

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення
комп'ютерних технологій
Стефурак Н.А. / _____ /
підпис
«__» _____ 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту
освітньо-кваліфікаційного рівня «молодший спеціаліст»
зі спеціальності 122 «Комп'ютерні науки»

на тему: «Інтернет магазин сільськогосподарської продукції»

Студент групи КН-41 Демчак.Д.А
(підпис)

Керівник проєкту Чубей О.О.
(підпис)

Консультанти:

з техніко-економічного Меленчук Л.І.
обґрунтування (підпис)

нормоконтролер Гавришків Н.Г.
(підпис)

Тернопіль – 2023

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення
комп'ютерних технологій

Стефурак Н.А. / _____ /
підпис

«__» _____ 2023 р.

ЗАВДАННЯ

на дипломне проектування

на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»

студенту _____
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту _____

затверджена наказом по коледжу від "16" грудня _____ 2022 р., №1193-н

2. Термін здачі студентом завершеного проєкту "___" _____ 2023 р.

3. Вихідні дані до проєкту _____

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина _____

б) техніко-економічне обґрунтування _____

5. Перелік графічного матеріалу _____

6. Консультанти проекту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко- економічного обґрунтування	_____ (вчена ступень, звання П.І.Б. консультанта)		

КАЛЕНДАРНИЙ ПЛАН дипломного проектування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проектування.	28.11.2022 р.	12.12.2022 р.
2.	Огляд типових рішень та написання відповідного розділу ПЗ	13.12.2022 р.	25.12.2022 р.
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	23.01.2023 р.	13.02.2023 р.
4.	Розробка функціональних вимог до проекту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ	14.02.2023 р.	24.02.2023 р.
5.	Встановлення на налаштування середовища реалізації	1.03.2023 р.	23.03.2023 р.
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту)	27.03.2023 р.	10.04.2023 р.
7.	Реалізація та налаштування програмного засобу	11.04.2023 р.	05.05.2023 р.
8.	Доопрацювання модулів	08.05.2023 р.	29.05.2023 р.
9.	Тестування на налагодження продукту	30.05.2023 р.	05.06.2023 р.
10.	Опрацювання економічного розділу	06.06.2023 р.	12.06.2023 р.
11.	Робота над оформленням пояснювальної записки	12.06.2023 р.	17.06.2023 р.
12.	Попередній захист дипломного проекту, доопрацювання	15.06.2023 р.	
13.	Підготовка до захисту дипломного проекту	16.06.2023 р.	25.06.2023 р.
14.	Захист дипломного проекту	26.06.2023 р.	27.06.2023 р.

7. Дата видачі завдання “___” _____ 2022 р.

Керівник _____ /

Завдання прийняв до виконання _____ /

Реферат

Дипломний проєкт. Тема: «Інтернет магазин сільськогосподарської продукції» містить сторінок – 63, рисунків – 41, таблиць – 4, джерел – 4, блок-схеми – 2.

Метою проєкту забезпечити повнофункціональний інтернет-магазин, де клієнти можуть купувати товари та послуги.

Об'єктом дослідження для подальшої розробки є інтернет магазин сільськогосподарської продукції.

Головним завданням є розробка функціонального інтернет магазину з робочим функціоналом.

Результатом дипломного проєктування став інтернет магазин розроблений на Magento 2 та віртуальному хості Apache2.

Під час реалізації проєкту використовувалась платформа Magento, віртуальний хост Apache2 та мова програмування PHP8.1. Інтернет магазин сільськогосподарської продукції був розроблений з метою створення зручної та ефективної платформи для продажу сільськогосподарських товарів онлайн.

MAGENTO 2, PHP8.1, АНАЛІЗУВАННЯ, СИСТЕМА, ІНТЕРНЕТ, МАГАЗИН,

СІЛЬСЬКОГОСПОДАРСЬКА ПРОДУКЦІЯ, КОНТРОЛЬ ЯКОСТІ.

Abstract

Diploma project. Topic: "Internet magazin sil's'kohospodars'koyi produktsiyi" mistyt' storinok – 63, rysonkiv – 41, tablits' – 4, dzherel – 4, blok-skhemy – 1. Metoyu proyektu zabespechity povnofunktsional'nyy internet-magazyn, de kliyenty mozhut' kupuvaty tovary ta posluhy. Ob'ektom doslidzhennya dlya podal'shoi rozrobky ye internet magazin sil's'kohospodars'koyi produktsiyi. Holovnym zavdanniam ye rozrobka funktsional'noho internet magazynu z robochym funktsionalom. Rezul'tatom diplomnoho proyektuvannya stav internet magazin rozrobleny na Magento 2 ta virtual'nomu khosti Apache2. Pid chas realizatsiyi proyektu vykorystovuvalasya platforma Magento, virtual'nyy khost Apache2 ta mova prohramuvannya PHP8.1. Internet magazin sil's'kohospodars'koyi produktsiyi buv rozrobleny z metoyu stvorennia zruchnoyi ta efektyvnoyi platformy dlya prodazhu sil's'kohospodars'kykh tovariv onlayn.

MAGENTO 2, PHP8.1, ANALIZUVANNYA, SYSTEMA, INTERNET, MAGAZYN, SIL'S'KOHOSPADRS'KA PRODUKTSIYA, KONTROL' YAKOSTI.

Зміст

Вступ	7
1 Аналіз існуючих рішень і постановка завдань	9
1.1 Загальні правила та аналіз цільових секторів	9
1.2 Аналіз існуючих рішень	10
2 Проектування програмної системи	13
2.1 Перелік вимог до функціоналу системи	13
2.2 Проектування структури системи	14
3 Реалізація та тестування програмної системи	21
3.4 Архітектура системи та розроблені програмні засоби	32
3.3 Реалізація функціоналу системи	38
3.4. Розробка спеціалізованого функціоналу	45
4 Техніко-економічне обґрунтування	57
4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт	57
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи	58
4.4 Розрахунок суми амортизаційних відрахувань	60
4.5 Обчислення накладних витрат	60
4.6 Складання кошторису витрат та визначення собівартості робіт	60
4.7 Розрахунок ціни робіт	61
4.9 Висновки	62
Висновки	63
Перелік джерел посилань	64
Додатки	65

					ДП.КН.23.503.01.000 ПЗ			
Змн	Арк	№ документа	Підпис	Дата	Розробка проекту Інтернет магазин сільсько господарської продукції	Літ.	Арк.	Аркушів
Розроб.		Демчак.Д.А						
Перевірів		Чубей О.О.						63
Реценз.		Посвятовська ОБ..				ГФКімВЧ ЦК ІКД КН-41		
Н. Контр		Гавришків Н.Г.						
Зав.від.		Стефурак Н.А.						

Вступ.

Платформи електронної комерції – це програмні додатки для покупок і продажів в Інтернеті.

Вони дозволяють продавцям керувати своїми віртуальними активами, цифровими продажами та маркетингом на одній центральній платформі. Існує кілька типів рішень для платформ електронної комерції, кожне з яких має свої переваги та вигоди.

Мета платформи електронної комерції – забезпечити повнофункціональний інтернет-магазин, де клієнти можуть купувати товари та послуги.

Існують різні типи платформ електронної комерції. Багато з них випускаються у вигляді конструкторів сайтів, що дозволяє заощадити багато часу. Деякі конструктори сайтів дозволяють вам зробити все самостійно з мінімальними зусиллями, тоді як інші мають функцію перетягування і спрощують створення сайту.

Існує три основні типи платформ електронної комерції:

- на основі підписки;
- з відкритим кодом;
- без підписки.

Бізнес-моделі електронної комерції на основі підписки є дорожчими, ніж традиційні платформи електронної комерції, але мають багато корисних функцій.

Серед переваг – можливість використання платіжних шлюзів в Інтернеті, податкова обробка, постійний дохід і масштабованість.

З іншого боку, платформи електронної комерції на основі підписки зазвичай вимагають багато технічних ноу-хау і можуть бути дорогими, якщо у вас інтернет-магазин з високим трафіком.

Варіанти платформ електронної комерції SaaS набувають все більшої популярності, але також зростають і в ціні. Компанії шукають способи зберегти переваги ведення великого сайту електронної комерції без високих витрат, і саме тут на допомогу приходять платформи з відкритим кодом. Платформи з

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

відкритим кодом є безкоштовними і простими у використанні.

Платформи з відкритим кодом безкоштовні, прості у використанні, мають багато функцій і легко налаштовуються. Недоліком є те, що вони не пропонують клієнтської підтримки, як рішення на основі підписки.

Насправді, платформи електронної комерції з відкритим кодом (наприклад, Magento) вимагають значних технічних ноу-хау. За допомогою цієї платформи можна встановити різні варіанти комерційних хмарних плагінів, але ви повинні вміти встановлювати їх самостійно.

Сьогодні практично всі продавці різних товарів, як оптові, так і роздрібні, намагаються мати власний інтернет-магазин. У зв'язку з цим, розробка інтернет-магазину, в якому всі хочуть купувати, є дуже підходящою темою для дипломного проекту!

					ДП.КН 23.503.01.000. ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ І ПОСТАНОВКА ЗАВДАНЬ

1.1 Загальні правила та аналіз цільових секторів

Розвиток електронної комерції є одним з найперспективніших напрямків економічного розвитку в Україні.

За останні роки ринок електронної комерції значно розширився, з'явилося багато нових компаній та великих інтернет-магазинів. Попит на різні види продукції також значно зріс, і власники сайтів електронної комерції прагнуть утримати та залучити клієнтів, оновлюючи свій асортимент продукції та впроваджуючи нові рішення у свої сервісні пропозиції.

Електронна комерція зростає за кількістю замовлень, а не за середнім чеком. Хоча кількість людей, які здійснюють покупки онлайн, продовжує зростати, кількість онлайн-замовлень зростає переважно за рахунок збільшення частоти покупок. Відповідно до цього, конкуренція серед власників сайтів електронної комерції та темпи виконання завдань також зросли, що свідчить про динамічний розвиток галузі.

Magento 2 – провідна платформа для електронної комерції, випущена в 2015 році і є стандартною платформою для новітніх практик розробки інтернет-магазинів як основи для майбутніх бізнес-можливостей. Magento 2 є продуктом Adobe і становить приблизно 30% від загальної кількості .

Це високоефективне, масштабне рішення з потужним функціоналом "з коробки" та великою спільнотою, а нещодавно розроблені можливості кастомізації будуть продовжувати додавати нові функції та розширювати функціональність платформи в майбутньому. Magento 2 пропонує значно покращену масштабованість. Нова версія Magento 2 пропонує новий рівень масштабованості, зменшену кількість зчитувань з диска та використання мережі, а також пропонує ряд спеціальних функцій, які забезпечують оптимізацію маркетингових та аналітичних технологій.

Magento 2 – це найновіша технологія, яка була додана до HTML 5 та

					ДП.КН 23.503.01.000. ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Magento 2 може похвалитися нещодавно доданими технологіями, такими як HTML 5 та оновленими процесами CSS.

Magento 2 може похвалитися нещодавно доданими технологіями, такими як HTML 5 та оновленими процесами CSS, а також простими у використанні налаштуваннями та візуальними стилями.

Створення цього формату є передовим рішенням у світі електронної комерції.

В рамках даного дипломного проекту було проведено процес встановлення та налаштування Magento 2 та інших програмних інструментів для забезпечення коректної та повноцінної роботи програмного додатку. Крім того, були розроблені кастомні моделі бекенду та фронтенду для розширення функціональності системи та задоволення вимог дипломної роботи. Ці моделі доповнюють існуючу функціональність Magento 2 та надають додаткові можливості та функціональність для задоволення вимог дипломного проекту. Аналіз існуючих програмних рішень дозволяє визначити найбільш підходящу платформу для реалізації завдань і вимог диплому

1.2 Аналіз існуючих рішень

Агромаркет – один з найпопулярніших інтернет-магазинів сільськогосподарської продукції. Він пропонує широкий асортимент товарів та послуг, включаючи насіння, добрива, засоби захисту рослин, техніку та обладнання. На сайті можна знайти потрібний товар за різними критеріями, такими як категорія, бренд, ціновий діапазон та інші фільтри. Крім того, AgroMarket24 дозволяє замовляти товари онлайн і використовувати різні способи оплати, такі як накладений платіж, банківський переказ тощо.

Важливим аспектом є безпека даних та операцій; AgroMarket24 захищає ваші персональні дані та використовує найсучасніші технології шифрування даних для забезпечення конфіденційності та безпеки платежів.

Крім того, AgroMarket24 пропонує своїм клієнтам широкий спектр послуг підтримки, включаючи поради щодо вибору продуктів, технічну підтримку та післяпродажне обслуговування.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Завдяки всім цим функціям та новітнім технологіям шифрування даних, AgroMarket24 гарантує високий рівень безпеки платежів для своїх клієнтів. Деякі з найпоширеніших технологій шифрування даних, що використовуються для забезпечення конфіденційності та безпеки платежів, включають: SSL/TLS - протокол шифрування, який використовується для захисту від перехоплення даних під час передачі даних між клієнтом і сервером.

3D-Secure – стандарт, який вимагає від користувача введення додаткового пароля для підтвердження транзакції.

Токенізація – метод заміни конфіденційної інформації унікальним токеном або кодом-символом.

Рис. 1.1 – Веб-сайт Agromarket24

Астарта Т веб-сайт <https://astartat.com.ua/ua/>, де публікуються статті та новини про сільське господарство, сільськогосподарські технології та інновації в аграрному секторі. Крім того, портал надає фермерам і садівникам інформацію, яка допоможе їм вибрати насіння, засоби захисту рослин, добрива та іншу сільськогосподарську продукцію.

Він також включає каталог сільськогосподарської техніки, де користувачі можуть знайти огляди та порівняння різних моделей тракторів, зернозбиральних комбайнів та іншої сільськогосподарської техніки.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім того, на сайті можна знайти <https://astartat.com.ua/ua/> можна придбати насіння, добрива, засоби захисту рослин та ряд інших сільськогосподарських товарів.

Спочатку рекомендується ознайомитися з наступними технологіями, які можуть бути корисними для розробки інтернет-магазину

Веб-фреймворки: веб-фреймворки: наприклад, Django, Ruby on Rails, Laravel, Symfony. Вони надають візуальні інструменти для розробки веб-додатків, включаючи аутентифікацію, макетування, моделі баз даних і створення користувацького інтерфейсу.

Бази даних: наприклад, MySQL, PostgreSQL, MongoDB, SQLite. Вони гарантують зберігання даних, необхідних для роботи інтернет-магазину, таких як інформація про товари, замовлення, користувачів тощо.

Фронтенд-фреймворки React, Vue.js, Angular тощо. Надає візуальні інструменти, необхідні для розробки користувацького інтерфейсу, включаючи створення компонентів, стилізацію та динамічні ефекти.

Операційні системи та веб-сервери: наприклад, Linux, Apache, Nginx. Забезпечують роботу веб-сервера для відображення сайту користувачеві.

Електронні платіжні сервіси: наприклад, Stripe, PayPal, Braintree. Вони дозволяють інтернет-магазинам приймати оплату за товари.

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

2 ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

2.1 Перелік вимог до функціоналу системи.

- Вивчення сучасних тенденцій та технологій, що використовуються в інтернет-магазинах сільськогосподарської продукції.
- Розгляд особливостей маркетингу та продажу сільськогосподарської продукції в Інтернеті.
- Дослідження ринку сільськогосподарської продукції в Інтернеті: обсяги, конкуренція, тренди.
- Вивчення існуючих інтернет-магазинів сільськогосподарської продукції: їх функціональні можливості, переваги та недоліки.
- Визначення функціональних вимог до інтерфейсу та функціоналу інтернет-магазину.
- Розробка структури бази даних для зберігання інформації про товари, замовлення та користувачів.
- Вибір технологічного стеку для реалізації інтернет-магазину (наприклад, веб-фреймворки, мови програмування, системи управління базами даних і т.д.).
- Розробка дизайну інтерфейсу користувача, з урахуванням зручності використання та візуальної привабливості.
- Розробка адаптивного дизайну для оптимального відображення на різних пристроях (комп'ютери, планшети, мобільні телефони).
- Реалізація процесу замовлення, оплати та доставки продукції.
- Виконання тестування розробленого інтернет-магазину на наявність помилок та недоліків.
- Здійснення коригувань та виправлення виявлених проблем.
- Оцінка швидкодії та продуктивності інтернет-магазину під час одночасної роботи багатьох користувачів.
- Зіставлення результатів реалізованого проекту з поставленими завданнями та вимогами.

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

– Представлення рекомендацій щодо подальшого вдосконалення інтернет-магазину.

– Обґрунтування можливих напрямків розширення та покращення функціональності.

2.2 Проектування структури системи.

Для розробки інтерфейсу веб-сайту були використані такі технології, як Magento з відкритим вихідним кодом.

Головна сторінка веб-сайту показана на (Рис. 2.1) і виглядає так:

Рис. 2.1 – Головна сторінка веб-сайту.

Головне меню сайту включає логотип, слоган, банер та рядок заголовка. Прокрутивши головне меню вниз, знайдемо розділ "Нові товари" (Рис. 2.2), де можемо побачити товари, додані до магазину.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 2.2 – Нові товари на веб-сайті

Прокрутивши головне меню вниз дійдемо до розділу "Інформація про магазин" (Рис. 2.3), де ви зможемо побачити реквізити та контактні дані магазину.

Рис. 2.3 – Інформація про магазин

					ДП.КН 23.503.01.000. ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

Внизу головного меню є можливість розмістити рекламу від спонсора магазину (Рис. 2.4).

Рис. 2.4 – Місце для майбутньої реклами.

Рис. 2.5 – Інша підтримка.

У нижній частині головного меню знаходяться посилання, за якими можна отримати підтримку, отримати доступ до свого облікового запису.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 2.6 – Кнопка "Трафік" та каталог продуктів.

При натисканні на кнопку категорії відбувається перехід до каталогу категорій, де доступна бічна панель. Ця бічна панель має багато цікавих і корисних функцій, які допоможуть вам легше і швидше знайти те, що ви хочете в магазині, а також ви можете скористатися конкуренцією у верхньому правому куті, щоб вибрати товари за популярністю, ціною і т.д. (Рис. 2.6).

Рис. 2.7 – Іконка трактора

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

При наведенні курсору миші на кнопку трактора відкриваються категорії трактора, а при натисканні на одну з категорій можна вибрати категорію каталогу (Рис. 2.7).

Рис. 2.8 – Інформація про товар.

Опинившись у вікні опису товару, ви можете прочитати інформацію про товар, додати його до кошика, порівняти з іншими товарами або додати до списку обраного (Рис. 2.8).

Рис. 2.9 – Кошик.

Перегляньте товари та натисніть на кнопку "Додати в кошик", потім натисніть на тіло кошика у спливаючому меню, щоб переглянути вибрані товари

					ДП.КН 23.503.01.000. ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

Натискання на кнопку "Додати в кошик" переносить вас до форми замовлення, де ви вводите інформацію про покупця, якщо він ще не зареєстрований на сайті (Рис. 2.9; Рис. 2.10).

Рис. 2.10 – Форма замовлення.

Рис. 2.11 – Сторінка пошуку товару.

Сторінка пошуку товарів, яка допомагає швидко знайти те, що потрібно, ввівши потрібну назву товару (рис. 2.11).

Рис. 2.12 – Сортувати за.

У підсумку даного розділу було показано, що вимоги до функціоналу системи

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

були ретельно вивчені та зазначені. Проектування структури системи було здійснено з урахуванням цих вимог з метою забезпечення ефективності та надійності роботи.

Перш за все, у проекті було описано перелік функціональних вимог, які система повинна виконувати. Це включало в себе аналіз потреб користувачів та вимог до функціоналу, які були ретельно проаналізовані та узагальнені. Зазначені вимоги були конкретизовані і перетворені на детальні технічні специфікації, що дозволило чітко визначити, як система повинна працювати.

Далі, було розглянуто проектування структури системи. Цей етап включав у себе розробку архітектури системи, визначення компонентів та їх взаємодії. Проектування структури системи було здійснено з урахуванням принципів модульності, розширюваності та зручності обслуговування. Були використані сучасні технології та практики для створення оптимальної архітектури, що забезпечує ефективну роботу системи.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Вибір інструментів розробки системи

Magento Open Source – це безкоштовне програмне забезпечення для електронної комерції, засноване на PHP і розповсюджується під ліцензією Open Software Licence 3.0. Воно розповсюджується під номером 0 і дозволяє розробникам та інтернет-магазинам створювати та керувати веб-сайтами електронної комерції. Спочатку відома як Magento Community Edition, вона була перейменована на Magento Open Source 28 червня 2017 року тодішніми власниками Permira. Зараз Magento належить компанії Adobe, яка придбала платформу у 2018 році за 1,68 мільярда доларів США. Власником компанії є корпорація Adobe Inc.



Рис. 3.1 – Відкритий код Magento

Magento Open Source – одна з двох основних версій програмного забезпечення Magento. Його аналог, Magento Commerce, є програмним забезпеченням на основі підписки, яке можна придбати як локальне рішення або як платформу як послугу (PaaS). Magento Commerce є Magento Open з відкритим вихідним кодом, Magento Commerce побудований з використанням того ж основного коду, що і Magento Open Source, модифікований для забезпечення додаткової функціональності та об'єднаний з іншими сервісами для надання повнофункціонального рішення для електронної комерції для великого бізнесу [Magento Open Source, раніше відома як Magento Community Edition, є

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

платформою електронної комерції з відкритим вихідним кодом.

Розробники можуть впроваджувати основні файли та розширювати її функціональність, додаючи нові моделі плагінів, надані іншими розробниками. З моменту свого першого публічного бета-релізу в 2007 році, Magento Open Source була розроблена, щоб забезпечити базову платформу електронної комерції. Поточна та всі попередні версії Magento Open Source 1.X та 2.X доступні на сайті Magento Commerce, Inc. Розробка гілки з відкритим вихідним кодом публічно управляється на GitHub. Magento 1.9.4 є останньою версією Magento 1.X, термін дії якої закінчується 30 червня 2020 р. Останні активно підтримувані версії Magento з відкритим кодом.

Запущений у 2008 році, Magento швидко завоював популярність у галузі, обігнавши свого головного конкурента, osCommerce, вже через рік після запуску. Незважаючи на популярність, архітектурні недоліки призвели до того, що розробники платформи почали розробляти оновлену версію програмного забезпечення у 2011 році [14].

Поточна версія Magento, широко відома як Magento 2, була запущена 17 листопада 2015 року разом з оригінальною версією, з високопродуктивними архітектурними покращеннями та підтримкою новітніх технологій. Після її придбання компанією Adobe у 2018 році, компанія оголосила, що припиняє підтримку Magento 1 і встановила крайній термін до 30 червня 2020 року. Однак підтримка старої платформи була припинена, як і планувалося, і після липня 2020 року Magento 2 є єдиною офіційною версією, яка наразі доступна на ринку [15].

Можливості відкритого коду Magento

Magento відома своєю багатотою функціональністю та високою можливістю налаштування, що робить її придатною для малих та середніх підприємств для управління та розширення своїх онлайн-операцій.

Управління каталогом

Magento має потужну систему управління каталогом, яка може підтримувати майже будь-який продукт, від стандартних продуктів до продуктів, що

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

налаштовуються. Вона підтримує все, від простих, кастомізованих проектів до більш складних ігор, ігор і навіть віртуальних проектів. Його система каталогів також може обробляти необмежену кількість продуктів, що дозволяє підтримувати бізнес з великими запасами. Глобальні маски для статей, метаназви, метаключові слова та метаописи можна створювати та автоматично генерувати зі значень-заповнювачів. Створення продуктів для конференцій також спрощено, що дозволяє створювати низку пов'язаних простих продуктів з одного продукту для конференцій.

Функціональність просування продуктів

Magento Store підтримує просування товарів за допомогою правил ціноутворення на рівні кошика і на рівні кошика. Правила ціноутворення для каналів можна використовувати, щоб пропонувати продукти зі знижкою на основі різних критеріїв. Вони працюють на рівні каталогу і не вимагають коду купона.

Заздалегідь визначені атрибути прав можна використовувати для резервування знижок або зміни ціни на товари в каталозі магазину, використовуючи майже всі атрибути магазину, такі як група клієнтів, категорія товару, колір і розмір. Цінову політику кошика можна налаштувати так, щоб застосовувати знижки до товарів, доданих до кошика, на основі набору критеріїв або шляхом ручного введення коду купона.

При застосуванні знижок ціна буде знижена, а на сторінці кошика відображатиметься серія за ціною нижче середньої. Його можна використовувати для створення користувацьких або автоматично згенерованих купонів, безкоштовної доставки, знижок на мінімальну покупку, пропозицій "купи X – отримай Y" тощо.

Magento з відкритим вихідним кодом надає негайну підтримку SEO. Ви можете встановлювати мета-назви, мета-описи та інші метрики на рівні продукту або створювати глобальні шаблони для автоматичного оновлення метрик продукту. Він також підтримує канонічні URL-адреси, гарантуючи, що дублікати URL-адрес не будуть проіндексовані пошуковими системами. За замовчуванням

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

до всіх продуктів додається розмітка схеми даних на основі schema.org. Ця інформація може бути надана пошуковим системам для додавання прем'єрного фрагмента до списку продуктів на сторінці результатів пошуку. Крім того, Magento підтримує створення карт сайту та додавання даних користувача до файлу robots.txt, що може допомогти контролювати роботу пошукових систем під час сканування сайту.

Підтримка PWA

Magento підтримує створення PWA-сайтів за допомогою проекту з відкритим вихідним кодом на GitHub під назвою Magento PWA Studio. Цей проект являє собою набір інструментів для розробників, які допомагають створювати та підтримувати PWA-магазин поверх вашого магазину Magento; PWA-сайти є безпечними, чистими, швидкими, сумісними з багатьма пристроями та працюють безперебійно.

Рис. 3.2 – Шаблон MVC при роботі з Magento

Magento 2 використовує Model-View-Controller (MVC) як програмний патерн для архітектури, який, як показано на рисунку 3.2, організовує потік процесів всієї системи на рівні кодування і робить систему гнучкою та розширюваною. Основна роль MVC полягає у встановленні інструментів. 3.2. MVC розрізняє три рівні програмної діяльності з метою високої згуртованості та вільного зв'язку моделей, що відповідають за роботу системи Magento 2.

Модель: рівень моделювання системи. Його основними сферами є бізнес-правила системи та збір постійних даних.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

View: відповідає за рівень, який відображає інформацію про користувача, наприклад, сторінки продуктів та контактні форми, які управляються файлами верстки, що збирають блоки, контейнери та компоненти користувацького інтерфейсу на сторінці для відображення у веб-браузері. Блоки підтримуються кодом PHP для створення динамічного вмісту сторінки. Зазвичай вони поєднуються з такими файлами, як .phtml. Файли-шаблони можна легко налаштувати для створення фрагментів HTML, які будуть об'єднані на сторінці.

Контролери: рівень, що визначає основні дії, запити та відповіді клієнта, які можуть змінювати стан моделі та створювати представлення даних на рівні моделі. Контролер керує організацією потоку сторінок і поданням форм. Magento слідує стандартному фреймворку MVC, який забезпечує хороше розділення проблем для покращення модульності. Magento використовує шаблон програмної архітектури, який називається Модель -View-Controller (MVC) архітектура. Цей фреймворк відповідає за організацію потоку процесів системи на рівні кодування та надає засоби для масштабування та розширення системи.

Модель MVVM – це клас, який дозволяє передавати дані та додаткову функціональність з тієї частини системи, де реалізована бізнес-логіка. Також можна надавати дані, завантажені з об'єктів, таких як продукти. Це дозволяє розробникам спростити класи таблиць і блоків і надати єдиний клас, який легко тестувати і підтримувати.

Рис. 3.3 – Шаблон MVVM при роботі з Magento

Логіка ViewModel може бути максимально простою, оскільки клас ViewModel може містити логіку для обчислень, пошуку даних, вибору об'єктів і т.д. і може бути переданий безпосередньо до View, як показано на рисунку 3.3. ViewModels зазвичай, у каталозі моделей з назвою ViewModel.

Однак розташування класів ViewModel не обмежується каталогом ViewModel. Можна створити будь-яку кількість ViewModels, щоб забезпечити необхідний єдиний набір класів з власною логікою. Magento 2 використовує архітектуру Model View Model (MVVM). Ця архітектура MVVM тісно пов'язана з Model-View-Controller (MVC), що забезпечує більш сильне розділення між рівнями моделі та представлення.

Модель: містить бізнес-логіку додатку і спирається на пов'язану з нею модель класу – ресурсу для доступу до бази даних; подібно до MVC, це логіка управління даними, яка описує основні дані, необхідні для запуску додатку. Модель відповідає на запити до представлень.

Представлення – графічний інтерфейс, який користувач бачить на екрані. Він відповідає за відображення відповідей на запити користувача. Представлення – це структура або макет, який відображає дані в певному форматі. Представлення показують, "як виглядають дані"; Представлення мають підписки на зміну властивостей і командних значень, що надаються моделлю представлення.

ViewModel: ViewModel є похідною від Model і надає лише інформацію, необхідну для подання. Вона обробляється класом ModelBlock у Magento 2. Слід зазначити, що це зазвичай є частиною ролі MVC-контролера: у MVVM контролер відповідає лише за обробку потоку користувачів. Тобто, він отримує запит, інформує систему про представлення і перенаправляє користувача на інший фреймворк, який містить модель, перетворену в Presenter, а також містить команди, за допомогою яких можна взаємодіяти з моделлю.

ViewModel фактично є додатком Presenter, що надає загальнодоступні властивості та команди.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2 Вибір інтерфейсу для системи

jQuery вважається однією з найпопулярніших бібліотек JavaScript з відкритим вихідним кодом. Вона забезпечує швидкий спосіб реалізації вибірки запитів, обробки HTML, обробки подій, анімації, спецефектів та AJAX. Це бібліотека з відкритим вихідним кодом, з досить повною документацією по API і регулярною підтримкою.

Рис. 3.4 – Компоненти бібліотеки jQuery.

Javascript – це наступна найкраща річ для додавання динамічних елементів на сайт Magento 2. Важливо зробити інтерфейс динамічним та інтерактивним за допомогою JavaScript у Magento 2, jQuery та jQuery UI для створення різних динамічних елементів, таких як випадаючі меню, логотипи, іконки, кнопки, селектори дат, спливаючі вікна тощо. За замовчуванням Magento 2 додає необхідні бібліотеки JavaScript. Це означає, що Magento 2 поєднує в собі jQuery та JS, як показано на рисунку 3.4. Крім того, Magento 2 встановлює значення базової URL-адреси у файлі RequireJS на вимогу, тому немає необхідності додавати окрему базову URL-адресу. Базова URL-адреса – це розширений префікс веб-адреси, який додає базову URL-адресу до префіксу URL-адреси моделі.

Використання вбудованої бібліотеки jquery Magento вирішує цю проблему і надає наступні переваги jquery:

- візуальний, оскільки використовує одну бібліотеку в якості функції;
- кросбраузерна сумісність без необхідності додаткової підтримки;
- відносно простий в реалізації;
- прискорює AJAX запити;

- відкритий вихідний код;
- простий і зрозумілий API.

Magento 2 використовується для веб-розробки і працює відповідно до стратегії, заснованої на моделях, завантажуючи моделі за потреби. За замовчуванням Magento 2 також включає jQuery, оскільки Magento 2 додає необхідні бібліотеки JavaScript. Magento 2 також встановлює базову URL-адресу у файлі RequireJS, тому немає необхідності додавати окрему базову URL-адресу.

Це тому, що він встановлює файл RequireJS за замовчуванням. Базова URL-адреса – це тип префікса веб-адреси для модельної URL-адреси, який додає базову URL-адресу до префікса модельної URL-адреси. Якщо ви перевірите вихідний код будь-якої веб-сторінки, ви побачите конфігурацію базової URL-адреси за замовчуванням [23].

Knockout JS – це бібліотека Javascript, яка використовує шаблон MVVM для прив'язки даних до певного елементу DOM. Magento зазвичай визначає View-Model (= .js файл) і Template (= .html файл), а файл шаблону пов'язані з View-Model, тому, коли дані View-Model змінюються, вони також оновлюються в шаблоні [24]. чи можу я використовувати бібліотеку, таку як jQuery, щоб вручну розбирати DOM? Звичайно, ні. jQuery не є обов'язковим. Всі операції з DOM також можна виконати за допомогою базового JavaScript. Однак такі бібліотеки можуть допомогти вам писати стандартизований, зрозумілий і красивий код. Це простий приклад, наприклад, сторінки, яка відображає кількість відвідувачів у трьох окремих розділах.

Zend Framework є основним вибором для розробки додатків корпоративного рівня, оскільки йому надають перевагу PHP-розробники і використовують під час розробки, оскільки він відповідає всім основним вимогам ідеального фреймворку, включаючи безпеку, продуктивність і масштабованість. Варто зазначити, що Magento 1 побудована з використанням компонентів Zend Framework і не повністю базується на додатку Zend; перша стабільна версія Zend Framework 2 була випущена в 2012 році. Таким чином, Magento 2 не є єдиним фреймворком, а

включає в себе фреймворки та компоненти з інших фреймворків, таких як Laminas та Symfony. Розробка Magento 2 була завершена в 2015 році. Початковий план полягав у тому, щоб вирішити існуючі проблеми з юзабіліті та продуктивністю фреймворку, але проект еволюціонував протягом багатьох років.

Замість того, щоб використовувати існуючий зрілий фреймворк, такий як Symfony 2, команда розробників вирішила створити та впровадити власну платформу, фреймворк Magento. Починаючи з Magento 2.3.5, Adobe почала міграцію непідтримуваних клієнтів Zend Framework на Laminas, розширення того ж фреймворку.

Оскільки фреймворк призначений для підтримки великих підприємств, основна команда розробників Magento дотримується стандартів кодування Magento і заохочує розробників, які використовують фреймворк, робити це. Розробникам також рекомендується використовувати такі інструменти, як PHP_CodeSniffer для перевірки відповідності коду при використанні фреймворку.

Стандарти кодування Magento охоплюють такі області, як:

- Використання небезпечних функцій
- Синтаксис коду PHP
- Відповідність стандарту PHP (PSR);
- Неправильна обробка винятків.
- Угоди про імена методів та змінних;
- Неочікуване завершення;
- Використання застарілих функцій PHP; і
- Багато проблем з кодом, пов'язаних з PHP та Magento, таких як

неправильно розміщені блоки коду;

Повний набір правил можна знайти у файлі ruleset.xml стандарту кодування Magento;

					ДП.КН 23.503.01.000. ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.5 – Приклад компонентів Knockout JS.

Використання Magento PHP пропонує користувачам велику гнучкість та розширюваність. Добре розуміючи структуру, стандарти кодування та PHP, розробники можуть максимізувати його потенціал в управлінні електронною комерцією [25].

Для яких використовуються компоненти користувацького інтерфейсу Magento 2.

Відображення окремих елементів інтерфейсу користувача, таких як таблиці, кнопки, вкладки та діалогові вікна. Компоненти користувацького інтерфейсу складаються з XML-декларацій, що визначають їх склад та параметри внутрішньої структури, класів JavaScript та шаблонів посилань, зв'язаних за допомогою Knockout (рис. 3.5). Компонент інтерфейсу користувача реалізує часто використовувані елементи дизайну MVVM. Моделі.

Рисунок 3.6 – Архітектурний зв'язок MySQL та Magento 2.

Доступний безкоштовно, MySQL – це інструмент з відкритим вихідним кодом, заснований на системі управління реляційними базами даних (СКБД).

					ДП.КН 23.503.01.000. ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

MySQL базується на реляційній системі управління базами даних (СКБД), яка використовує `mySQL.SQL` – це мова для маніпулювання та зберігання даних у базах даних. Це популярний інструмент, який широко використовується в різноманітних додатках для зберігання та пошуку даних

Підготовка до перенесення даних Magento до MySQL

Вам потрібно встановити сервер MySQL і створити базу даних з відповідним ім'ям. Ви можете налаштувати це, дотримуючись інструкцій тут. Для блогу необхідно встановити та налаштувати Python у вашій системі, щоб вручну перенести дані з Magento до MySQL за допомогою Python, як показано на рисунку 3.5 для отримання даних користувач повинен мати обліковий запис Magento, підключений до одного з сервісів; для отримання додаткової інформації про Magento Як отримати дані з Magento та перенести їх до MySQL

Ми використовуємо Composer для управління залежностями та хмарними оновленнями для Adobe Commerce, а також для надання контексту про включені пакети, їх функціональність та їх комбінації. Ми виявили, що це дуже рекомендує досвід роботи з Composer, який керує бібліотеками та залежностями, необхідними для вашого проекту, і встановлює їх у вашу поштову скриньку.

Для Adobe Commerce Cloud ви можете використовувати файл `composer.json` у корені проекту, а файли `composer.lock` використовуються для управління моделями, пакетами, списками залежностей тощо, а також для виявлення оновлень, патчів, виправлень тощо. Розробники розширень використовують файл `composer.json` для керування інсталяціями та оновленнями продукту. Файл `composer.lock` містить версію залежності, яка відповідає всім обмеженням версій для кожної вимоги кожного пакета у дереві залежностей проекту. Він містить точний набір залежностей.

Робочий процес виглядає наступним чином:

- Внесіть зміни до `composer.json`. Наприклад, ви можете відредагувати цей файл під час встановлення розширення або моделі.
- Ініціюйте оновлення композитора.

- Додайте або оновіть composer.lock у хмарному Git-репозиторії.
- Внесіть зміни до хмарного середовища, які дозволять Cloud збирати та розгортати середовище.

– На етапі збірки хмарне середовище запустить install composer на свіжому клоні вашої гілки Git'a, щоб отримати решту залежностей.

Програмне забезпечення, розроблене у вигляді моделі програмного забезпечення для електронної комерції на платформі Magento 2, призначене для розширення функціональності Magento 2 та надання унікальних можливостей для інтернет-магазинів сільськогосподарської техніки. Ця модель дозволяє користувачам записатися на огляд обладнання, яке вони бажають придбати, на сайті, тим самим забезпечуючи наочність та персональний підхід до послуги.

3.4 Архітектура системи та розроблені програмні засоби

Архітектура системи, у вигляді інтегрованої програмної моделі для платформи Magento, складається з таких основних компонентів

- Серверна частина – платформа електронної комерції Magento 2 та попередньо встановлений програмний модуль розробки "Lixer_LiveReview". – Lixer_LiveReview.

- Інтернет-магазин на платформі Magento 2 з встановленою темою Pixtron.
- База даних інтернет-магазину на Magento 2.
- Сервіс створення електронних записів календаря.

Архітектура Magento 2 базується на патерні Model-View-ViewModel (MVVM). Архітектура MVVM тісно пов'язана з Model-View-Controller (MVC), що забезпечує краще розділення шарів моделі та представлення. Цей патерн описано в розділі 2.4.

Детальна діаграма MVVM показана на рисунку 4.1, а детальна діаграма дворівневої архітектури Magento показана на рисунку 4.1 [1].

Рис. 3.7 – Детальна діаграма дво-рівневої архітектури Magento.

Розглянемо кожен елемент (шар) системи окремо:

– рівень презентації є верхнім рівнем архітектури Magento 2; при взаємодії з веб-інтерфейсом Magento ми контактуємо з кодом рівня презентації. Цей рівень містить всі контролери та елементи подання, такі як макети, шаблони, блоки та `css.js`. Рівень презентації зазвичай викликає рівень обслуговування за допомогою контрактів на обслуговування. Це може перевизначати бізнес-логіку.

– користувачами презентаційного рівня є веб-користувачі та системні адміністратори.

– сервісний рівень є середнім рівнем між рівнем представлення та доменним рівнем. Він забезпечує зв'язок між рівнем представлення і доменним рівнем та даними, пов'язаними з ресурсами. Сервісний рівень реалізує сервісні контракти, визначені за допомогою інтерфейсу PHP. Ці контракти дозволяють додавати або змінювати модель бізнес-логіки ресурсу без порушення роботи системи. Це робиться за допомогою конфігураційного файлу ін'єкції залежностей (`di.xml`).

Сервісний рівень забезпечує доступ до API (REST/SOAP або інших моделей). Він розташований над доменним рівнем і нижче рівня представлення.

Доменний рівень – доменний рівень відповідає за бізнес-логіку. Він не

містить інформації, пов'язаної з конкретними базами даних або ресурсами. Доменний рівень визначає загальні об'єкти даних або моделі Magento, які містять бізнес-логіку. Ця логіка описує, які операції можна виконувати над певними типами даних, наприклад, над об'єктами клієнтів.

Для отримання даних з бази даних за допомогою викликів MySQL, кожна модель доменного рівня має посилання на модель ресурсів.

Персистентний рівень – описує моделі ресурсів, які відповідають за отримання та модифікацію даних в базі даних за допомогою CRUD-запитів (Create, Read, Update та DELETE). Для виконання цих запитів ресурсна модель містить SQL-код. Також реалізовані додаткові функції бізнес-логіки.

Magento базується на архітектурі моделювання, яка дозволяє розширювати та перевизначати функціональність платформи шляхом додавання та модифікації моделей.

Основними модулями Magento є

- Модуль каталогу: відповідає за управління та відображення товарів у магазині. Ця модель надає можливість додавати, редагувати та видаляти продукти, категорії, варіанти продуктів та інші атрибути.

- Клієнтська модель: відповідає за управління обліковими записами клієнтів, авторизацію, реєстрацію та персональні дані клієнтів.

- Модуль каси (обробка замовлень): забезпечує такі функції, як обробка замовлень, розрахунок вартості, вибір способу доставки та оплата.

- Модель продажів: відповідає за управління замовленнями, управління історією замовлень, обробку платежів і повернення.

- Модель системи управління контентом (CMS): надає функції для управління контентом веб-сайту, таким як сторінки, статті та банери.

Структура модуля.

Рис. 3.8 – Структура модуля cms

Розглянемо окремо деякі основні частини модуля:

- registration.php

Файл registration.php є один з двох основних файлів, які повинні буде створеними, для реєстрації модуля. Нижче наведено лістинг файлу для модуля module-cms.

лістинг 3.1. register.php для модуля module-cms

```
<?php
use Magento\Framework\Component\ComponentRegistrar;
ComponentRegistrar::register(ComponentRegistrar::MODULE, 'Magento_
Cms', __DIR__);
```

Тут для реєстрації модуля в системі Magento необхідно використати функцію класу Magento\Framework\Component\ComponentRegistrar - register(). У функцію необхідно передати тип компонента, що реєструється, його ім'я, та абсолютний шлях до нього.

Файл розміщується в корені модуля.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

- module.xml

Файл module.xml є другим файлом, обов'язковим для реєстрації модуля. Нижче наведено лістинг файлу для модуля module-cms.

лістинг 3.2. module. xml для модуля module-cms

```
<?xml version="1.0"?>
<config      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/modu
le.xsd">
    <module name="Magento_Cms" >
        <sequence>
            <module name="Magento_Store"/>
            <module name="Magento_Theme"/>
            <module name="Magento_Variable"/>
        </sequence>
    </module>
</config>
```

Тут xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd" вказано шлях до xsd файлу, в якому описано простір імен для даного .xml файлу.

В ноді module name необхідно вказати назву модуля ідентичну тій, що в register.php. Ноди

```
<sequence>
    <module name=""/>
```

Є необов'язковими, вони вказують на порядок загрузки модулів, при компіляції. Тобто якщо створений модуль використовує функціонал вже існуючого модуля – то він повинен бути додай в ноду <sequence> для збереження цілості.

- Арі

В цій папці повинні бути розміщені інтерфейси майбутніх класів Моделей, чи Сервісних класів.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.9 – Структура папки Api модуля module-cms

- Model

В цій папці описуються класи описані в папці Api, а також Колекції та Ресурсні Моделі.

Моделі – це відображення об’єкта, вони відповідають за бізнес логіку. Ресурсні моделі – це класи відповідальні за зв’язок об’єкта з базою даних. Колекції – це клас який містить набір об’єктів Моделей, він відповідальний за зручне маніпулювання багатьма об’єктами одночасно. До прикладу Колекція реалізовує такі функції як: getAllItems(), getCount(), setSize() та інші.

Рис. 3.10 – Структура папки Model модуля module-cms

					ДП.КН 23.503.01.000. ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

3.3 Реалізація функціоналу системи

Основним інструментом керування для користувача є панель адміністратора інтернет – магазину, де він має можливість керувати магазином: створювати та редагувати продукти, створювати, та видаляти користувачів, створювати, та видаляти замовлення, налаштувати всі наявні опції програмних модулів. Для входу адміністратор повинен ввести свої дані.

Рис. 3.11 – Вхід у панель адміністратора

Якщо адміністратор використає неправильні дані тричі – він повинен буде ввести каптчу перш ніж увійти. Після входу адміністратор потрапляє на панель управління, тут він може бачити меню на лівій частині екрану, взаємодіючи з яким, він може почати працювати. Кожен елемент меню зазвичай представляє окремий модуль. Якщо адміністратор немає прав для керування певним функціоналом – він не буде бачити ці елементи меню.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.12 – Елементи управління адміністратора.

На головній сторінці адміністратора буде панель управління де можна відстежувати:

- довічні продажі;
- середнє замовлення;
- прибуток;
- податок;
- достаку;
- кількість;
- лідери продажів;
- найбільше переглядаємі товари;
- нових покупців;
- покупців;

					ДП.КН 23.503.01.000. ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

- останнє замовлення;
- останні пошукові фрази;
- часті пошукові фрази.

Рис. 3.13 – Панель управління.

Якщо адміністратор наведеться курсором миші на вікно продажі він зможе побачити підпункти даного меню а саме:

- замовлення;
- рахунки;
- білінгові угоди;
- платіжні операції.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.14 – Підпункти вікна продажі.

Якщо адміністратор перейде на підпункт замовлення, він зможе переглянути замовлення які очікують своєї обробки

У такому списку зазвичай можуть бути замовлення, які були оформлені клієнтами, але ще не були оброблені або не перевезені. Це може стосуватися різних типів замовлень, наприклад, покупок товарів, доставки послуг або будь-яких інших замовлень, що потребують обробки або виконання.

Рис. 3.15 – Підпункти вікна продажі – замовлення.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо адміністратор перейде у вікно розрахунків, він зможе побачити різноманітну інформацію та дані, пов'язані з фінансовими розрахунками і платежами. Точний обсяг і тип інформації, яку адміністратор може побачити, залежить від конкретної системи або програми, яку вони використовують.

Рис. 3.16 – Підпункти вікна продажі – рахунки.

При наведенні курсором миші на пункт каталог адміністратор побачить підпункти: Товари; Категорії

Рис. 3.17 – Підпункти вікна каталог.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо адміністратор перейде у вікно каталогу, він зможе побачити інформацію про доступні товари, які можуть бути пропоновані в системі.

Новий товар

Набір атрибутів: Default

Назва продукту *

Артикул *

Ціна *

Клас податку: Taxable Goods

Кількість [global]

Стан складу [global]: В наявності

Вага: kgs

Категорії: Виберіть...

Видимість: Каталог, Пошук

Рис. 3.18 – Підпункт вікна каталог – товари.

Натиснувши кнопку додати товар адміністратора перекине на вікно де він зможе додати товар на сайт з переліком функції а саме:

- набір атрибутів;
- назва продукту;
- артикул;
- ціна;
- клас податку;
- кількість;
- стан складу;
- вага;
- категорії;
- видимість.

Перейшовши на вікно категорії адміністратор зможе додавати категорії для товари та створювати підпункти для категорії щоб зручніше було здійснювати пошук товару.

Рис. 3.19 – Підпункт вікна каталог – категорії.

Перейшовши у вікно покупці адміністратор зможе переглянути інформації про різних покупців а саме:

- назву покупця;
- електрону пошту;
- телефон;
- zip;
- країну;
- штат/область;
- дату реєстрації.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.20 – Вікно категорії покупці.

3.4. Розробка спеціалізованого функціоналу

У процесі розробки спеціалізованого функціоналу для магазину сільськогосподарської продукції була створена функція електронного запису на відеодзвінок з менеджером. Цей функціонал є важливим додатком до магазину, оскільки він надає покупцям можливість забезпечити собі індивідуальний огляд техніки, яку вони планували хочуть придбати.

Основна причина розробки цього функціоналу полягала в тому, що покупці часто мають потребу у перевірці стану техніки перед покупкою, особливо у випадку дороговартісних товарів, таких як сільськогосподарська техніка. Завдяки системі електронного запису на відеодзвінок з менеджером, покупці могли вирішити цю проблему, забезпечивши собі віддалений огляд та консультацію.

Такий функціонал дозволяє покупцям самостійно записуватись на відеодзвінок з менеджером прямо на сторінці продукта. Вони могли вибрати зручну для них дату та час з варіантів, доступних менеджеру.

Після підтвердження запису, в системі автоматично створюється запис з обраним часом та датою, а також зберігається email та телефон користувача для

					ДП.КН 23.503.01.000. ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

подальшого зв'язку. Блок схема алгоритму наведена нижче:

Рис. 3.21 – Блок схема алгоритму створення Онлайн Запису

					ДП.КН 23.503.01.000. ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.22 – Діаграма прецендетів

Створено файли registration.php та module.xml

Лістинг 3.23. Лістинг module.xml

```
<?php
use Magento\Framework\Component\ComponentRegistrar;
ComponentRegistrar::register(ComponentRegistrar::MODULE,
'Lixer_LiveReview', __DIR__);
Лістинг 3.4.1 Лістинг registration.php
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module
le.xsd">
<module name="Lixer_LiveReview">
```

					ДП.КН 23.503.01.000. ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        <sequence>
        <module name="Magento_Catalog"/>
        </sequence>
    </module>
</config>

```

3.23. Розробка структури спеціальної таблиці БД та її створення.

Розроблено схему нової таблиці, що має зберігати інформацію про онлайн запис.

Рис. 3.23 – Структура таблиці live_review_request

Таблиця має назву "lixer_live_review_request" і зберігає дані про запити на онлайн записи про продукти. Основні поля таблиці включають:

live_review_id (тип: smallint): Унікальний ідентифікатор запиту на онлайн запис (ID сутності).

products_ids (тип: varchar): Список ідентифікаторів продуктів, пов'язаних з запитом.

status (тип: varchar): Статус запиту на онлайн запис.

customer_name (тип: varchar): Ім'я клієнта, який подав запит.

customer_email (тип: varchar): Електронна пошта клієнта.

customer_phone (тип: varchar): Номер телефону клієнта.

scheduled_at (тип: varchar): Дата та час запланованої онлайн запису.
 created_at (тип: varchar): Дата та час створення запиту на онлайн запис.
 updated_at (тип: varchar): Дата та час оновлення запиту на онлайн запис.
 products_names (тип: varchar): Назви простих або конфігурованих продуктів, пов'язаних з запитом.

customer_id (тип: int): Ідентифікатор клієнта, пов'язаний з запитом.

Таблиця також має декілька обмежень (constraints):

PRIMARY: Основний ключ таблиці, який базується на полі live_review_id.

LIVE_REVIEW_REQUEST_CUSTOMER_ID_CUSTOMER_ENTITY_ENTITY_ID: Зовнішній ключ, який пов'язує поле customer_id таблиці "lixer_live_review_request" з полем entity_id таблиці "customer_entity". У разі видалення запису в таблиці "customer_entity", значення customer_id у таблиці "lixer_live_review_request" буде встановлено в NULL.

Користуючись вбудованими можливостями Magento 2 створено файл db_schema.xml який дозволяє оголосити таблицю в файлі .xml після чого вона буде створена в БД

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

Розробка інтерфейсів основних класів.

Після цього описано інтерфейси основних класів, що реалізують бізнес логіку:

CalendlyApiInterface.php – описує функцію event() яка відповідальна за оновлення даних про існуючий запис. Ця функція спрацює, коли Calendly буде відправляти запити на сторону Magento.

LiveReviewRepositoryInterface.php – Відповідальний за CRUD (Create, Read, Update, Delete) операції над об'єктом Онлайн Запису.

LiveReviewInterface.php – Представлення об'єкту Онлайн Запису.

Лістинг 3.4.4 LiveReviewInterface.php

```
<?php
declare(strict_types=1);
namespace Lixer\LiveReview\Api;
```

```

interface CalendlyApiInterface
{
    final const EVENT_CREATED      = 'invitee.created';
    final const EVENT_CANCELED    = 'invitee.canceled';
    /**
     * @return void
     */
    public function event() : void;
}

```

Лістинг 3.4.3 CalendlyApiInterface.php

```

<?php
declare(strict_types=1);
namespace Lixer\LiveReview\Api;
use Lixer\LiveReview\Api\Data\LiveReviewInterface;
public function setCreatedAt(string $createdAt) : LiveReview;
    /**
     * @return string|null
     */
    public function getCreatedAt() : ?string;
    /**
     * @param string $updatedAt
     * @return LiveReview
     */
    public function setUpdatedAt(string $updatedAt) : LiveReview;
    /**
     * @return string|null
     */
    public function getUpdatedAt() : ?string;
    /**
     * @param string[] $productNames
     * @return LiveReview
     */
    public function setProductNames(array $productNames) :
LiveReview;
    /**

```

					ДП.КН 23.503.01.000. ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        * @return string[]|null
    */
    public function getProductNames() : ?array;
}

```

Оголошення маршруту для прийому запитів Calendly

Користуючись вбудованими можливостями Magento 2. Було створено файл `webapi.xml` та оголошено

Лістинг 3.4.5 `webapi.xml`

```

<?xml version="1.0"?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Webapi:etc
/webapi.xsd">
    <route url="/V1/calendly/webhook" method="POST">

<service class="Lixer\LiveReview\Api\CalendlyApiInterface"
method="event"/>
        <resources>
            <resource ref="anonymous"/>
        </resources>
    </route>
</routes>

```

У Magento 2, маршрути (routes) використовуються для визначення, які дії повинні відбутися при отриманні запиту на певний URL. У даному випадку, маршрут має URL `"/V1/calendly/webhook"` і метод запиту `"POST"`.

У цьому маршруті також визначено сервіс (service) та метод (method), які пов'язані з обробкою запиту. Клас `"Lixer\LiveReview\Api\CalendlyApiInterface"` вказує на сервіс, який буде використовуватися для обробки запиту, а `"event"` – це метод цього сервісу, який буде викликаний при отриманні запиту.

Також вказано ресурс (resource) з посиланням `"anonymous"`. Це вказує на те, що доступ до цього маршруту є анонімним, тобто не потрібна авторизація або аутентифікація для доступу до нього.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

У загальному, цей фрагмент коду визначає маршрут у Magento 2, який прив'язаний до URL "/V1/calendly/webhook" і методу "POST". При отриманні запиту на цей маршрут, викликається метод "event" сервісу "Lixer\LiveReview\Api\CalendlyApiInterface" для обробки запиту.

Реалізація основних класів

Реалізовано інтерфейси описані в п. 3.4.3. Нижче наведено листінг.

```
<?php
declare(strict_types=1);
namespace Lixer\LiveReview\Model;
use Lixer\LiveReview\Api\CalendlyApiInterface;
use Lixer\LiveReview\Api\Data\LiveReviewInterface;
use Lixer\LiveReview\Api\LiveReviewRepositoryInterface;
use Lixer\LiveReview\Model\ResourceModel\LiveReview\Collection;
use Lixer\LiveReview\Logger\Logger;
use Lixer\LiveReview\Helper\Data;
use Magento\Framework\Stdlib\DateTime;
use Magento\Framework\Stdlib\DateTime\TimezoneInterface;
use Magento\Framework\Webapi\Rest\Request;
use Magento\Framework\HTTP\Client\Curl;
use Magento\Framework\Serialize\Serializer\Json;
use Exception;
class CalendlyApi implements CalendlyApiInterface
    /**
     * @param Request $request
     * @param Collection $liveReviewCollection
     * @param LiveReviewRepositoryInterface $liveReviewRepository
     * @param TimezoneInterface $timezone
     * @param Curl $curl
     * @param Json $json
     * @param Data $helper
     * @param Logger $logger
     */
    public function __construct(
        Request $request,
```

```

Collection $liveReviewCollection,
LiveReviewRepositoryInterface $liveReviewRepository,
TimezoneInterface $timezone,
Curl $curl,
Json $json.

```

Інтеграція “Calendly” у систему.

В системі Magento створено меню, куди необхідно додати ідентифікатор користувача Calendly та Personal Access Token. Ці дані використовуються для вбудовування форми Calendly в сайт магазину.

Рис. 3.24 – Конфігурація доступів

Лістинг 3.4.8 Вбудовування форми Calendly в сторінку продукту

```

<?php if ($viewModel->getCustomer()) :
    $customer = $viewModel->getCustomer(); ?>
    <link
href="https://assets.calendly.com/assets/external/widget.css"
rel="stylesheet">
    <script                                type="text/javascript"
src="https://assets.calendly.com/assets/external/widget.js"></script
>
    <button type="button" class="action secondary live-
review__modal-trigger" data-bind="click: openModal">
        <?= $escaper->escapeHtml(__( 'Записатись на огляд
товару')) ?>
    </button>
<?php endif; ?>

```

Тестування розробленого функціоналу

Якщо користувач авторизований – йому буде доступна кнопка “Записатись на огляд товару”(рис 3.25.)

Рис. 3.25 – Вікно запису на огляд.

Натиснувши на неї відкривається модальне вікно, де він за бажанням може обрати додатковий товар, та продовжити, натиснувши кнопку “Вибрати час та дату” (рис. 3.26.)

Рис. 3.26 – Вікно запису на огляд.

Натиснувши кнопку “Вибрати час та дату” користувач бачить форму з сервісу Calendly, де може обрати вільні дати. (Рис. 3.27)

					ДП.КН 23.503.01.000. ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.27 – Запис на огляд товару

Після чого в панелі адміністратора з'явиться інформація про запис.

Рис. 3.28 – Інформація про онлайн запис в панелі адміністратора.

Висновки розробленого функціоналу

Функція онлайн запису на відеодзвінок була піддана успішному тестуванню. Під час тестування було перевірено різні сценарії взаємодії з функцією, включаючи вибір дати та часу, заповнення електронної пошти та підтвердження запису.

В процесі тестування було перевірено, чи працює вбудована форма з сервісу Calendly на сторінці продукта, як покупці здійснюють вибір дати та часу, як коректно зберігаються дані у базі даних Magento 2 та як відбувається взаємодія

					ДП.КН 23.503.01.000. ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

з API Calendly для створення запису.

В результаті тестування було підтверджено, що функція онлайн запису на відеодзвінок працює належним чином і відповідає очікуванням. Покупці можуть успішно записуватись на відеодзвінок з менеджером, обираючи зручну для них дату та час, а дані про запис коректно зберігаються та відображаються у системі.

Тестування функції онлайн запису дозволило переконатись у її надійності та функціональності, що сприяє впевненості в успішному впровадженні цього функціоналу в живу систему магазину.

В цьому розділі детально описана архітектура розроблюваного веб-додатка на основі фреймворку Magento. Вивчено всі моделі програмного забезпечення, які зараз використовуються в Magento, а саме MVC і MVVM. Вирішено, що для проекту буде використовуватися модель MVVM разом з бібліотекою knockout, замість простого js. Детально показано як будуть працювати і взаємодіяти фронтенд і бек-енд частини проекту. Ось чому ми будемо використовувати модель MVVM і використовувати бібліотеку knockout замість простого js. Друга частина детально показує архітектуру Magento, як працюють і взаємодіють фронтенд і бек-енд частини.

Спочатку детально описано роботу моделей розробки MVVM і MVC. Описано їх основні переваги та недоліки, основні принципи роботи та основні бібліотеки, на яких базуються ці моделі. Описано дві обрані моделі розробки, які гарантують виконання технічних вимог проекту.

Потім було розглянуто питання вибору мови програмування для написання проекту. Найбільш підходящими варіантами є мови PHP та JS, які детально розглянуто в цьому розділі. Зокрема, в цьому розділі розглядаються загальні відомості про такі мови програмування, їх численні переваги та недоліки. На підставі цього було зроблено висновок, що мова повністю відповідає вимогам нашого проекту, і було прийнято рішення використовувати її, оскільки в цьому випадку розроблена система отримає масу переваг.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Темою даного дипломного проекту є “Інтернет магазин сільсько господарської продукції”. Мета економічної частини в дипломному проекті – це здійснення економічних розрахунків, спрямованих на визначення економічної ефективності розробки сайту і прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної роботи.

- Об’єктом розробки є сайт.
- Розрахунок вартості розробки виконується в декілька етапів.
- Опис технологічного процесу розробки із зазначенням трудомісткості кожної операції:

- визначення суми витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- визначення суми матеріальних затрат;
- обчислення витрат на електроенергію для науково-виробничих цілей;
- розрахунок транспортних витрат;
- нарахування суми амортизаційних відрахувань;
- визначення суми накладних витрат;
- складення кошторису та визначення собівартості робіт;
- розрахування ціни робіт;
- визначення економічної ефективності та терміну окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт

Базуючись на даних сформованого технічного завдання виділено 9 стадій розробки сайту.

Дані про витрати часу на проведення робіт наведені в таблиці 4.1.

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

Таблиця 4.1 – Середній час виконання робіт по обслуговуванню та стадії
(операції) технологічного процесу.

№	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Постановка задачі та формування технічного завдання	Засновник сайту	11
2.	Розробка дизайну сайту	Програміст	16
3.	Затвердження дизайну	Засновник сайту	3
4.	Розробка frontend та backend	Програміст	21
5.	Тестування сайту	Менеджер	4
6.	Документування	Менеджер	4
Разом		-	59

Сумарний час виконання операцій технічного процесу становить 59 годин.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи
Заробітна плата, відповідно до Закону України “Про оплату праці”, – це “винагорода, обчислена, як правило, у грошовому виразі.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично

Зм.	Арк.	№ докум.	Підпис	Дата	
-----	------	----------	--------	------	--

відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна заробітна плата розраховується за формулою:

$$Z_{осн.} = T_c \cdot K_z,$$

де: T_c – тарифна ставка, грн.;

K_z – кількість відпрацьованих годин.

Отже основна заробітна плата для:

- Засновника сайту $Z_{осн1} = 100 \cdot 14 = 1400$ грн.
- програміст $Z_{осн2} = 200 \cdot 37 = 7400$ грн.
- лаборанта $Z_{осн3} = 70 \cdot 8 = 1120$ грн.

Сумарна основна заробітна плата становить:

$$Z_{осн} = 1400 + 7400 + 1120 = 18340 \text{ грн.}$$

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробіт на плата, грн.	Нарахув. на ФОП, грн.	Всього витрат на оплату праці, грн.
		Тариф на ставка, грн.	К-сть відпрацьов. год.	Фактична нарах. з/пл., грн.			
1	Кер. проекту	100	14	1400	160	-	-
2	Програміст	200	37	7400	900	-	-
3	Лаборант	70	8	1120	70	-	-
Разом				18340	1130	1567	21037

Отже, загальні витрати на оплату праці становлять 21037 грн.

4.3 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S ,$$

де: W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Час роботи ПК над даним проектом становить 59 годин, споживана потужність - 0.5 кВт/год.

$$Z_e = 0.5 \cdot 59 \cdot 1,68 = 49,56 \text{ грн.}$$

Загальні витрати на електроенергію становлять $Z_e = 49,56$ грн.

4.4 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20 – 60 % від суми основної та додаткової заробітної плати працівників.

$$H_v = B_{o.l.} \cdot 0,2 \dots 0,6 ,$$

де H_v – накладні витрати.

$$H_v = 11200 \cdot 0,3 = 6311,1 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості робіт

Результати проведених розрахунків витрат зведемо у таблицю нижче.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 – Кошторис витрат

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	21037	87,11
Відрахування на соціальні заходи	1567	9,21
Витрати на електроенергію	49,56	0,28
Накладні витрати	6311	38,65
Собівартість	28964	171,3

Собівартість (C_B) НДР розрахуємо за формулою:

$$C_B = B_{o.n.} + B_{c.z.} + Z_{m.v.} + Z_e + T_v + A + H_v$$

Отже, собівартість дорівнює $C_B = 28964$ грн.

4.7 Розрахунок ціни робіт

$$Ц = \frac{C_B \cdot (1 + P_{рен}) + K \cdot B_{i.н.} \cdot (1 + ПДВ)}{K},$$

де: $P_{рен.}$ – рівень рентабельності, 30 %;

K – кількість замовлень, од.;

$B_{i.н.}$ – вартість носія інформації, грн.;

ПДВ – ставка податку на додану вартість, (20 %).

$$Ц = 21037 \cdot (1 + 0,3) \cdot (1 + 0,2) = 41022,15$$

де: $P_{рен.}$ – рівень рентабельності, 30 %;

K – кількість замовлень, од.;

$B_{i.н.}$ – вартість носія інформації, грн.;

ПДВ – ставка податку на додану вартість, (20 %).

$$Ц = 21037 \cdot (1 + 0,3) \cdot (1 + 0,2) = 41022,15$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на

підприємстві за певний проміжок часу.

де K_B – затрати на проект;

Γ_{π} – грошовий потік за t – ий рік;

t – відповідний рік проекту;

i - величина дисконтної ставки (10...15%).

Якщо ЧТВ ≥ 0 , то проект може бути рекомендований до впровадження.

$$\text{ЧТВ} = -28964,13 + \frac{9300}{(1 + 0,1)} + \frac{9300}{(1 + 0,1)^2} + \frac{9300}{(1 + 0,1)^3} = 7554,67 \text{ грн}$$

Термін окупності визначається за формулою:

$$T_{\text{ок}} = T_{\text{пв}} + \frac{H_B}{\Gamma_{\text{пв}}},$$

де ТПВ – період до повного відшкодування витрат, років;

НВ – невідшкодовані витрати на початок року, грн.;

ГПВ – грошовий потік на початок року, грн.

4.9 Висновки

Таблиця 4.5 – Техніко-економічні показники розробки програми

№ п/п	Показник	Значення
1.	Собівартість, грн.	28964
2.	Плановий прибуток, грн	9300
3.	Ціна, грн.	41022,15
4.	Чиста вартість, грн	7554,67
5.	Термін окупності, рік	0,8

В економічному розділі описано фінансову доцільність проведення науково-дослідних робіт. В результаті проведених розрахунків можна зробити висновок: оскільки собівартість даного проекту становитиме 28964 грн. при очікуваному плановому прибутку 9300 грн. то термін окупності даного сайту складатиме менше ніж рік.

ВИСНОВКИ

У даній дипломному проєкті було досліджено розробку інтернет-магазину сільськогосподарської продукції на основі Magento Open Source. Отримані результати демонструють ефективність використання цієї платформи для розробки електронних торгових платформ, спрямованих на продаж сільськогосподарської продукції.

Процес розробки включав аналіз вимог та потреб клієнтів, проектування архітектури системи, налаштування та інтеграцію Magento, розробку унікального дизайну, створення каталогу товарів, та інші важливі етапи.

Magento Open Source надавав широкі можливості для налаштування та розширення функціональності магазину. Багатий набір модулів та розширень дозволив додавати нові функції та вдосконалювати існуючі, забезпечуючи гнучкість та масштабованість системи. Також, мною з допомогою Magento забезпечено високий рівень безпеки та надійності, що є особливо важливим у сфері електронної комерції.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Вступ в HTML [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/html/html_intro.asp – Дата доступу: 13.04.2023р.
2. Основні поняття мови HTML та структура документів [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/vivcaemowebdizajndistancijno/html/lekcja-3-osnovni-ponatta-movi-html-ta-struktura-dokumentiv> – Дата доступу: 13.04.2023р.
3. Застосування CSS-технології в сайті, основні поняття [Електронний ресурс] – Режим доступу до ресурсу: http://antibotan.com/file.html?work_id=529477 – Дата доступу: 13.04.2023р.
4. Керівництво по SASS. Як верстати сайти вдвічі швидше? [Електронний ресурс] – Режим доступу до ресурсу: <https://tokar.ua/read/6672> – Дата доступу: 13.04.2023р.
5. SASS [Електронний ресурс] – Режим доступу до ресурсу: <https://sass-lang.com/> – Дата доступу: 13.04.2023р.
6. Поняття про мови веб-скриптів способи їх використання [Електронний ресурс] – Режим доступу до ресурсу: http://comscienceatschool.blogspot.com/p/26_17.html – Дата доступу: 13.04.2023р.
7. React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/> – Дата доступу: 13.04.2023р.
8. ReactJS курс [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/reactjs/index.htm> – Дата доступу: 13.04.2023р.
9. Vue.js | вступ та інсталяція [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/vue-js-introduction-installation/> – Дата доступу: 13.04.2023р.

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТКИ

Додаток А. Текст скрипта щоб оголосити маршрут для прийому запитів Calendly.

Було створено файл webapi.xml та оголошено:

```
<?xml version="1.0"?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Webapi:etc/webapi.xsd">
    <route url="/V1/calendly/webhook" method="POST">
        <service
class="Lixer\LiveReview\Api\CalendlyApiInterface"
method="event"/>
        <resources>
            <resource ref="anonymous"/>
        </resources>
    </route>
</routes>
```

Лістинг 4.1 – webapi.xml

Також реалізовані інтерфейси Calendly лістинг наведено нижче:

```
<?php
declare(strict_types=1);
namespace Lixer\LiveReview\Model;
use Lixer\LiveReview\Api\CalendlyApiInterface;
use Lixer\LiveReview\Api\Data\LiveReviewInterface;
use Lixer\LiveReview\Api\LiveReviewRepositoryInterface;
use
Lixer\LiveReview\Model\ResourceModel\LiveReview\Collection;
use Lixer\LiveReview\Logger\Logger;
use Lixer\LiveReview\Helper\Data;
use Magento\Framework\Stdlib\DateTime;
use Magento\Framework\Stdlib\DateTime\TimezoneInterface;
use Magento\Framework\Webapi\Rest\Request;
use Magento\Framework\Http\Client\Curl;
use Magento\Framework\Serialize\Serializer\Json;
use Exception;
class CalendlyApi implements CalendlyApiInterface
{
    /**
     * @param Request $request
     * @param Collection $liveReviewCollection
     * @param LiveReviewRepositoryInterface
$liveReviewRepository
     * @param TimezoneInterface $timezone
     * @param Curl $curl
     * @param Json $json
```

					ДП.КН 23.503.01.000. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

```

        * @param Data $helper
        * @param Logger $logger
        */
    public function _construct(
        Request $request,
        Collection $liveReviewCollection,
        LiveReviewRepositoryInterface
$liveReviewRepository,
        TimezoneInterface $timezone,
        Curl $curl,
        Json $json,
        Data $helper,
        Logger $logger
    ) {
        $this->request = $request;
        $this->liveReviewCollection =
$liveReviewCollection;
        $this->liveReviewRepository =
$liveReviewRepository;
        $this->timezone = $timezone;
        $this->curl = $curl;
        $this->json = $json;
        $this->helper = $helper;
        $this->logger = $logger;
    }
    /**
     * This function gets triggered by Calendly webhook
    when even is created or canceled
     * Find existing Live Review Request and update it
    data
     *
     * @return void
     */
    public function event(): void
    {
        $data = $this->request->getRequestData();
        $liveReview = $this->getLiveReviewByEmail($data);
        if($liveReview === null) {
            $this->logger->alert(
                "Not found entity with email " .
$data['payload']['email'] . "\n" .
                "Aborting scheduling"
            );
            return;
        }
        $this->changeStatus($data, $liveReview);
        $this->setStartTime($data['payload']['event'],
$liveReview);
        $this->liveReviewRepository->save($liveReview);
    }

    /**

```

```

        * Change status of Live Review Request
        *
        * @param array $data
        * @param LiveReviewInterface $liveReview
        * @return void
        */
        private function changeStatus(array $data,
LiveReviewInterface $liveReview) : void
        {
            $status = $data['event'];

            if($status === self::EVENT_CANCELED) {
                $liveReview->setStatus(
                    LiveReviewInterface::STATUS_CANCELED
                );
            } else if($status === self::EVENT_CREATED) {
                $liveReview->setStatus(
                    LiveReviewInterface::STATUS_SCHEDULED
                );
            }
        }

        /**
        * Find existing LiveReview entity with the same
email
        * and return it
        * If found more than one - return the last one
        *
        * @param array $data
        * @return LiveReviewInterface|null
        */
        private function getLiveReviewByEmail(array $data) :
?LiveReviewInterface
        {
            $email = $data['payload']['email'];
            $this->liveReviewCollection
                ->addFieldToFilter('customer_email', $email)
                ->addFieldToFilter('status',
LiveReviewInterface::STATUS_WAITING);
            if ($this->liveReviewCollection->count() > 1 ) {
                $this->logger->notice(
                    "User with email:" . $email . "\n" .
                    "has few Live Review Requests with ids:"
. implode(', ', $this->liveReviewCollection->getAllIds()) .
"\n" .
                    "the last one with id " . $this-
>liveReviewCollection->getLastItem()->getId() . " used"
                );
            }
            if($this->liveReviewCollection->count() === 0) {
                return null;
            }
        }
    }

```

```

        return $this->liveReviewCollection->getLastItem();
    }

    /**
     * Request Calendly API for getting event start time
     and set it to LiveReview
     *
     * @param string $uuid
     * @param LiveReviewInterface $liveReview
     * @return bool
     */
    private function setStartTime(string $uuid,
LiveReviewInterface $liveReview) : bool
    {
        $this->curl->setHeaders(
            [
                'Content-Type' => 'application/json',
                'Authorization' => 'Bearer ' . $this->helper->getAccessToken()
            ]
        );

        try {
            $this->curl->get($uuid);
            $response = $this->curl->getBody();

            if($this->curl->getStatus() !== 200) {
                $this->logger->error($response . "\n URI:
" . $uuid);

                $liveReview->setScheduledAt("ERROR");
                return false;
            }

            $response = $this->json->unserialize($response);
            $startTime = $response['resource']['start_time'];
            $liveReview->setScheduledAt($startTime);
        } catch (Exception $exception) {
            $this->logger->error($exception->getMessage());
            return false;
        }
        return true;
    }
}

```

ЛІСТИНГ 4.2 – ЛІСТИНГ CalendlyApi.php

Дотаток Б. Блок схема можливостів наведено нижче:

Рис. 4.1 – Блок схема можливостей адміністратора та менеджера.

					ДП.КН 23.503.01.000. ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

РЕЦЕНЗІЯ
на дипломний проект
студента відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола

студента IV курсу групи КН-41

Демчака Дмитра Андрійовича

Спеціальність 122 „Комп'ютерні науки ”

Обсяг дипломного проекту: стор.

Тема: Інтернет магазин сільськогосподарської продукції

1. Актуальність теми: Тема ДП є досить актуальною, оскільки інтернет-магазини на теперішній час є зручним засобом для порівняння цін та здійснення покупок різноманітних товарів, не залежно від місця перебування. Відповідно, створення інтернет-магазину сільськогосподарської продукції є актуальним для розширення кола клієнтів та ринку збуту, здійснення маркетингових заходів та забезпечення простоти і зручності керування бізнес-процесами.

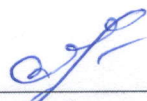
2. Практична або теоретична цінність опрацьованих питань: виконано ґрунтовне дослідження інтернет-магазинів, проаналізовано їхні функціональні можливості, переваги та недоліки, досліджено ряд інструментів для розробки подібних проєктів, спроектовано та реалізовано інтернет магазин сільськогосподарської продукції. Проєкт готовий до практичного застосування.

3. Недоліки роботи: виявлено не значні недоліки в розділі проєктування та оформлення пояснювальної записки до дипломного проєкту.

4. Загальний висновок: Дипломний проєкт відповідає всім поставленим вимогам і заслуговує на позитивну оцінку

Рецензент: Наталя КУЛЬЧИНСЬКА

« 24 » 06 2023 р.


(підпис)

ВІДГУК
дипломний проєкт
відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола
студента IV курсу групи КН-41

Демчака Дмитра
(прізвище та ініціали)

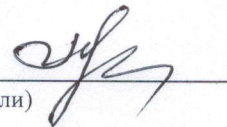
Спеціальність 122 „Комп'ютерні науки”

Керівник ДП Чубей О.О.

Тема: *Інтернет магазин сільськогосподарської продукції*

1. Загальна характеристика студента: *студент на достатньому рівні володіє засобами проектування, проявив себе старанним та відповідальним. Робота проводилась із дотриманням усіх відповідних термінів.*
2. Практична або теоретична цінність опрацьованих питань: *Робота полягає у проектуванні веб-сайту Інтернет-магазину, з метою розвитку приватного бізнесу. Тема є практичною і значимою на даному етапі розвитку комерційного підприємництва.. Проблематику проаналізовано, здійснено огляд існуючих рішень. Даний проєкт готовий до застосування за призначенням, оскільки є нескладним у використанні.*
3. Недоліки роботи: *пояснювальна записка не містить в тексті посилань на використані джерела літератури.*
4. Загальний висновок: *дипломний проєкт виконано згідно вимог і заслуговує на задовільну оцінку.*

Керівник дипломного проєкту Чубей О.О.
(прізвище та ініціали)



Ім'я користувача:
Василь Кузик

Дата перевірки:
15.06.2023 23:40:49 EEST

Дата звіту:
15.06.2023 23:43:04 EEST

ID перевірки:
1015619877

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012366

Назва документа: Демчак

Кількість сторінок: 63 Кількість слів: 8136 Кількість символів: 61736 Розмір файлу: 7.65 MB ID файлу: 1015267100

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

14.4%
Схожість

Найбільша схожість: 7.56% з Інтернет-джерелом (https://ela.kpi.ua/bitstream/123456789/49930/1/Boichuk_bakalavr.pdf)

14.4% Джерела з Інтернету

424

Сторінка 65

0.1% Джерела з Бібліотеки

6

Сторінка 67

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

4075

Підозріле форматування

34
сторінки