

Державний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка онлайн-платформи адміністрування

замовлень»

Студента 4 курсу, 12 групи,
спеціальності
126 «Інформаційні системи та
технології»

підпис студента

Козловський
Дмитро
Олександрович

Науковий керівник
доктор фізико-математичних наук,
професор

підпис керівника

Краскевич
Валерій
Євгенович

Гарант освітньої програми
кандидат технічних наук, доцент

підпис керівника

Козлов Валерій
Володимирович

Київ 2024

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 126 «Інформаційні системи та технології»

Затверджую

Зав. кафедри _____ Пурський О.І.
«18» грудня 2023р.

Завдання **на випускню кваліфікаційну роботу студентці** **Козловський Дмитро Олександрович** (прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи.

“Розробка онлайн-платформи адміністрування замовлень”

Затверджена наказом ректора від «27» листопада 2023 р. № 4150

2. Строк здачі студентом закінченої роботи 31 травня 2024 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: метою даної роботи є розробка онлайн-платформи для ефективного адміністрування замовлень

Об'єкт дослідження: процеси обробки та управління замовленнями в сучасному бізнес-середовищі, включаючи отримання, підтвердження, обробку, відправлення та відстеження замовлень.

Предмет дослідження: методи, інструменти та технології, що використовуються для розробки онлайн-платформ адміністрування замовлень, включаючи вибір технологічного стеку, проектування архітектури та інтерфейсу платформи, а також розробку та впровадження функціональності платформи.

4. Перелік графічного матеріалу:

Рис.1.1. - Shopify.....	14
Рис.1.2. - Magento.....	15
Рис.1.3. - WooCommerce.....	16
Рис.1.4. - KeyCRM.....	16
Рис.3.1. - Вигляд головної сторінки сайту.....	36
Рис.3.2. - ”Про нас”.....	36

Рис.3.3. - Сторінка "Клієнтам".....	37
Рис.3.4. - Послуги.....	37
Рис.3.5. - Форма замовлення.....	38
Рис.3.6. - Сторінка Веб Розробники.....	38

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Краскевич В.Є	22.12.2023 р.	22.12.2023 р.
2	Краскевич В.Є	22.12.2023 р.	22.12.2023 р.
3	Краскевич В.Є	22.12.2023 р.	22.12.2023 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ВИМОГ ТА ДИЗАЙН ПЛАТФОРМИ:

1.1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПЛАТФОРМ АДМІНІСТРУВАННЯ
ОНЛАЙН-ЗАМОВЛЕНЬ

1.2 ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ ДО ПЛАТФОРМИ

1.3 АРХІТЕКТУРА ПЛАТФОРМИ ТА ДИЗАЙН ІНТЕРФЕЙСУ

РОЗДІЛ 2. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ОСНОВНИХ МОДУЛІВ
ПЛАТФОРМИ

2.1. РОЗРОБКА МОДУЛЯ УПРАВЛІННЯ ЗАМОВЛЕННЯМИ

2.2. РЕАЛІЗАЦІЯ МОДУЛЯ УПРАВЛІННЯ КОРИСТУВАЧАМИ

2.3. РОЗРОБКА МОДУЛЯ ПРОЦЕСИНГУ ПЛАТЕЖІВ

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПЛАТФОРМИ

3.1. ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ТА НАДІЙНОСТІ ПЛАТФОРМИ

3.2. Вигляд сайту в браузері

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Додаток

коду

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	05.10.2023	05.10.2023
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	18.12.2023	18.12.2023
3	<i>Вступ</i>	02.02.2024	02.02.2024
4	<i>РОЗДІЛ 1. Теоретичне дослідження механізмів оцінювання рівня соціально-економічного розвитку</i>	26.02.2024	26.02.2024
5	<i>РОЗДІЛ 2. Моделі, методи, алгоритми оцінювання рівня соціально-економічного розвитку</i>	05.04.2024	05.04.2024
6	<i>РОЗДІЛ 3. Алгоритм визначення інтегрального показника соціально-економічного розвитку регіон</i>	10.05.2024	10.05.2024
7	<i>Висновки</i>	15.05.2024	15.05.2024
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	20.05.2024	20.05.2024
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	27.05.2024	27.05.2024
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	28.05.2024	28.05.2024
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	31.05.2024	31.05.2024
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «22» грудня 2023 р.

9. Керівник випускної кваліфікаційної роботи Краскевич В.Є
(прізвище, ініціали, підпис)

10. Гарант освітньої програми Козлов В.В
(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент Козловський Д.О
(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи

Анотація

У дипломній роботі комплексно розглянуто актуальну проблему розробки онлайн-платформи адміністрування замовлень, що є невід'ємним інструментом для оптимізації та автоматизації бізнес-процесів сучасних підприємств, незалежно від їх розміру та сфери діяльності. Зростаючий попит на онлайн-сервіси та необхідність ефективної взаємодії з клієнтами зумовлюють актуальність дослідження та розробки спеціалізованих платформ для управління замовленнями.

Метою роботи є створення інтуїтивно зрозумілої, масштабованої та функціональної онлайн-платформи, що забезпечує повний цикл адміністрування замовлень: від їх оформлення та обробки до відстеження статусу, аналізу даних та підтримки комунікації з клієнтами. Платформа має бути адаптивною до потреб різних бізнес-моделей та галузей, надаючи можливість налаштування під індивідуальні вимоги.

Об'єктом дослідження є процес адміністрування замовлень в онлайн-середовищі, включаючи етапи оформлення, обробки, відстеження, аналізу та взаємодії з клієнтами. Предметом дослідження є сучасні методи та інструменти розробки онлайн-платформ, архітектурні підходи, технології забезпечення безпеки та інтеграції з іншими системами.

Практичне значення дослідження полягає у можливості використання розробленої онлайн-платформи підприємствами різних сфер діяльності для підвищення ефективності роботи, зниження витрат на обробку замовлень, покращення якості обслуговування клієнтів та отримання конкурентних переваг на ринку.

Ключові слова: онлайн-платформа, адміністрування замовлень, управління замовленнями, автоматизація бізнес-процесів, взаємодія з клієнтами, веб-розробка, електронна комерція, CRM-системи.

Anotation

This thesis comprehensively addresses the актуальну problem of developing an online order management platform, an essential tool for optimizing and automating business processes in modern enterprises, regardless of their size or industry. The growing demand for online services and the need for effective customer interaction underscore the relevance of research and development of specialized order management platforms.

The aim of this work is to create an intuitive, scalable, and functional online platform that provides a full cycle of order management: from placement and processing to status tracking, data analysis, and customer communication support. The platform should be adaptable to the needs of various business models and industries, providing customization options to meet individual requirements.

The object of research is the process of order management in an online environment, including the stages of placement, processing, tracking, analysis, and customer interaction. The subject of research is modern methods and tools for developing online platforms, architectural approaches, security technologies, and integration with other systems.

The practical significance of this research lies in the potential use of the developed online platform by businesses across various industries to improve operational efficiency, reduce order processing costs, enhance customer service quality, and gain a competitive advantage in the market.

Keywords: online platform, order management, order processing, business process automation, customer interaction, web development, e-commerce, CRM systems.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ВИМОГ ТА ДИЗАЙН ПЛАТФОРМИ.....	12
1.1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПЛАТФОРМ АДМІНІСТРУВАННЯ ОНЛАЙН-ЗАМОВЛЕНЬ.....	12
1.2 ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ ДО ПЛАТФОРМИ...	19
1.3 АРХІТЕКТУРА ПЛАТФОРМИ ТА ДИЗАЙН ІНТЕРФЕЙСУ.....	22
РОЗДІЛ 2. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ОСНОВНИХ МОДУЛІВ ПЛАТФОРМИ.....	24
2.1. РОЗРОБКА МОДУЛЯ УПРАВЛІННЯ ЗАМОВЛЕННЯМИ.....	24
2.2. РЕАЛІЗАЦІЯ МОДУЛЯ УПРАВЛІННЯ КОРИСТУВАЧАМИ.....	25

2.3. РОЗРОБКА МОДУЛЯ ПРОЦЕСИНГУ ПЛАТЕЖІВ.....	28
РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПЛАТФОРМИ	30
3.1. ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ТА НАДІЙНОСТІ ПЛАТФОРМИ	30
3.2. Вигляд сайту в браузері.....	35
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТОК КОДУ.....	41

ВСТУП

Актуальність. Тема кваліфікаційної роботи є вкрай актуальною, оскільки ефективне адміністрування замовлень є ключовим фактором успіху будь-якого підприємства, що працює в онлайн-середовищі. Зростаючий обсяг онлайн-торгівлі та збільшення кількості замовлень вимагають впровадження сучасних технологічних рішень для оптимізації та автоматизації процесів обробки замовлень. Онлайн-платформа адміністрування замовлень дозволяє підвищити ефективність роботи, зменшити кількість помилок, забезпечити прозорість процесів та покращити взаємодію з клієнтами.

Мета. Метою даної роботи є розробка функціональної та зручної онлайн-платформи для адміністрування замовлень, яка б забезпечувала повний цикл обробки замовлень, починаючи від їх оформлення та закінчуючи доставкою та аналізом результатів. Платформа має бути адаптивною, масштабованою та відповідати потребам різних типів бізнесу.

Об'єкт дослідження. Об'єктом дослідження є процес адміністрування замовлень в онлайн-середовищі, включаючи такі етапи, як прийом, обробка, відстеження, виконання та аналіз замовлень.

Предмет дослідження. Предметом дослідження є методи, технології та інструменти, що використовуються для розробки та впровадження онлайн-платформ адміністрування замовлень. Це включає вивчення архітектурних підходів, вибір технологічного стеку, проектування інтерфейсу користувача та розробку функціоналу платформи.

Інформаційна база дослідження. Дослідження буде базуватися на аналізі наукових публікацій, монографій, статей у фахових виданнях, матеріалах конференцій, а також огляді існуючих онлайн-платформ адміністрування замовлень та їх функціональних можливостей.

Методи дослідження. У роботі будуть використані такі методи дослідження, як аналіз та синтез, системний підхід, порівняльний аналіз, моделювання, проектування та тестування програмного забезпечення.

Практичне значення дослідження. Практичне значення дослідження полягає у можливості використання розробленої онлайн-платформи для покращення ефективності бізнес-процесів, зниження витрат на обробку

замовлень, підвищення рівня задоволеності клієнтів та збільшення прибутковості підприємства. Результати дослідження можуть бути використані як основа для подальшого розвитку та вдосконалення онлайн-платформ адміністрування замовлень.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ ТА ДИЗАЙН ПЛАТФОРМИ:

1.1. ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПЛАТФОРМ АДМІНІСТРУВАННЯ ОНЛАЙН-ЗАМОВЛЕНЬ

Дослідження існуючих платформ для адміністрування онлайн-замовлень

Сучасне бізнес-середовище вимагає від підприємств та організацій ефективного управління процесом замовлення. Платформи для адміністрування онлайн-замовлень є одним з найбільш затребуваних інструментів для автоматизації та оптимізації цього процесу. У цьому розділі представлено дослідження існуючих онлайн-платформ для адміністрування замовлень, а також дизайн платформи, яка буде розроблена в рамках даної роботи.

Аналіз існуючих платформ

Перш ніж розпочати розробку нової платформи, необхідно провести ґрунтовний аналіз наявних на ринку рішень. Це дозволить виявити їх сильні та слабкі сторони, визначити ключові тенденції та функціональні можливості, які мають бути реалізовані у новій платформі, а також зрозуміти потреби цільової аудиторії.

Огляд популярних платформ:

- **Shopify:** Популярна хмарна платформа, що спеціалізується на створенні інтернет-магазинів. Вона відрізняється інтуїтивно зрозумілим інтерфейсом, широким вибором шаблонів дизайну та додаткових модулів (додатків), а також інтеграцією з різноманітними платіжними системами та службами доставки. Shopify підходить для малого та середнього бізнесу, особливо для тих, хто не має глибоких технічних знань, оскільки не вимагає складного налаштування та

обслуговування. Крім того, Shopify пропонує вбудовані інструменти маркетингу (email-маркетинг, SEO, відновлення покинутих кошиків), що дозволяють ефективно просувати товари та збільшувати продажі. Також Shopify надає можливість створювати багатомовні магазини та підтримує дропшипінг, що є важливим для бізнесів, які працюють на міжнародному ринку або використовують цю модель продажів.

- **Magento:** Потужна платформа з відкритим вихідним кодом, що дозволяє створювати складні інтернет-магазини з індивідуальним дизайном та функціоналом. Magento пропонує гнучкі можливості налаштування, розширені функції для маркетингу, управління складними каталогами товарів (наприклад, з великою кількістю атрибутів та варіацій товарів) та багатомовними магазинами. Вона підтримує різні бізнес-моделі, включаючи B2B та оптову торгівлю, надаючи можливість налаштування індивідуальних цін та знижок для різних груп клієнтів. Magento також має потужні інструменти для SEO-оптимізації та аналітики, що дозволяють відстежувати ефективність магазину та приймати обґрунтовані рішення щодо його розвитку.
- **WooCommerce:** Безкоштовний плагін для WordPress, що перетворює сайт на повноцінний інтернет-магазин. WooCommerce відрізняється простотою використання, великою кількістю додаткових плагінів, що розширюють функціонал (від розширених опцій доставки до інтеграції з CRM-системами), та гнучкістю налаштування. Він ідеально підходить для малого бізнесу та стартапів, які вже використовують WordPress для своїх сайтів. WooCommerce також пропонує вбудовані інструменти для SEO-оптимізації та аналітики, що дозволяють відстежувати ефективність магазину та приймати обґрунтовані рішення щодо його розвитку.
- **Bitrix24:** CRM-система з вбудованим модулем інтернет-магазину, що дозволяє об'єднати управління продажами, маркетингом та клієнтською базою в одній системі. Bitrix24 пропонує широкий спектр інструментів для автоматизації бізнес-процесів, включаючи управління проектами, документообіг, комунікації, календар та завдання. Крім того, Bitrix24 надає можливість створювати власні додатки та інтегрувати їх з платформою, що дозволяє налаштувати систему під унікальні потреби бізнесу. Bitrix24 також включає в себе інструменти для email-маркетингу, телефонії та проведення вебінарів.

Детальний аналіз платформ:

Shopify:

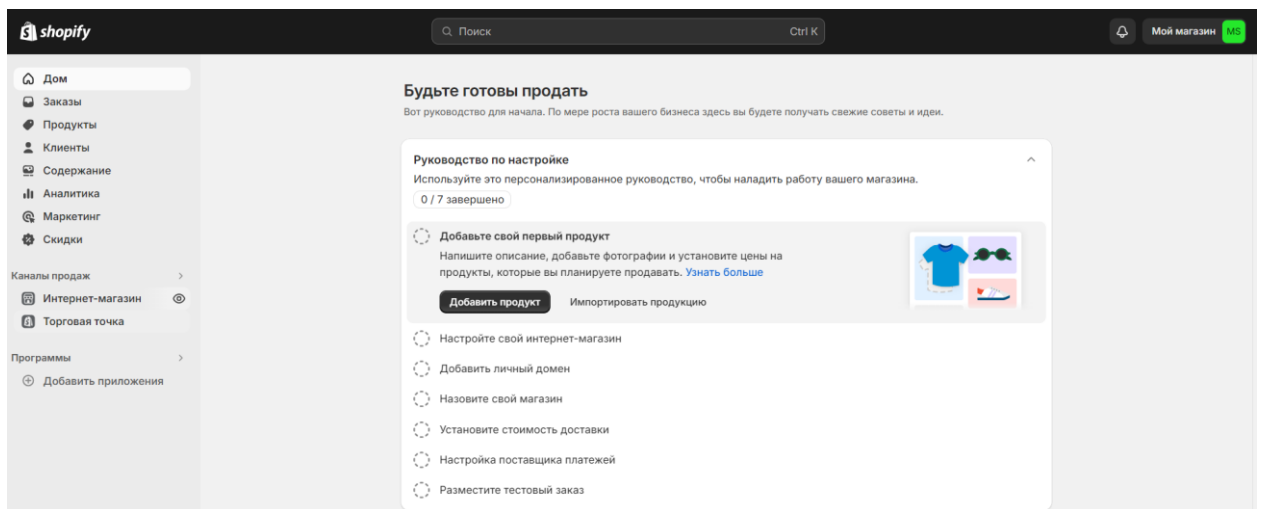


Рис.1.1 - Shopify

- **Преваги:** Простота використання, широкий вибір шаблонів та додатків, інтеграція з платіжними системами та службами доставки, SEO-оптимізація, надійна хостинг-інфраструктура, цілодобова технічна підтримка, вбудовані інструменти маркетингу та аналітики, можливість створення багатомовних магазинів, підтримка дропшипінгу, зручний мобільний додаток для управління магазином.
- **Недоліки:** Обмежена кастомізація, вартість (щомісячна абонентська плата та комісії за транзакції, вартість додаткових модулів), обмеження у функціоналі для складних бізнес-моделей, залежність від платформи Shopify.
- **Цільова аудиторія:** Малий та середній бізнес, який шукає просте та швидке рішення для запуску інтернет-магазину, підприємці-початківці, які не мають глибоких технічних знань, компанії, які займаються дропшипінгом.

Magento:

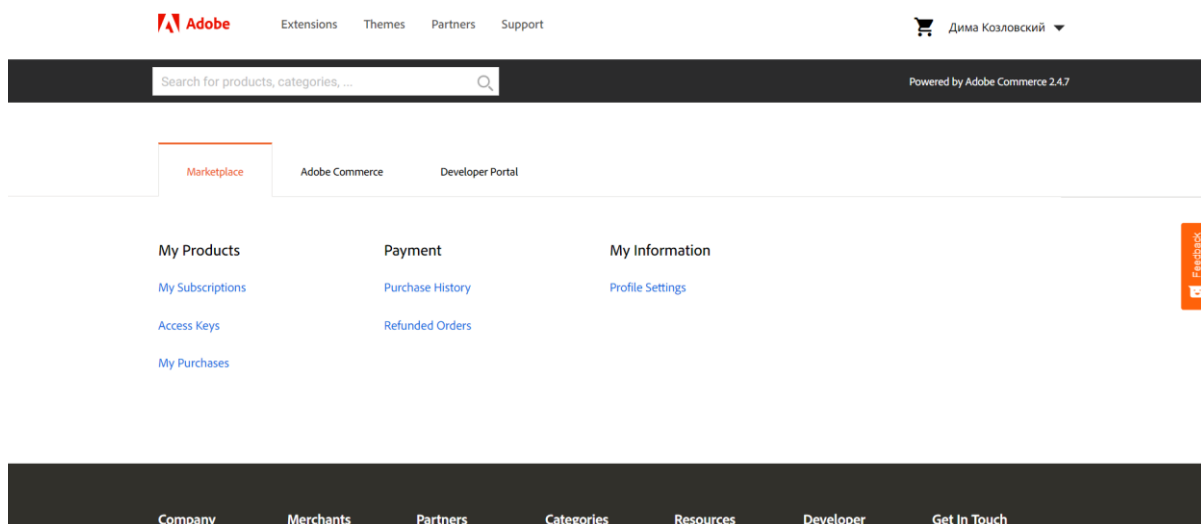


Рис.1.2 - Magento

- **Переваги:** Гнучкість та кастомізація, масштабованість, широкий функціонал, SEO-дружелюбність, велике співтовариство розробників, можливість створення багатомовних та мультивалютних магазинів, підтримка складних бізнес-моделей (B2B, оптова торгівля), розширені можливості управління каталогом товарів, інтеграція з різними ERP-системами.
- **Недоліки:** Складність, висока вартість розробки та підтримки, вимоги до хостингу, необхідність технічних знань для налаштування та управління, тривалий процес впровадження.
- **Цільова аудиторія:** Великі підприємства та компанії з високими вимогами до кастомізації та складними бізнес-моделями, які мають достатньо ресурсів для розробки та підтримки платформи.

WooCommerce:

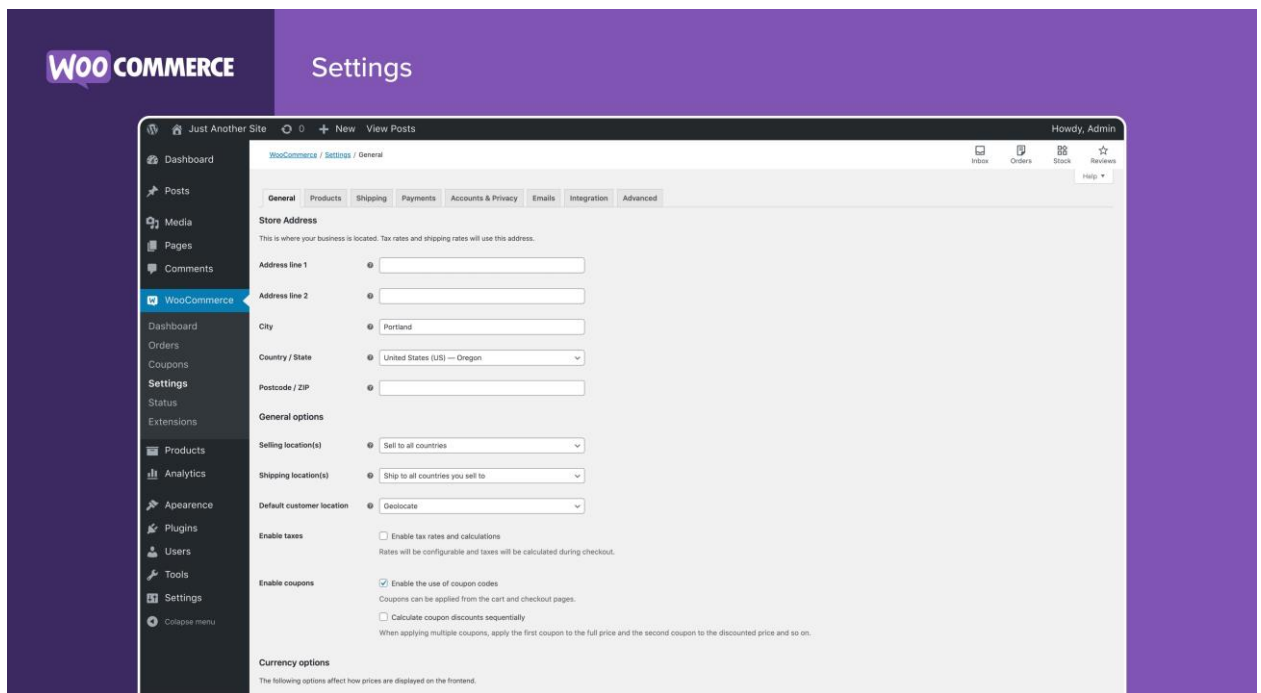


Рис.1.3 - WooCommerce

- **Переваги:** Безкоштовна, проста у використанні, велика кількість додаткових плагінів, гнучкість налаштування, інтеграція з WordPress, відкритий вихідний код, можливість використання на власному хостингу, велика спільнота користувачів та розробників.
- **Недоліки:** Обмеження у продуктивності при великому обсязі замовлень, менше можливостей для кастомізації порівняно з Magento, вимагає знань WordPress для ефективного використання, може вимагати додаткових витрат на хостинг та безпеку.
- **Цільова аудиторія:** Малий бізнес та стартапи, які вже використовують WordPress для своїх сайтів, підприємці з обмеженим бюджетом, які готові самостійно налаштовувати та підтримувати платформу.

KeyCRM:

№ заказа	Источник	Время создания	Дата доставки/отправки	Статус	Менеджер	Покупатель	Служба доставки	Трекинг код	Статус доставки	Товары	Статус оплаты	Сумма	Действия
191	GNZ UA	Сегодня 00:42		НОВЫЙ	Евгений Администратор	Флэкова Ана	Новою Поштою на відділення (Україна)		Добавить	1 1444-шт. Крон-тол Алпа, Яркен Желтый 2 1387-шт. Селест Тап, Яркен Желтый	Оплачено	1 000,00 грн	
190	GNZ UA	Вчера 21:20		НОВЫЙ	Евгений Администратор	Лаврская Анастас	Новою Поштою на відділення (Україна)		Добавить	10115-шт. Пальме Natali, Сней кобель	Оплачено	1 250,00 грн	
189	GNZ UA	22.04 11:27		НОВЫЙ	Евгений Администратор	Евгений Топчата	Новою Поштою на відділення (Україна)		Добавить	1398-шт. Носки, Белый	Оплачено	100,00 грн	
188	GNZ UA	22.04 02:35		НОВЫЙ	Евгений Администратор	Иван Ильяков	Новою Поштою на відділення (Україна)		Добавить	1398-шт. Носки, Белый	Оплачено	100,00 грн	
187	GNZ UA	22.04 01:56		НОВЫЙ	Евгений Администратор	Евгений Вареница	Новою Поштою на відділення (Україна)		Добавить	1398-шт. Носки, Белый	Оплачено	100,00 грн	
186	GNZ UA	22.04 01:44		НОВЫЙ	Евгений Администратор	Вареница Евгений	Новою Поштою на відділення (Україна)		Добавить	1398-шт. Носки, Белый	Не оплачено	0,00 грн	

Рис.1.4 - KeyCRM

Українська CRM-система, спеціально розроблена для автоматизації бізнес-процесів в сфері онлайн-торгівлі та послуг. KeyCRM пропонує інтеграцію з популярними маркетплейсами, соціальними мережами та месенджерами, що дозволяє збирати замовлення з різних каналів в єдиному інтерфейсі. Система також надає інструменти для управління замовленнями, клієнтською базою, складом, фінансами та аналітикою. KeyCRM відрізняється простотою використання, гнучкістю налаштування та доступною ціною,

Переваги:

- **Спеціалізація на українському ринку:** Система враховує особливості українського законодавства та бізнес-процесів, інтегрується з популярними українськими маркетплейсами (Prom.ua, Rozetka, Zakupka тощо), платіжними системами (LiqPay, Приват24) та службами доставки (Нова Пошта, Укрпошта).
- **Оmnіканальність:** Збір замовлень з різних каналів (маркетплейси, сайт, соціальні мережі, месенджери) в єдиному інтерфейсі.
- **Автоматизація рутинних операцій:** Автоматичне створення ТТН, відправка повідомлень клієнтам, оновлення статусів замовлень.
- **Зручний інтерфейс:** Інтуїтивно зрозумілий та простий у використанні інтерфейс, що не вимагає спеціальних технічних знань.
- **Гнучкість налаштування:** Можливість налаштування під індивідуальні потреби бізнесу.
- **Доступна ціна:** Гнучка система тарифів, що підходить для бізнесу різного розміру.
- **Якісна підтримка:** Оперативна технічна підтримка українською мовою.

Недоліки:

- **Обмежений функціонал порівняно з деякими західними аналогами:** Відсутність деяких специфічних функцій, які можуть бути потрібні великим підприємствам.
- **Менша популярність порівняно з деякими глобальними CRM:** Менша кількість користувачів та менша кількість доступних інтеграцій з іншими сервісами.
- **Цільова аудиторія:** Український малий та середній бізнес, що займається онлайн-торгівлею, особливо ті, хто продає товари на українських маркетплейсах.

Дизайн платформи

На основі аналізу існуючих платформ було прийнято рішення розробити нову платформу адміністрування замовлень. Основний функціонал, який має бути реалізований у новій платформі, включає:

1. Управління замовленнями: можливість створювати, редагувати та видаляти замовлення, а також відстежувати їх статус та історію змін.

2. Інтеграція з платіжними процесорами: платформа повинна підтримувати інтеграцію з популярними платіжними процесорами, такими як PayPal, Stripe тощо. Це дозволить автоматизувати процес оплати замовлень.

3. Аналітика та звітність: Платформа повинна надавати можливість аналізувати дані про замовлення, такі як обсяг продажів, популярність продукту та інші показники. Також потрібно вміти формувати звіти для аналізу ефективності.

4. Інтеграція з іншими системами: платформа повинна бути гнучкою та підтримувати інтеграцію з іншими системами, такими як CRM-системи, системи управління складом та інші. Це дозволить автоматизувати обмін даними між різними системами та спростити процеси всередині організації.

5. Інтерфейс користувача: Платформа повинна мати інтуїтивно зрозумілий і зручний інтерфейс, який дозволить користувачам легко звикнути до системи та виконувати необхідні операції.

Дослідження існуючих платформ адміністрування онлайн-замовлень виявило їх переваги та недоліки, а також визначило ключові функціональні можливості, які мають бути реалізовані на новій платформі. Виходячи з цього, було виконано проектування платформи, яка буде розроблятися в рамках даної роботи. Він включатиме управління замовленнями, інтеграцію з платіжними системами, аналітику та звітність, інтеграцію з іншими системами, а також зручний інтерфейс користувача. У наступному розділі мова піде про те, як розгорнути і протестувати розроблену платформу.

1.2. ВИЗНАЧЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ ДО ПЛАТФОРМИ

Визначення функціональних вимог до платформи

У цьому розділі ми проаналізуємо вимоги до платформи адміністрування замовлень та визначимо функціональні вимоги, які необхідно реалізувати в рамках розробки онлайн-платформи.

1. Реєстрація та авторизація користувача:

- Можливість реєстрації нових користувачів за допомогою логіна, пароля та контактної інформації.

- Авторизація користувача за допомогою логіна і пароля.

- Відновлення пароля в разі втрати.

2. Управління профілем користувача:

- Можливість зміни інформації в профілі користувача, такої як контактна інформація, адреса доставки і т.д.

- Можливість перегляду історії замовлень користувача.

3. Управління каталогом товарів:

- Можливість додавати нові товари з їх назвою, описом, ціною та зображенням.

- Можливість редагувати і видаляти існуючі товари.

- Можливість перегляду списку всіх товарів.

4. Управління замовленнями:

- Можливість створення нового замовлення із зазначенням обраної

продукції, кількості та адреси доставки.

- Можливість перегляду списку всіх замовлень із зазначенням їх статусу (в обробці, виконаних і т.д.).

- Можливість змінити статус замовлення (наприклад, підтвердження замовлення, скасування замовлення тощо).

- Можливість відслідковувати статус доставки замовлення.

5. Управління акціями та знижками:

- Можливість створювати акції та знижки на певні товари або категорії товарів.

- Можливість вказати термін дії акції або знижки.

- Можливість перегляду списку всіх акцій і знижок.

6. Управління користувачами:

- Можливість перегляду списку всіх користувачів.

- Можливість блокування або видалення користувачів, які порушують правила платформи.

- Можливість призначення користувачів на різні ролі (адміністратор, менеджер, оператор і т.д.).

7. Управління відгуками та рейтингами:

- Можливість перегляду списку всіх відгуків і оцінок товарів і послуг.

- Можливість додати відповідь до відгуку.

- Можливість видалення відгуків або рейтингів, які порушують правила платформи.

8. Управління платіжною системою:

- Можливість інтеграції з платіжними системами для прийому онлайн-платежів.

- Можливість перегляду списку всіх платежів та їх статусу.

9. Управління системою доставки:

- Можливість інтеграції з системами доставки для автоматичного розрахунку вартості доставки та відстеження статусу доставки замовлень.

10. Управління повідомленнями:

- Можливість надсилати користувачам повідомлення про статус замовлення, акції, знижки тощо.

- Можливість налаштування способів отримання повідомлень (e-mail, SMS і т.д.).

11. Аналітика та управління звітністю:

- Можливість перегляду статистики продажів, замовлень, популярності товару і т.д.

- Можливість створення та експорту звітів за різними параметрами.

12. Управління системою безпеки:

- Можливість захистити платформу від несанкціонованого доступу та злому.

- Можливість резервного копіювання даних і відновлення системи в разі збою.

Це лише деякі з функціональних вимог до платформи адміністрування замовлень. Остаточний перелік функцій та їх детальний опис можуть бути уточнені в ході подальшого аналізу та обговорення з замовником. Важливо враховувати, що розробка платформи повинна бути гнучкою і

масштабованою, щоб в майбутньому була можливість додавати нові функції і модулі відповідно до потреб бізнесу .

1.3. АРХІТЕКТУРА ПЛАТФОРМИ ТА ДИЗАЙН ІНТЕРФЕЙСУ

Проектування архітектури та інтерфейсу платформи є одним із важливих етапів розробки платформи для адміністрування онлайн-замовлень. У цьому розділі ми розглянемо основні принципи та підходи до проектування архітектури, а також детально опишемо інтерфейс платформи.

Проектування архітектури платформи починається з аналізу вимог, які пред'являються до системи. У разі розробки платформи онлайн-адміністрування замовлень основними вимогами є: можливість створення та управління замовленнями, автоматизація процесу обробки замовлень, інтеграція з іншими системами, забезпечення безпеки даних та високої продуктивності системи.

Для задоволення цих вимог пропонується використовувати трирівневу архітектуру, що складається з клієнтського, серверного і базового рівнів. Клієнтський рівень відповідає за представлення та взаємодію з даними користувачеві. Серверний рівень відповідає за обробку клієнтських запитів, взаємодію з базою даних та інтеграцію з іншими системами. Базовий рівень відповідає за зберігання даних і забезпечення їх безпеки [16].

На клієнтському рівні пропонується розробити веб-інтерфейс, який забезпечить зручну та інтуїтивно зрозумілу взаємодію користувача з системою. Веб-інтерфейс повинен бути адаптивним для коректного відображення на різних пристроях, таких як комп'ютери, планшети та смартфони. Він повинен містити всі необхідні елементи управління, такі як форми для створення і редагування замовлень, таблиці для відображення списку замовлень, кнопки для виконання різних операцій і т.д. .

Серверний рівень повинен бути реалізований з використанням сучасних технологій і дозволяти обробляти велику кількість запитів одночасно. Для забезпечення високої продуктивності системи пропонується використовувати масштабовану архітектуру на основі мікросервісної архітектури. Це дозволить розділити функціонал системи на окремі сервіси, кожен з яких відповідатиме за певне завдання. Наприклад, буде створено окремий сервіс для управління замовленнями, окремий сервіс для авторизації та аутентифікації користувачів тощо.

Базовий рівень повинен забезпечувати надійне зберігання даних і забезпечувати їх безпеку. Для зберігання даних пропонується використовувати реляційну базу даних, таку як PostgreSQL або MySQL. Реляційна база даних забезпечить структуроване зберігання даних і дозволить ефективно виконувати запити. Для забезпечення безпеки даних пропонується використовувати механізми шифрування даних, аутентифікації та авторизації, такі як OAuth або JWT.

Таким чином, проектування архітектури платформи включає в себе визначення трирівневої архітектури, проектування клієнтського інтерфейсу, реалізацію серверного рівня з використанням мікросервісної архітектури, забезпечення безпеки даних на базовому рівні. Це дозволить створити ефективну та надійну платформу адміністрування замовлень, яка відповідатиме вимогам користувачів та забезпечуватиме високу продуктивність та безпеку системи.

РОЗДІЛ 2. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ОСНОВНИХ МОДУЛІВ ПЛАТФОРМИ:

2.1. РОЗРОБКА МОДУЛЯ УПРАВЛІННЯ ЗАМОВЛЕННЯМИ

Розробка та впровадження основних модулів платформи є ключовим етапом у створенні платформи для адміністрування онлайн-замовлень. Одним з таких модулів є модуль управління замовленнями, який відповідає за весь життєвий цикл замовлень, від їх створення до їх виконання і доставки.

При розробці модуля управління замовленнями необхідно враховувати ряд функціональних вимог, які дозволять ефективно управляти замовленнями і забезпечувати їх успішну обробку. Однією з таких вимог є можливість створення замовлень. Система повинна надавати зручний інтерфейс, що дозволяє швидко і легко створювати нові замовлення. Для цього можна скористатися формою, в якій користувач заповнить необхідну інформацію про замовлення, таку як його назва, опис, вартість, адреса доставки та інші реквізити .

Після створення замовлення система повинна надати можливість його редагування. Можливість редагувати замовлення дозволяє вносити зміни в інформацію про замовлення, наприклад, змінювати адресу доставки, додавати або видаляти товари, змінювати вартість та інші параметри. Це особливо важливо, якщо користувач припустився помилки при створенні замовлення або якщо виникла необхідність внести зміни в процесі обробки замовлення .

Щоб ефективно керувати замовленнями, ви повинні мати можливість переглядати та фільтрувати замовлення. Система повинна надавати користувачеві зручний інтерфейс, який дозволяє швидко знаходити необхідні замовлення за допомогою різних фільтрів, таких як статус замовлення, дата створення, вартість та інші параметри. Це дозволить користувачеві легко

відстежувати статус замовлень і швидко знаходити потрібну йому інформацію [20].

Для забезпечення більш ефективного управління замовленнями необхідно передбачити можливість призначення відповідальних осіб за виконання доручень. Система повинна дозволяти призначати співробітників, які будуть відповідати за обробку і виконання замовлень. Це дозволить розподілити роботу між співробітниками і забезпечить більш ефективну обробку замовлень.

Одним з важливих аспектів управління замовленнями є контроль і відстеження їх виконання. Система повинна надавати можливість відстежувати статус кожного замовлення і отримувати повідомлення про його зміну. Наприклад, якщо статус замовлення змінюється з «оброблений» на «виконаний», користувач повинен бути повідомлений про це. Це дозволить користувачам завжди бути в курсі статусу своїх замовлень і своєчасно реагувати на зміни.

Для забезпечення успішної доставки замовлень необхідно передбачити модуль управління доставкою. Цей модуль повинен дозволяти контролювати процес доставки замовлень, відстежувати їх місцезнаходження та отримувати повідомлення про доставку. Наприклад, система може надавати можливість відстежувати місцезнаходження кур'єра, який доставляє замовлення, і відправляти повідомлення клієнту при наближенні кур'єра.

Важливим аспектом розробки модуля управління замовленнями є його інтеграція з іншими модулями платформи. Наприклад, модуль управління замовленнями повинен інтегруватися з модулем управління клієнтами, щоб мати доступ до інформації про клієнтів та їхні замовлення. Також модуль управління замовленнями повинен інтегруватися з модулем управління складом, щоб мати доступ до інформації про наявність товару на складі та забезпечувати успішну обробку замовлень.

Отже, розробка модуля управління замовленнями є важливим кроком у створенні онлайн-платформи для управління замовленнями. При його розробці необхідно враховувати функціональні вимоги, такі як можливість створювати та редагувати замовлення, переглядати та фільтрувати замовлення, призначати відповідальних за виконання замовлень, контролювати та відстежувати виконання замовлень, керувати доставкою, інтегруватися з іншими модулями платформи. Це дозволить створити ефективну систему управління замовленнями, яка забезпечить успішну обробку та доставку замовлень.

2.2. РЕАЛІЗАЦІЯ МОДУЛЯ УПРАВЛІННЯ КОРИСТУВАЧАМИ

Розробка та впровадження основних модулів платформи є ключовим етапом у створенні платформи для адміністрування онлайн-замовлень. Одним з таких модулів є модуль управління користувачами, який відіграє важливу роль у забезпеченні безпеки та ефективності роботи платформи.

Основним завданням модуля управління користувачами є надання можливості реєстрації, авторизації та управління користувачами на платформі. Для цього необхідно впровадити функціонал, який дозволяє користувачам створювати облікові записи, авторизуватися, змінювати свої дані, керувати правами доступу до функціоналу платформи .

Першим кроком в розробці модуля управління користувачами є проектування бази даних, в якій буде зберігатися інформація про користувачів. База даних повинна містити таблиці для зберігання облікових записів користувачів, їх персональних даних, а також інформації про права доступу .

Для реєстрації нових користувачів необхідно надати форму, в яку користувач буде вводити свої дані, такі як ім'я, прізвище, адреса електронної

пошти та пароль. Після заповнення форми дані необхідно перевірити на правильність і зберегти в базі даних. Для забезпечення безпеки паролів користувачів рекомендується використовувати алгоритми хешування [22].

Після реєстрації користувач повинен мати можливість авторизуватися на платформі. Для цього передбачена форма входу, в яку користувач повинен ввести свою адресу електронної пошти та пароль. Після звірки введених даних з даними з бази даних користувачеві надається доступ до функціоналу платформи.

Для забезпечення безпеки даних та захисту від несанкціонованого доступу необхідно впровадити механізм аутентифікації та авторизації користувачів. Аутентифікація дозволяє перевірити справжність користувача, а авторизація визначає його права доступу до різних розділів платформи. Для цього можна використовувати сесії або токени доступу.

Після успішної автентифікації користувач повинен мати можливість керувати своїм профілем. Для цього передбачений особистий кабінет, де користувач може переглядати і редагувати свої дані, змінювати пароль, додавати і видаляти свої адреси доставки і т.д.

Для забезпечення безпеки та захисту даних користувачів необхідно впровадити механізми шифрування та захисту інформації. Наприклад, паролі користувачів можуть зберігатися в зашифрованому вигляді, а дані, що передаються між клієнтом і сервером, можуть бути захищені за допомогою протоколу HTTPS.

Щоб керувати правами доступу користувачів, потрібно надати приклад для наслідування. Ролі можуть бути різними, наприклад, адміністратор, модератор, користувач. Кожна роль має свій набір прав доступу до функціоналу платформи. Адміністратор повинен мати повний доступ до всіх функцій платформи, модератор повинен мати обмежений доступ, а користувач повинен мати доступ лише до свого профілю та функцій

замовлення.

Реалізація модуля управління користувачами вимагає використання різних технологій та інструментів розробки. Ви можете використовувати HTML, CSS і JavaScript для створення інтерфейсу користувача. Для обробки даних на сервері можна використовувати мову програмування, таку як PHP, Python або Java. Для роботи з базою даних можна використовувати бібліотеки SQL або ORM.

Підсумовуючи, можна сказати, що модуль керування користувачами є важливою частиною платформи адміністрування замовлень. Розробка та впровадження цього модуля вимагає ретельного проектування баз даних, реалізації функціоналу реєстрації та авторизації користувачів, а також безпеки даних та управління правами доступу. Грамотне впровадження модуля управління користувачами забезпечить безпеку та ефективність роботи платформи.

2.3. РОЗРОБКА МОДУЛЯ ПРОЦЕСИНГУ ПЛАТЕЖІВ

Розробка та впровадження основних модулів платформи є одним із ключових завдань у створенні платформи для адміністрування онлайн-замовлень. У даній роботі ми розглянемо модуль процесингу платежів, який відіграє важливу роль у процесі здійснення фінансових операцій на платформі.

Модуль процесингу платежів відповідає за прийом та обробку фінансових операцій, пов'язаних із замовленнями користувачів. Він забезпечує зручний і безпечний інтерфейс для здійснення платежів, а також забезпечує надійність і конфіденційність передачі фінансових даних [25].

Перш ніж приступити до розробки модуля процесингу платежів, необхідно проаналізувати вимоги та визначити функціонал, який він повинен

забезпечувати. Важливо враховувати особливості платіжних систем, з якими буде взаємодіяти платформа, а також забезпечувати сумісність з різними платіжними шлюзами .

Одним з основних завдань модуля процесингу платежів є можливість приймати платежі від клієнтів. Для цього потрібно надати користувачеві зручний інтерфейс, що дозволяє вибрати спосіб оплати та ввести необхідні дані. Модуль повинен підтримувати різні види платежів, такі як банківські картки, електронні гаманці та інші платіжні системи .

Після отримання платіжних даних модуль процесингу платежів повинен їх валідувати та перевірити на відповідність вимогам платіжної системи. Сюди входить перевірка правильності введених даних, наявності достатнього балансу на рахунку клієнта та інших параметрів, визначених платіжною системою.

Після успішної валідації даних модуль обробки платежів повинен здійснити фактичну операцію переказу коштів. Для цього потрібно взаємодіяти з платіжним шлюзом, який забезпечує зв'язок з банками та іншими фінансовими установами. Модуль повинен передати необхідні дані в платіжний шлюз і обробити отриману від нього відповідь.

При обробці платежів необхідно також враховувати можливість помилок або невдалих транзакцій. Модуль процесингу платежів повинен передбачати механізми обробки помилок і відновлення даних, щоб мінімізувати ризики втрати коштів або некоректної обробки платежів.

Крім базового функціоналу, модуль процесингу платежів повинен забезпечувати безпеку фінансових операцій. Це і захист даних клієнтів, і використання шифрування при передачі інформації, і механізми виявлення та запобігання шахрайським транзакціям.

Для розробки модуля процесингу платежів можуть бути використані

різні технології та інструменти. Наприклад, ви можете використовувати API або SDK, що надаються відповідними сервісами, для взаємодії з платіжними шлюзами. Для забезпечення безпеки платежів можуть використовуватися протоколи шифрування, такі як SSL або TLS, а також системи аутентифікації та авторизації.

Підсумовуючи, модуль обробки платежів є важливою частиною платформи адміністрування замовлень, яка забезпечує фінансові операції. Розробка та впровадження цього модуля вимагає врахування особливостей платіжних систем, забезпечення безпеки та надійності операцій, а також сумісності з різними платіжними шлюзами.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПЛАТФОРМИ:

3.1. ПЛАНУВАННЯ ТА ПРОВЕДЕННЯ ФУНКЦІОНАЛЬНОГО ТЕСТУВАННЯ

У сучасному світі онлайн-платформи відіграють все більш важливу роль у різних сферах діяльності. Однією з таких платформ є платформа адміністрування замовлень, яка надає можливість керувати замовленнями, відстежувати їх виконання та взаємодіяти з клієнтами. Розробка такої платформи вимагає не тільки створення функціонального і зручного інтерфейсу, а й тестування для забезпечення його якості та надійності.

Метою даного розділу статті є вивчення та опис процесу планування та проведення функціонального тестування платформи адміністрування замовлень. Функціональне тестування є одним з найважливіших видів тестування, оскільки дозволяє перевірити відповідність функціоналу платформи вимогам і очікуванням користувачів [32].

Планування функціонального тестування

Першим кроком у плануванні функціонального тестування платформи адміністрування замовлень є визначення цілей тестування. Цілі можуть бути

різними, наприклад, перевірка виконання основних функцій платформи, виявлення та виправлення помилок, перевірка відповідності вимогам замовника тощо.

Далі йде план тестування, який включає опис функцій платформи, які потрібно протестувати, як і як вони будуть тестуватися, а також ресурсів, необхідних для проведення тестування. Тест-план також може включати опис тестових сценаріїв, які дозволяють відтворити типові ситуації використання платформи та перевірити її роботу в цих умовах [26].

Підбір методів та інструментів тестування

Після того, як у вас є план тестування, вам потрібно вибрати відповідні методи та інструменти для тестування. У функціональному тестуванні широко використовуються такі методи, як тестування чорної скриньки, тестування білої скриньки, тестування граничних значень, тестування варіантів використання тощо.

Тестування «чорної скриньки» є одним із найпоширеніших методів функціонального тестування. Він заснований на перевірці функціональності платформи без знання внутрішньої реалізації. Для проведення тестування чорних скриньок можна використовувати автоматизовані інструменти, такі як Selenium, які дозволяють записувати та відтворювати тестові сценарії.

Тестування White-box, у свою чергу, ґрунтується на знанні внутрішньої структури та логіки платформи. Для проведення тестування білого ящика можна використовувати інструменти, що дозволяють перевірити програмний код на відповідність заданим вимогам, наприклад, статичний аналізатор коду.

Оптимізуйте тестові сценарії

Одним з важливих аспектів функціонального тестування є оптимізація тестових сценаріїв. Оптимізація дозволяє скоротити час, витрачений на тестування, і підвищити його ефективність. Для оптимізації тест-кейсів

можна використовувати такі прийоми, як видалення дублікатів тестів, об'єднання тест-кейсів, використання параметризації і так далі.

Видалення дублікатів тестів запобігає повторному запуску одних і тих самих тестових сценаріїв. Для цього необхідно проаналізувати існуючі тести і виявити їх дублікати. Після цього можна виділити один з дублікатів і видалити інші.

Об'єднання тестових сценаріїв дозволяє зменшити кількість тестових сценаріїв і скоротити час, витрачений на їх виконання. Для цього необхідно проаналізувати існуючі тести і виявити їх загальні частини. Потім ви можете об'єднати ці частини в один тест.

Використання параметризації дозволяє проводити тестування з різними наборами даних, що дозволяє перевірити працездатність платформи в різних умовах. Для цього потрібно визначити параметри, які будуть використовуватися в тестових сценаріях, і встановити різні значення для цих параметрів.

Тестування та оптимізація платформи адміністрування замовлень є важливими етапами її розробки. Планування та проведення функціонального тестування дозволяє переконатися, що платформа відповідає вимогам та очікуванням користувачів. Підбравши правильні методи та інструменти тестування, а також оптимізувавши тестові скрипти, можна підвищити ефективність тестування та скоротити час, витрачений на тестування.

3.2. ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ТА НАДІЙНОСТІ ПЛАТФОРМИ

При розробці платформи для адміністрування онлайн-замовлень одним із ключових аспектів є оптимізація продуктивності та надійності платформи. Створюючи таку платформу, слід враховувати багато факторів, які можуть

вплинути на те, як вона працює та взаємодіє з користувачами. У цьому розділі ми розглянемо основні методи та інструменти, які допоможуть оптимізувати продуктивність та надійність платформи [4].

Одним з перших кроків в оптимізації продуктивності платформи є аналіз системних вимог. Потрібно розуміти, які завдання та функціонал потрібно виконувати платформі, щоб визначитися з необхідними ресурсами та конфігурацією сервера. Наприклад, якщо платформа буде обробляти велику кількість замовлень одночасно, то для ефективного зберігання і обробки інформації потрібна потужна оптимізація сервера і бази даних [3].

Одним з найважливіших аспектів оптимізації продуктивності є робота з базою даних. База даних — це основне сховище інформації про замовлення, клієнтів, продукти та інші сутності, які використовуються на платформі. Правильне проектування бази даних, оптимізація запитів та індексів, а також масштабування бази даних залежно від навантаження є важливими кроками для забезпечення високої продуктивності платформи [6].

Ще одним важливим аспектом оптимізації продуктивності є кешування даних. Кешування дозволяє прискорити доступ до часто доступних даних, таких як інформація про продукт або клієнта. Не потрібно виконувати одні й ті самі запити щоразу, коли ви запитуєте базу даних, якщо дані не змінилися. Замість цього ви можете зберігати результати запиту в кеші і використовувати їх в наступних запитах. Такий підхід значно знижує навантаження на базу даних і прискорює обробку запитів.

Ще одним важливим аспектом оптимізації продуктивності є оптимізація коду. Неefективний код може уповільнити роботу платформи та споживати більше ресурсів. Слід приділити увагу оптимізації алгоритмів, використанню ефективних структур даних та алгоритмів пошуку. Також важливо стежити за використанням пам'яті та ресурсів вашого сервера, щоб уникнути витоку пам'яті або перевантаження сервера.

Щоб забезпечити надійність платформи, необхідно проводити тестування на різних рівнях. По-перше, вам потрібно протестувати кожен компонент платформи. Модульні тести дозволяють переконатися в коректній роботі кожного окремого модуля і виявити можливі помилки або проблеми. По-друге, вам потрібно провести інтеграційне тестування, щоб перевірити взаємодію між різними компонентами платформи. Такі тести дозволяють виявити можливі проблеми із взаємодією компонентів. По-третє, необхідно провести навантажувальні випробування, щоб перевірити роботу платформи при високому навантаженні. Такі тести дозволяють виявити вузькі місця та визначити, яке навантаження може витримати платформа.

Оптимізація продуктивності та надійності платформи – це безперервний процес, який потребує постійного моніторингу та аналізу. Необхідно регулярно перевіряти продуктивність платформи, аналізувати логи та метрики продуктивності, щоб виявити можливі проблеми та покращити продуктивність платформи. Крім того, необхідно стежити за оновленнями та патчами компонентів та інструментів, що використовуються для забезпечення безпеки та стабільності платформи.

Підсумовуючи, оптимізація продуктивності та надійності платформи є важливим аспектом розробки платформи для адміністрування онлайн-замовлень. Потрібно враховувати системні вимоги, оптимізувати базу даних, використовувати кешування даних, оптимізувати код і проводити тестування на різних рівнях. Це допоможе створити стабільну та ефективну платформу, яка відповідатиме потребам користувачів та забезпечуватиме надійну роботу.

3.2 Вигляд сайту.

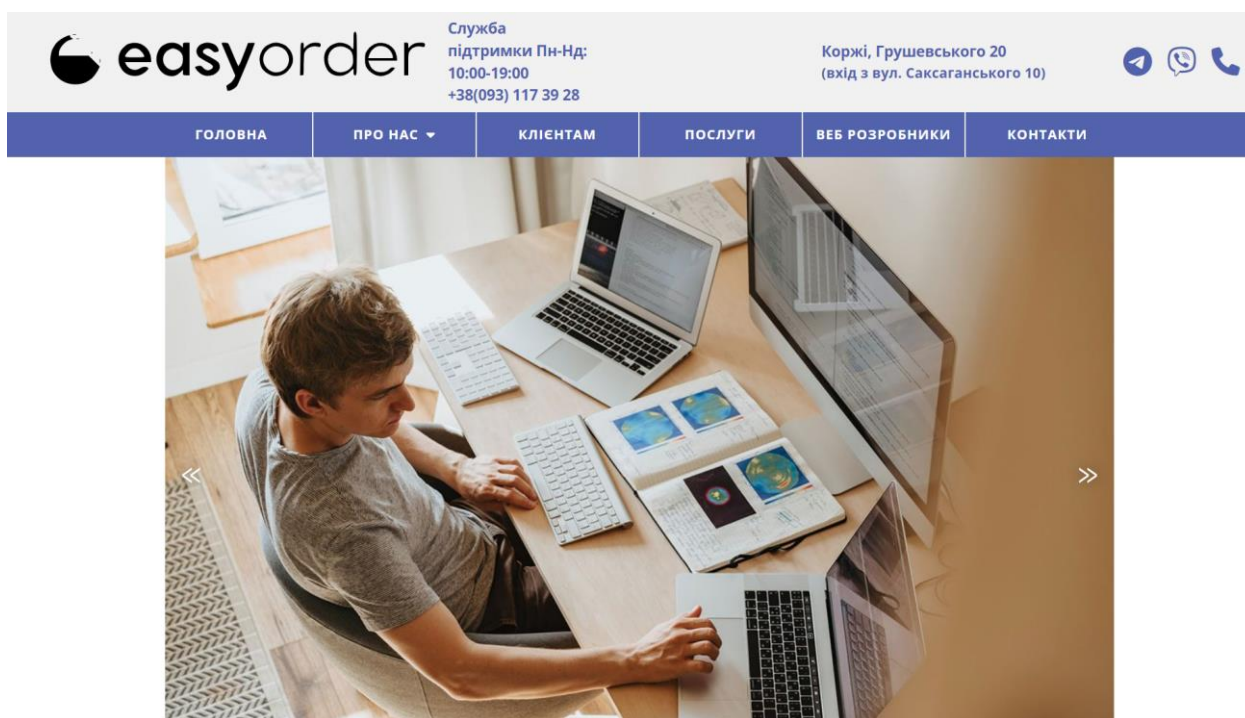


Рис.3.1 -Вигляд головної сторінки сайту.

Заходячи на сайт, людина відразу попадає на головну сторінку. Де зверху вказано таку інформацію: графік роботи служби підтримки, номер телефону, адресу.

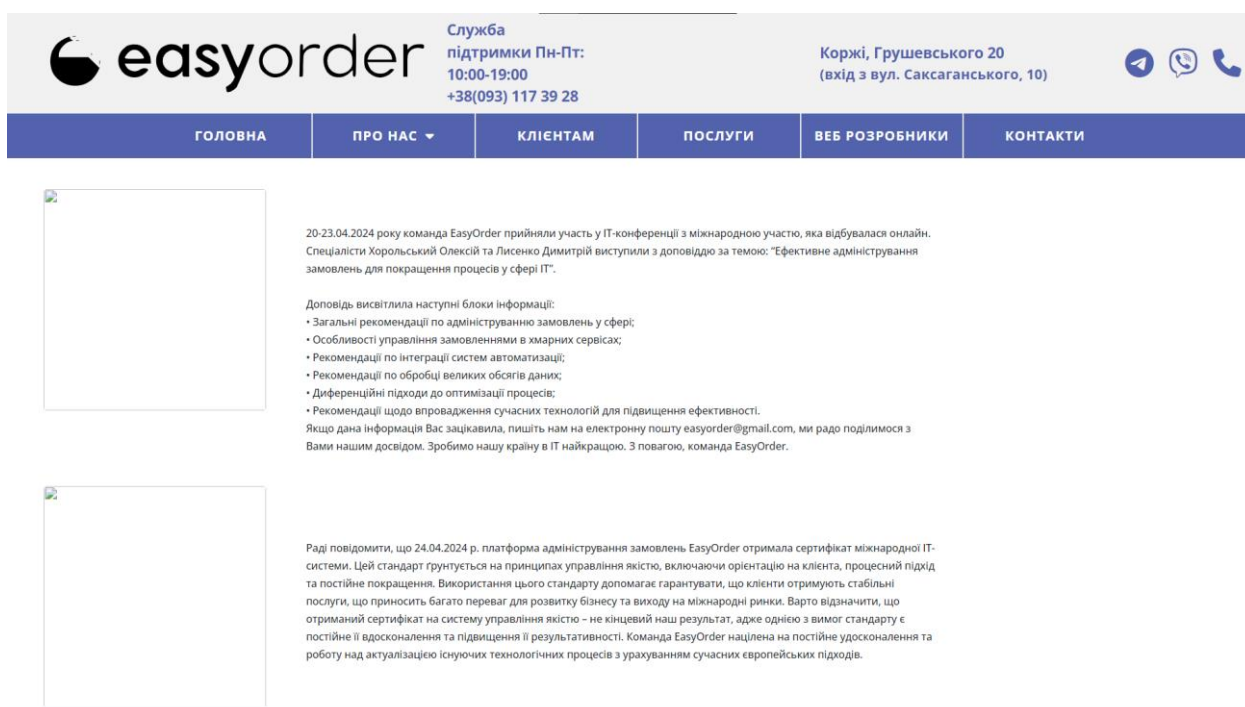


Рис. 3.2. - ”Про нас”

Також на сайті є інформація ”Про нас” на цій сторінці люди можуть ознайомитися з компанією та її успіхами.

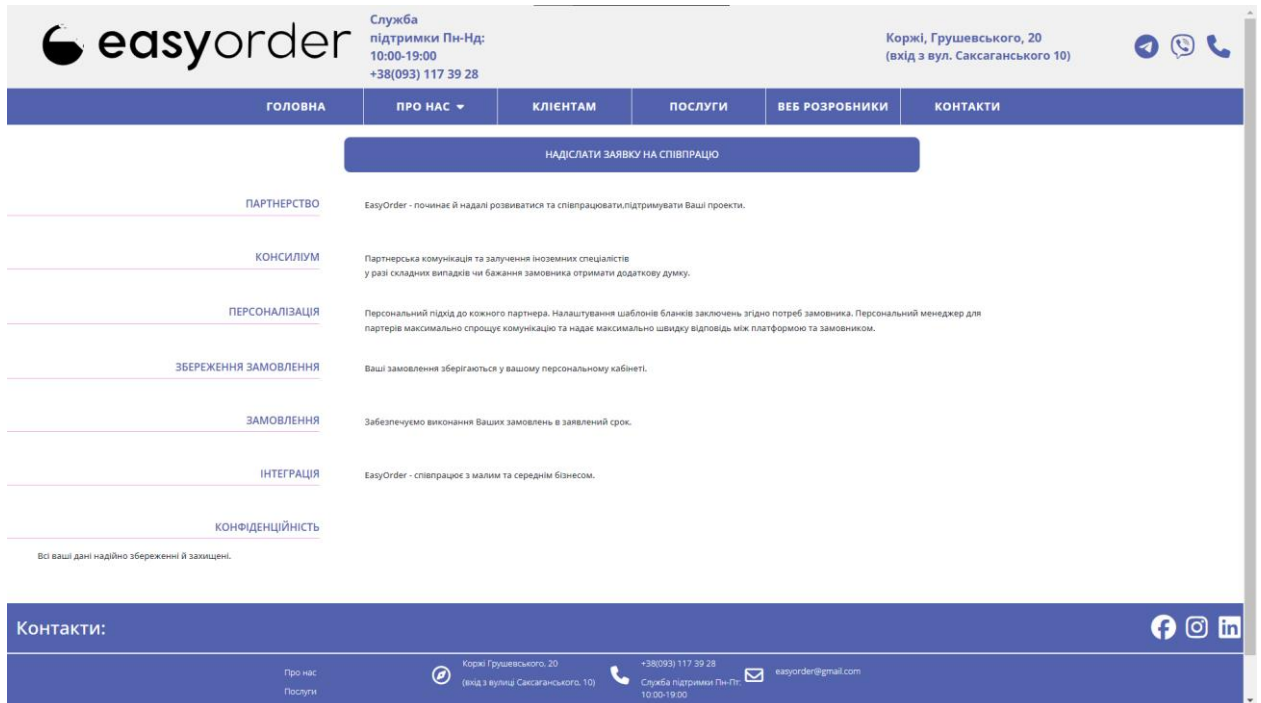


Рис.3.3 - Сторінка ”Клієнтам”

Переходячи на сторінку ”Клієнтам” наші клієнти можуть ознайомитися з партнерством та іншою інформацією, а також відправити заявку на співпрацю з платформою.

ГОЛОВНА	ПРО НАС ▾	Клієнтам	ПОСЛУГИ	ВЕБ РОЗРОБНИКИ	КОНТАКТИ
Вид роботи	Термін виконання (повних робочих днів)	Ціна	Замовити послугу		
Контроль та оновлення інформації про наявність товарів на складі, автоматичне оновлення кількості товарів після продажу.	2 дня	100	ЗАМОВИТИ		
Створення звітів про продажі, ефективність доставки, управління запасами, аналіз даних.	1-2 тижня	200 USD	ЗАМОВИТИ		
: Доступ до платформи через мобільні пристрої для управління замовленнями та перегляду звітів.	14 днів	150	ЗАМОВИТИ		
Синхронізація з маркетплейсами (Amazon, eBay, Shopify тощо) для автоматичного оновлення інформації про товари та замовлення.	14 днів	1900	ЗАМОВИТИ		
Використання алгоритмів машинного навчання для персоналізації рекомендацій товарів і маркетингових повідомлень для кожного клієнта.	2-4 тижня	700	ЗАМОВИТИ		
Інтегрована система управління взаємовідносинами з клієнтами (CRM), що дозволяє відстежувати взаємодії з клієнтами, керувати контактами та аналізувати дані про клієнтів.	14 днів	850	ЗАМОВИТИ		
Підтримка багатоканальної торгівлі через різні канали (веб-сайт, мобільний додаток, маркетплейси, фізичні магазини) з єдиною системою управління замовленнями.	14 днів	1000	ЗАМОВИТИ		
Автоматизація обробки повторюваних платежів, створення інвойсів, управління підписками та проведення фінансових операцій.	7 днів	200	ЗАМОВИТИ		

Рис. 3.4.- Послуги.

На цій сторінці клієнти платформи можуть ознайомитися з видом роботи, терміном виконання, ціною та замовити послугу. Зараз спробуємо замовити якусь послугу.

ФОРМА ЗАМОВЛЕННЯ ×

Заповніть коротку анкету і натисніть «надіслати»
 Оператор зв'яжеться з Вами найближчим часом для уточнення інформації.

: Доступ до платформи через мобільні пристрої для управління замовленнями та перегляду звітів. | 14 днів |

Рис.3.5 -. Форма замовлення

Я натиснув на послугу, де я вказав своє ім'я і контактний номер телефону. Натискаємо "Надіслати заявку" Так як я не розробив базу

даних, нічого відбуватися не буде.

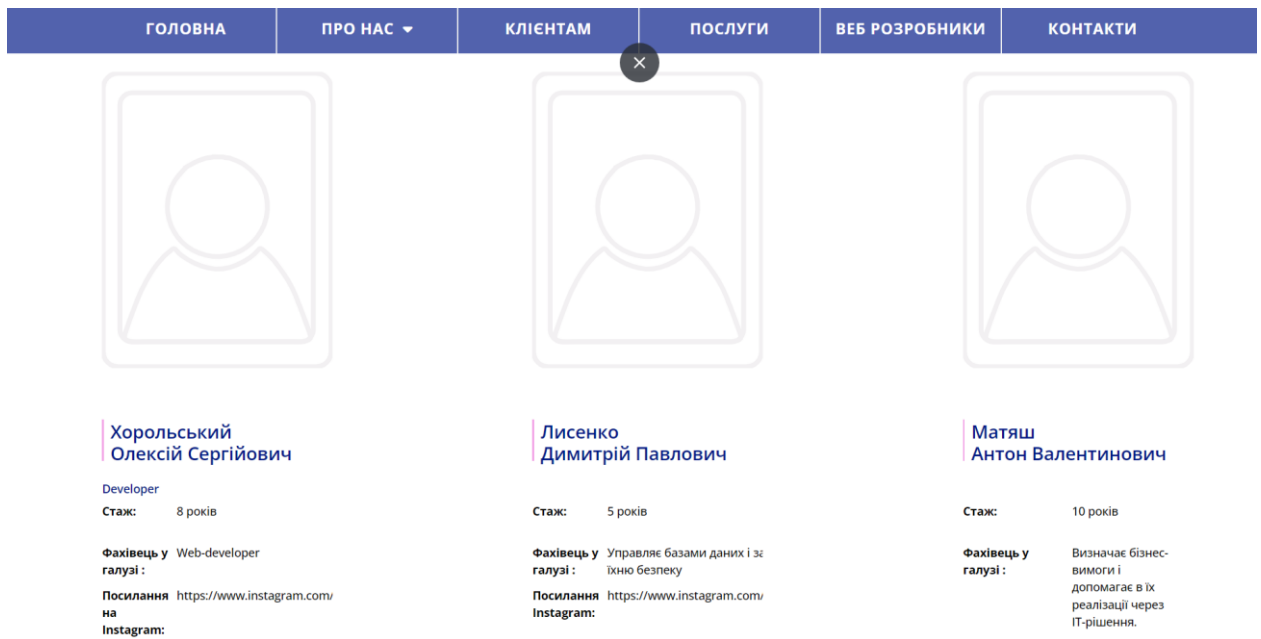


Рис.3.6. Сторінка Веб Розробники

На цій сторінці сайту можна побачити коротку інформацію про Веб Розробників ця сторінка показує: стаж, галузь в якій працює спеціаліст та посилання на сторінку.

Висновок

Результати виконаної дипломної роботи дозволяють зробити наступні висновки:

1. **Аналіз вимог та дизайн платформи:** В ході дослідження було проведено аналіз існуючих платформ адміністрування онлайн-замовлень, визначено функціональні вимоги та розроблено архітектуру платформи і дизайн інтерфейсу. Це дозволило створити основу для розробки ефективної та зручної у використанні системи.
2. **Розробка основних модулів:** Було розроблено та впроваджено основні модулі платформи, такі як модуль управління замовленнями, модуль управління користувачами та модуль процесингу платежів. Кожен модуль забезпечує необхідну функціональність для повного циклу адміністрування замовлень.
3. **Тестування та оптимізація:** Проведено тестування платформи з метою оптимізації продуктивності та надійності. Результати тестування показали, що платформа відповідає встановленим вимогам та може бути використана у реальних умовах.
4. **Практична значущість:** Розроблена онлайн-платформа має потенціал для застосування у різних галузях бізнесу, що сприятиме підвищенню ефективності операційної діяльності, зниженню витрат на обробку замовлень, покращенню якості обслуговування клієнтів та отриманню конкурентних переваг на ринку.

В майбутньому планується додати базу даних та CRM-систему для автоматизації процесів зв'язку з потенційними клієнтами.

Таким чином, проведення дослідження та розробка онлайн-платформи адміністрування замовлень є важливим кроком у напрямку автоматизації бізнес-процесів та покращення взаємодії з клієнтами.

Список використаної літератури

1. Книжкові джерела

- Freeman E., Robson E. Head First Design Patterns. - O'Reilly Media, 2020.

- Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. - Addison-Wesley, 1994.
- Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship. - Prentice Hall, 2008.
- Sommerville I. Software Engineering. - Pearson, 2016.
- Shneiderman B., Plaisant C., Cohen M., Jacobs S. Designing the User Interface: Strategies for Effective Human-Computer Interaction. - Pearson, 2016.

2.Статті та публікації

- Bass L., Clements P., Kazman R. Software Architecture in Practice // Addison-Wesley Professional. - 2012.
- Boehm B., Turner R. Balancing Agility and Discipline: A Guide for the Perplexed // Addison-Wesley Professional. - 2004.
- Cockburn A. Agile Software Development: The Cooperative Game // Addison-Wesley Professional. - 2006.
- Nielsen J. Usability Engineering // Morgan Kaufmann. - 1994.
- Fowler M. Refactoring: Improving the Design of Existing Code // Addison-Wesley Professional. - 2018.

3.Інтернет-джерела

- Mozilla Developer Network. JavaScript documentation [Електронний ресурс]. - Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- W3C. Web Content Accessibility Guidelines (WCAG) 2.1 [Електронний ресурс]. - Режим доступу: <https://www.w3.org/TR/WCAG21/>.
- Stack Overflow. Community-driven Q&A platform for developers [Електронний ресурс]. - Режим доступу: <https://stackoverflow.com>.
- Google Developers. Web Fundamentals [Електронний ресурс]. - Режим доступу: <https://developers.google.com/web/fundamentals/>.
- Smirnov O. Best practices for web application development [Електронний ресурс]. - Режим доступу: <http://www.webdevpractices.com>.

Додаток коду index.html

```
<html lang="uk"><div id="in-page-channel-node-id" data-channel-name="in_page_channel_9qjtdO"></div><head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>EasyOrder - Платформа адміністрування систем</title>
```

```
<link rel="icon" type="image/x-icon" href="DIALOG-mini-03.svg">
```

```
<meta name="keywords" content="EasyOrder, EasyOrder, Order, Easy Order, EasyOrder, Easy-Order, Легке  
Замовлення, Легке Замовлення, Лаба гісто, Платформа адміністрування замовлень, Плафформа адміністрування  
Легке Замовлення, Легке, Легке замовлення Платформа, Легке замовлення адміністрування, Адміністрування  
замовлень, Гістологія лабораторія, Платформа Легке замовлення, Легке замовлення, Платформа  
адміністрування замовлень, Платформа адміністрування замовлень Легке замовлення, Адміністрування  
замовлень, Платформа Легке замовлення, Легке замовлення, Легке замовлення адміністрування, Легке  
замовлення , Легке замовлення систем, Дослідження, Платформа, Адміністрування Легке замовлення,  
Дослідження системи, адміністрування замовлень, Платформа адміністрування замовлень Легке  
замовлення">
```

```
<meta name="description" content="Ласкаво просимо на EasyOrder, вашу провідну платформу для  
адміністрування замовлень. Ми пропонуємо широкий спектр інструментів та рішень, що спрощують процес  
обробки замовлень, забезпечують ефективну комунікацію з клієнтами та допомагають оптимізувати ваш  
бізнес. Наша команда досвідчених фахівців готова надати вам професійну підтримку та консультації з усіх  
питань, пов'язаних з управління замовленнями. Звертайтеся до нас, щоб отримати доступ до сучасних  
технологій та інноваційних рішень, які допоможуть вам досягти успіху у вашій галузі.">
```

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
```

```
<script src="https://kit.fontawesome.com/d003f22ee1.js" crossorigin="anonymous"></script><style media="all"  
id="fa-v4-font-face"></style>
```

```
<link rel="preconnect" href="https://fonts.googleapis.com">
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin="">
```

```
<link  
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;500;600;700;800&display=swap"  
rel="stylesheet">
```

```
<link rel="stylesheet" href="about.css">
```

```
</head>
```

```
<body>
```

```
<div class="wrapper">
```

```
<div class="header">
```

```
<div class="navigation-stuff">
```

```
<div class="left">
```

```

```

```
<div class="date-and-number" id="location">
```

```
<span class="date">Служба підтримки Пн-Нд: 10:00-19:00</span>
```

```
<br>
```

```
<span>+38(093) 117 39 28</span>
```

```
</div>
```

```
</div>
```

```
<div class="right">
```

```
<div class="date-and-number">
```

```
<span>Коржі, Грушевського 20</span>
```

```
<br>
```

```
<span id="min-text">(вхід з вул. Сакаганського 10)</span>
```

```
</div>
```

```
<a id="facebook"><i class="fa-brands fa-telegram"></i></a>
```

```
<a id="linkedin" style="font-size: 40px;"><i class="fa-brands fa-viber"></i></a>
```

```
<a href="tel:+380931173928" id="linkedin"><i class="fa-solid fa-phone"></i></a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="nav">
```

```
<div class="under-navigation">
```

```
<div class="dropdown" data-dropdown="">
```

```
<a href="contacts.html"><button class="link last" id="info-bar" data-dropdown-button="">КОНТАКТИ</button></a>
```

```
</div>
```

```
<div class="dropdown" data-dropdown="">
```

```

        <a href="web.html"><button class="link" id="info-bar" data-dropdown-button="">БЕБ
РОЗРОБНИКИ</button></a>

    </div>

    <div class="dropdown" data-dropdown="">

        <a href="home.html"><button class="link" id="info-bar" data-dropdown-
button="">ПОСЛУГИ</button></a>

    </div>

    <div class="dropdown" data-dropdown="">

        <a href="clients.html"><button class="link too-big-btn" id="info-bar" data-dropdown-
button="">КЛІЄНТАМ</button></a>

    </div>

    <div class="dropdown" id="dropdown" data-dropdown="">

        <a><button class="link" id="info-bar" data-dropdown-button="">ПІРО НАС <i class="fa-solid fa-
caret-down" style="margin-left: 5px;"></i></button></a>

        <div class="dropdown-content" id="dropdown-content"
onmouseover="dropdown_color_changer_mouse_over()" onmouseout="dropdown_color_changer_mouse_out()">

            <div class="dropdown-content-link" style="vertical-align: center; padding: 5px;"><a
href="news_home.html" style="vertical-align: center; padding: 5px;">Новини</a></div>

            <div class="dropdown-content-link dropdown-content-link-last" style="vertical-align: center; padding:
5px;"><a href="laws.html" style="padding: 5px;">Публічний договір</a></div>

        </div>

    </div>

    <div class="dropdown" data-dropdown="">

        <a href="index.html"><button class="link" id="info-bar" data-dropdown-
button="">ГОЛЮБНА</button></a>

    </div>

</div>

<div class="navbar-responsive-left">

```

```

<a href="https://www.facebook.com/frayzzzyy"><i class="fa-brands fa-facebook"></i> </a>

<a href="#" id="instagram-under"><i class="fa-brands fa-instagram"></i> </a>

<a href="tel:"><i class="fa-solid fa-phone"></i></a>

</div>

<div class="dropdown" data-dropdown="">

  <button class="link burger-btn" id="burger-knopka"><i class="fa-solid fa-bars"></i></button>

</div>

</div>

<main>

<ul class="szirovaja-otkladina" id="tayler-szirden">

  <li><a href="contacts.html">КОНТАКТИ</a></li>

  <li><a href="clients.html">КЛІЄНТАМ</a></li>

  <li><a href="web.html">ВЕБ РОЗРОБНИКИ</a></li>

  <li><a href="home.html">ПОСЛУГИ</a></li>

  <li id="mobile-dropdown-1"><a class="mobile-dropdown"><button
onclick="mobile_dropdown_function()" class="dropdown-btn" id="dropdown-btn">ПІРО НАС <i class="fa-solid
fa-caret-down" style="margin-left: 5px;"></i></button></a></li>

  <div class="mobile-dropdown-content" id="mobile-dropdown-content">

    <li class="dropdown-link"><a href="news_home.html">Новини</a></li>

  </div>

  <li><a href="index.html">ГОЛОВНА</a></li>

</ul>

<div class="content_info" style="margin-top: 0;">

<div id="slider" class="desktop-slider slider">

<ul id="slideWrap" class="slideWrap" style="height: 50vh;">

  <li></li>

```

```

    <li></li>

    <li></li>

    <li></li>

</ul>

<a id="prev" class="prev" href="#"><<</a>

<a id="next" class="next" href="#">>>/a>

</div>

<div id="slider" class="mobile-slider slider">

    <ul id="slideWrap" class="slideWrap" style="height: 40vh;">

        <li style="height: 40vh;"></li>

        <li style="height: 40vh;"></li>

        <li style="height: 40vh;"></li>

        <li style="height: 40vh;"></li>

    </ul>

    <a id="prev" class="prev" href="#"><<</a>

    <a id="next" class="next" href="#">>>/a>

</div>

</div>

<script>

let responsiveSlider = function(){

    let slider = document.querySelector(".slider");

    let sliderWidth = slider.offsetWidth;

    let slideList = document.querySelector(".slideWrap");

    let count = 1;

    let items = slideList.querySelectorAll("li").length;

    let prev = document.querySelector(".prev");

```

```
let next = document.querySelector(".next");
```

```
window.addEventListener('resize', function() {  
    sliderWidth = slider.offsetWidth;  
})
```

```
let prevSlide = function() {  
    if(count > 1){  
        count = count -2;  
        slideList.style.left = "-" + count * sliderWidth + "px";  
        count++;  
    }  
    else if(count = 1){  
        count = items - 1;  
        slideList.style.left = "-" + count * sliderWidth + "px";  
        count++;  
    }  
};
```

```
let nextSlide = function() {  
    if(count < items){  
        slideList.style.left = "-" + count * sliderWidth + "px";  
        count++;  
    }  
    else if(count = items){
```

```

        count = items - 1;

        slideList.style.left = "0px";

        count = 1;
    }
};

next.addEventListener("click", function(){
    nextSlide();
});

prev.addEventListener("click", function(){
    prevSlide();
});

setInterval(function(){
    nextSlide();
}, 5000);

// Additional code for mobile slider

let mobileSlider = document.querySelector(".mobile-slider");
let mobileSlideList = mobileSlider.querySelector(".slideWrap");
let mobileItems = mobileSlideList.querySelectorAll("li").length;
let mobilePrev = mobileSlider.querySelector(".prev");
let mobileNext = mobileSlider.querySelector(".next");

let mobileCount = 1;

```

```
let mobileSliderWidth = mobileSlider.offsetWidth;
```

```
window.addEventListener('resize', function() {  
    mobileSliderWidth = mobileSlider.offsetWidth;  
});
```

```
let mobilePrevSlide = function() {  
    if (mobileCount > 1) {  
        mobileCount -= 2;  
        mobileSlideList.style.left = "-" + mobileCount * mobileSliderWidth + "px";  
        mobileCount++;  
    } else if (mobileCount === 1) {  
        mobileCount = mobileItems - 1;  
        mobileSlideList.style.left = "-" + mobileCount * mobileSliderWidth + "px";  
        mobileCount++;  
    }  
};
```

```
let mobileNextSlide = function() {  
    if (mobileCount < mobileItems) {  
        mobileSlideList.style.left = "-" + mobileCount * mobileSliderWidth + "px";  
        mobileCount++;  
    } else if (mobileCount === mobileItems) {  
        mobileCount = mobileItems - 1;  
        mobileSlideList.style.left = "0px";  
        mobileCount = 1;  
    }  
};
```

```

    }
};

mobileNext.addEventListener("click", function() {
    mobileNextSlide();
});

mobilePrev.addEventListener("click", function() {
    mobilePrevSlide();
});

setInterval(function() {
    mobileNextSlide();
}, 5000);
};

window.onload = function(){
    responsiveSlider();
};
</script>
</main>
<div id="contact" class="container-fluid bg-primary mobile-footer">
    <div class="container-fluid px-0 py-0"><h2 class="pull-left">Контакты:</h2></div>
    <div class="row">
        <div class="col-sm-7">
            <p class="mb-0"><span class="glyphicon glyphicon-map-marker"><i class="fa-solid fa-location-dot"
style="margin-right: 5px;"></i></span></p>

```

<p> </p>

<p><i class="fa-solid fa-clock" style="margin-right: 3px;"></i>
 Служба підтримки Пн-Пт: 10:00-19:00</p>

<p><i class="fa-solid fa-phone" style="margin-right:
5px;"></i> +38(093) 117 39 28</p>

<p><i class="fa-solid fa-envelope" style="margin-right:
5px;"></i> easyorder@gmail.com</p>

</div>

</div>

<div class="container-fluid border-top px-0 pt-3 pb-0" >

<h2 class=".pull-left mb-3">СЛІДКУЙТЕ ЗА НАМИ:</h2>

<h1 class=".pull-left" style="font-size: 35px; margin-bottom: 0;"><i class="fa-brands fa-facebook"></i> <i
class="fa-brands fa-instagram"></i> <i class="fa-brands fa-linkedin"></i></h1>

</div>

</div>

<!-- Image of location/map -->

<footer class="container-fluid text-center mobile-footer">

<p id="cookies-rights" style="padding-bottom: 5px;">© EASYORDER 2024 Всі права захищені| Цей
сайт використовує політику Cookies. Дізнатися більше.</p>

</footer>

<footer class="footer">

<div class="upper-footer">

<div class="footer-col">

```
<div class="social-links">
  <li><h4>Контакти: </h4></li>
  <li class="social-networks">
    <a href="https://www.facebook.com/frayzzzyy"><i class="fa-brands fa-facebook"></i></a>
    <a href="https://www.instagram.com/frayzzzyy/" id="instagram-under"><i class="fa-brands fa-
instagram"></i></a>
    <a href="#" id="linked-in-under"><i class="fa-brands fa-linkedin"></i></a>
  </li>
</div>
</div>
</div>
</div>
<div class="container">
  <div class="row">
    <div class="footer-col">
      <ul>
        <li><a href="news_home.html">Про нас</a></li>
        <li><a href="home.html">Послуги</a></li>
        <li><a href="corporative_clients.html">Клієнтам</a></li>
      </ul>
    </div>
    <div class="footer-col footer-adresses">
      <h4><i class="fa-regular fa-compass"></i></h4>
      <ul>
      </ul>
    </div>
    <div class="footer-col phones-footer">
      <h4><i class="fa-solid fa-phone"></i></h4>
```

```
<ul>
  <li><a href="#">+38(093) 117 39 28</a></li>
  <li><a href="#"> Служба підтримки працює Пн-Нд: 10:00-19:00 </a></li>
</ul>
</div>
<div class="footer-col footer-email">
  <h4><i class="fa-regular fa-envelope"></i></h4>
  <ul class="footer-email-ul">
    <li><a href="#">easyorder@gmail.com</a></li>
  </ul>
</div>
</div>
</div>
<div class="upper-footer under-footer">
  <div class="footer-col">
    <div class="social-links">
      <div class="grid-container">
        <div class="item2">
          <p class="cookeis-and-confidence">
            © EASYORDER 2024 | Цей сайт використовує політику Cookies. Дізнатися більше.
          </p>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
</div>
```

```
</footer>

</div>

<script>

const burger_menu = document.getElementById("taylor-szirden");

const open_button = document.getElementById("burger-knopka");

let counter_of_fat_clicks = 0;

open_button.addEventListener("click", function() {

  if (counter_of_fat_clicks === 0) {

    burger_menu.style.display = "block";

    counter_of_fat_clicks++;

  } else {

    burger_menu.style.display = "none";

    counter_of_fat_clicks = 0;

  }

});

</script>
```