

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «СТВОРЕННЯ ПЗ ДЛЯ ЗБОРУ ДОДАТКОВИХ
ПЛАТЕЖІВ З КЛІЄНТІВ ІНТЕРНЕТ-МАГАЗИНУ НА
ОСНОВІ ГЕОГРАФІЧНОЇ ЛОКАЦІЇ»

Виконав: студент _____ 2 _____ курсу, групи _____ 8.1262
спеціальності _____ 126 інформаційні системи та технології
(шифр і назва спеціальності)

інформаційні системи та штучний
освітньої програми _____ інтелект
(назва освітньої програми)

_____ А.В. Котюх
(ініціали та прізвище)

Керівник _____ кандидат технічних наук, доцент, д.т.н. Борю С.Ю.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент _____ кандидат фізико-математичних наук, доцент кафедри
програмної інженерії, д.т.н. Горбенко В.І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний
Кафедра комп'ютерних наук
Рівень вищої освіти магістр
Спеціальність 126 інформаційні системи та технології
(шифр і назва)
Освітня програма інформаційні системи та штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

“ _____ ” _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Котюху Артему Валерійовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи створення пз для збору додаткових платежів з клієнтів
інтернет-магазину на основі географічної локації
- керівник роботи Борю Сергій Юрійович, д.т.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)
- затверджені наказом ЗНУ від « 01 » травня 2023 року № 642-с
2. Строк подання студентом роботи 10.11.2023
3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розробка пз для збору додаткових платежів з клієнтів інтернет-магазину
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 10.05.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	18.05.2023	
2.	Збір вихідних даних.	27.06.2023	
3.	Обробка методичних та теоретичних джерел.	19.07.2023	
4.	Розробка першого та другого розділу.	12.08.2023	
5.	Розробка третього розділу.	13.10.2023	
6.	Оформлення та нормоконтроль кваліфікаційної роботи магістра.	01.12.2023	
7.	Захист кваліфікаційної роботи.	13.12.2023	

Студент _____
(підпис)

А.В. Котюх
(ініціали та прізвище)

Керівник роботи _____
(підпис)

С.Ю. Борю
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.Г. Спиця
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Створення ПЗ для збору додаткових платежів з клієнтів інтернет-магазину на основі географічної локації»: 41 с., 7 рис., 0 табл., 17 джерел, 0 додатків.

Архітектура веб-додатку, веб-магазин, географічна локація, додаткові платежі, збір платежів, інтернет-торгівля, клієнтський досвід, програмне забезпечення.

Об'єкт дослідження – сегмент інтернет-торгівлі у різних країнах, програмне забезпечення, спрямоване на ефективний збір додаткових платежів від клієнтів інтернет-магазину, заснований на аналізі їх географічного розташування.

Мета роботи: розробити та впровадити програмний модуль Magento 2 для реалізації збору додаткових платежів в інтернет-магазинах на основі географічної локації клієнтів, зокрема в Канаді.

Метод дослідження – описовий аналітичний.

Кваліфікаційна робота розглядає розробку модулю для Magento 2 , який не лише збирає додаткові платежі в інтернет-магазині, але і враховує географічну локацію клієнтів. За допомогою цього модулю бізнес стає більш гнучким та прибутковим. Модуль був успішно розроблений, демонструючи позитивний вплив на фінансовий результат і задоволення клієнтів. Перед встановленням рекомендується вивчити специфіку бізнесу та встановити параметри збору платежів відповідно до географічних властивостей аудиторії. Сфера застосування отриманих результатів охоплює електронну комерцію у всіх країнах, де діють магазини на платформі Magento 2.

Висновки дослідження підтверджують досягнення цілей, пропозиції щодо розвитку передбачають розширення функціоналу, включаючи соціальні та економічні характеристики клієнтів.

SUMMARY

Master's qualifying paper «Development of Software for Collecting Additional Payments from Online Store Customers Based on Geographical Location»: 41 pages, 7 figures, 0 tables, 17 references, 0 supplements.

Web application architecture, online store, geographic location, additional tax, payment tax, e-commerce, customer experience, software.

Object of investigation – the segment of e-commerce in various countries, software designed for efficient collection of additional payments tax from web store customers, based on the analysis of their geographic location.

Purpose of the paper: develop and implement a Magento 2 software module to facilitate the collection of additional payments in online stores based on the geographical location of clients, particularly in Canada.

Method of investigation – descriptive analytical.

Master's qualifying paper explores the development of a module for Magento 2, which not only gathers additional payments in an online store but also considers the geographic location of customers. Through this module, businesses become more flexible and profitable. The module has been successfully developed, showcasing a positive impact on financial results and customer satisfaction. Before installation, it is recommended to understand the specifics of the business and set payment collection parameters based on the geographic characteristics of the audience. The application scope of the obtained results covers e-commerce in all countries where stores operate on the Magento 2 platform.

The investigation findings confirm the achievement of set objectives, and development proposals involve expanding functionality, including social and economic characteristics of customers.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary.....	5
Вступ.....	7
1 Особливості інтернет-торгівлі в різних країнах	9
1.1 Основні задачі інтернет-магазинів залежно від географії.....	9
1.2 Основні методи визначення місцезнаходження клієнта	10
1.3 Огляд прикладів пз для збору платежів з урахуванням географічної локації клієнтів.....	12
1.4 Постановка задачі.....	15
2 Розробка пз для реалізації збору додаткових платежів.....	16
2.1 Розробка вимог до функціональності програмного забезпечення.....	16
2.2 Розробка uml моделі	17
2.3 Розробка інформаційної моделі	19
2.4 Розробка алгоритмів.....	20
2.5 Розробка інструкції по експлуатації та впровадженню.....	22
3 Реалізація пз.....	24
3.1 Вибір інструментарію для реалізації пз	24
3.2 Реалізація налаштувань в адміністративній панелі	24
3.3 Реалізація відображення додаткового податка	34
Висновки.....	39
Перелік посилань	40

ВСТУП

Інтернет-торгівля, яка стала важливим аспектом сучасного способу життя, надає споживачам зручність та широкий вибір у сфері покупок. Разом із зростанням популярності інтернет-магазинів виникають нові завдання, включаючи ефективний збір додаткових платежів від клієнтів. Географічна локація клієнтів стає ключовим фактором у цьому процесі. Шляхом збору додаткових платежів на основі географічної локації, інтернет-магазини отримують можливість персоналізувати підхід до кожного клієнта та оптимізувати свою діяльність. Для досягнення цього мети необхідне спеціалізоване програмне забезпечення.

У цьому дослідженні розглянуті особливості інтернет-торгівлі в різних країнах, визначимо основні виклики, з якими стикаються інтернет-магазини в залежності від географії, та проведемо огляд існуючих рішень у цій області, щоб визначити сучасні тенденції та найефективніші підходи. Фокусуючись на постановці задачі та зборі теоретичної інформації, визначено потенційні сфери впровадження розробленого програмного модуля в практиці інтернет-торгівлі та суміжних галузях. Враховуючи реальні потреби бізнесу та споживачів, розглянуті перспективи впровадження нашого рішення, зокрема його вплив на оптимізацію бізнес-процесів, покращення взаємодії з клієнтами, забезпечення безпеки фінансових операцій та створення конкурентних переваг для інтернет-магазинів. Також спробуємо удосконалити процес збору додаткових платежів від клієнтів інтернет-магазину з урахуванням їхньої географічної локації.

Одним з важливих викликів є адаптація бізнесу до специфіки кожної країни, включаючи мови, валюти, законодавство та правила платежів. Використання програмного забезпечення з урахуванням географічної локації дозволить ефективно вирішити ці завдання та забезпечити зручні та безпечні платежі для клієнтів інтернет-магазину.

Основні проєктні рішення, які обрано, базуються на комплексному аналізі вимог інтернет-торгівлі та унікальних викликів, пов'язаних із збором додаткових платежів від клієнтів за їхньою географічною локацією. На цьому етапі обґрунтовані технічні вирішення, вибір алгоритмів та програмних інструментів, враховуючи вимоги до швидкодії, масштабованості та безпеки. Розробка оптимальних архітектурних рішень буде фокусуватися на забезпеченні максимальної ефективності та гнучкості програмного модуля.

Отже, метою цього дослідження є розробка та впровадження програмного модуля для оптимізації збору додаткових платежів в інтернет-магазинах, зокрема, на основі географічної локації клієнтів.

1 ОСОБЛИВОСТІ ІНТЕРНЕТ-ТОРГІВЛІ В РІЗНИХ КРАЇНАХ

1.1 Основні задачі інтернет-магазинів залежно від географії

Основні задачі інтернет-магазинів залежно від географічного розташування можуть включати різні аспекти, які варто враховувати при розробці та веденні бізнесу в конкретній країні.

Локалізація товарів і послуг: Інтернет-магазинам необхідно адаптувати свою пропозицію товарів та послуг до потреб і вподобань місцевих клієнтів. Це може включати надання місцевих варіантів товарів, забезпечення інформації про розміри, вагу та одиниці виміру, які використовуються в даній країні.

Коректне ціноутворення та валюта: Країни можуть мати різні валюти та вимоги до ціноутворення. Інтернет-магазинам потрібно забезпечити можливість відображення цін у місцевій валюті та розрахунку вартості товарів та послуг відповідно до місцевих вимог. Наприклад, в ЄС, де використовується євро, інтернет-магазинам необхідно пропонувати ціни в євро та використовувати правильні курси обміну валют для конвертації цін. [5]

Ефективна логістика та доставка – важливі аспекти інтернет-торгівлі. В країнах з розтягнутим територіальним розташуванням або на віддалених територіях інтернет-магазинам потрібно забезпечити доступність товарів та послуг для клієнтів. Це може включати партнерство з надійними службами доставки, врахування географічних особливостей для визначення термінів доставки та належну упаковку для захисту товарів під час транспортування. [1]

Інтернаціоналізація веб-сайту та мовний аспект. Розробка міжнародного веб-сайту з можливістю перекладу на мови різних країн та забезпечення зручності використання мовного інтерфейсу є важливими кроками для привернення та утримання місцевих клієнтів. [1]

Особливості маркетингу та реклами: Залежно від географічного положення інтернет-магазину, маркетингові стратегії та рекламні кампанії

повинні враховувати культурні, соціальні та економічні відмінності між країнами. Розробка ефективних маркетингових стратегій, які враховують ці особливості, допомагає забезпечити максимальний вплив на цільову аудиторію та підвищити конверсію. [1]

Специфікації щодо платіжних систем: Різні регіони можуть мати власні уподобання та популярні методи оплати. Врахування цих особливостей та надання різноманітних варіантів платіжних систем, таких як місцеві кредитні картки чи електронні гроші, покращує зручність для клієнтів та сприяє підвищенню рівня довіри.

Розглядаючи ці виклики, визначено кілька ключових проблем, які можуть бути вирішені за допомогою спеціалізованого програмного забезпечення для збору додаткових платежів від клієнтів в залежності від їхньої географічної локації. Наявність такого інструменту дозволить інтернет-магазинам ефективно збирати додаткові платежі, забезпечуючи оптимізований та персоналізований підхід до кожної аудиторії. Оптимізувати процеси, забезпечити адаптацію до місцевих умов та покращити взаємодію з клієнтами, що в свою чергу позитивно впливає на конкурентоспроможність та успіх інтернет-магазину на різних ринках.

1.2 Основні методи визначення місцезнаходження клієнта

Способи локалізації клієнта в інтернет-магазині включають в себе використання різних технологій та стратегій для визначення його місцезнаходження та адаптації обслуговування відповідно до місцевих законів та вимог.

Геолокація IP-адреси: визначення місцезнаходження клієнта за його IP-адресою є одним з найпоширеніших методів. Це забезпечує можливість адаптації мови, валюти та інших параметрів сайту.

Вибір мови веб-сайту: користувачам надається можливість вручну обрати мову веб-сайту, що дозволяє їм вибрати мову відповідно до їхніх вподобань.

Наприклад, інтернет-магазин може пропонувати розгортання випадуючого списку з вибором доступних мов, таких як англійська, французька, німецька тощо.

Автоматичний вибір валюти: Інтернет-магазин може автоматично визначати регіон користувача та пропонувати валюту, яка є стандартною для цього регіону. Наприклад, для користувачів з США може бути встановлено долар США як основну валюту.

Паспортні дані акаунта: Запит від користувача на введення паспортних даних або адреси може також слугувати методом локалізації. Наприклад, запит про введення поштового індексу може допомогти точніше визначити його регіон.

Використання GPS на мобільних пристроях: Додатки для мобільних пристроїв можуть використовувати GPS для визначення місцезнаходження клієнта.

Введення користувачем даних про доставку: В інтерфейсі замовлення інтернет-магазину користувач може вручну вказати адресу доставки, введення якої дозволяє точно визначити географічне розташування клієнта. Дані, такі як поштовий індекс, місто та країна, можуть служити важливими вказівниками для локалізації. Цей метод особливо корисний у випадках, клієнт хоче змінити адресу доставки вручну.

У контексті розробки модуля важливо враховувати не лише географічне розташування користувачів, але й різноманітність законодавства в різних країнах. Наприклад, в Європейському Союзі діє Загальний регламент з питань захисту даних (GDPR), який ставить високі стандарти щодо обробки та захисту особистих даних громадян. Зокрема, він визначає права осіб на контроль над своїми особистими даними та вимагає від компаній забезпечувати високий рівень конфіденційності та безпеки цих даних. [2]

Закон штату Каліфорнія з конфіденційності споживачів (CCPA) – надає жителям Каліфорнії право знати, які особисті дані про них збираються. Закон визначає геолокацію як інформацію про фізичне місце знаходження особи. Збір,

використання та розкриття геолокаційних даних підпадає під обов'язок розкриття та надання споживачеві можливості виключити їх з обробки. Це означає, що компанії, які збирають такі дані, повинні повідомити споживача про мету їх збору та надати можливість відмовитися від цього виду обробки. [8]

У Канаді правове регулювання електронної комерції відрізняється в залежності від штату. Наприклад, є штати, де існують спеціальні податки на Інтернет-торгівлю або вимоги щодо обов'язкового зберігання певних даних про клієнтів.

У розробці модуля для інтернет-магазину вибір Канади як об'єкта для дослідження і впровадження вибору «Введення користувачем даних про доставку» має свої підстави. Цей підхід дозволяє створити гнучкий та індивідуалізований підхід до обслуговування клієнтів, забезпечуючи їм можливість самостійно вводити необхідні дані для точної локалізації та оптимальної обробки замовлень. Збір додаткових платежів на основі цих даних може бути налаштований для врахування конкретних особливостей географічного розташування.

1.3 Огляд прикладів ПЗ для збору платежів з урахуванням географічної локації клієнтів

На сьогоднішній день існує значна кількість рішень для збору додаткових платежів з клієнтів інтернет-магазинів на основі географічної локації. Ці рішення розроблені з метою забезпечення зручності та ефективності оплати для клієнтів, а також сприяють успішній експлуатації бізнесу у різних регіонах світу.

Деякі платіжні платформи пропонують готові рішення, які враховують особливості кожної країни, такі як популярні методи оплати, валюти та правові вимоги. Це дозволяє інтернет-магазинам швидко запуснути оплату для клієнтів у різних країнах без необхідності розробки власного рішення з нуля.

Налаштування індивідуальних параметрів: Деякі платіжні рішення надають можливість налаштування індивідуальних параметрів залежно від

потреб бізнесу. Це означає, що інтернет-магазини можуть встановлювати різні методи оплати, валюти, обмеження та правила оплати в залежності від конкретних вимог свого бізнесу та ринку. [5]

Забезпечення безпеки та надійності. Багато платіжних рішень забезпечують високий рівень безпеки та захисту персональних даних клієнтів. Вони використовують шифрування та інші заходи безпеки для захисту платіжних транзакцій від шахрайства та несанкціонованого доступу. [7]

Міжнародна підтримка. Багато з цих рішень працюють у багатьох країнах та підтримують різні валюти. Це дозволяє інтернет-магазинам працювати на міжнародному рівні та здійснювати оплату замовлень з усього світу.

Ось декілька прикладів таких рішень:

– stripe – це платіжна платформа, яка надає гнучкі інструменти для прийому платежів в Інтернеті. Вона працює в багатьох країнах світу та підтримує різні валюти. Stripe надає API та готові плагіни для інтеграції з популярними платформами електронної комерції, такими як WooCommerce, Shopify, Magento тощо; [3]

– braintree – це платіжна платформа, яка належить компанії PayPal. Вона пропонує широкі можливості для обробки платежів в Інтернеті, включаючи різні методи оплати та підтримку міжнародних платежів. Braintree також має зручне API та готові плагіни для інтеграції з популярними платформами електронної комерції; [3]

– 2Checkout – це міжнародна платіжна платформа, яка дозволяє приймати платежі з усього світу та підтримує багато різних способів оплати. Вона пропонує інтеграцію з популярними електронними платформами та надає засоби для налаштування географічної сегментації та управління платежами; [3]

– payu – це платіжна платформа, яка активно працює в країнах Європи, Латинської Америки, Азії та Африки. Вона підтримує різні способи оплати, включаючи кредитні картки, банківські платежі та мобільні платежі. PayU надає

інтеграцію з популярними електронними платформами та має велику географічну охопленість; [3]

– amazon pay – це платіжна система, розроблена Amazon. Вона дозволяє клієнтам здійснювати платежі за допомогою свого облікового запису Amazon. Amazon Pay підтримує міжнародні операції та надає інструменти для налаштування різних параметрів оплати в залежності від географічної локації; [3].

– wechat pay та Alipay найпопулярніші методи платежів в Китаї. Це означає, що в інтернет-магазинах, спрямованих на китайський ринок, необхідно включити ці методи оплати, оскільки багато покупців не відкривають банківські рахунки та не користуються кредитними картками; [5]

Проте, при всіх перевагах використання існуючих рішень, виникають певні обмеження, які зумовили рішення розробити власний модуль:

Неадаптованість до специфіки Канади: Багато з існуючих рішень можуть не повністю враховувати особливості географічного розташування та правові норми, які характерні для Канади, включаючи податкові ставки, методи оплати та регіональні обмеження. [4]

Необхідність індивідуалізації: Інтернет-магазини можуть виявити потребу в більшій гнучкості та індивідуальних налаштуваннях, які можуть бути важко або неможливо реалізувати з використанням готових рішень.

Бажання досягти максимальної ефективності: Розробка власного модуля дозволить забезпечити максимально точну локалізацію та ефективну обробку замовлень, враховуючи конкретні особливості географічного розташування клієнтів у Канаді.

Отже, рішення розробити власний модуль для оптимізації збору платежів є обґрунтованим і відповідає специфіці завдань, які стоять перед інтернет-магазинами у Канаді.

1.4 Постановка задачі

Метою даної роботи є розробка та впровадження програмного модуля Magento 2 для реалізації збору додаткових платежів в інтернет-магазинах на основі географічної локації клієнтів, зокрема в Канаді.

Для досягнення цієї мети передбачено вирішення наступних завдань:

- аналіз проблеми;
- розробка вимог до функціональності ПЗ;
- розробка UML моделі;
- розробка інформаційної моделі;
- розробка алгоритмів;
- розробка інструкції по експлуатації та впровадженню;

2 РОЗРОБКА ПЗ ДЛЯ РЕАЛІЗАЦІЇ ЗБОРУ ДОДАТКОВИХ ПЛАТЕЖІВ

Аналіз проблеми: метою цього дослідження є розв'язання труднощів, пов'язаних із конфігурацією зборів за доставку в електронному магазині, зокрема, у контексті різних провінцій Канади. Насамперед, проблема виникає через неможливість налаштування різних зборів для кожної окремої провінції та складнощі управління цією процедурою в існуючій системі Magento 2.

Важливі аспекти проблеми включають відсутність можливості стягнення зборів для кожного. Відзначається, що відсутність ефективного механізму налаштування може призвести до втрати конкурентоспроможності та обмежень стратегічних можливостей магазину. Враховуючи це, виникає необхідність впровадження нового модуля, позначеного як «JWA_ProvinceSurcharge». Цей модуль буде спроектований для гнучкого управління зборами за доставку в залежності від розташування покупців у провінціях.

2.1 Розробка вимог до функціональності програмного забезпечення

Основна функціональність:

Можливість визначення індивідуального збору за доставку для кожного провінційного регіону.

- визначення налаштувань модуля через адміністративний інтерфейс;
- зберігання та управління налаштуваннями збору за провінційним регіоном у вигляді конфігураційних параметрів;

Адміністративний інтерфейс:

- можливість активації/деактивації модуля збору за провінцією;
- введення та збереження значень зборів за провінцією через адміністративний інтерфейс;
- наявність інтуїтивно зрозумілих інтерфейсів для додавання, редагування та видалення налаштувань зборів для конкретних провінцій;

Інтеграція з іншими модулями:

- сумісність із засобами розрахунку податків та доставки;

Захист та безпека:

- забезпечення конфіденційності налаштувань зборів за провінцією;
- перевірка та валідація введених користувачем значень для попередження можливих помилок;

Документація:

- створення докладної та зрозумілої документації, яка включає в себе інструкції з використання модуля та налаштування;
- надання прикладів використання та сценаріїв взаємодії з іншими компонентами системи;

Локалізація та підтримка мов:

- підтримка локалізації для різних мов та регіональних стандартів;
- забезпечення інтерфейсу для зміни мови в адміністративному інтерфейсі;

Тестування:

- створення тестів для валідації основної функціональності;
- забезпечення стабільної роботи після змін та оновлень;

Модульність та розширюваність:

- забезпечення можливості розширення функціональності для майбутніх потреб;
- дотримання стандартів Magento щодо створення розширень;

Ці вимоги визначають ключові функції та характеристики модуля, спрямовані на забезпечення його ефективної роботи та легкості у використанні.

2.2 Розробка UML моделі

В роботі було проаналізовано взаємозв'язки між сутностями додатку і на основі цих даних пропущється розглянути компоненти та їх взаємодію. Забезпечення

наложної конфігурації провінцій та збору платежів (Рис. 1), обробку інформації про провінції та визначення додаткової вартості для певних провінцій (Рис. 2), розрахунок та представлення додаткового збору в інтерфейсі магазину (Рис. 3).

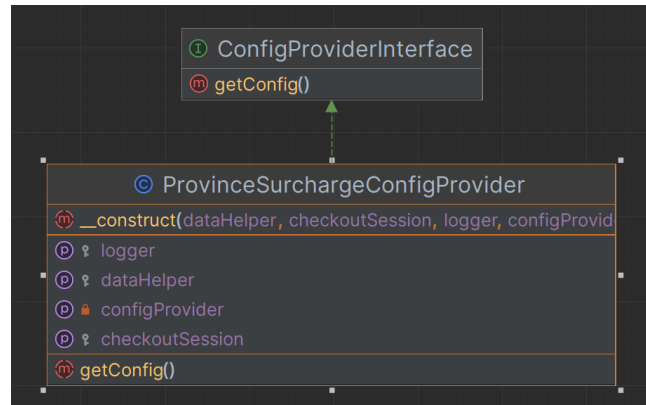


Рис. 1 – UML-діаграма ProvinceSurchargeConfigProvider моделі, що надає дані для розрахунку збору за доставку в інтернет-магазині

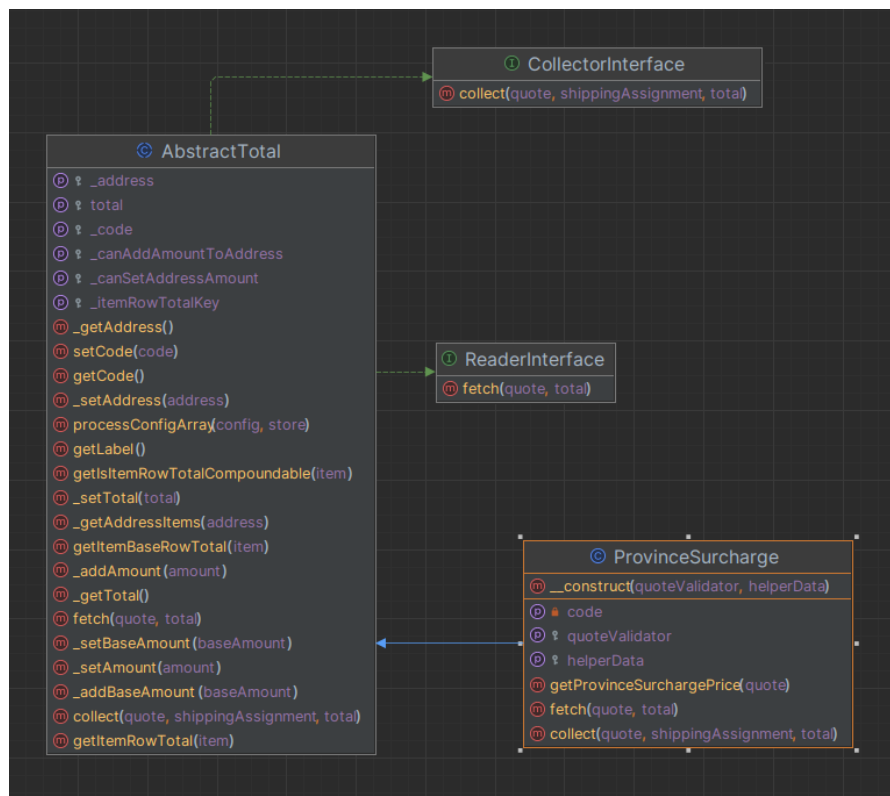


Рис. 2 – UML-діаграма моделі ProvinceSurcharge яка додає додатковий збір платежів залежно від географічного розташування клієнтів у інтернет-магазині

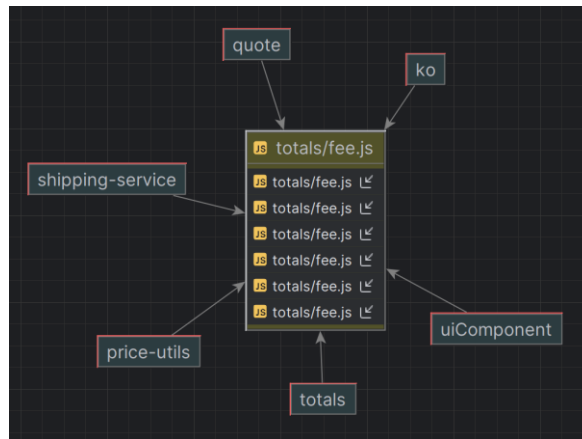


Рис. 3 – UML-діаграма скрипту для обчислення та відображення значення додаткового платежу

2.3 Розробка інформаційної моделі

Система Magento 2 використовує EAV (Entity-Attribute-Value) для зберігання та організації даних. Прямий зв'язок модуля визначається через взаємодію з п'ятьма таблицями з ядра Magento 2.

Одна з цих таблиць – `core_config_data` – служить для зберігання конфігураційних значень сум податків. Відзначимо, що таблиця `core_config_data` не має прямого зв'язку з іншими таблицями в системі (Рис. 4).

Інші чотири таблиці – `sales_order`, `customer_entity`, `quote`, `sales_creditmemo` – містять дані про націнку на товар в залежності від конфігураційних налаштувань, пов'язаних із геолокацією (Рис. 5).

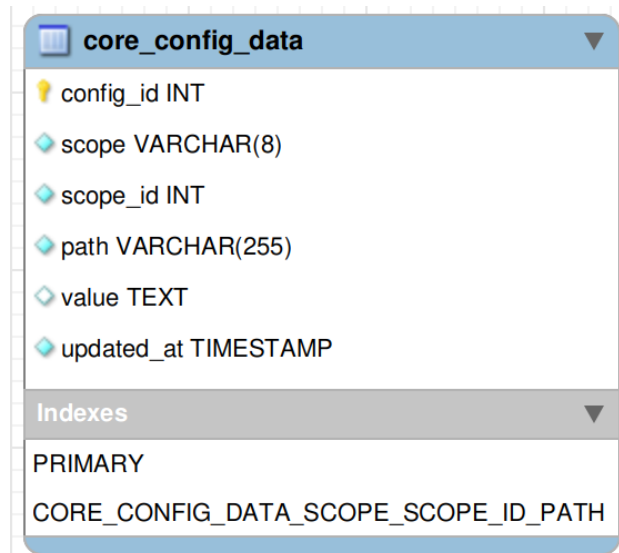


Рис. 4 – ER-діаграма, що відображає структуру `core_config_data`

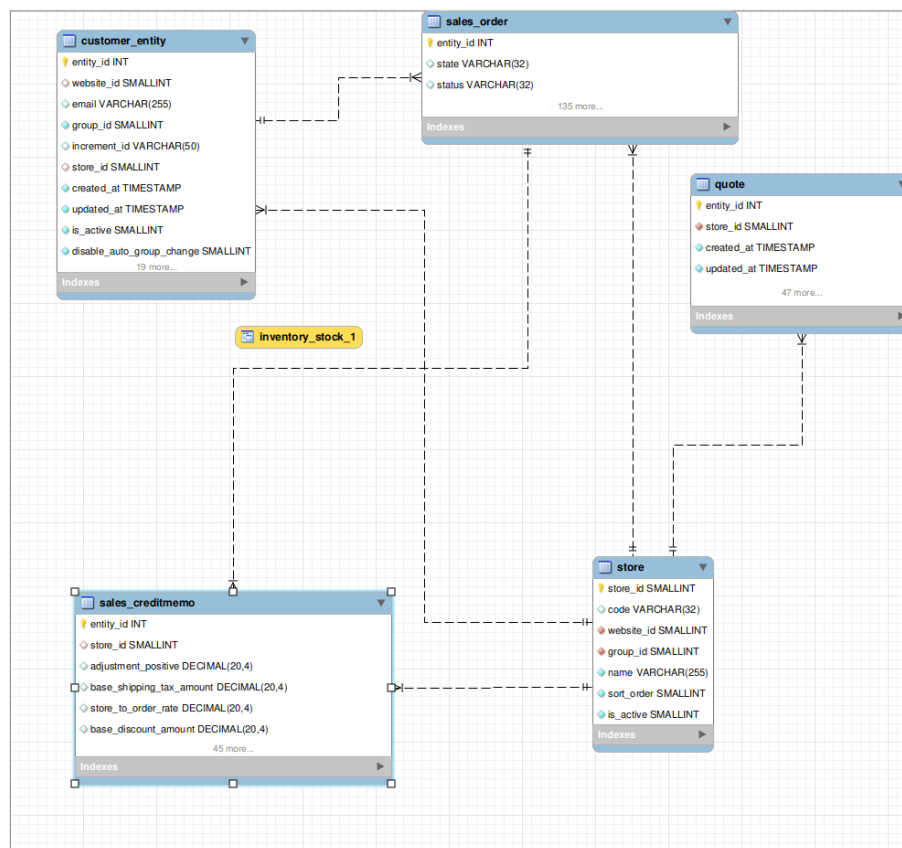


Рис. 5 – EER-діаграма взаємозв'язків між таблицями `sales_order`, `customer_entity`, `quote` та `sales_creditememo`

2.4 Розробка алгоритмів

Алгоритм визначення місцезнаходження клієнта при введенні адреси доставки

- 1) клієнт вводить адресу доставки під час оформлення замовлення;
- 2) система перевіряє, чи введена адреса достатня для визначення географічних координат;
 - а) якщо адреса не повна або не забезпечує необхідної точності, система повідомляє клієнта про необхідність докладніших введень;
 - б) якщо адреса достатня для визначення геолокації, переходьте до кроку 3);
- 3) система запитує веб-служби геолокації для отримання актуальних географічних координат на основі введеної адреси;
 - а) якщо веб-служби геолокації не доступні або надають неповні або невірні дані, система повідомляє клієнта про можливі проблеми та рекомендує перевірити введені дані;
 - б) якщо отримані географічні координати відповідають вимогам, переходьте до кроку 4);
- 4) система обробляє отримані географічні координати для визначення конкретного географічного регіону чи провінції;
 - а) збереження інформації про визначений географічний регіон у системі для подальшого використання;
 - б) повідомлення клієнта про успішне визначення місцезнаходження та продовження оформлення замовлення;
- 5) завершення алгоритму.

Алгоритм розрахунку додаткових платежів на основі геолокації клієнта

- 1) система отримує дані конфігурації провінції з адміністративного інтерфейсу, зокрема ставки додаткових платежів;
- 2) система визначає географічні координати клієнта згідно алгоритму визначення місцезнаходження;
- 3) система використовує отримані географічні координати для ідентифікації відповідного регіону чи провінції з конфігурації;

- a) якщо відповідний регіон не знайдено або дані неправильні, система повідомляє клієнта про можливі проблеми та рекомендує звернутися до служби підтримки;
 - b) якщо відповідний регіон ідентифіковано успішно, переходьте до кроку 4);
- 4) система отримує ставку додаткового платежу для визначеного регіону;
 - 5) система звертається до замовлення та отримує загальну суму товарів, до якої буде застосовано додатковий платіж;
 - 6) система застосовує ставку додаткового платежу до загальної суми товарів для розрахунку додаткового платежу;
 - 7) збереження інформації про розрахований додатковий платіж у системі та повідомлення клієнта про його включення до загальної суми замовлення;
 - 8) завершення алгоритму;

2.5 Розробка інструкції по експлуатації та впровадженню

Дана інструкція призначена для ефективного впровадження та користування модулем «JWA_ProvinceSurcharge» в Magento 2. Вона включає в себе ключові кроки для встановлення, налаштування та ефективного використання функціоналу модуля.

- 1) встановити модуль використовуючи команду `composer require jwa/module-province-surcharge`;
- 2) виконати команду `bin/magento setup:upgrade` в кореневій директорії проекту Magento 2 для того щоб увімкнути модуль;
- 3) перейти в адмін-панель Magento 2 та налаштувати збори за доставку для різних провінцій в розділі конфігурацій модуля «JWA_ProvinceSurcharge»;
- 4) додати та редагувати збори за доставку для кожної провінції за допомогою адміністративного інтерфейсу Magento;

- 5) перевірити відображення нового податку на сторінці сплати;
- 6) регулярно оновлювати та звертатися до документації модуля;

3 РЕАЛІЗАЦІЯ ПЗ

3.1 Вибір інструментарію для реалізації ПЗ

Magento 2 було обрано через її потужність та гнучкість у створенні електронних магазинів. Вона забезпечує широкі можливості налаштування, величезну спільноту розробників та підтримку розширень для розвитку функціоналу. Також, Magento 2 вважається стандартом для електронної комерції, що гарантує його надійність та актуальність.

Оскільки основною метою є розробка модуля для Magento 2, то був обраний такий інструментарій:

- php;
- apache2;
- js;
- jQuery;
- knockout.js;
- mysql;
- composer;
- docker;
- git;
- elasticsearch;

Цей інструментарій широко використовується у модулях Magento. [6]

3.2 Реалізація налаштувань в адміністративній панелі

Розроблено функціонал, який дозволяє адміністраторам налаштовувати податкові параметри товарів в залежності від географічного розташування для конкретних провінцій через адміністративну панель (Рис. 6).

Province Surcharge

Apply Province Surcharge [website] Yes

Province Surcharge Label [website] Province Surcharge

Province Surcharge [store view]

Price	Province	Action
19	Alberta	
54	Ontario	
23	Manitoba	
34	British Columbia	
59	New Brunswick	
11	Northwest Territories	

Add

Рис. 6 – Параметри налаштування податків для певної провінції Канади

Програмний код - XML-конфігураційний файл

```
<?xml version="1.0"?>
```

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Config/etc/system_file.xsd">
```

```
  <system>
```

```
    <section id="province_surcharge" translate="label" type="text" sortOrder="310" showInDefault="1" showInWebsite="1" showInStore="1">
```

```
      <label>Province Surcharge</label>
```

```

<tab>sales</tab>
<resource>JWA_ProvinceSurcharge::province_surcharge</resource>
<group id="province_surcharge" translate="label" type="text" sortOrder="1"
showInDefault="1" showInWebsite="1" showInStore="0">
  <label>Province Surcharge</label>
  <field id="enable" translate="label" type="select" sortOrder="10" showInD
efault="1" showInWebsite="1" showInStore="0">
    <label>Apply Province Surcharge</label>
    <source_model>Magento\Config\Model\Config\Source\Yesno</source_
model>
  </field>
  <field id="label" translate="label" type="text" sortOrder="20" showInDefa
ult="1" showInWebsite="1" showInStore="0">
    <label>Province Surcharge Label</label>
    <depends>
      <field id="enable">1</field>
    </depends>
  </field>
  <field id="province_surcharge_rows" translate="label" sortOrder="30" sho
wInDefault="1" showInWebsite="1" showInStore="1">
    <label>Province Surcharge</label>
    <frontend_model>TGK\ProvinceSurcharge\Block\Adminhtml\Form\Fiel
d\ProvinceSurcharges</frontend_model>
    <backend_model>Magento\Config\Model\Config\Backend\Serialized\Ar
raySerialized</backend_model>
    <depends>
      <field id="enable">1</field>
    </depends>
  </field>
</group>

```

```

    </section>
</system>
</config>

```

PHP блоки для додатку, який дозволяє адміністраторам встановлювати оплату за провінцію для товарів у канадському регіоні через адміністративну панель. [10]

```
<?php
```

```
declare(strict_types=1);
```

```
namespace TGK\ProvinceSurcharge\Block\Adminhtml\Form\Field;
```

```
use TGK\ProvinceSurcharge\Service\Validator;
```

```
use Magento\Directory\Api\CountryInformationAcquirerInterface;
```

```
use Magento\Framework\Exception\NoSuchEntityException;
```

```
use Magento\Framework\View\Element\Context;
```

```
use Magento\Framework\View\Element\Html>Select;
```

```
/**
```

```
 * Province column block for displaying a select input with Canadian provinces.
```

```
*/
```

```
class ProvinceColumn extends Select
```

```
{
```

```
    /**
```

```
     * @var CountryInformationAcquirerInterface
```

```
     */
```

```
    private $countryInformationAcquirer;
```

```

/**
 * @param CountryInformationAcquirerInterface $countryInformationAcquirer
 * @param Context $context
 * @param array $data
 */
public function __construct(
    CountryInformationAcquirerInterface $countryInformationAcquirer,
    Context $context,
    array $data = []
) {
    parent::__construct($context, $data);
    $this->countryInformationAcquirer = $countryInformationAcquirer;
}

/**
 * Set "name" for <select> element.
 *
 * @param string $value
 *
 * @return $this
 */
public function setInputName(string $value): self
{
    return $this->setData('name', $value);
}

/**
 * Set "id" for <select> element.
 *
 * @param string $value

```

```
*  
* @return $this  
*/  
public function setInputId(string $value): self  
{  
    return $this->setId($value);  
}  
  
/**  
* Render block HTML.  
*  
* @throws NoSuchEntityException  
*  
* @return string  
*/  
public function _toHtml(): string  
{  
    if (!$this->getOptions()) {  
        $this->setOptions($this->getCanadianProvinces());  
    }  
  
    return parent::_toHtml();  
}  
  
/**  
* Get available Canadian provinces.  
*  
* @throws NoSuchEntityException  
*  
* @return array
```

```

*/
private function getCanadianProvinces(): array
{
    $countryCanadaInfo = $this->countryInformationAcquirer
        ->getCountryInfo(Validator::CANADA_COUNTRY_CODE);

    $provinces = [];
    if ($availableProvinces = $countryCanadaInfo->getAvailableRegions()) {
        foreach ($availableProvinces as $province) {
            $provinces [] = [
                'label' => $province->getName(),
                'value' => $province->getId()
            ];
        }
    }

    return $provinces;
}
}

```

```
<?php
```

```
declare(strict_types=1);
```

```
namespace TGK\ProvinceSurcharge\Block\Adminhtml\Form\Field;
```

```
use Magento\Config\Block\System\Config\Form\Field\FieldArray\AbstractFieldArra
y;
```

```
use Magento\Framework\DataObject as MagentoObject;
```

```
use Magento\Framework\Exception\LocalizedException;
```

```

/**
 * Block that renders an array of fields for configuring surcharges by province.
 */
class ProvinceSurcharges extends AbstractFieldArray
{
    /**
     * @var ProvinceColumn
     */
    private $provinceRenderer;

    /**
     * Add columns to the array of fields.
     *
     * @throws LocalizedException
     *
     * @return void
     */
    protected function _prepareToRender(): void
    {
        $this->addColumn('price', [
            'label' => __('Price'),
            'class' => 'required-entry'
        ]);

        $this->addColumn('province', [
            'label' => __('Province'),
            'renderer' => $this->getProvinceRenderer()
        ]);
    }
}

```

```

    $this->_addAfter = false;
    $this->_addButtonLabel = __('Add');
}
/**
 * Prepare existing row data object.
 *
 * @param MagentoObject $row
 *
 * @throws LocalizedException
 *
 * @return void
 */
protected function _prepareArrayRow(MagentoObject $row): void
{
    $options = [];

    $province = $row->getData('province');
    if ($province !== null) {
        $options = $this->getProvinceOptions($province);
    }

    $row->setData('option_extra_attrs', $options);
}
/**
 * Get options for a province.
 *
 * @param string $province
 *
 * @throws LocalizedException
 *

```



```

* @return array
*/
private function getProvinceOptions(string $province): array
{
    $optionHash = $this->getProvinceRenderer()->calcOptionHash($province);
    $options ['option_' . $optionHash] = 'selected="selected"';

    return $options;
}
/**
 * Get the ProvinceColumn renderer for the 'Province' select field.
 *
 * @throws LocalizedException
 *
 * @return ProvinceColumn
 */
private function getProvinceRenderer(): ProvinceColumn
{
    if (!$this->provinceRenderer) {
        $this->provinceRenderer = $this->getLayout()->createBlock(
            ProvinceColumn::class,
            "",
            ['data' => ['is_render_to_js_template' => true]]
        );
    }

    return $this->provinceRenderer;
}
}

```

3.3 Реалізація відображення додаткового податка

Додана функціональність, яка дозволяє відобразити додаткові податки на товари на сторінці оплати як додаткову суму до сплати (Рис. 7).

The screenshot shows a payment page with two main sections: 'Payment Method' and 'Order Summary'.

Payment Method: Under 'Check / Money order', there is a checked checkbox 'My billing and shipping address are the same'. Below it, the address is listed: Artem TGK, Sweet 19, Ontario, Ontario A1B2C3, Canada, 2505550199. A blue 'Place Order' button is visible.

Order Summary: A table showing the order breakdown:

Cart Subtotal	\$49.00
Province Surcharge	\$54.00
Shipping Flat Rate - Fixed	\$5.00
Order Total	\$108.00

The 'Province Surcharge' row is highlighted with a red border. Below the summary, the shipping address is repeated under 'Ship To:' and the shipping method is 'Flat Rate - Fixed'.

Рис. 7 – Відображення додаткової суми до сплати

Програмний код

Реалізації компоненту, який виводить додатковий збір за провінцію в розділі оформлення замовлення.

```
<tr class="totals fee excl" data-bind="visible: isVisible">
  <th class="mark" colspan="1" scope="row" data-bind="text:
getProvinceSurchargeLabel()"></th>
  <td class="amount">
    <span class="price" data-bind="text: getFormattedPrice()"></span>
  </td>
</tr>
```

```

define( [
    'ko',
    'uiComponent',
    'Magento_Checkout/js/model/quote',
    'Magento_Catalog/js/price-utils',
    'mage/translate',
    'Magento_Checkout/js/model/totals'
], function (ko, Component, quote, priceUtils, $t, totals) {
    'use strict';

    return Component.extend({
        defaults: {
            isVisible: ko.observable(false),
            formattedPrice: ko.observable(""),
            provinceSurchargeLabel: ko.observable("")
        },
        totals: quote.getTotals(),

        /**
         * Initializes component
         */
        initialize: function () {
            this._super();
            this._initProvinceSurcharge();

            quote.totals.subscribe(this._initProvinceSurcharge.bind(this));

            return this;
        },
    });

```

```

/**
 * Initializes the Province Surcharge component
 *
 * @private
 */
_initProvinceSurcharge: function () {
  const config = window.checkoutConfig.province_surcharge_config || {};
  const label = config.label || "";

  const amount = (this.totals() &&
totals.getSegment('province_surcharge_amount'))
  ? totals.getSegment('province_surcharge_amount').value
  : 0.0;

  if (amount > 0 && label) {
    this.isVisible(true);
    this._setFormattedPrice(amount);
    this._setProvinceSurchargeLabel(label);
  } else {
    this.isVisible(false);
  }
},

/**
 * Sets the formatted province surcharge price
 *
 * @param {Number} amount - The amount of the province surcharge
 * @private
 */
_setFormattedPrice: function (amount) {

```

```

const priceFormat = quote.getPriceFormat();
try {
    this.formattedPrice(priceUtils.formatPrice(amount, priceFormat));
} catch (e) {
    console.error('Error formatting price: ', e);
    this.formattedPrice(null);
}
},

/**
 * Sets the province surcharge label
 *
 * @param {String} label - The label of the province surcharge
 * @private
 */
_setProvinceSurchargeLabel: function (label) {
    this.provinceSurchargeLabel($t(label));
},

/**
 * Returns formatted province surcharge price
 *
 * @return {String}
 */
getFormattedPrice: function () {
    return this.formattedPrice();
},

/**
 * Returns province surcharge label

```

```
*  
* @return {String}  
*/  
getProvinceSurchargeLabel: function () {  
    return this.provinceSurchargeLabel();  
}  
});  
});
```

Наданий код використовує бібліотеку Knockout.js (означену як 'ko' відповідно до конвенцій Magento) для реалізації функціональності на сторінці оформлення замовлення в Magento 2.

Knockout.js використовується для зручного та ефективного зв'язування даних із користувацьким інтерфейсом, включаючи відображення та автоматичне оновлення значень. [9]

ВИСНОВКИ

В результаті виконання роботи розроблений та впроваджений програмний модуль Magento 2 для реалізації збору додаткових платежів в інтернет-магазинах на основі географічної локації клієнтів, зокрема в Канаді. Розробка використовує передові технології, такі як PHP 8.1, JavaScript, та MySQL, і заснована на модульній структурі, що полегшує інтеграцію з існуючими системами.

Важливим компонентом розробки є також фронтенд-частина, реалізована з використанням технологій Knockout.js та jQuery, яка забезпечує високу швидкість та зручність для кінцевого користувача. Крім того, наявність інтуїтивно зрозумілого інтерфейсу для адміністраторів полегшує налаштування податкових параметрів відповідно до вимог різних регіональних юрисдикцій.

Зокрема, впровадження модульної архітектури, яка дозволяє розширювати функціональність за необхідністю, сприяє простоті обслуговування та адаптації до змін потреб бізнесу. Враховуючи вищевказані аспекти, можна визначити, що розробка відповідає високим стандартам електронної комерції та є конкурентоздатною на ринку.

Таким чином, розроблене програмне забезпечення, побудоване на основі Magento 2, не лише відповідає вимогам сучасного електронного бізнесу, а й надає інтернет-магазинам значущі конкурентні переваги в умовах канадського ринку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Директива 2000/31/ЄС Європейського Парламенту і Ради «Про правові аспекти інформаційних послуг щодо електронної комерції на внутрішньому ринку» від 08.06.2000 р. // Правова інформатика. – № 2(6)/2005. – С. 72 – 89. Tembhurnekar C.M. Role and challenges of e-commerce in the economic development of India. International Journal of Multidisciplinary, 2021. Vol. 8. P. 70 – 77.
2. General Data Protection Regulation URL: <https://gdpr-info.eu/> (дата звернення : 29.06.2023)
3. Frank Kehl. The best payment gateway providers for small businesses. *Merchant Maverick*. URL: <https://www.merchantmaverick.com/best-payment-gateways-online-payment-processing/> (дата звернення : 01.07.2023)
4. Mueller S. C. Measuring and mapping the emergence of the digital economy: a comparison of the market capitalization in selected countries. Digital policy, regulation and governance. 2017. 19(5). P. 367 – 382.
5. Magento, now Adobe Commerce | eCommerce Software. URL: <https://business.adobe.com/products/magento/magento-commerce.html> (дата звернення : 04.07.2023)
6. Крегул Ю. Правове регулювання міжнародної електронної комерції. Зовнішня торгівля: економіка, фінанси, право. – 2018. – № 2. – С. 136 – 147.
7. Gazieva L.R. The impact of e-commerce on the digital economy. L. R. International Conference on Finance, Entrepreneurship and Technologies in Digital Economy, 2021. – P. 121 – 126.
8. California Privacy Protection Agency. URL: <https://coppa.ca.gov/> (дата звернення : 04.07.2023)
9. Knockout : Home. URL: <https://knockoutjs.com/> (дата звернення : 05.07.2023)

10. Magento 2 Form component documentation. URL: <https://developer.adobe.com/commerce/frontend-core/ui-components/components/form/> (дата звернення : 20.07.2023)
11. Magento 2 System requirements. URL: <https://experienceleague.adobe.com/docs/commerce-operations/installation-guide/system-requirements.html> (дата звернення : 20.07.2023)
12. Tembhonekar C.M. Role and challenges of e-commerce in the economic development of India. International Journal of Multidisciplinary, 2021. Vol. 8. P. 70 – 77.
13. Matt Weisfeld. The Object-Oriented Thought Process. – Fourth Edition. – Addison-Wesley Professional, 2013. – 336 с.
14. Schonberger, R.J. (2001). Let's Fix It! Overcoming the Crisis in Manufacturing. New York: Free Press. pp. 70 – 71.
15. Rodgers, Peter. «Service-Oriented Development on NetKernel – Patterns, Processes & Products to Reduce System Complexity». CloudComputingExpo. SYS-CON Media. Retrieved 19 August 2015.
16. А.А. Мазаракі. Глобалізаційні виклики розвитку національних економік. Київ. Нац. торг. – екон. ун-т. 2016. 1190 с.
17. Balalaie, A. (2016-05-01). «Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture». IEEE Software 33 (3): 42 – 52.