.

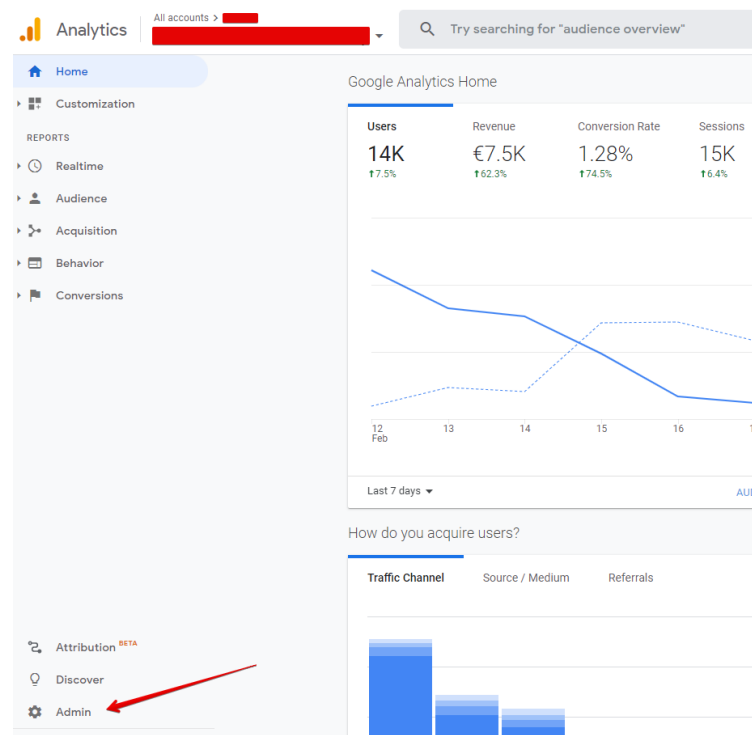


Рис. 2.14. Кнопка «Admin»

Після натискання буде виконано перехід в інше вікно, де ви можна знайти кнопку «Goals» як показано на рис. 2.15. [25]

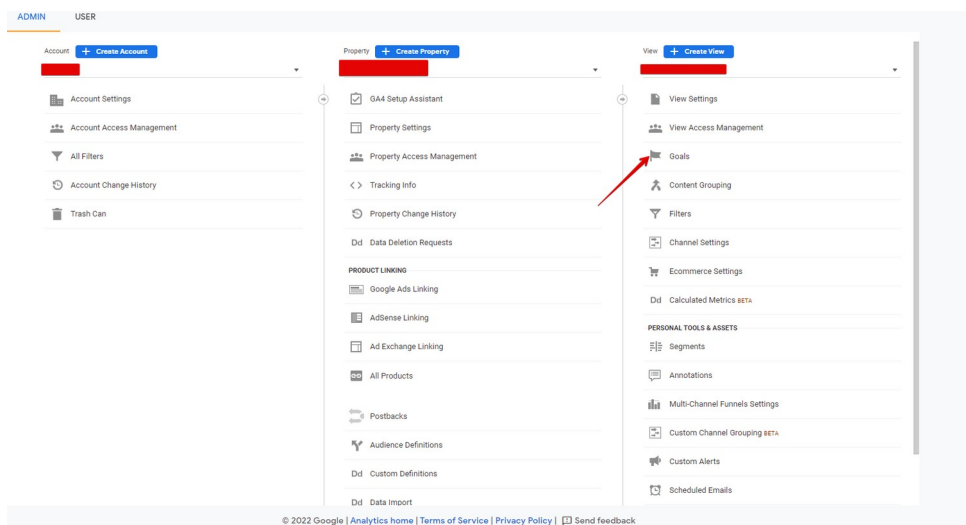


Рис. 2.15. Кнопка «Goals»

Натиснувши кнопку як показано на рис. 2.15, буде відкрита інформаційна панель «Goals» де буде надана можливість створити нову ціль як показано на рис. 2.16. [25]

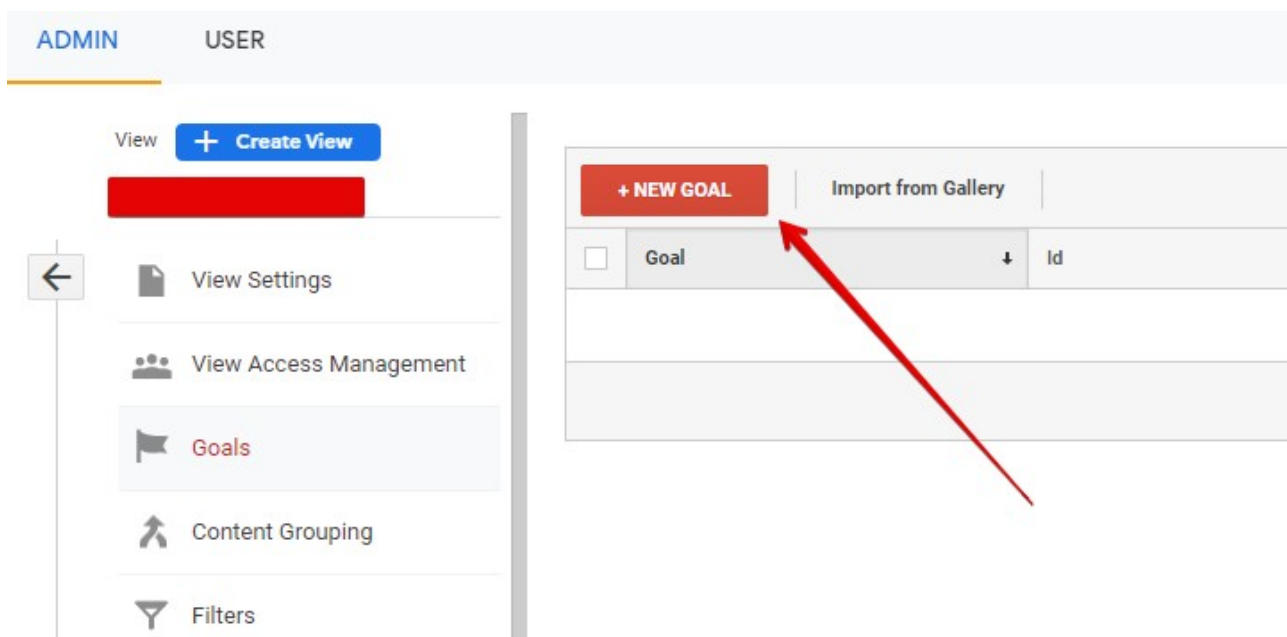


Рис. 2.16. Кнопка «New goals»

Звідси користувач зможе переглядати різні шаблони цілей, щоб побачити, чи відповідає один з них передбачуваний цілі. Далі потрібно буде вибрати тип цілі, яку потрібно досягти [25]:

- **destination.** Наприклад, якщо метою було перехід користувача на певну веб-сторінку;
- **duration.** Наприклад, якщо метою було, щоб користувачі проводили певну кількість часу на веб-сайті;
- **pages/screens per session.** Наприклад, якщо мета полягала в тому, щоб користувачі перейшли на певну кількість сторінок;
- **event.** Наприклад, якщо метою було змусити користувачів відтворити відео або натиснути посилання.

The image shows a 'Goal setup' interface with two main sections: 'Goal setup' (marked with a green checkmark) and 'Goal description' (marked with a blue '2').

- Goal setup:** Includes a 'Template:' label.
- Goal description:**
 - Name:** A text input field containing 'Duration'.
 - Goal slot ID:** A dropdown menu showing 'Goal Id 1 / Goal Set 1'.
 - Type:** A list of radio buttons:
 - Destination ex: thanks.html
 - Duration ex: 5 minutes or more
 - Pages/Screens per session ex: 3 pages
 - Event ex: played a video
 - Smart Goal Smart Goal not available.

At the bottom, there is a note: 'Measure the most engaged visits to your website and automatically turn those visits into Goals. Then use those Goals to improve your Google Ads bidding. [Learn more](#)'

Рис. 2.17. Шаблон цілі що включає налаштування та опис

На рис. 2.17. видно, що надається можливість ще точніше визначити свої цілі, наприклад, визначити, скільки часу користувачі мають провести на веб-сайті, щоб вважати його успішним. В кінці необхідно зберегти цілі, тоді вони будуть приблизно виглядати як на рис. 2.18. [25]

<input type="checkbox"/>	Goal	Id	Success	Past 7 day conversions	Recording
<input type="checkbox"/>	Duration	Goal ID 1 / Goal Set 1	Success	0	<input checked="" type="checkbox"/>

19 goals left

Рис. 2.18. Зберегти ціль у Google Analytics

2.5 Підключення до Google Search Console

Google Search Console — це потужний інструмент, який допомагає маркетологам і веб-майстрам отримати безцінні показники та дані пошуку. З його допомогою можна робити такі речі, як [26]:

- дізнатись швидкість сканування веб-сайту через пошук;
- дізнатись коли Google аналізує веб-сайт;
- дізнатись, які внутрішні та зовнішні сторінки посилають на веб-сайт;
- дізнатись запити за ключовими словами, за якими ранжується веб-сайт в результатах пошукової системи.

Щоб налаштувати Google Search Console, необхідно натиснути на кнопку «Admin» в лівому куті головної панелі, як показано на рис. 2.19. [27]

Кафедра інтелектуальних інформаційних систем
Аналіз статистичного ядра е-commerce платформи на основі Shopify

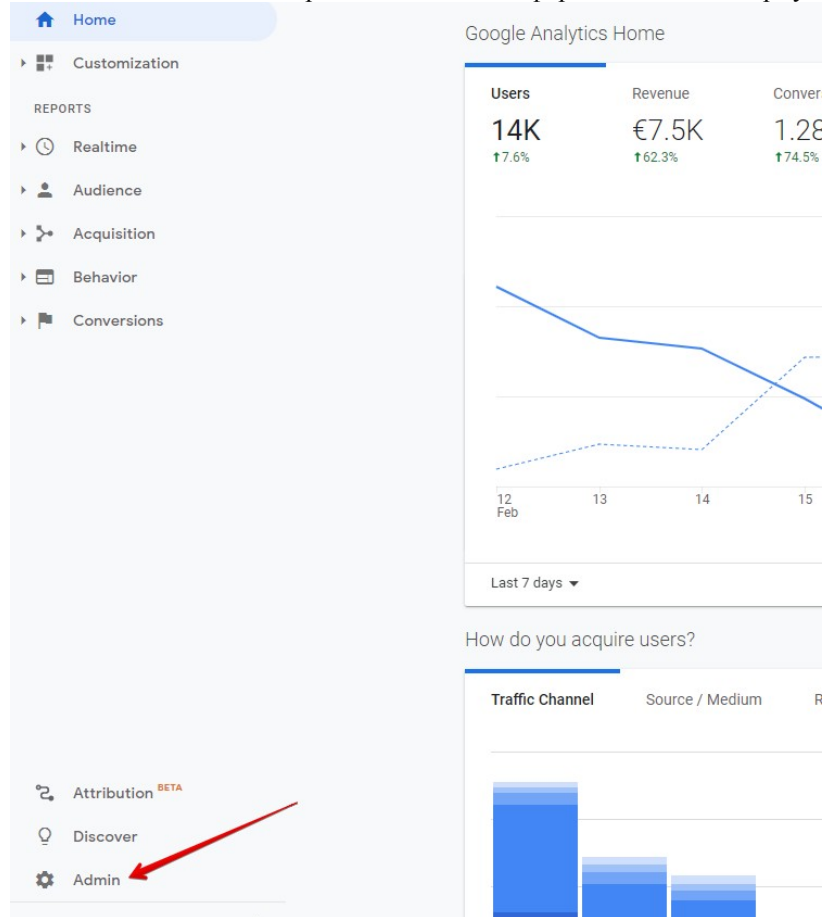


Рис. 2.19. Кнопка «Admin»

У вікні, що відкрилося, потрібно натиснути на кнопку «Property settings» яка показана на рис. 2.20, у середній колонці, щоб перейти до вікна додавання веб-сайту в Google Search Console [27].

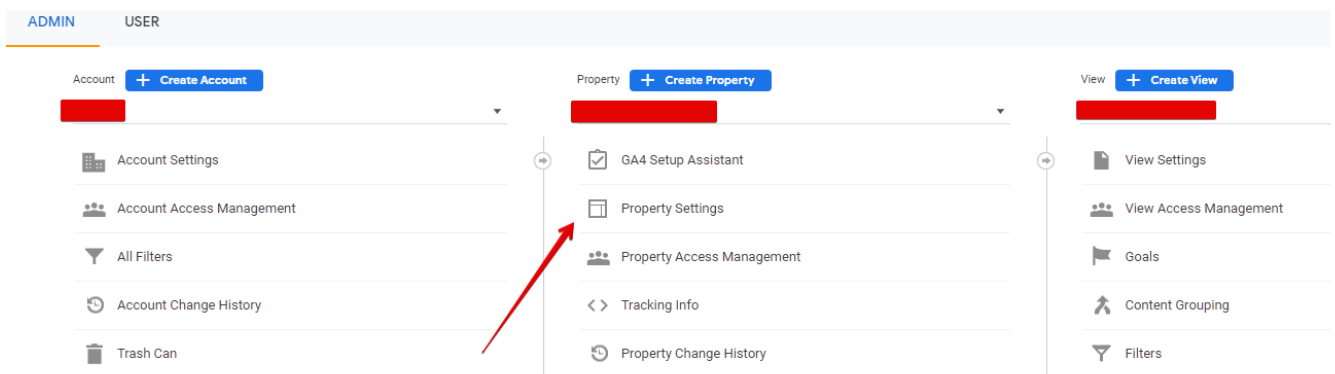


Рис. 2.20 Кнопка «Property settings»

У вікні, що відкрилось на панелі справа, необхідно прокрутити сторінку донизу і натиснути кнопку «Adjust Search Console», як показано на рис. 2.21. [27]

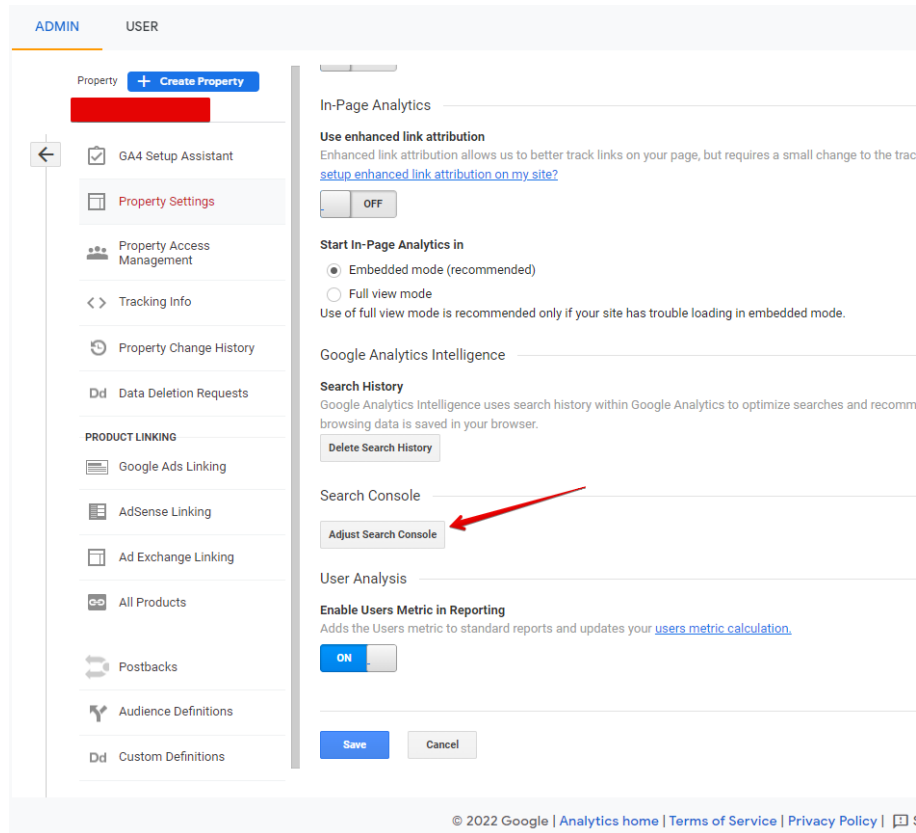


Рис. 2.21. Кнопка «Adjust Search Console»

На рис. 2.22. показано, що необхідно натиснути на кнопку «Add», яка призведе до того, що аккаунт до якого підключена аналітика буде вважатись власником веб-сайту [27].

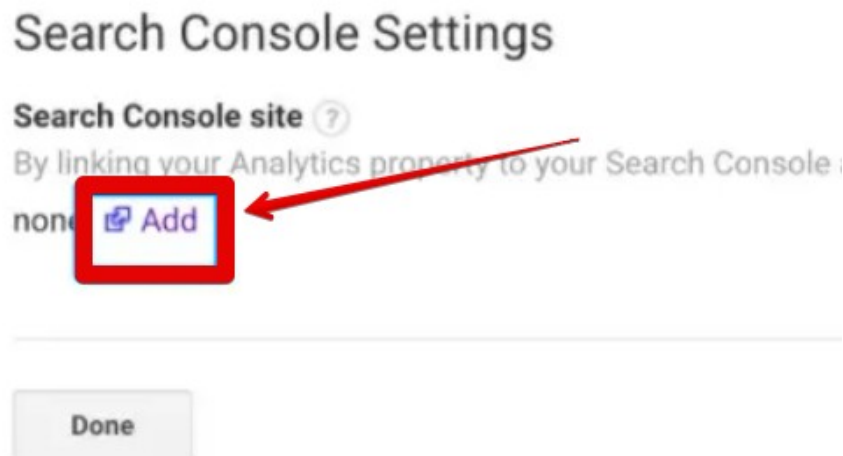


Рис. 2.22. Кнопка «Add»

Для того, щоб додати новий веб-сайт, потрібно натиснути на кнопку «Add a site to Search Console» як це показано на рис. 2.23.

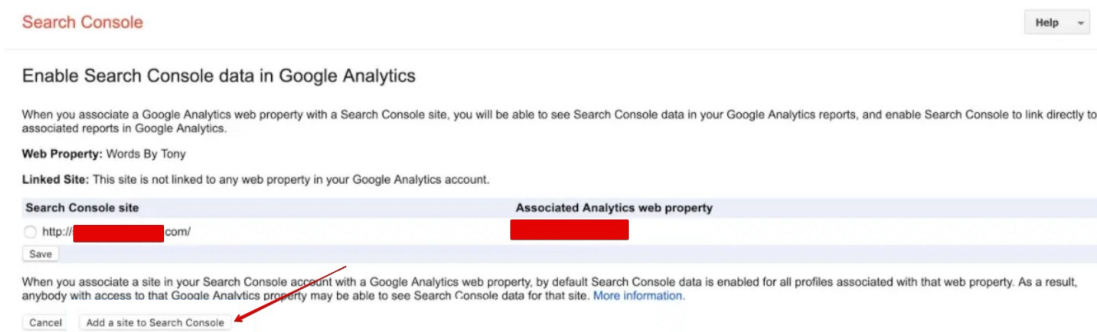


Рис. 2.23. Кнопка «Add a site to Search Console»

Після виконання попереднього кроку, повинно було відкритись нове вікно, яке показано на рис. 2.24., в якому потрібно вказану URL адресу бажаного сайту и натиснути кнопку «Add», після цього веб-сайт буде доданий до Google Search Console [27].

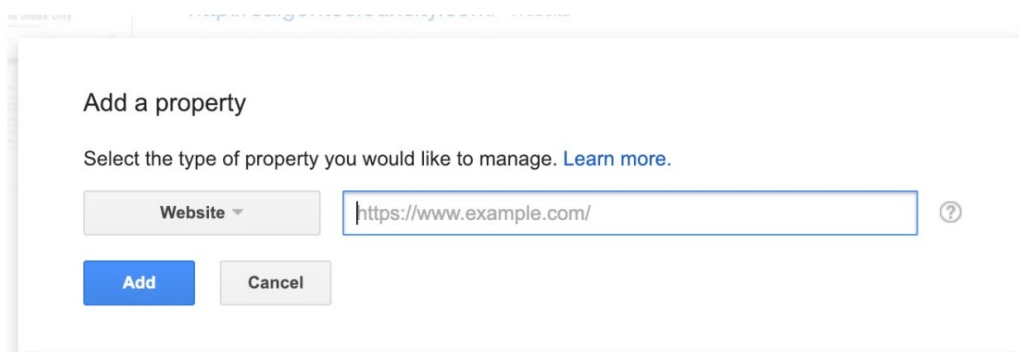


Рис. 2.24. Додавання нового веб-сайту до Google Search Console

Далі будуть надані інструкції, щоб додати HTML-код на веб-сайт. Після цього потрібно натиснути на кнопку «Save», що збереже виконані дії і перенаправить на аналітичну панель Google Analytics [27].

Слід зазначити, що після проведених маніпуляцій, дані з'являться не відразу. Все це тому що пошуковим роботам Google треба проіндексувати веб-сайт. Зазвичай, перевірка веб-сайту може бути виконана за термін від декількох днів до двох тижнів, отже слід періодично перевіряти дані в Google Search Console [27].

Висновки до розділу 2

Google Analytics необхідний для будь-якого цифрового маркетолога. Це допоможе відстежувати ефективність веб-сайту та всіх цифрових маркетингових кампаній. З його допомогою можна визначити рентабельність інвестицій і дізнатися більше про свою аудиторію. З однієї інформаційної панелі можна керувати всіма профілями в соціальних мережах та відстежувати дані вебсайту.

Існує широкий спектр даних, які можна відстежувати за допомогою Менеджера тегів Google і Google Analytics. Легко заплутатися в усіх показниках, які може відстежувати Google Analytics у поєднанні з Google Tag Manager. Для початківців рекомендується почати відстежувати тільки значущі показники, як тільки збільшиться досвід, слід поглиблюватись відстежуючи нові дані та ставлячи нові цілі.

3 РОЗРОБКА WEB-ДОДАТКУ ДЛЯ АНАЛІТИКИ ІНТЕРНЕТ МАГАЗИНУ

3.1 Інструментальні засоби для створення web-додатку. Мови програмування та розмітки тексту

Коли люди говорять про інструменти веб-розробки (або інструменти розробки в бізнесі), вони зазвичай мають на увазі програми та програмне забезпечення, яке дозволяє веб-розробникам тестувати та налагоджувати код та інтерфейс веб-сайту чи веб-програми [28].

Технічно цей термін не відноситься до інструментів, які насправді допомагають створити веб-сторінку або додаток. Але ця відмінність не допомагає. Веб-розробникам потрібен цілий ряд інструментів, які виходять за рамки налагодження та тестування – чи-то інструмент віддаленої співпраці, як-от Slack, інструмент проектування, як-от Figma, чи навіть просто онлайн-форум, як StackOverflow [28].

Веб-розробка – це всеосяжний термін, який позначає купу ролей. Веб-розробку можна розділити на два напрями: frontend та backend:

– **Fronend** – стосується всього, що користувач може бачити або з чим взаємодіяти – гарним прикладом є програма або веб-сайт. Мова йде про забезпечення першокласного користувацького досвіду та дружнього інтерфейсу (зазвичай із комбінацією HTML, CSS та різних бібліотек JavaScript) [29].

– **Backend** – стосується речей за лаштунками програм і веб-сайтів, які користувачі не можуть побачити. Наприклад, PHP для роботи з сервером та MySQL для роботи з базами даних [29].

У більшості випадків веб-розробники спеціалізуються на одному з двох напрямів. Однак є розробники, які обирають обидві спеціалізації одночасно – їх називають Full-stack розробник [29].

3.1.1 PHP

PHP (Hypertext Preprocessor) – це мова сценаріїв, призначена для розробки динамічного веб-контенту і виконується на стороні сервера. Через свій синтаксис він успадкував від мови програмування C, а завдяки своїй простоті PHP отримав велику популярність. Минав час, мова розширювалася і з'явилася можливість об'єктно-орієнтованого програмування. Він має можливість процедурного програмування, але також дозволяє використовувати класи та інші концепції об'єктно-орієнтованого програмування [30, 31].

PHP можна використовувати для програмування консольних додатків і графічних інтерфейсів (бібліотека PHP-GTK), а його основне використання полягає в програмуванні динамічних сторінок в Інтернеті [30, 31].

Сьогодні PHP використовують кілька сотень тисяч розробників і кілька мільйонів сайтів, включаючи деякі з найбільш відвідуваних сайтів у світі, такі як Facebook, Wikipedia, Yahoo, YouTube і Flickr [30, 31].

Програма, написана на PHP, не вимагає компіляції, але інтерпретується при кожному виконанні. Інтерпретатор PHP може працювати за принципом PHP CGI, тобто інтерпретатор буде існувати як зовнішня програма, яка виконує заданий скрипт щоразу, коли його запитує користувач, а також може бути встановлений як модуль веб-служби. На сьогоднішній день найбільше використовується другий варіант, оскільки він забезпечує значно вищу швидкість виконання – таким чином інтерпретатор завжди завантажується в пам'ять, і не потрібна наявність зовнішньої програми [30, 31].

Звичайний сценарій, за яким виконуються скрипти PHP, такий:

Користувач запитує сторінку PHP від сервера, сервер пересилає запит до веб-служби (програма веб-сервера на сервері), веб-сервер розпізнає, що запитується файл PHP, не надсилає його вміст клієнту, але виконує його як програму за допомогою модуля PHP, вихідний текст програми (стандартний вихід) відправляється клієнту, в результаті запиту клієнт розпізнає тип

результатів (HTML-код, зображення, вміст PDF, архів тощо), результат відображається клієнту відповідним чином [30, 31].

За останні роки мова програмування PHP була піднята на більш високий рівень. Існують пакети (php framework), які служать інструментами для швидшої та безпечнішої розробки Інтернет-додатків. Мета цих пакетів — об'єднати готові інструменти з однієї технології в практичний каркас, щоб отриманий пакет використовувався як один інструмент. Таким чином, розробник мав би у своєму розпорядженні основні функціональні можливості використовуваної технології і присвятив би себе лише логіці створеного ним додатка, проблемам організації файлів, класифікації об'єктів та безпеки [30, 31].

Найбільш відомі і популярні пакети PHP: Laravel, Phalcon, Symfony2, CodeIgniter, Yii, Aura, Zend Framework 1 і 2, Cake PHP, Kohana та багато інших [30, 31].

3.1.2 JavaScript

Javascript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки [32, 33, 34].

Це так звана мова об'єктних сценаріїв, основна мета використання якої – покращити динаміку певної сторінки. Насправді справа в тому, що під час використання HTML-коду його роль полягає лише у формуванні певних елементів сторінки, таких як текст, таблиці, посилання чи форми, а потім їх редагування, але цей код не має можливості динамічно керувати цими елементами. Саме тоді вступає JavaScript, який відіграє значну роль у створенні

поведінки цих елементів. А при поєднанні з HTML і CSS створюється так званий динамічний HTML (DHTML) [32, 33, 34].

Коли кажуть, що мова сценаріїв є об'єктною, це фактично означає, що людина, яка працює з нею, має вказати тип даних, а також визначити, яка саме функція буде застосована, яка має бути визначена на основі структури конкретних даних. Такий принцип призводить до того, що структура певних даних фактично починає розглядатися як об'єкт, всередині якого ці дані та певні функції розташовані [32, 33, 34].

Javascript складається з численних команд, які необхідно прочитати в спеціальній програмі під назвою Interpreter. Він служить для читання цих команд, не перекладаючи їх одночасно на так звану «машинну» мову. Точніше, команди зчитуються безпосередньо з коду, і тому можна сказати, що мова програмування JavaScript виконується на стороні клієнта. Простіше кажучи, все це робиться на комп'ютері користувача без відправки даних на сервер (якщо не потрібно отримувати дані з сервера) [32, 33, 34].

Вважається, що однією з найпопулярніших мов сценаріїв, що використовуються сьогодні, є JavaScript, і особливо добре, що він підтримується всіма браузерами Google Chrome, Safari, Mozilla Firefox, Opera, Edge, тощо [32, 33, 34].

Після офіційного створення World Wide Web Netscape фактично створив першу мову сценаріїв, основною метою якої було підвищення динаміки конкретних веб-сайтів або їх інтерактивності. Це була так звана мова сценаріїв Mocha, яку пізніше назвали LiveScript. Однак виникла велика плутанина, тому що люди думали, що Java – це те саме, що JavaScript. Сьогодні вважається, що ця компанія використовувала цю назву виключно в маркетингових цілях, але Microsoft все ж використала момент і розмістила так звану мову сценаріїв VBScript, і в той період так званий JScript був інтегрований в поточний Internet Explorer 3.0 . Щоб вирішити всі ці суперечки, компанія, яка створила JavaScript,

надіслала запит до Європейської організації зі стандартизації JavaScript ECMA (European Computer Manufacturers Association), і з тих пір стандартизована версія цієї мови називається ECMAScript [32, 33, 34].

Між Java і JavaScript майже немає подібності, за винятком нюансів, і вони вперше проявляються в синтаксисі, оскільки обидві мови програмування певною мірою дотримуються правил мови C. Однак між цими двома мовами сценаріїв є дуже помітні відмінності, і вони, в-першу чергу, виражаються в тому, що Java характеризується як об'єктно-орієнтована мова, тоді як JavaScript є суто об'єктною. Є також різниця в тому, що Java вимагає попереднього визначення типів змінних, тоді як це не стосується JavaScript [32, 33, 34].

3.1.3 HTML

HTML (англ. Hypertext Markup Language) – це код, який використовується для структурування веб-сторінки та її вмісту. Наприклад, вміст може бути структурований в межах набору абзаців, списку маркованих пунктів або за допомогою зображень і таблиць даних [35, 36].

HTML – це мова розмітки, яка визначає структуру вмісту сайту. HTML складається з ряду елементів, які використовуються, щоб укласти або обернути різні частини вмісту, щоб він виглядав певним чином або діяв певним чином. Теги можуть створювати гіперпосилання слова або зображення на інше місце, можуть виділяти слова курсивом, можуть збільшити або зменшити шрифт тощо. [35, 36] Наприклад, наступний рядок – «ІТ – професія майбутнього»

Якщо є задача, щоб рядок стояв сам по собі, можна вказати, що це абзац, уклавши його в теги абзацу:

```
<p> ІТ – професія майбутнього </p>
```

Основні частини наведеного елемента такі:

- **початковий тег:** складається з назви елемента (у даному випадку p), загорнутого у відкриваючу та закриваючу кутові дужки. Це вказує, де починається або починає діяти елемент — у наведеному випадку, де починається абзац;
- **закриваючий тег:** це те саме, що і початковий тег, за винятком того, що він містить косу риску перед назвою елемента. Це вказує, де закінчується елемент — у наведеному випадку, де закінчується абзац. Відсутність закриваючого тегу є однією зі стандартних помилок початківців і може призвести до дивних результатів;
- **вміст:** це вміст елемента, який у наведеному випадку є просто текстом;
- **елемент:** початковий тег, закриваючий тег і вміст разом складають елемент.

Елементи також можуть мати атрибути, які показані на рис. 3.1.



Рис. 3.1. Атрибут в HTML

Атрибути містять додаткову інформацію про елемент, яка не буде відображатись в фактичному вмісті. Тут class – це ім'я атрибута, а editor-note – значення атрибута. Атрибут class дозволяє надати елементу неунікальний ідентифікатор, який можна використовувати для його стилізації (та будь-які інші елементи з таким же значенням класу) [35, 36].

Атрибут завжди повинен включати в себе наступне:

- пробіл між ним і назвою елемента (або попереднім атрибутом, якщо елемент уже має один або кілька атрибутів);
- ім'я атрибута із знаком рівності;

– значення атрибута, укладене в лапки, що відкривають і закривають.

3.1.3 CSS

CSS (англ. Cascading Style Sheets) – розшифровується як каскадні таблиці стилів. Це мова таблиць стилів, яка використовується для опису зовнішнього вигляду та форматування документа, написаного мовою розмітки. Він надає додаткову функцію для HTML. Зазвичай він використовується з HTML для зміни стилю веб-сторінок та інтерфейсу користувача. Його також можна використовувати з будь-якими документами XML, включаючи звичайний XML, SVG і XUL [37, 38, 39].

CSS використовується разом з HTML і JavaScript на більшості веб-сайтів для створення інтерфейсів користувача для веб-програм і інтерфейсів для багатьох мобільних додатків [37, 38, 39].

Три основні переваги для використання CSS [37, 38, 39]:

1) CSS вирішує велику проблему. До CSS такі теги, як шрифт, колір, стиль фону, вирівнювання елементів, межі та розміри, потрібно було повторювати на кожній веб-сторінці. Це був дуже тривалий процес. Наприклад, якщо розробляти великий веб-сайт, де на кожній сторінці додається інформація про шрифти та кольори, це стане довгим і дорогим процесом. CSS був створений для вирішення цієї проблеми. Це була рекомендація W3C.

2) CSS економить багато часу. Визначення стилів CSS зберігаються у зовнішніх файлах CSS, тому можна змінити весь веб-сайт, змінивши лише один файл.

3) CSS надає більше атрибутів. CSS надає більш детальні атрибути, ніж звичайний HTML, для визначення зовнішнього вигляду веб-сайту.

3.1.4 SQL

SQL (Structured Query Language) означає структуровану мову запитів. SQL використовується для зв'язку з базою даних. Згідно з ANSI (Американський національний інститут стандартів), це стандартна мова для систем управління реляційними базами даних. Операції SQL використовуються для виконання таких завдань, як оновлення даних у базі даних або отримання даних із бази даних. Деякі поширені системи керування реляційними базами даних, які використовують SQL: Oracle, Sybase, Microsoft SQL Server, Access, Ingres тощо [40, 41].

Хоча більшість систем баз даних використовують SQL, більшість із них також мають власні фреймворки, які зазвичай використовуються лише в їхній системі. Однак стандартні команди SQL, такі як «Вибрати», «Вставити», «Оновити», «Видалити», «Створити» та «Скинути», можна використовувати для виконання майже всього, що потрібно зробити з базою даних [40, 41].

SQL управляє великою кількістю даних, особливо якщо є багато даних, які одночасно записуються, і є занадто багато транзакцій даних [40, 41].

Існують різні версії та фреймворки для SQL, найчастіше використовується MySQL. MySQL — це рішення з відкритим вихідним кодом, яке полегшує роль SQL в управлінні серверними даними для веб-додатків. Такі компанії, як Facebook, Instagram, WhatsApp тощо, використовують SQL для внутрішніх рішень для зберігання та обробки даних. Коли запит SQL записується та виконується (або аналізується), він обробляється оптимізатором запитів. Запит досягає сервера SQL, де він компілюється в три етапи [40, 41]:

- **parsing** – процес перевірки синтаксису;
- **binding** – процес перевірки семантики запиту;
- **optimization** – процес створення плану виконання запиту.

На третьому кроці генеруються всі можливі перестановки та комбінації, щоб знайти найефективніший план виконання запиту за розумний час. Чим коротший запит, тим він краще [40, 41].

3.2 Створення додатку аналітики для інтернет-магазину на платформі Shopify

Shopify призначений для того, щоб продавці почали працювати з «коробки». Але кожен продавець унікальний, тому вони використовують додатки для вирішення своїх конкретних завдань [42, 43].

У контексті Shopify програми є веб-додатками, які розширюють функціональність магазинів Shopify. Вони дозволяють продавцям адаптувати свій досвід Shopify відповідно до своїх потреб [42, 43].

За рівнем доступності додаток може мати два різні рівні [42, 43]:

- **загальнодоступні додатки:** програми, які доступні багатьом продавцям і можуть бути викладені в Shopify App Store. Ці додатки створюються на інформаційній панелі партнера;
- **приватні додатки:** програми, створені на замовлення для конкретного продавця і не можуть бути викладені в Shopify App Store. Ці додатки також створюються на інформаційній панелі партнера.

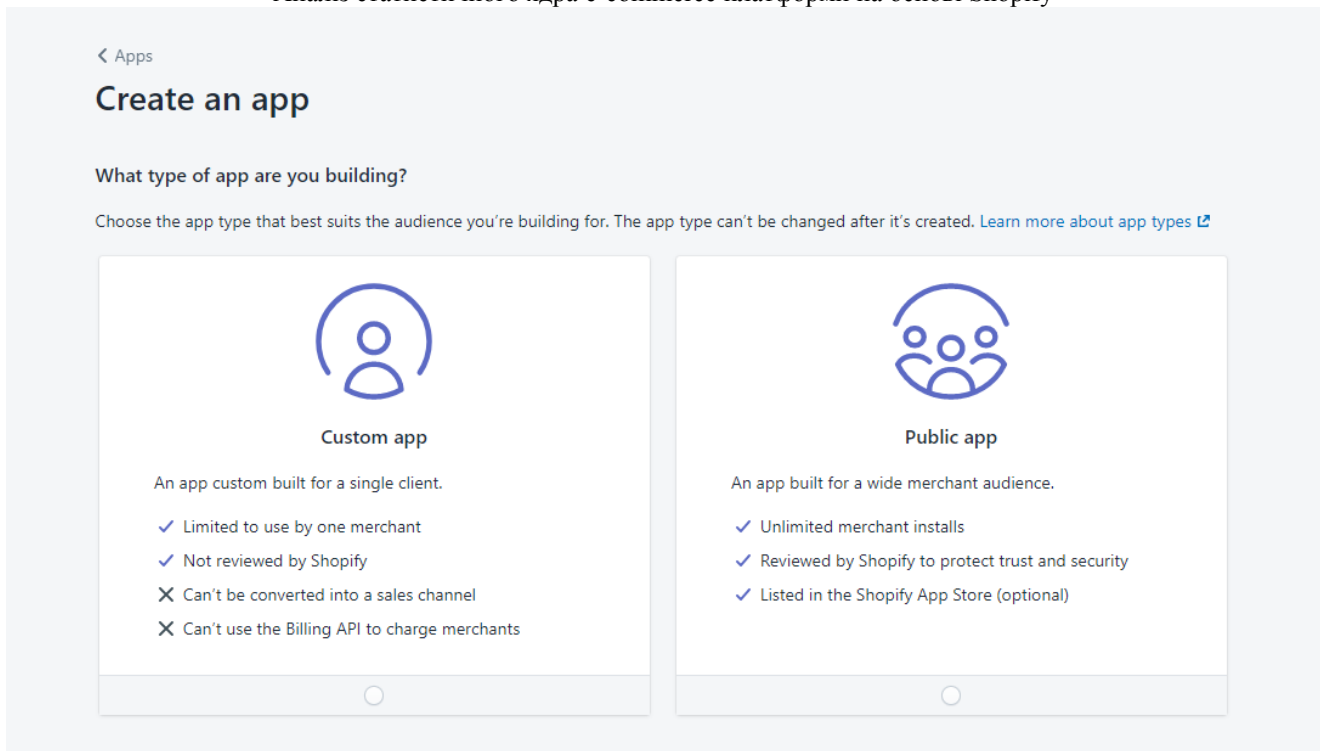


Рис. 3.2. Вибір доступності додатку перед створенням

Окрім свого типу, додатки можуть мати дві різні форми: окремі програми та вбудовані програми.

Автономні додатки мають свій власний інтерфейс користувача на власному веб-домені та не мають доступу до розширень програм. Вбудовані програми розміщуються у власній інфраструктурі, але глибше інтегруються з Shopify за допомогою Shopify App Bridge і відображаються безпосередньо в адміністраторі продавця Shopify або в точці продажу (POS) всередині iframe [42, 43].

Вбудовані додатки також дозволяють додавати функції до певних частин інтерфейсу користувача (UI) Shopify та онлайн-магазинів за допомогою розширень додатка. Коли використовується розширення додатка, Shopify передає інформацію в додаток, коли продавець взаємодіє з Shopify, а потім повертає додаток продавцеві через розширення [42, 43].

Після того, як був обраний тип додатку, що створюється, потрібно дати йому назву та заповнити два додаткові поля, які показані на рис. 3.3:

- **URL-адреса додатку:** URL-адреса, на якій розміщено додаток. Якщо офіційно не налаштований хостинг для додатку, можна використовувати ngrok (або подібне програмне забезпечення) для забезпечення середовища тестування;
- **URL-адреси переспрямування з білого списку:** це мають бути надійні URL-адреси, які додаток використовує для прийняття запитів від Shopify або для надсилання запитів до Shopify. Запити, що надходять з різних URL-адрес, не приймаються, навіть якщо інші облікові дані правильні.

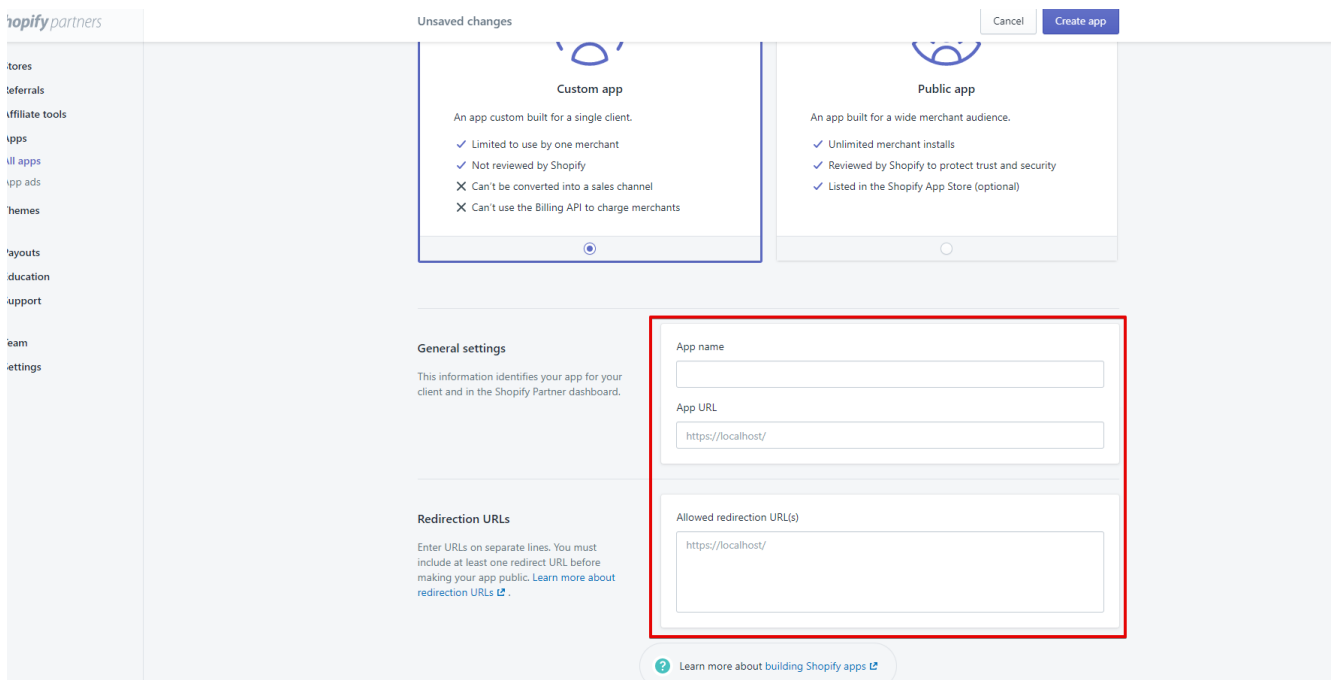


Рис. 3.3. Форма створення додатку на інформаційній панелі користувача

3.3 Розробка головної сторінки додатку

З будь-яким додатком повинен бути спосіб взаємодії. Додаток що створюється в даній роботі не є виключенням. Для того щоб користувачу було легше користуватись додатком, він повинен мати мінімалістичний дизайн в якому повинен бути пропрацьований інтерфейс користувача (UI), а також враховані головні вимоги до користувацького досвіду (UX), тобто додаток не повинен викликати труднощі при взаємодії з ним.

Кафедра інтелектуальних інформаційних систем
Аналіз статистичного ядра e-commerce платформи на основі Shopify

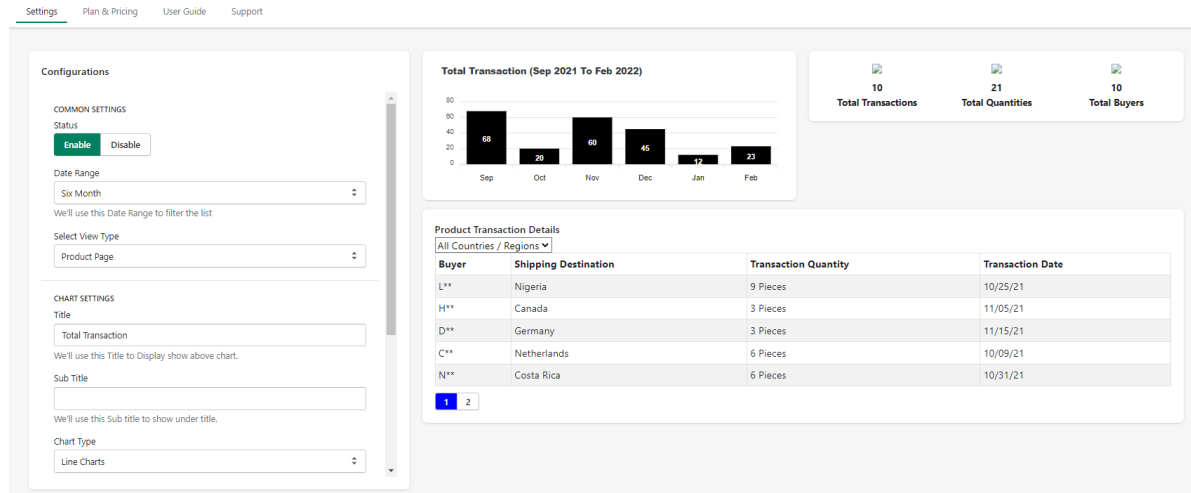


Рис. 3.4. Головна сторінка додатку

На рис. 3.4. можна бачити створений інтерфейс головної сторінки додатку. За «канонами UX», користувача не потрібно переправляти поміж багатьма сторінками, тому що у великому обсязі сторінок та інформації що там розміщена можна легко заплутатись, особливо, якщо користувачу необхідно провести певні дії і одразу побачити результат. Враховуючи вище описане, було прийнято рішення створити одну основну сторінку, на якій будуть розміщені налаштування додатка, а також інфографіка аналітики, яку збирає додаток.

Як показано на рис. 3.4, в лівій частині головної сторінки розміщено блок з налаштуваннями які відповідають за працездатність додатку, всі налаштування показані на рис.3.5.

Рис. 3.5. Блок налаштувань додатку

3.3.1 Група налаштувань «Common settings»

Для зручності користувача, всі налаштування були зібрані в одному місці, але розділені на блоки за категоріями.

Рис. 3.6. Група «Common settings» в блоці налаштувань

Першою є категорія «Common Settings» (загальні налаштування), яка показана на рис. 3.6. Ця категорія включає в себе три (п'ять, за умови, якщо для налаштування «Select view type» обраний варіант «Modal») налаштування(нь):

– **Status.** Це налаштування відповідає за те чи увімкнено додаток у даний момент. Ця функція буде дуже корисною коли, наприклад, потрібно буде провести тестові платежі та ін., щоб додаток не збирав ці дані. Як виглядає налаштування показано на рис. 3.7.

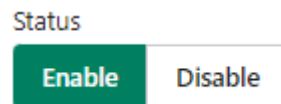


Рис. 3.7. Вигляд налаштування «Status»

– **Date range.** Це налаштування відповідає за період вибірки даних. Для вибору доступні чотири варіанти: три місяці, пів року, шість місяців і один рік, як показано на рис. 3.8.

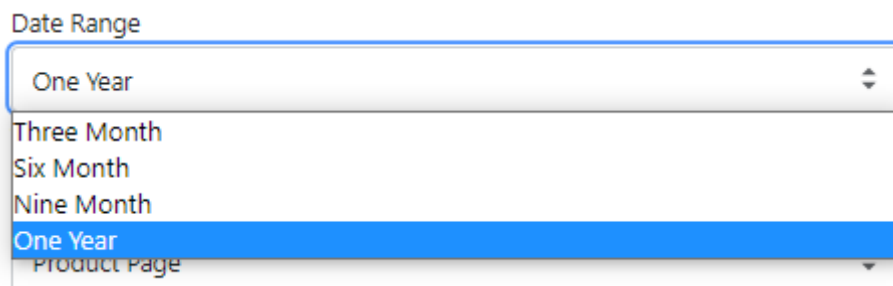


Рис. 3.8. Доступні варіанти вибору для налаштування «Data range»

– **Select view type.** Це налаштування відповідає за те, чи буде показуватись інформація аналітики безпосередньо на сторінці продукту. Налаштування має два варіанти: «Product page» та «Modal» які показані на рис. 3.3.1.4. Якщо обрати «Modal», вигляд спливаючого вікна на сторінці продукту буде виглядати як показано на рис. 3.3.1.5., але якщо лишити варіант «Product page», тоді аналітика буде виводитись лише на головній сторінці додатку;

Select View Type

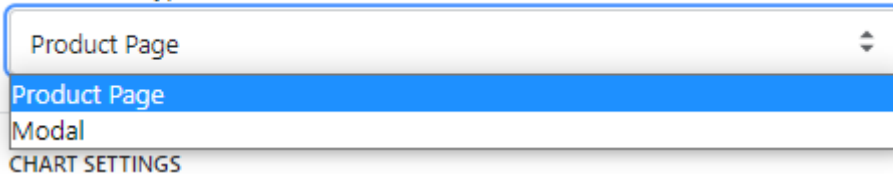


Рис. 3.9. Доступні варіанти вибору для налаштування «Select view type»

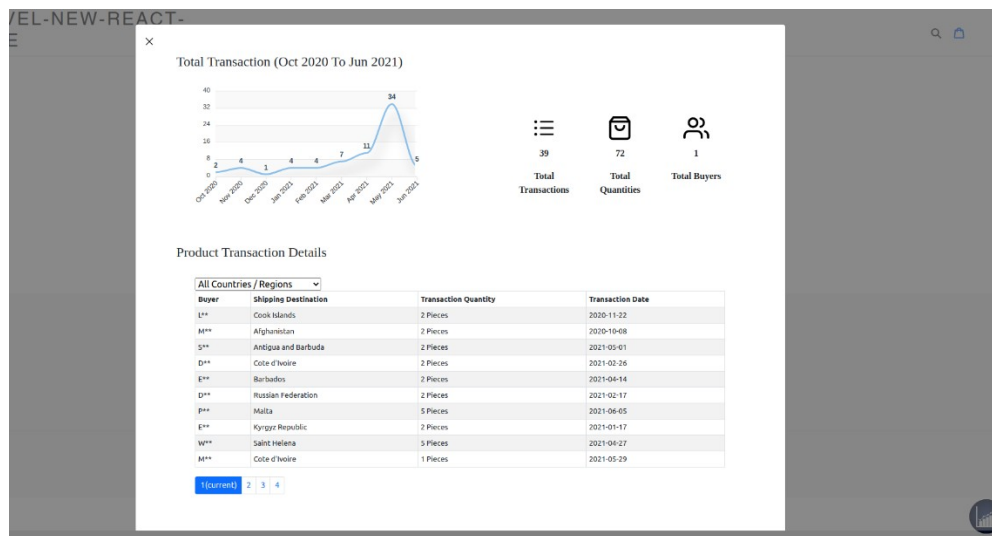


Рис. 3.10. Вигляд спливаючого вікна, якщо в налаштуванні «Select View Type» обрано «Modal»

– **Placement.** Дане налаштування буде доступне за умови варіанту «Modal» для налаштування «Select view type». Воно відповідає за те, з якої сторони (зліва чи справа) буде виводитись іконка для відкриття модального вікна з аналітикою про продукт. Налаштування має два варіанти: «Left bottom» та «Right bottom». Як виглядає дане налаштування, показано на рис. 3.11;

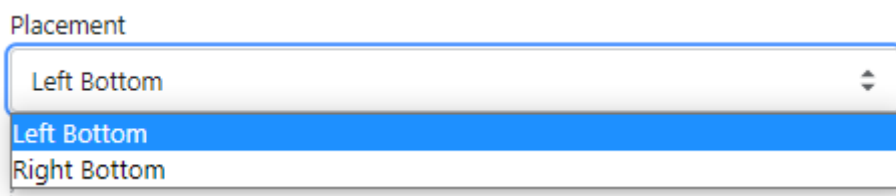


Рис. 3.11. Доступні варіанти вибору для налаштування «Placement»

– **Select Icon.** Дане налаштування буде доступне за умови варіанту «Modal» для налаштування «Select view type». Воно відповідає за те, яка іконка

буде виводитись для відкриття модального вікна з аналітикою про продукт. Як виглядає дане налаштування, показано на рис. 3.12.

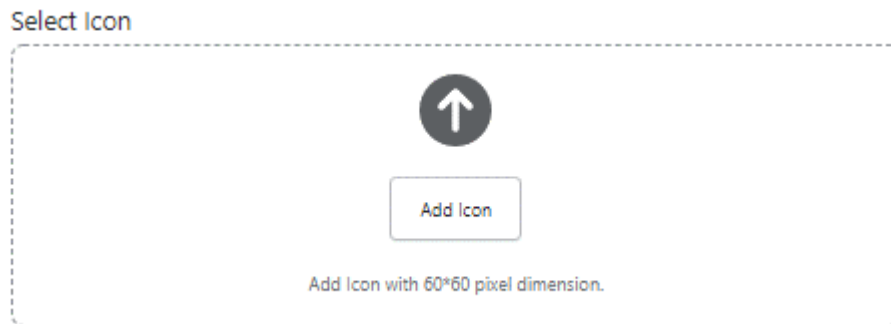


Рис. 3.12. Поле завантаження зображення для налаштування «Select Icon»

3.3.2 Група налаштувань «Chart settings»

Другою є категорія «Chart Settings» (налаштування діаграми), яка показана на рис. 3.13.

CHART SETTINGS

Title

We'll use this Title to Display show above chart.

Sub Title

We'll use this Sub title to show under title.

Chart Type


Chart Color 

Рис. 3.13. Група «Chart settings» в блоці налаштувань

Ця категорія включає в себе чотири (три, за умови, якщо для налаштування «Chart type» обраний варіант «Distibuted Columns») налаштування:

– **Title.** Це налаштування відповідає за те, який заголовок буде виводитись для діаграми. Як виглядає налаштування показано на рис. 3.14. Як

виглядає доданий заголовок до діаграми показано на рис. 3.15. Якщо налаштування «Title» не буде заповнено, діаграма матиме наступний вигляд, як показано на рис. 3.16;

Title

We'll use this Title to Display show above chart.

Рис. 3.14. Вигляд налаштування «Title»

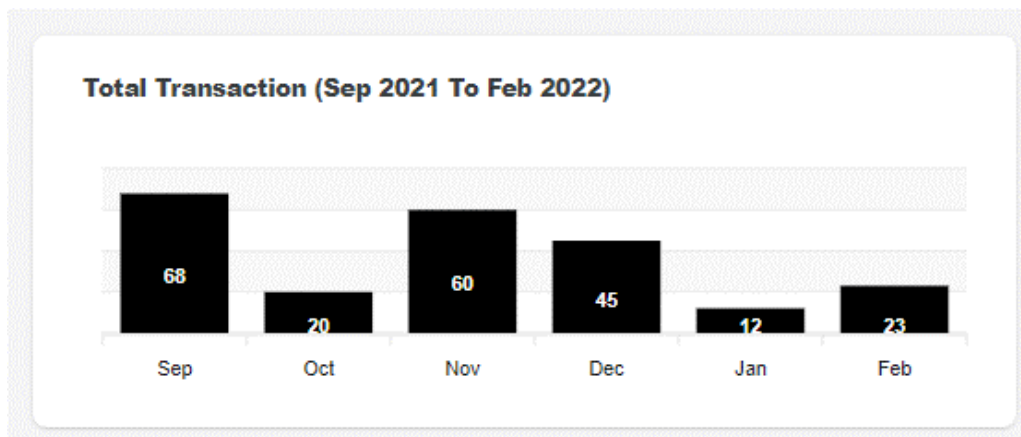


Рис. 3.15. Вигляд діаграми, якщо налаштування «Title» заповнено

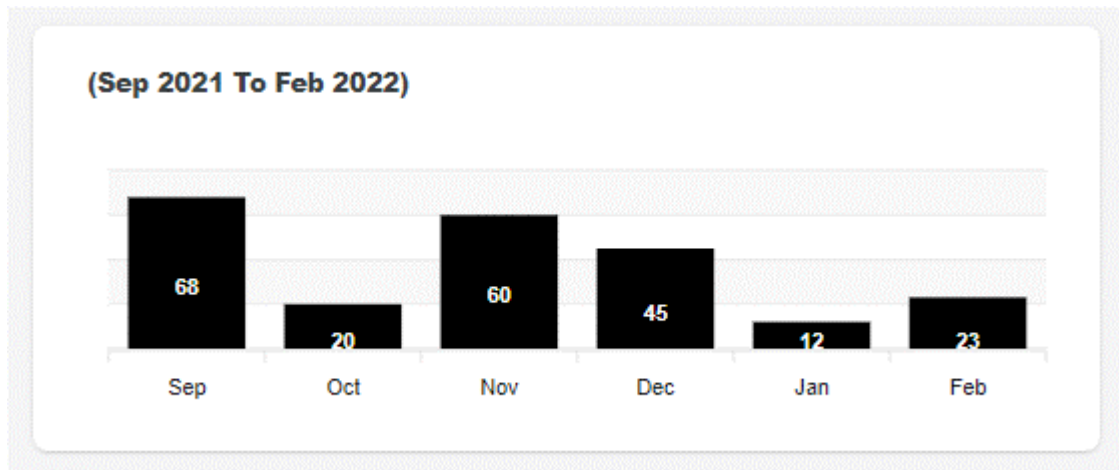


Рис. 3.16. Вигляд діаграми, якщо налаштування «Title» не заповнено

– **Sub title.** Це налаштування відповідає за те, який підзаголовок буде виводитись для діаграми. Як виглядає налаштування показано на рис. 3.17. Як виглядає доданий заголовок до діаграми показано на рис. 3.18. Якщо

налаштування «Sub title» не буде заповнено, діаграма матиме наступний вигляд, як показано на рис. 3.19.

Sub Title

We'll use this Sub title to show under title.

Рис. 3.17. Вигляд налаштування «Sub title»

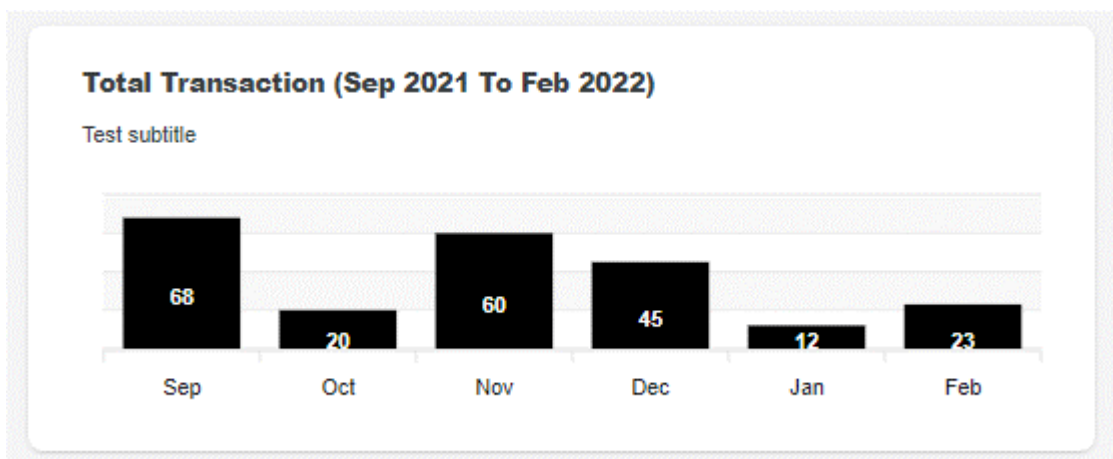


Рис. 3.18. Вигляд діаграми, якщо налаштування «Title» заповнено

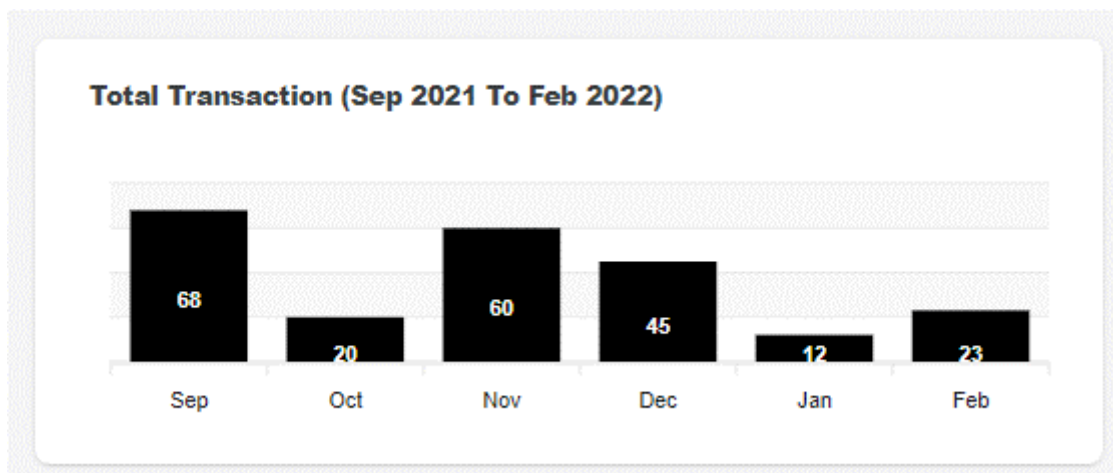


Рис. 3.19. Вигляд діаграми, якщо налаштування «Sub title» не заповнено

– **Chart Type.** Налаштування має шість варіантів: «Line Charts», «Basic Area Charts», «Column with Data Labels», «Column with Rotated Labels», «Line with Data Labels» та «Distributed Columns». Як виглядає дане налаштування, показано на рис. 3.20. Як буде виглядати діаграма в залежності від обраного варіанту показано на рис. 3.21. – рис. 3.26.

Chart Type

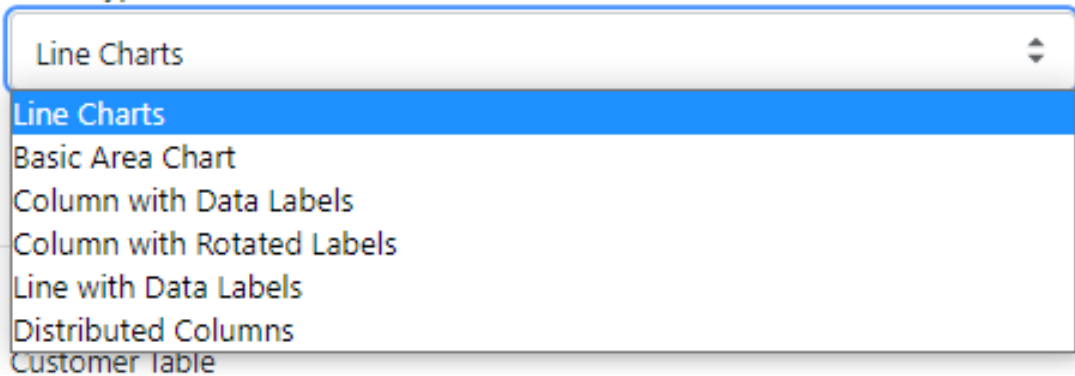


Рис. 3.20. Доступні варіанти вибору для налаштування «Placement»

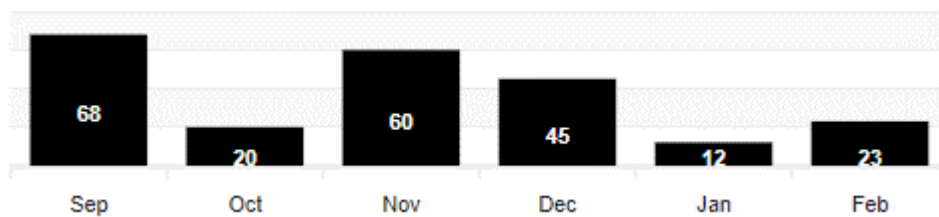


Рис. 3.21. Вигляд діаграми при обраному варіанті «Line Charts»

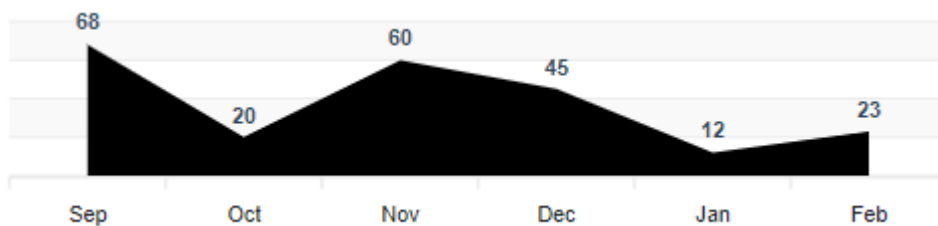


Рис. 3.22. Вигляд діаграми при обраному варіанті «Basic Area Chart»

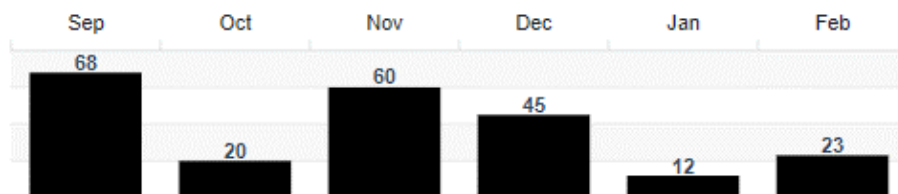


Рис. 3.23. Вигляд діаграми при обраному варіанті «Column with Data Labels»



Рис. 3.24. Вигляд діаграми при обраному варіанті «Column with Rotated Labels»



Рис. 3.25. Вигляд діаграми при обраному варіанті «Line with Data Labels»

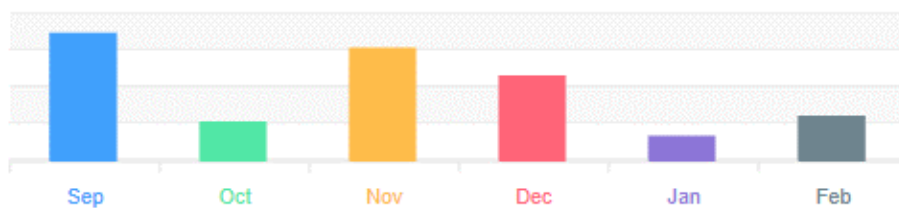


Рис. 3.26. Вигляд діаграми при обраному варіанті «Distributed Columns»

– **Chart Color.** Дане налаштування буде доступне за будь-якого варіанту окрім «Distibuted Columns» для налаштування «Chart type». За допомогою цього налаштування можна змінити колір діаграми. Як виглядає налаштування, можна побачити на рис. 3.27. Як виглядає діаграма на прикладі діаграми «Line with Data Labels» показано на рис. 3.28.

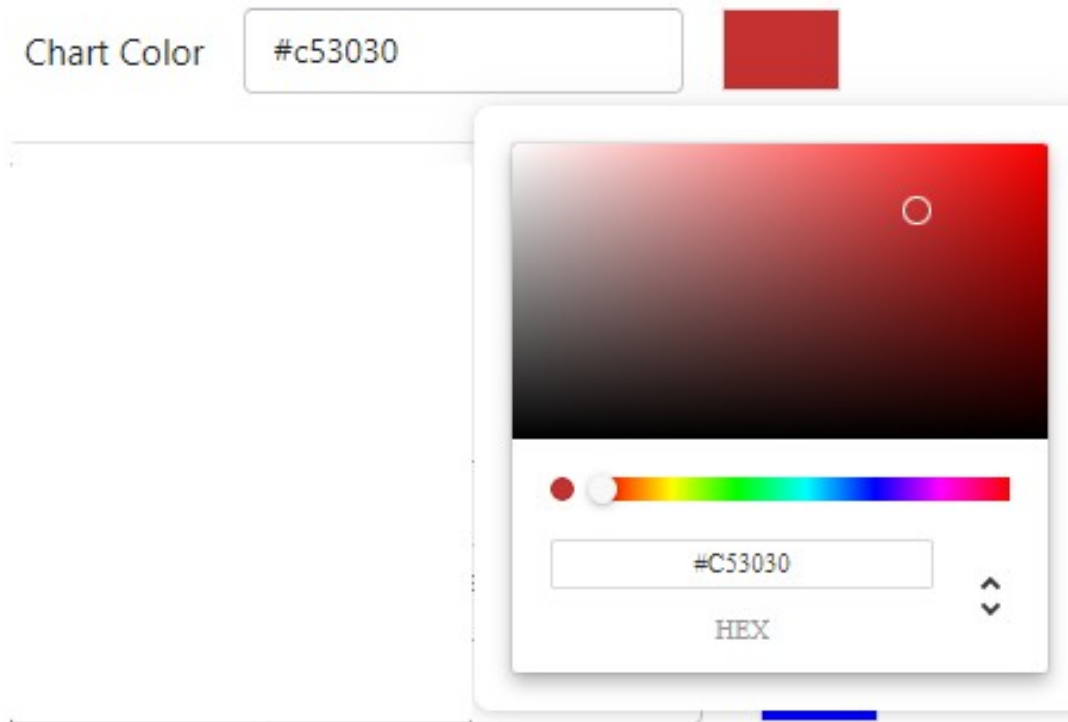


Рис. 3.27. Вигляд налаштування «Chart color»

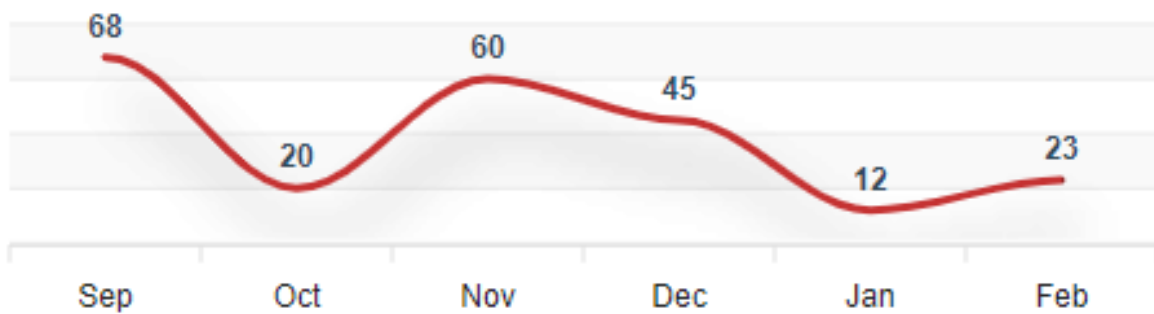


Рис. 3.28. Вигляд діаграми на прикладі діаграми «Line with Data Labels» зі зміненим кольором

3.3.3 Група налаштувань «Table settings»

Третьою є категорія «Table Settings» (налаштування таблиці), яка показана на рис. 3.29.

TABLE SETTINGS

Customer Table

Show Hide

List Title

Product Transaction Details

We'll use this list title to show as a table title.


Button Color #0000FF 

Рис. 3.29. Група «Table settings» в блоці налаштувань

Ця категорія включає в себе три налаштування:

– **Customer Table.** Дане налаштування відповідає за те буде показана таблиця покупців з інформацією про їх покупки на головній сторінці чи ні. В даного налаштування є два варіанти: «Show», «Hide». Як виглядає налаштування, показано на рис. 3.30. Як виглядає таблиця покупців показано на рис. 3.31;

Customer Table

Show Hide

Рис. 3.30. Вигляд налаштування «Customer table»

Product Transaction Details			
All Countries / Regions			
Buyer	Shipping Destination	Transaction Quantity	Transaction Date
L**	Nigeria	9 Pieces	10/25/21
H**	Canada	3 Pieces	11/05/21
D**	Germany	3 Pieces	11/15/21
C**	Netherlands	6 Pieces	10/09/21
N**	Costa Rica	6 Pieces	10/31/21

1 2

Рис. 3.31. Таблиця покупців на головній сторінці

– **List Title.** Дане налаштування відповідає за те, який заголовок буде виведено до таблиці. Вигляд таблиці з доданим заголовком можна подивитись на рис. 3.31. Як виглядає налаштування «List Title» показано на рис. 3.32. Вигляд таблиці без доданого заголовку показано на рис. 3.33;

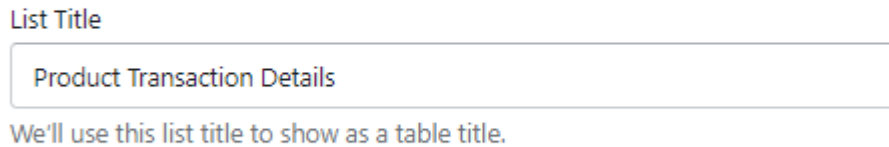


Рис. 3.32. Вигляд налаштування «List title»

Buyer	Shipping Destination	Transaction Quantity	Transaction Date
L**	Nigeria	9 Pieces	10/25/21
H**	Canada	3 Pieces	11/05/21
D**	Germany	3 Pieces	11/15/21
C**	Netherlands	6 Pieces	10/09/21
N**	Costa Rica	6 Pieces	10/31/21

Рис. 3.33. Таблиця покупців на головній сторінці без заголовку

– **Button Color.** Дане налаштування служить для зміни кольору кнопки в таблиці, яка відповідає за зміну сторінки виведених даних. Як виглядає налаштування можна подивитись на рис. 3.34. Вигляд таблиці зі зміненим кольором кнопки показано на рис. 3.35;

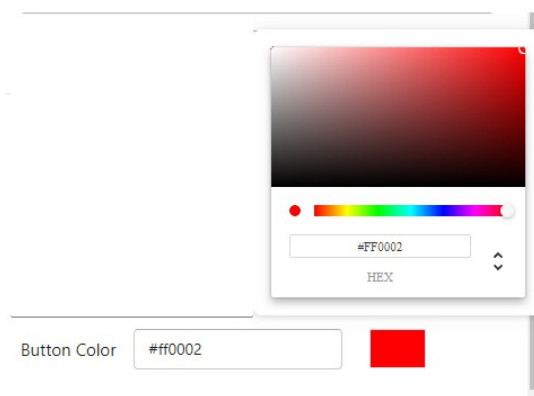


Рис. 3.34. Вигляд налаштування «Button color»

Buyer	Shipping Destination	Transaction Quantity	Transaction Date
L**	Nigeria	9 Pieces	10/25/21
H**	Canada	3 Pieces	11/05/21
D**	Germany	3 Pieces	11/15/21
C**	Netherlands	6 Pieces	10/09/21
N**	Costa Rica	6 Pieces	10/31/21

Рис. 3.35. Таблиця покупців на головній сторінці зі зміненим кольором кнопок

3.4 Розробка сторінки з тарифними планами

Серед додатків розміщених в магазині додатків Shopify можна знайти як платні додатки, так і безкоштовні. Веб-додаток, що розроблюється в даній роботі, замислювався бути безкоштовним, що не передбачало існування даної сторінки в додатку, але довелось її додати. Політика Shopify говорить про те, що будь-який додаток, що планується розмістити в магазині додатків Shopify повинен мати інформацію про вартість використання того чи іншого додатка. Якщо додаток є платним, в ньому повинна бути сторінка або спливаюче вікно з повною інформацією про те, які можливості надає користувачу конкретний тарифний план, а також можливість придбати інший. В безкоштовному додатку логічно припустити, що не буде можливості змінити тарифний план, але все ж таки повинна бути інформація про поточний план на окремій сторінці або у спливаючому вікні, як виглядає сторінка з тарифними планами показано на рис. 3.36.

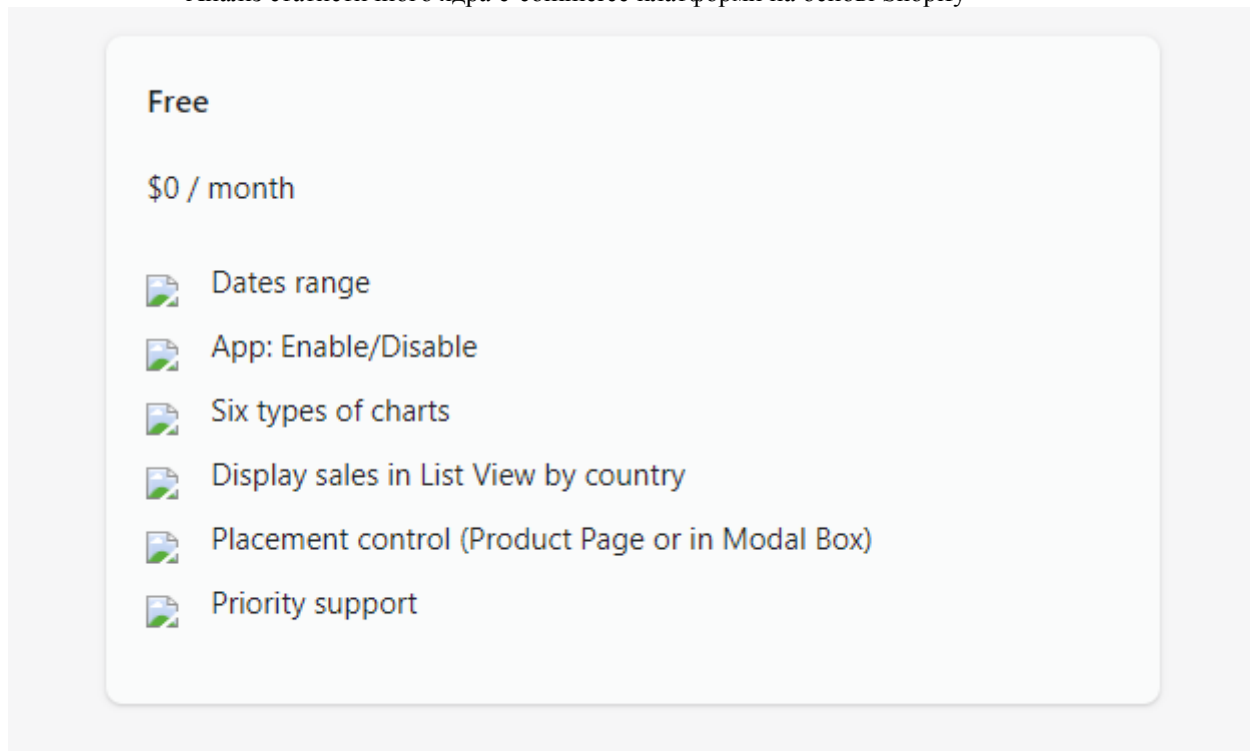


Рис. 3.36. Сторінка з тарифними планами

3.5 Розробка сторінки з загальною інформацією про додаток

Часто буває так, що придбавши будь-яку річ, наприклад, шафу яку потрібно власноруч зібрати з багатьох частин, спочатку всі намагаються впоратись опираючись на свій досвід, логіку чи щось інше. Нажаль, частіше за все це не вдається і доводиться звертатись за допомогою до інструкції. Після цього справа іде значно легше. Така сама аналогія має місце бути і з додатками. Буває так, що людина встановлює додаток, починає розбиратись, але нічого не виходить і певні функції не працюють так, як хотів би користувач чи щось інше. Для цього зазвичай і створюються сторінки-інструкції, на яких є вся необхідна інформація. Часто такі сторінки називаються «FAQ page» (Frequently Asked Questions – часто задавані питання). Додаток, що розроблюється, має таку сторінку. Для зручності користувачів, вся інформація була поміщена в так звані «аккордеони».

«Аккордеон» являє собою кілька кнопок або ярликів, розташованих один над одним. При натисканні на заголовок, контент «аккордеону» повинен плавно з'явитись і так само зникнути при повторному натисканні на заголовок. Є два

типи «аккордеонів», в першому випадку кожний «аккордеон» є незалежним, тобто відкриваючи інший, поточний не повинен зникати. В другому випадку натискаючи на один «аккордеон», інші повинні зникнути. Нема чіткого описання як потрібно правильно реалізовувати даний функціонал, тому кожен розробник робить на власний розсуд чи побажання клієнта. Побачити як виглядає сторінка «User Guide» можна на рис. 3.37. Як виглядає сторінка з «аккордеоном» у відкритому стані показано на рис. 3.38.



Рис. 3.37. Вигляд сторінки «User Guide» із закритими «аккордіонами»

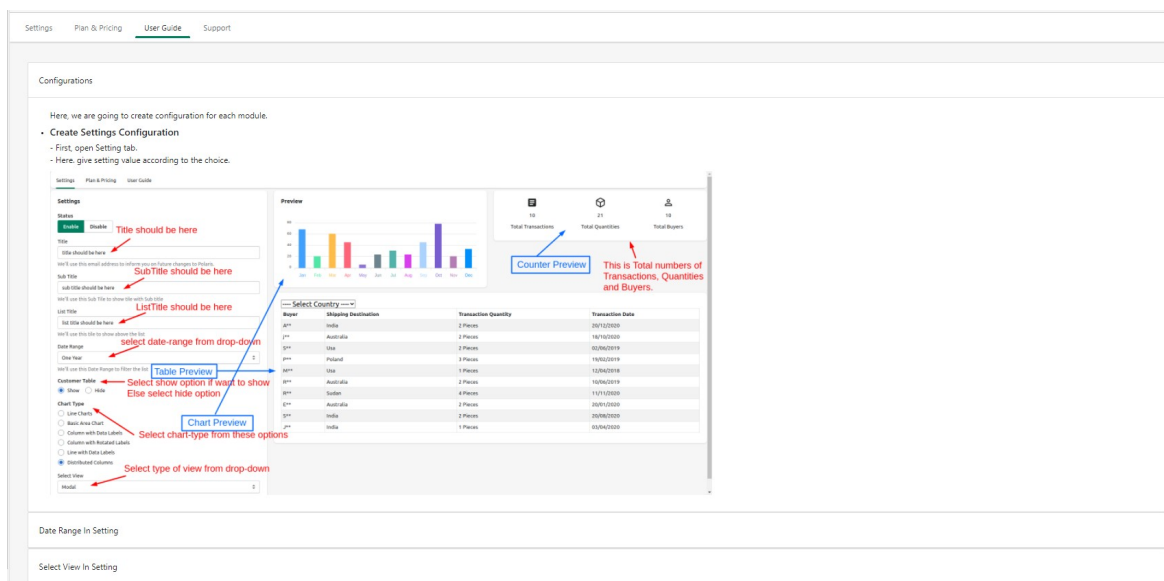


Рис. 3.38. Вигляд сторінки «User Guide» із першим відкритим «аккордіоном»

3.6 Розробка сторінки для зворотного зв'язку

Мабуть не буде новиною, що для будь-якої виконаної роботи потрібен зворотній зв'язок, наприклад, як це відбувається в університеті після пройденого семестру. Дана процедура є дуже важливою, адже зібравши інформацію про те, наскільки корисним був матеріал, чи легко він засвоюється, врешті-решт чи був

він цікавим і т.д., можна проаналізувати ці дані і покращити курс, або прибрати його. Так само відбувається в усьому світі, додаток, якому присвячена ця робота не є винятком. Вірогідно, що в додатку є «баги» (помилки в роботі) про які потрібно сповістити, хтось не зміг розібратись із додатком і чекає на допомогу, або бажає додати якусь нову функцію, яка би на його погляд покращила додаток, тому і була додана сторінка зворотного зв'язку. Як виглядає дана сторінка показано на рис. 3.39.

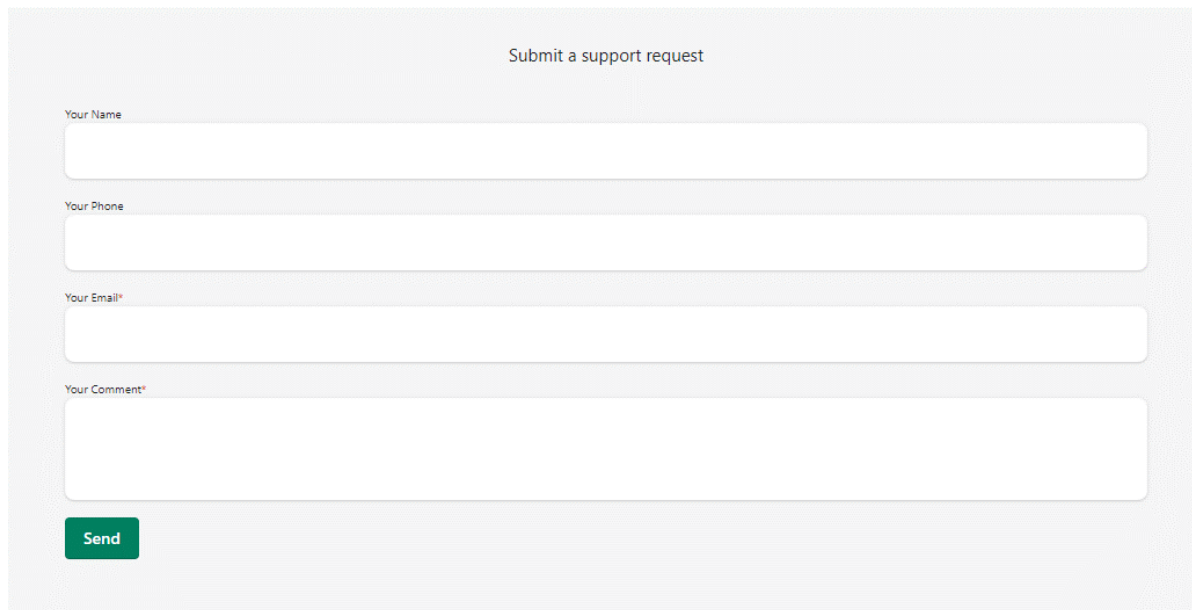
The image shows a web form titled "Submit a support request". It contains four input fields: "Your Name", "Your Phone", "Your Email*", and "Your Comment*". Below the fields is a green "Send" button. The form is set against a light gray background.

Рис. 3.39. Вигляд сторінки зворотного зв'язку

Форма зворотного зв'язку має всього чотири поля: «Your name», «Your Phone», «Your Email», «Your Comment», два останні мають бути обов'язково заповнені, без них форма не буде відправлена.

3.7 База даних для веб-додатку

PostgreSQL, також відомий як Postgres, є безкоштовною системою керування реляційною базою даних (RDBMS) з відкритим вихідним кодом, яка робить акцент на розширюваності та відповідності SQL. Спочатку вона називалась POSTGRES, посилаючись на походження як наступник бази даних

Ingres, розробленої в Каліфорнійському університеті в Берклі. У 1996 році проект був перейменований на PostgreSQL, щоб відобразити його підтримку SQL. Після огляду в 2007 році команда розробників вирішила зберегти назву PostgreSQL та псевдонім Postgres [44, 45].

PostgreSQL містить транзакції з властивостями Atomicity, Consistency, Isolation, Durability (ACID), автоматично оновлюваними представленнями, матеріалізованими представленнями, тригерами, зовнішніми ключами та збереженими процедурами. Система призначена для обробки різноманітних робочих навантажень, від окремих комп'ютерів до сховищ даних або веб-сервісів з багатьма одночасними користувачами. Це стандартна база даних для macOS Server, а також доступна для Windows, Linux, FreeBSD і OpenBSD. На рис. 3.40. показана структура створеної бази даних. Подивитись як виглядає головна сторінка створеної бази даних можна на рис. 3.41. Таблиця із даними про продуктахми показана на рис. 3.42. Таблиця з даними про варіанти продуктів показана на рис. 3.43. [44, 45]

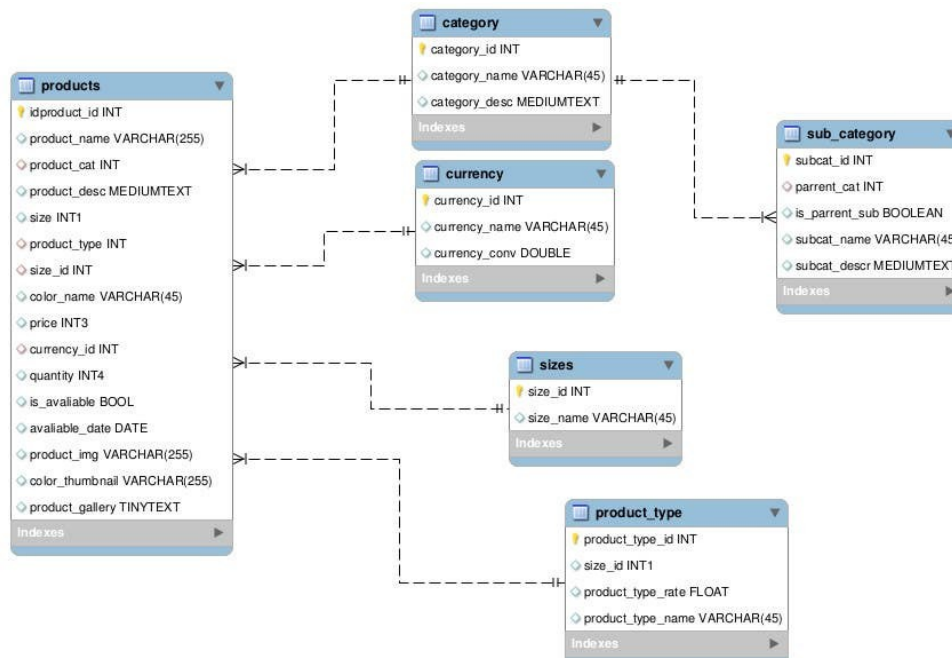


Рис. 3.40. Структура створеної бази даних

Кафедра інтелектуальних інформаційних систем
Аналіз статистичного ядра e-commerce платформи на основі Shopify

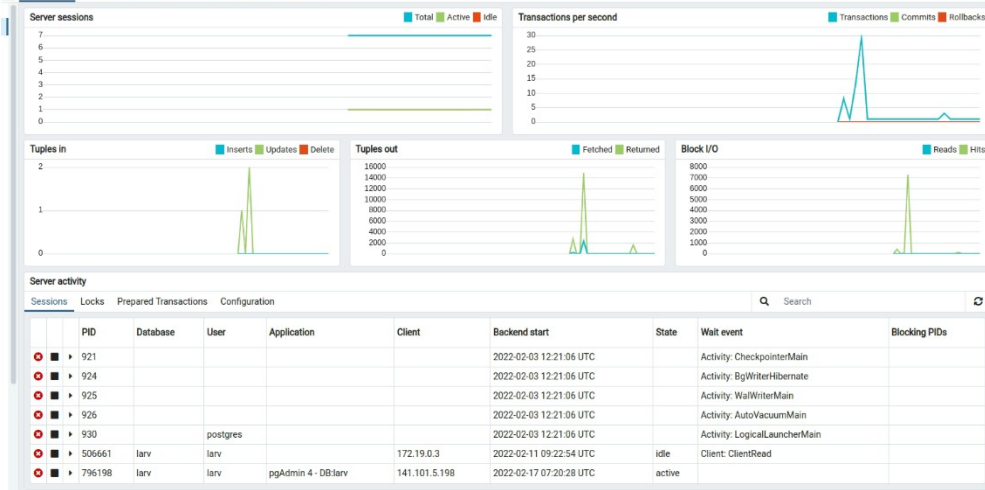


Рис. 3.41. Головна сторінка бази даних в «PgAdmin»

uid	[PK] bigint	id	body_html
21	6789169807551		<meta charset="utf-8">
25	7027782385855		
63	1328944054343		<p>Designer AGOLDE</p>
108	7047058620607		High-waisted pencil skirt with invisible side zipper. Pair with our Alta Top or your favorite fitted top and boots. Made from a blend of upcycled fabrics to stay true to our commitment to repurpose and give new life to leftover materials.
112	7047058555071		Mock neck babydoll dress with a slip liner and a relaxed fit.
115	7047057801407		Cropped bustier top with bra-friendly straps that pairs well with our Alice Skirt or your favorite high-rise denim. Made from a blend of upcycled fabrics to stay true to our commitment to repurpose and give new life to leftover materials.
146	6929811898559		
178	6776273731775		<meta charset="utf-8">
240	6955861082303		
244	6867043516607		<meta charset="utf-8">
246	6955846336703		Relaxed fit, long line blazer with pockets
320	6619274346687		<p>span data-mca-fragment="1">Our bestselling Berkeley Tee has a new sister in town. Made from our signature-soft Rwear Knit fabric, this perfectly tapered t-shirt dress pairs well with both sneakers or sandals.</p>
352	7005362051327		Long sleeve, relaxed fit crewneck sweater. Floral jacquard design hand drawn in house by our AV designer. Length: 22" Made with Organic Cotton.
373	7027783172287		Mock neck, embroidered midi dress with scallop sleeve and hem, detachable belt and a slip lining. Made from a blend of upcycled fabrics to stay true to our commitment to repurpose and give new life to leftover materials.
384	6789169905855		<meta charset="utf-8">
392	6733053001919		<meta charset="utf-8">
423	7027788673111		Midi dress featuring a mock neck, long sleeves, and an adjustable waist tie. Made from our premium blend of Organic Cotton and Modal for a soft and comfortable fit. Model is 5'9" and wears size Small.
457	6789170263903		<meta charset="utf-8">
475	7047058666143		
487	1618011324487		Designer: Coclico Coclico believes that luxury isn't the ability to purchase endlessly, but the privilege of choosing wisely. Materials are sourced locally in Europe, near their small, family-run factory in Mallorca, Spain. At their NYC he
503	747592324855		<p>Designer: Coclico</p>
511	1673631465543		<p>Designer: Coclico Coclico believes that luxury isn't the ability to purchase endlessly, but the privilege of choosing wisely. Materials are sourced locally in Europe, near their small, family-run factory in Mallorca, Spain. At their NYC he
542	6867034570943		<meta charset="utf-8">
575	6955849679039		<p>Washable silk dress</p>
587	178188615699		<p>Designer: Deadwood</p>

Рис. 3.42. Таблиця з даними про продукти в «PgAdmin»

Кафедра інтелектуальних інформаційних систем
Аналіз статистичного ядра e-commerce платформи на основі Shopify

uid	id	barcode	compare_at_price	fulfillment_service	created_at	updated_at	grams	image_id	inventory_item_id	inventory_managed
[PK] bigint	bigint	character varying (1024)	double precision	character varying (1024)	timestamp without time zone	timestamp without time zone	integer	bigint	bigint	character varyi
1	123	40162311733439		128 manual	2021-07-06 15:09:17	2022-02-11 11:19:53	318	29489832067263	42257440604351	shopify
2	124	40162311766207		128 manual	2021-07-06 15:09:18	2022-02-04 12:26:40	318	29489832067263	422574406937119	shopify
3	125	40162311798975		128 manual	2021-07-06 15:09:18	2022-02-07 15:59:21	318	29489832067263	422574406969887	shopify
4	126	40162311831743		128 manual	2021-07-06 15:09:18	2022-02-04 12:26:40	318	29489832067263	42257440702655	shopify
5	127	40162311864511		128 manual	2021-07-06 15:09:18	2022-02-15 09:21:57	318	29489832067263	42257440735423	shopify
6	137	40901872255167		228 manual	2021-10-18 15:09:10	2022-02-04 12:28:34	181	30911181619391	42998783475903	shopify
7	138	40901872287935		228 manual	2021-10-18 15:09:10	2022-02-16 11:13:26	181	30911181619391	42998783508671	shopify
8	139	40901872320703		228 manual	2021-10-18 15:09:10	2022-02-12 10:53:06	181	30911181619391	42998783541439	shopify
9	140	40901872353471		228 manual	2021-10-18 15:09:10	2022-02-12 17:38:23	181	30911181619391	42998783574207	shopify
10	141	40901872386239		228 manual	2021-10-18 15:09:10	2022-02-04 12:28:34	181	30911181619391	42998783606975	shopify
11	994	12510819254343	883435934641	188 manual	2018-08-14 13:01:10	2022-02-07 15:59:25	454	13750370238535	125681261816327	shopify
12	995	12510819188807	883435934658	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:24	454	13750370238535	125681261750791	shopify
13	996	12510819221575	883435934665	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:24	454	13750370238535	125681261783559	shopify
14	997	12510819156029	883435934672	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:24	454	13750370238535	125681261798023	shopify
15	998	12510819123271	883435934680	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:24	454	13750370238535	125681261686255	shopify
16	999	12510819090503	883435934696	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:24	454	13750370238535	125681261653487	shopify
17	600	12510819024967	883435934702	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:23	454	13750370238535	125681261586951	shopify
18	601	12510819057735	883435934719	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:24	454	13750370238535	125681261519719	shopify
19	602	12510818992199	883435934726	188 manual	2018-08-14 13:01:10	2022-02-04 12:15:23	454	13750370238535	125681261554183	shopify
20	1134	40953169019071	69051839	88 manual	2021-10-29 12:11:28	2022-02-04 12:29:05	227	30911208718527	43050245062847	shopify
21	1135	40953169051839	69051839	88 manual	2021-10-29 12:11:28	2022-02-04 12:29:06	227	30911208718527	43050245099615	shopify
22	1136	40953169084607	69084607	88 manual	2021-10-29 12:11:28	2022-02-04 12:29:06	227	30911208718527	43050245128383	shopify
23	1137	40953169117375	69117375	88 manual	2021-10-29 12:11:28	2022-02-04 12:29:06	227	30911208718527	43050245161151	shopify
24	1138	40953169150143	69150143	88 manual	2021-10-29 12:11:28	2022-02-04 12:29:06	227	30911208718527	43050245193919	shopify
25	1153	40953168658623	68658623	178 manual	2021-10-29 12:11:23	2022-02-04 12:29:05				

Successfully run. Total query runtime: 706 msec. 3939 rows affected.

Рис. 3.43. Таблиця з даними про варіанти продукту в «PgAdmin»

Висновки до розділу 3

В даному розділі було розглянуто:

- інструментальні засоби для створення веб-додатку;
- покроково створений веб-додаток на інформаційній панелі користувача Shopify;
- описані основні сторінки та можливості створеного веб-додатку;
- створена структура бази даних;
- створена база даних.

ВИСНОВКИ

У кваліфікаційній магістерській роботі розроблено веб-додаток для аналітики на основі e-commerce платформи Shopify.

Першочергово був проведений аналіз наявних додатків та платформ з аналітики веб-сайтів, що допомогло виділити основний функціонал, який повинен бути інтегрований у розроблюваний веб-додаток в e-commerce платформі на основі Shopify

При розробці веб-додатку були проаналізовані сучасні інструментальні засоби для створення веб-додатку, мови програмування та розмітки тексту.

У процесі роботи над проектом були отримані практичні навички в дослідженні предметної області, опису проектного рішення. У роботі була використана методологія об'єктно-орієнтованого проектування.

Розроблений веб-додаток для e-commerce платформи Shopify задовольняє всім вимогам, поставленим на етапі постановки завдання.

Були вирішені наступні завдання:

- проведено дослідження і проаналізовано функціонування робочого процесу;
- реалізована база даних в середовищі «Postgres Admin»;
- розроблено веб-додаток для e-commerce платформи Shopify.

У методичній частині було розроблену лабораторну роботу на тему: «Перевірка статистичних гіпотез у середовищі Maple». Були надані вхідні дані для роботи, повністю описаний процес виконання та підбиті підсумки лабораторної роботи.

У спеціальній частині з охорони праці та безпеки в надзвичайних ситуаціях було проаналізовано приміщення ІТ компанії. Визначені норми шуму, вібрації та випромінювань. Визначені правила гігієни праці і виробничої санітарії техніки безпеки, пожежної безпеки, цивільного захисту, екологічної безпеки та безпеки життєдіяльності людини в умовах надзвичайних ситуацій.

Створений веб-додаток є засобом збору, зберігання та аналізу інформації для користувачів.

Веб-додаток дозволяє полегшити процес аналітики веб-сайту клієнта. В майбутньому веб-додаток повинен бути виведений у загальний доступ в магазині додатків «Shopify App Store», буде створено більше можливостей для комфортної роботи та аналізу інформації, буде додана мультимовність щоб задовольняти потреби користувачів з різних країн, додаватиметься новий функціонал, буде проведена робота з просування даного веб-додатку в магазині додатків «Shopify App Store».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кондратюк І. В., Воробйова А. І. «Конверсія електронної торгівлі в Shopify», Інформаційні технології та інженерія: Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів : тези доп., м. Миколаїв, 9-11 лютого 2022 р., ЧНУ ім. Петра Могили, 2022. С. 154-156.
2. Офіційний сайт Асоціації Ретейлерів України. URL: – <https://rau.ua/uk/novyni/onlajn-diktaturaE-commerce/>. (дата звернення: 03.01.2022)
3. The Statistics Portal. Amazoncom. URL: – <https://www.statista.com/statistics/266282/annual-net-revenue-of-amazoncom/>. (дата звернення: 03.01.2022)
4. The Statistics Portal. Ebay URL: – <https://www.statista.com/statistics/507881/ebays-annual-net-revenue/>. (дата звернення: 03.01.2022)
5. Global Ecommerce Report. URL: – <https://www.ecommercefoundation.org/reports>. (дата звернення: 04.01.2022)
6. Офіційний сайт Державної служби статистики України. URL: – <http://www.ukrstat.gov.ua>. (дата звернення: 04.01.2022)
7. What Is Electronic Commerce (Ecommerce)? URL: – <https://www.investopedia.com/terms/e/ecommerce.asp/> (дата звернення: 04.01.2022)
8. What is Shopify? URL: – <https://www.shopify.com/blog/what-is-shopify>
9. What is ecommerce? URL: – <https://www.shopify.com/encyclopedia/what-is-ecommerce/> (дата звернення: 04.01.2022)
10. What is Google Analytics, and why is it important to my business? URL: – <https://medium.com/analytics-for-humans/what-is-google-analytics-and-why-is-it-important-to-my-business-8c083a9f81be/> (дата звернення: 05.01.2022)
11. Why you should consider a Google Analytics alternative. URL: – <https://matomo.org/google-analytics-alternative/> (дата звернення: 05.01.2022)

12. Matomo vs. Google Analytics – Which is a Better Web Analytics Tool?
 URL: – <https://www.monsterinsights.com/matomo-vs-google-analytics>. (дата звернення: 06.01.2022)
13. Why we woopra. URL: – <https://www.woopra.com/company/about/> (дата звернення: 06.01.2022)
14. Web analytics showdown: Woopra vs. Google Analytics. URL: – <https://www.pingdom.com/blog/web-analytics-showdown-woopra-vs-google-analytics/> (дата звернення: 06.01.2022)
15. We enable subscription businesses to grow. URL: – <https://www.gosquared.com/about/> (дата звернення: 06.01.2022)
16. Compare GoSquared vs Google Analytics. URL: – <https://crozdesk.com/compare/gosquared-vs-google-analytics/> (дата звернення: 07.01.2022)
17. How Customer Analytics Can Change The Way Your Business Grows. URL: – <https://www.foxmetrics.com/blog/customer-analytics-change-business/> (дата звернення: 07.01.2022)
18. FoxMetrics vs Google Analytics 360. URL: – <https://www.capterra.com/web-analytics-software/compare/156016-177042/FoxMetrics-vs-Google-Analytics/> (дата звернення: 07.01.2022)
19. We help the world learn from its data. URL: – <https://mixpanel.com/about/> (дата звернення: 07.01.2022)
20. Mixpanel vs. Google Analytics: Which is best for your product team?
 URL: – <https://heap.io/topics/mixpanel-vs-google-analytics-which-is-best-for-your-product-team> (дата звернення: 08.01.2022)
21. What Is Google Tag Manager and Why Use It? URL: – <https://www.orbitmedia.com/blog/what-is-google-tag-manager-and-why-use-it/> (дата звернення: 08.01.2022)
22. The Beginners Guide to Google Tag Manager. URL:– <https://www.semrush.com/blog/beginners-guide-to-google-tag-manager/> (дата звернення: 08.01.2022)

23. A Beginners Tour of Google Analytics. URL: – <https://www.bluehost.com/resources/a-beginners-tour-of-google-analytics/> (дата звернення: 08.01.2022)
24. Setup and install Tag Manager URL: – <https://support.google.com/tagmanager/answer/6103696?hl=en/> (дата звернення: 09.01.2022)
25. How to Create Google Analytics Goals to Track Conversions. URL: – <https://www.monsterinsights.com/how-to-create-a-goal-in-google-analytics-to-track-conversions/> (дата звернення: 09.01.2022)
26. Google Search Console. URL: – <https://www.searchmetrics.com/glossary/google-search-console/> (дата звернення: 09.01.2022)
27. Configuring Search Console data in Analytics. URL: – <https://support.google.com/analytics/answer/1308621?hl=en/> (дата звернення: 09.01.2022)
28. Best web development tools in 2022. URL: – <https://www.techradar.com/news/best-web-development-tool/> (дата звернення: 10.01.2022)
29. Frontend vs. backend: what's the difference? URL: – <https://www.pluralsight.com/blog/software-development/front-end-vs-back-end/> (дата звернення: 10.01.2022)
30. What is PHP? URL: – <https://www.php.net/manual/en/intro-what-is.php/> (дата звернення: 10.01.2022)
31. PHP Introduction URL:– https://www.w3schools.com/php/php_intro.asp/ (дата звернення: 11.01.2022)
32. JavaScript Overview. URL: – https://www.tutorialspoint.com/javascript/javascript_overview.html/ (дата звернення: 11.01.2022)
33. JavaScript Tutorial URL:– <https://www.w3schools.com/js/> (дата звернення: 11.01.2022)

34. What is JavaScript? URL: – https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript/ (дата звернення: 12.01.2022)
35. What is HTML URL: – <https://www.javatpoint.com/what-is-html/> (дата звернення: 12.01.2022)
36. What Is HyperText Markup Language (HTML)? URL: – <https://www.investopedia.com/terms/h/html.asp/> (дата звернення: 12.01.2022)
37. What is CSS? URL: – https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS/ (дата звернення: 13.01.2022)
38. What is CSS? URL: – https://www.tutorialspoint.com/css/what_is_css.htm
39. What Is CSS and Why Should You Use It? URL: – <https://blog.devmountain.com/what-is-css-and-why-use-it/> (дата звернення: 13.01.2022)
40. The Ultimate Guide on SQL Basics URL: – <https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sql/> (дата звернення: 14.01.2022)
41. Introduction to SQL URL: – https://www.w3schools.com/sql/sql_intro.asp/ (дата звернення: 14.01.2022)
42. How to Build a Shopify App: The Complete Guide URL:– <https://www.shopify.com/partners/blog/how-to-build-a-shopify-app/> (дата звернення: 14.01.2022)
43. Build Shopify apps URL:– <https://shopify.dev/apps/> (дата звернення: 15.01.2022)
44. What is PostgreSQL? URL: – https://www.tutorialspoint.com/postgresql/postgresql_overview.html/ (дата звернення: 16.01.2022)

45. PostgreSQL Administration URL: – <https://www.postgresqltutorial.com/postgresql-administration/> (дата звернення: 16.01.2022)
46. В.З. Аладьев, В.К. Бойко, Е.А. Ровба "Программирование в пакетах Maple и Mathematica: Сравнительный аспект" / Монография / Гродно: Гродненский Госуниверситет, 2,011, 517 с.
47. М. Н. Кирсанов. "Практика программирования в системе Maple" М.: Издательский дом МЭИ, 2011, 208с.
48. М. Н. Кирсанов. Графы в Maple. М.: Физматлит, 2007, 105-168с.
49. В.З. Аладьев. Основы программирования в Maple. Таллин, 2006., 13-76с.
50. Інструкція з охорони праці при роботі з персональним комп'ютером URL: – <http://trudova-ohrana.ru/primery-dokumentov/prikladi-nstrukcij-z-ohoroni-prac-ukranskoju/1681-nstrukcija-z-ohoroni-prac-pri-robot-z-personalnim-kompjuterom.html>
51. Ергономічні вимоги для організації робочого місця URL: – <https://www.gpp.in.ua/robota/ergonomichni-vimogi-dlya-organizatsiji-robochogomistsya.html/> (дата звернення: 10.01.2022)
52. Про освітлення виробничих приміщень URL: – <https://oppb.com.ua/news/pro-osvitlennya-vyrobnychyh-prymishchen/> (дата звернення: 10.01.2022)
53. Мікроклімат виробничих приміщень та його вплив на працездатність URL: – <https://oppb.com.ua/news/mikroklimat-vyrobnychyh-prymishchen-ta-yogo-vplyv-na-pracездatnist/> (дата звернення: 11.01.2022)
54. Виробнича санітарія і гігієна праці URL: – https://pidru4niki.com/17660603/turizm/virobnicha_sanitariya_gigiyena_pratsi/ (дата звернення: 12.01.2022)
55. Пожежна безпека на підприємстві URL: – <https://esop.mcfr.ua/704079/> (дата звернення: 13.01.2022)

56. Електромагнітне випромінювання. Частина 1 URL: –

<https://oppb.com.ua/news/elektromagnitne-vyprominyuvannya-chastyna-1/> (дата

звернення: 14.01.2022)

57.

ДОДАТОК А

Код програми. Головний файл `Statistic.php`

```
<?php
namespace Statistic;
use Propel\Runtime\Connection\ConnectionInterface;
use Thelia\Core\Template\TemplateDefinition;
use Thelia\Module\BaseModule;
class Statistic extends BaseModule
{
    const MESSAGE_DOMAIN = "statistic";
    public function getHooks()
    {
        return array(
            array(
                'type' => TemplateDefinition::BACK_OFFICE,
                'code' => 'statistic.tab',
                'title' => array(
                    'fr_FR' => 'Module de statistiques, onglets.',
                    'en_US' => 'Statistic module, tabs.'
                ),
                'active' => true,
                'block' => true,
                'module' => false
            ),
            array(
                'type' => TemplateDefinition::BACK_OFFICE,
                'code' => 'hook_home_stats',
                'title' => array(
                    'fr_FR' => 'Accueil des statistiques',
                    'en_US' => 'Home Statistics'
                ),
                'active' => true,
                'block' => false,
                'module' => false
            ),
            array(
                'type' => TemplateDefinition::BACK_OFFICE,
                'code' => 'statistic.footer.js',
                'title' => array(
                    'fr_FR' => 'Module de statistiques, insertion des JS.',
                    'en_US' => 'Statistic module, JS insertion'
                ),
                'active' => true,
                'block' => false,
                'module' => false
            )
        );
    }
    public function postActivation(ConnectionInterface $con = null)
    {
        self::setConfigValue('order_types', '2,3,4');
    }
}
```

ДОДАТОК Б

Код програми. Файл `StatisticController.php`

```
<?php
namespace Statistic\Controller;
use DateInterval;
use Statistic\Statistic;
use Thelia\Controller\Admin\BaseAdminController;
use Thelia\Core\HttpFoundation\JsonResponse;
use Thelia\Model\Base\ProductQuery;
use Thelia\Tools\MoneyFormat;
class StatisticController extends BaseAdminController
{
    public function toolShow()
    {
        return $this->render('statistic-tool');
    }
    public function statAverageCartAction()
    {
        if ($session = $this->getRequest()->getSession()) {
            $session->save();
        }
        $ghost = $this->getRequest()->query->get('ghost');
        $startDay = $this->getRequest()->query->get('startDay', date('d'));
        $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
        $startYear = $this->getRequest()->query->get('startYear', date('Y'));
        $endDay = $this->getRequest()->query->get('endDay', date('d'));
        $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
        $endYear = $this->getRequest()->query->get('endYear', date('Y'));
        $startDate = new \DateTime($startYear . '-' . $startMonth . '-' .
$startDate);
        $endDate = new \DateTime($endYear . '-' . $endMonth . '-' . $endDay);
        $result = $this->getStatisticHandler()->averageCart($startDate, $endDate);
        $average = new \stdClass();
        $average->color = '#5cb85c';
        $average->graph = $result['stats'];
        $average->graphLabel = $result['label'];
        $data = new \stdClass();
        if ($startDay === $endDay && $startMonth === $endMonth && $startYear ===
$endYear) {
            $data->title = $this->getTranslator()->trans("Stats between
%startDay/%startMonth/%startYear", array(
                '%startDay' => $startDay,
                '%startMonth' => $startMonth,
                '%startYear' => $startYear,
            ), Statistic::MESSAGE_DOMAIN);
        } else {
            $data->title = $this->getTranslator()->trans("Stats between
%startDay/%startMonth/%startYear and %endDay/%endMonth/%endYear", array(
                '%startDay' => $startDay,
                '%startMonth' => $startMonth,
                '%startYear' => $startYear,
                '%endDay' => $endDay,
                '%endMonth' => $endMonth,
                '%endYear' => $endYear
            ), Statistic::MESSAGE_DOMAIN);
        }
        $data->series = array(
```

```

    $average,
  );

  if ((int)$ghost === 1) {
    $ghostGraph = $this->getStatisticHandler()->averageCart(
      $startDate->sub(new DateInterval('P1Y')),
      $endDate->sub(new DateInterval('P1Y'))
    );
    $ghostCurve = new \stdClass();
    $ghostCurve->color = "#38acfc";
    $ghostCurve->graph = $ghostGraph['stats'];
    $data->series[] = $ghostCurve;
  }
  return $this->jsonResponse(json_encode($data));
}
public function statBestSalesAction()
{
  $startDate = $this->getRequest()->query->get('startDate', date('d'));
  $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
  $startYear = $this->getRequest()->query->get('startYear', date('Y'));
  $endDate = $this->getRequest()->query->get('endDate', date('d'));
  $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
  $endYear = $this->getRequest()->query->get('endYear', date('Y'));
  $startDate = new \DateTime($startYear . '-' . $startMonth . '-' .
$startDate);
  $endDate = new \DateTime($endYear . '-' . $endMonth . '-' . $endDate);
  $productRef = null;
  if ($productId = $this->getRequest()->query->get('productId')) {
    $productRef = ProductQuery::create()->findOneById($productId)-
>getRef();
  }
  $dateDiff = date_diff($startDate, (new \DateTime($endDate->format("Y-m-
d"))));
  $table = [];
  $locale = $this->getSession()->getLang()->getLocale();
  $results = $this->getStatisticHandler()->bestSales($startDate, $endDate,
$locale, $productRef);
  $results2 = $this->getStatisticHandler()->bestSales(
    (clone($startDate))->sub($dateDiff),
    (clone($endDate))->sub($dateDiff),
    $locale,
    $productRef
  );
  $results3 = $this->getStatisticHandler()->bestSales(
    (clone($startDate))->sub(new DateInterval('P1Y')),
    (clone($endDate))->sub(new DateInterval('P1Y')),
    $locale,
    $productRef
  );
  foreach ($results as $result) {
    $row = $result;
    $row['total_sold2'] = 0;
    $row['total_sold3'] = 0;
    $row['total_ttc2'] = 0;
    $row['total_ttc3'] = 0;
    $row['total_ttc'] = MoneyFormat::getInstance($this->getRequest()-
>formatByCurrency($row['total_ttc']);

    if (array_key_exists($result['product_ref'], $results2)) {
      $row['total_sold2'] = $results2[$result['product_ref']]
['total_sold'];
      $row['total_ttc2'] = MoneyFormat::getInstance($this-
>getRequest()->formatByCurrency($results2[$result['product_ref']]
['total_ttc']);

```

```

    unset($results2[$result['product_ref']]);
  }

  if (array_key_exists($result['product_ref'], $results3)) {
    $row['total_sold3'] = $results3[$result['product_ref']]
['total_sold'];
    $row['total_ttc3'] = MoneyFormat::getInstance($this-
>getRequest())->formatByCurrency($results3[$result['product_ref']]['total_ttc']);
    unset($results3[$result['product_ref']]);
  }
  if ($row) {
    $table[] = $row;
  }
}
foreach ($results2 as $result) {
  $row = $result;
  $row['total_sold'] = 0;
  $row['total_sold2'] = $result['total_sold'];
  $row['total_sold3'] = 0;
  $row['total_ttc'] = 0;
  $row['total_ttc2'] = MoneyFormat::getInstance($this->getRequest())-
>formatByCurrency($result['total_ttc']);
  $row['total_ttc3'] = 0;
  if (array_key_exists($result['product_ref'], $results3)) {
    $row['total_sold3'] = $results3[$result['product_ref']]
['total_sold'];
    $row['total_ttc3'] = MoneyFormat::getInstance($this-
>getRequest())->formatByCurrency($results3[$result['product_ref']]['total_ttc']);
    unset($results3[$result['product_ref']]);
  }
  if ($row) {
    $table[] = $row;
  }
}
foreach ($results3 as $result) {
  $row = $result;
  $row['total_sold'] = 0;
  $row['total_sold2'] = 0;
  $row['total_sold3'] = $result['total_sold'];
  $row['total_ttc'] = 0;
  $row['total_ttc2'] = 0;
  $row['total_ttc3'] = MoneyFormat::getInstance($this->getRequest())-
>formatByCurrency($result['total_ttc']);
  if ($row) {
    $table[] = $row;
  }
}
$bestSales = new \stdClass();
$bestSales->color = '#5cb85c';
$bestSales->mhead = [
  $this->getTranslator()->trans('tool.panel.general.bestSales.sales',
[], Statistic::MESSAGE_DOMAIN),
  $this->getTranslator()->trans('tool.panel.general.bestSales.totalTTC',
[], Statistic::MESSAGE_DOMAIN),
];
$bestSales->thead = array(
  'title' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.name', [], Statistic::MESSAGE_DOMAIN),
  'product_ref' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.reference', [], Statistic::MESSAGE_DOMAIN),
  'brand_title' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.brand', [], Statistic::MESSAGE_DOMAIN),
  'total_sold' => $this->getTranslator()-

```

```

>trans('tool.panel.general.bestSales.periode', [], Statistic::MESSAGE_DOMAIN),
    'total_sold2' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.periode-1', [], Statistic::MESSAGE_DOMAIN),
    'total_sold3' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.periodeN-1', [], Statistic::MESSAGE_DOMAIN),
    'total_ttc' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.periode', [], Statistic::MESSAGE_DOMAIN),
    'total_ttc2' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.periode-1', [], Statistic::MESSAGE_DOMAIN),
    'total_ttc3' => $this->getTranslator()-
>trans('tool.panel.general.bestSales.periodeN-1', [], Statistic::MESSAGE_DOMAIN),
);
$bestSales->table = $table;
$bestSales->totals = [
    $this->getTranslator()->trans('TOTALS', [],
Statistic::MESSAGE_DOMAIN),
    '',
    'total_sold',
    'total_sold2',
    'total_sold3',
    'total_ttc',
    'total_ttc2',
    'total_ttc3',
    ''
];
$data = new stdClass();
$data->series = array(
    $bestSales,
);
return $this->jsonResponse(json_encode($data));
}
public function getProductDetails()
{
    $productId = $this->getRequest()->query->get('productId');
    $startDay = $this->getRequest()->query->get('startDay', date('d'));
    $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endDay = $this->getRequest()->query->get('endDay', date('d'));
    $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $startDate = new DateTime($startYear . '-' . $startMonth . '-' .
$startDate);
    $endDate = new DateTime($endYear . '-' . $endMonth . '-' . $endDay);
    $locale = $this->getSession()->getLang()->getLocale();
    $result = $this->getStatisticHandler()->productDetails($startDate,
$endDate, $productId, $locale);
    return $this->jsonResponse(json_encode($result));
}
public function statDiscountCodeAction()
{
    $startDay = $this->getRequest()->query->get('startDay', date('d'));
    $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endDay = $this->getRequest()->query->get('endDay', date('d'));
    $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $startDate = new DateTime($startYear . '-' . $startMonth . '-' .
$startDate);
    $endDate = new DateTime($endYear . '-' . $endMonth . '-' . $endDay);
    $discount = new stdClass();
    $result = $this->getStatisticHandler()->discountCode($startDate,
$endDate);
    foreach ($result as &$coupon) {

```



```

/** @var \Thelia\Coupon\Type\CouponInterface $couponService */
$couponService = $this->getSpecificCouponService($coupon['type']);
$coupon['rule'] = $couponService->getName();
}
unset($coupon);
$discount->table = $result;
$discount->thead = array(
    'code' => $this->getTranslator()-
>trans('tool.panel.general.discountCode.code', [], Statistic::MESSAGE_DOMAIN),
    'rule' => $this->getTranslator()-
>trans('tool.panel.general.discountCode.rule', [], Statistic::MESSAGE_DOMAIN),
    'total' => $this->getTranslator()-
>trans('tool.panel.general.discountCode.nbUse', [], Statistic::MESSAGE_DOMAIN),
);
$data = new \stdClass();
$data->series = array(
    $discount,
);
return $this->jsonResponse(json_encode($data));
}
public function statMeansTransportAction()
{
    $startDate = $this->getRequest()->query->get('startDate', date('d'));
    $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endDate = $this->getRequest()->query->get('endDate', date('d'));
    $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $startDate = new \DateTime($startYear . '-' . $startMonth . '-' .
$startDate);
    $endDate = new \DateTime($endYear . '-' . $endMonth . '-' . $endDate);
    $local = $this->getSession()->getLang()->getLocale();
    $transport = new \stdClass();
    $transport->table = $this->getStatisticHandler()-
>meansTransport($startDate, $endDate, $local);
    $transport->thead = array(
        'code' => $this->getTranslator()-
>trans('tool.panel.general.meansTransport.means', [], Statistic::MESSAGE_DOMAIN),
        'title' => $this->getTranslator()-
>trans('tool.panel.general.meansTransport.description', [],
Statistic::MESSAGE_DOMAIN),
        'total' => $this->getTranslator()-
>trans('tool.panel.general.meansTransport.nbUse', [], Statistic::MESSAGE_DOMAIN),
);
    $data = new \stdClass();
    $data->series = array(
        $transport,
);
    return $this->jsonResponse(json_encode($data));
}
public function statMeansPaymentAction()
{
    $startDate = $this->getRequest()->query->get('startDate', date('d'));
    $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endDate = $this->getRequest()->query->get('endDate', date('d'));
    $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $startDate = new \DateTime($startYear . '-' . $startMonth . '-' .
$startDate);
    $endDate = new \DateTime($endYear . '-' . $endMonth . '-' . $endDate);

    $local = $this->getSession()->getLang()->getLocale();

```

Кафедра інтелектуальних інформаційних систем
 Аналіз статистичного ядра e-commerce платформи на основі Shopify

```

    $payment = new \stdClass();
    $payment->table = $this->getStatisticHandler()->meansPayment($startDate,
$endDate, $local);
    $payment->thead = array(
        'code' => $this->getTranslator()-
>trans('tool.panel.general.meansPayment.means', [], Statistic::MESSAGE_DOMAIN),
        'title' => $this->getTranslator()-
>trans('tool.panel.general.meansPayment.description', [],
Statistic::MESSAGE_DOMAIN),
        'total' => $this->getTranslator()-
>trans('tool.panel.general.meansPayment.nbUse', [], Statistic::MESSAGE_DOMAIN),
    );
    $data = new \stdClass();
    $data->series = array(
        $payment,
    );
    return $this->jsonResponse(json_encode($data));
}
public function statTurnoverAction()
{
    if ($session = $this->getRequest()->getSession()) {
        $session->save();
    }
    setlocale(LC_TIME, 'fr_FR.utf8', 'fra');
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $result[$startYear] = $this->getStatisticHandler()-
>getTurnoverYear($startYear);
    $turnoverStart = new \stdClass();
    $turnoverStart->color = '#adadad';
    $turnoverStart->graph = $result[$startYear]['graph'];
    $turnoverStart->graphLabel = $result[$startYear]['month'];
    $turnoverStart->table = $result[$startYear]['table'];
    $turnoverStart->thead = array(
        'month' => $this->getTranslator()-
>trans('tool.panel.general.turnover.month', [], Statistic::MESSAGE_DOMAIN),
        'TTCWithShipping' => $this->getTranslator()-
>trans('tool.panel.general.turnover.TTCWithShipping', [],
Statistic::MESSAGE_DOMAIN),
        'TTCWithoutShipping' => $this->getTranslator()-
>trans('tool.panel.general.turnover.TTCWithoutShipping', [],
Statistic::MESSAGE_DOMAIN),
    );
    $data = new \stdClass();
    $data->series = array(
        $turnoverStart,
    );
    if ($startYear !== $endYear) {
        $result[$endYear] = $this->getStatisticHandler()-
>getTurnoverYear($endYear);
        $turnoverEnd = new \stdClass();
        $turnoverEnd->color = '#F00';
        $turnoverEnd->graph = $result[$endYear]['graph'];
        $turnoverEnd->graphLabel = $result[$endYear]['month'];
        $turnoverEnd->table = $result[$endYear]['table'];
        $turnoverEnd->thead = array(
            'month' => $this->getTranslator()-
>trans('tool.panel.general.turnover.month', [], Statistic::MESSAGE_DOMAIN),
            'TTCWithShipping' => $this->getTranslator()-
>trans('tool.panel.general.turnover.TTCWithShipping', [],
Statistic::MESSAGE_DOMAIN),
            'TTCWithoutShipping' => $this->getTranslator()-
>trans('tool.panel.general.turnover.TTCWithoutShipping', [],

```

```

Statistic::MESSAGE_DOMAIN),
    );
    $data->series[] = $turnoverEnd;
    $data->title = $this->getTranslator()->trans("Stats on %startYear and
%endYear", array('%startYear' => $startYear, '%endYear' => $endYear),
Statistic::MESSAGE_DOMAIN);
    } else {
        $data->title = $this->getTranslator()->trans("Stats on %startYear",
array('%startYear' => $startYear), Statistic::MESSAGE_DOMAIN);
    }
    return $this->jsonResponse(json_encode($data));
}
public function statRevenueAction()
{
    if ($session = $this->getRequest()->getSession()) {
        $session->save();
    }
    $ghost = $this->getRequest()->query->get('ghost');
    $startDay = $this->getRequest()->query->get('startDay', date('d'));
    $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endDay = $this->getRequest()->query->get('endDay', date('d'));
    $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $startDate = new \DateTime($startYear . '-' . $startMonth . '-' .
$startDay);
    $endDate = new \DateTime($endYear . '-' . $endMonth . '-' . $endDay);
    $saleSeries = new \stdClass();
    if ($startDate->diff($endDate)->format('%a') === '0') {
        $result = $this->getStatisticHandler()-
>getRevenueStatsByHours($startDate);
    } else {
        $result = $this->getStatisticHandler()->getRevenueStats($startDate,
$endDate);
    }
    $saleSeries->color = '#adadad';
    $saleSeries->graph = $result['stats'];
    $saleSeries->graphLabel = $result['label'];
    $data = new \stdClass();
    if ($startDay === $endDay && $startMonth === $endMonth && $startYear ===
$endYear) {
        $data->title = $this->getTranslator()->trans("Stats between
%startDay/%startMonth/%startYear", array(
            '%startDay' => $startDay,
            '%startMonth' => $startMonth,
            '%startYear' => $startYear,
        ), Statistic::MESSAGE_DOMAIN);
    } else {
        $data->title = $this->getTranslator()->trans("Stats between
%startDay/%startMonth/%startYear and %endDay/%endMonth/%endYear", array(
            '%startDay' => $startDay,
            '%startMonth' => $startMonth,
            '%startYear' => $startYear,
            '%endDay' => $endDay,
            '%endMonth' => $endMonth,
            '%endYear' => $endYear
        ), Statistic::MESSAGE_DOMAIN);
    }
    $data->series = array(
        $saleSeries,
    );
    if ((int)$ghost === 1) {
        if ($startDate->diff($endDate)->format('%a') === '0') {

```

```

$ghostGraph = $this->getStatisticHandler()-
>getRevenueStatsByHours($startDate->sub(new DateInterval('P1Y')));
    } else {
        $ghostGraph = $this->getStatisticHandler()->getRevenueStats(
            $startDate->sub(new DateInterval('P1Y')),
            $endDate->sub(new DateInterval('P1Y'))
        );
    }
    $ghostCurve = new \stdClass();
    $ghostCurve->color = "#38acfc";
    $ghostCurve->graph = $ghostGraph['stats'];
    $data->series[] = $ghostCurve;
}
return $this->jsonResponse(json_encode($data));
}
public function statOrdersAction()
{
    if ($session = $this->getRequest()->getSession()) {
        $session->save();
    }
    $ghost = $this->getRequest()->query->get('ghost');
    $startDay = $this->getRequest()->query->get('startDay', date('d'));
    $startMonth = $this->getRequest()->query->get('startMonth', date('m'));
    $startYear = $this->getRequest()->query->get('startYear', date('Y'));
    $endDay = $this->getRequest()->query->get('endDay', date('d'));
    $endMonth = $this->getRequest()->query->get('endMonth', date('m'));
    $endYear = $this->getRequest()->query->get('endYear', date('Y'));
    $startDate = new \DateTime($startYear . '-' . $startMonth . '-' .
$startDay);
    $endDate = new \DateTime($endYear . '-' . $endMonth . '-' . $endDay);
    $saleSeries = new \stdClass();
    if ($startDate->diff($endDate)->format('%a') === '0') {
        $result = $this->getStatisticHandler()-
>getOrdersStatsByHours($startDate);
    } else {
        $endDate->add(new DateInterval('P1D'));
        $result = $this->getStatisticHandler()->getOrdersStats($startDate,
$endDate);
    }
    $saleSeries->color = '#d10d0d';
    $saleSeries->graph = $result['stats'];
    $saleSeries->graphLabel = $result['label'];
    $data = new \stdClass();
    if ($startDay === $endDay && $startMonth === $endMonth && $startYear ===
$endYear) {
        $data->title = $this->getTranslator()->trans("Stats between
%startDay/%startMonth/%startYear", array(
            '%startDay' => $startDay,
            '%startMonth' => $startMonth,
            '%startYear' => $startYear,
        ), Statistic::MESSAGE_DOMAIN);
    } else {
        $data->title = $this->getTranslator()->trans("Stats between
%startDay/%startMonth/%startYear and %endDay/%endMonth/%endYear", array(
            '%startDay' => $startDay,
            '%startMonth' => $startMonth,
            '%startYear' => $startYear,
            '%endDay' => $endDay,
            '%endMonth' => $endMonth,
            '%endYear' => $endYear
        ), Statistic::MESSAGE_DOMAIN);
    }
    $data->series = array(

```

```

    $saleSeries,
  );
  if ((int)$ghost === 1) {
    if ($startDate->diff($endDate)->format('%a') === '0') {
      $ghostGraph = $this->getStatisticHandler()-
>getOrdersStatsByHours($startDate->sub(new DateInterval('P1Y')));
    } else {
      $ghostGraph = $this->getStatisticHandler()->getOrdersStats(
        $startDate->sub(new DateInterval('P1Y')),
        $endDate->sub(new DateInterval('P1Y'))
      );
    }
    $ghostCurve = new \stdClass();
    $ghostCurve->color = "#38acfc";
    $ghostCurve->graph = $ghostGraph['stats'];
    $data->series[] = $ghostCurve;
  }
  return $this->jsonResponse(json_encode($data));
}
protected $statisticHandler;
protected function getStatisticHandler()
{
  if (!isset($this->statisticHandler)) {
    $this->statisticHandler = $this->getContainer()-
>get('statistic.handler.statistic');
  }
  return $this->statisticHandler;
}
protected $couponsServices = array();
protected function getSpecificCouponService($serviceId)
{
  if (!isset($this->couponsServices[$serviceId])) {
    $this->couponsServices[$serviceId] = $this->getContainer()-
>get($serviceId);
  }
  return $this->couponsServices[$serviceId];
}
}

```