

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему **Особливості побудови сайту надання послуг з використанням**
Python і Django

Виконав: студент 2 курсу, групи 8.1212-іпз-2
спеціальності 121 Інженерія програмного
забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного
забезпечення

(код і назва освітньої програми)

К.К. Лещенко

(ініціали та прізвище)

Керівник доцент, к.ф.-м.н., В.І.Попівший
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дискус

Р.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2023

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

Кафедра Електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення
(код та назва)

Освітня програма Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Т.В. Критська
“ 01 ” вересня 2023 року

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

Лещенку Костянтину Костянтинівичу

(прізвище, ім'я, по батькові)

1. Тема роботи Особливості побудови сайту надання послуг з використанням Python і Django

керівник роботи Попівший Василь Іванович, доцент, к.ф.-м.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від 02.06. 2023 р. №597-с

2. Строк подання студентом кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження технологій, методів та засоби проектування веб-сайтів надання послуг;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
12 слайдів презентації.

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.09.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-10.09. 23	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.09-12.09. 23	виконано
3	Ознайомлення із загальними положеннями інтернет-магазинів	13.09-14.09. 23	виконано
4	Аналіз існуючих методів рішення	15.09-20.09. 23	виконано
5	Аналіз методів розробки сайтів надання послуг	21.09-26.09. 23	виконано
6	Узгодження подальших дій з науковим керівником	27.09-28.09. 23	виконано
7	Реалізація сайту надання послуг з використанням Python і Django	29.09-13.10. 23	виконано
8	Реалізація користувацького інтерфейсу	14.10-16.10. 23	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.10-19.10. 23	виконано
10	Усунення додаткових проблем та «багів»	20.10-09.11. 23	виконано
11	Перевірка працездатності створеного програмного застосунку	10.11-17.11. 23	виконано
12	Аналіз платформ, що надають послуги хмарних сервісів	18.11- 23.11. 23	виконано
13	Оформлення звіту	23.11-27.11. 23	виконано

Студент _____ К.К.Лещенко
 (підпис) (ініціали та прізвище)

Керівник роботи _____ В.І. Попівций
 (підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ І.А. Скрипник
 (підпис) (ініціали та прізвище)

АНОТАЦІЯ

Сторінок: 80

Рисунків: 41

Таблиць: 4

Джерел: 29

Лещенко К. К. Особливості побудови сайту надання послуг з використанням Python і Django : кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник В. І. Попівщій. Запоріжжя : ЗНУ, 2023. 80 с.

Метою кваліфікаційної роботи є дослідження існуючих фреймворків на мові Python та розробка з використанням обраного фреймворку інтернет-магазину для надання послуг в сфері обслуговування та продажу вантажних шин.

В процесі дослідження фреймворків виявлені і проаналізовані їх архітектурні відмінності, необхідні бібліотеки та шаблони і як інструмент реалізації обрано фреймворк Django.

Спроектовано і реалізовано інтернет-магазин вантажних шин виходячи з аналізу існуючих рішень та регіональних вимог. Виконано тестування розробленого проекту, проаналізовано варіанти розгортання проекту в хмарних середовищах та розгорнуто і протестовано проєкт на одному з хмарних сервісів.

Ключові слова: веб-сайт, інтернет-магазин, фреймворки, Django, хмарні сервіси, MVP.

SUMMARY

Pages: 80

Figures: 41

Tables: 4

Sources: 29

Leshchenko K. K. Features of building a service provision site using Python and Django : qualification work of the master of specialty 121 "Software Engineering" / Science head Popivshchyi V.I. Zaporizhzhia: ZNU, 2023. p. 80.

The purpose of the qualifying work is to study existing frameworks in Python and to develop an online store to provide services in the field of servicing and selling truck tires using the selected framework.

In the process of researching frameworks, their architectural differences, necessary libraries and templates were identified and analyzed, as a result, the Django framework was chosen as an implementation tool.

An online store for truck tires was designed and implemented based on an analysis of existing solutions and regional requirements. The developed project was tested, options for deploying the project in cloud environments were analyzed, the project was deployed and tested on one of the cloud services.

Keywords: website, online store, frameworks, Django, cloud services, MVT.

ЗМІСТ

ВСТУП	8
1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗРОБКИ ВЕБ-САЙТІВ ІНТЕРНЕТ МАГАЗИНІВ	10
1.1 Електронна комерція	10
1.2 Аналіз основних типів інтернет-магазинів	14
1.3 Огляд видів веб-сайтів.....	16
1.4 Огляд основних методів створення сайтів	19
1.5 Аналіз ресурсів аналогічної тематики	23
2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ЗАСТОСУНКІВ.....	29
2.1 Вибір інструментів реалізації	29
2.2 Аналіз Python-фреймворків	33
2.3 Аналіз систем управління базами даних	35
2.4 Аналіз архітектури сучасних веб-додатків	37
2.5 Висновки до розділу 2	39
3 РОЗРОБКА ВЕБ-САЙТУ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ ВАНТАЖНИХ ШИН.....	40
3.1 Постановка завдання	40
3.2 Вимоги до інтернет-магазину	40
3.3 Проектування веб-сайту інтернет магазину з продажу вантажних шин	42
3.4 Архітектура інтернет-магазину вантажних шин	45
3.5 Програмна реалізація застосунку	50
3.6 Проектування інтерфейсу користувача	56
3.7 Висновки до розділу 3	65
4 ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ ТА АНАЛІЗ ХМАРНИХ ПЛАТФОРМ ДЛЯ ЙОГО РОЗГОРТАННЯ	66
4.1 Тестування функціональних можливостей додатку.....	66

4.2 Аналіз платформ, що надають послуги хмарних сервісів	71
4.3 Висновки до розділу 4	76
ВИСНОВКИ.....	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78

ВСТУП

Актуальність теми

У наш непростий час, коли у світі пандемія «Covid 19» та війна в Україні все більше зростає потреба купівлі та продажу будь-яких товарів (послуг) в Інтернет-магазинах, не контактуючи особисто. Сайти стали масовими, майже кожна фірма має свою сторінку в Інтернеті.

У сьогоднішньому світі є велика кількість Інтернет-магазинів різного роду спеціалізації, наприклад: Lamoda, Aliexpress, Amazon, Ebay. В даний час є величезний вибір Інтернет-магазинів, наприклад: магазини техніки, магазини одягу, спортивні магазини, вузькоспеціалізовані магазини та інші. Для залучення клієнтів Інтернет-магазини організовують чималу кількість акцій та здійснюють великі рекламні компанії. І за вдалої рекламної компанії вони залишаються не у мінусі, а навпаки в плюсі.

Електронна комерція, як і будь яка система має відповідні технології. Функціонал електронного магазину повинен відповідати реальному магазину, тобто в ньому буде реалізовано такі ж самі основні функції: перелік товарів або послуг, доступних для покупця, можливість обробки замовлень, продаж та доставка товарів.

Таким чином, завдання розробки Інтернет-магазину стає актуальним, оскільки:

- це дозволяє швидко подавати інформацію широкому колу осіб;
- наявний зворотний зв'язок з клієнтами;
- є можливість організації маркетингових досліджень та проводити рекламу, а також залучати нових покупців і клієнтів.

Мета роботи

Метою кваліфікаційної роботи є дослідження існуючих фреймворків на мові Python та розробка Інтернет-магазину на мові Python з використанням

фреймворку Django, що буде використовуватися для надання послуг в сфері обслуговування та продажу вантажних шин.

В ході дослідження треба вирішити набір завдань, а саме:

- 1) вивчити поняття веб-сайту;
- 2) визначити принцип роботи інтернет-магазинів;
- 3) проаналізувати існуючі системи та платформ, які використовуються для створення веб-сайтів;
- 4) розробити інтернет-магазин з продажу вантажних шин.

Об'єкт дослідження

Існуючі фреймворки на мові Python для розробки веб-сайтів.

Предмет дослідження

Веб-сайт інтернет магазину вантажних шин.

Методи дослідження

Аналіз існуючих фреймворків на мові Python для розробки веб-сайтів.
Синтез результатів дослідження фреймворків.

Наукова новизна

Наукова новизна дослідження полягає у виділенні і систематизації методики розробки веб-сайту інтернет-магазину вантажних шин.

Практичне значення одержаних результатів

Спроектовано і реалізовано інтернет-магазин вантажних шин виходячи з аналізу існуючих рішень та регіональних вимог. Виконано тестування розробленого проекту, проаналізовано варіанти розгортання проекту в хмарних середовищах та розгорнуто і протестовано проєкт на одному з хмарних сервісів

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗРОБКИ ВЕБ-САЙТІВ ІНТЕРНЕТ МАГАЗИНІВ

1.1 Електронна комерція

Можна з впевненістю сказати, що головною подією 20 століття у галузі управління економіки є виникнення комп'ютера та його карколомний вплив на зміни засобів передачі, накоплення та переробки інформації. Починаючи з 70-х років бізнес у всьому світі відчуває нові тенденції керування виробництвом, взаємодії між усіма суб'єктами економічного процесу, управління та комунікації між гравцями ринку. Працівники та замовники можуть знаходитись у різних країнах та навіть на різних континентах, ніколи не зустрічатися віч на віч, рекрутингові компанії працюють на усіх мовах світу, набираючи команди, які підходять до умов проекту. Новий рівень усіх цих процесів наступив з виникненням всесвітньої мережі, яка дозволила використовувати увесь комп'ютерний потенціал.

В Україні її електронна комерція активно розвивається з 90-х років минулого століття. Питанням формування і розвитку ринку електронної комерції в Україні присвячені роботи ряду вітчизняних авторів, зокрема Даныко Т., Яворської Н. Пильнової В., Гавриш О., Синявської О. та інших.

Досліджуючи роботу [2], порівняно окремі аспекти традиційної торгівлі та інтернет-торгівлі у сучасному стані. Автори акцентують увагу на головних недоліках та перевагах, які мають підприємства за рахунок впровадження електронної торгівлі як окремого виду електронної комерції. Проведено аналіз поширених видів електронної комерції B2B («Business to Business») та B2C («Business to Consumer») — бізнес для споживача, який саме і отримав широке розповсюдження в Україні [2].

Розглянувши особливості реалізації товару за допомогою Інтернету у роботі [3] доведена перспективність застосування інформаційних і телекомунікаційних технологій у різних сферах. Автори роботи говорять:

«проведений аналіз виявив ключові переваги Інтернет-магазину, такі як: низькі витрати на рекламу, ефективне представлення споживачеві будь-яких товарів; цілодобове функціонування; підвищений рівень сервісу; автоматизація та інтеграція фінансових звітів; висока швидкість відновлення даних про реалізований товар». У роботі подано вимоги до розробки сучасного Інтернет-магазину.

У статті [4] автор досліджує стан та перспективи розвитку електронної торгівлі в Україні. Описано тенденції у розвитку електронної торгівлі та її роль у сучасному бізнесі, розглянуто проблеми, пов'язані з розвитком електронної торгівлі в Україні, такі як відсутність розвинутої інфраструктури та недостатня свідомість споживачів про переваги онлайн-торгівлі.

На сьогоднішній день електронна комерція — це широкий спектр послуг, починаючи з доступу до карткових рахунків (Приват24) і закінчуючи електронними біржами, на яких можна купити автомобіль у іншій частині світу. Це беззаперечно зручно, оскільки як клієнти, так і продавці отримують потужний інструмент для бізнесу. До найбільш цікавих з можливостей, на яких хотілося б зупинитись детальніше, відносяться:

- клієнти можуть самостійно реєструватися у електронному магазині або у постачальника у зручний час. Клієнт може на власний розсуд створювати, коригувати та видаляти замовлення, обирати метод оплати та доставки. Продавець або постачальник має змогу розраховувати замовлення та миттєво отримувати інформацію про попит на продукт та проводити цільову рекламу;
- замовлення товару через інтернет дозволяє подивитись на його зовнішній вигляд, характеристики та відгуки інших покупців. Це заощаджує час під час пошуку необхідного товару чи послуги. Крім того, це дозволяє зробити порівняльний аналіз у різних постачальників.
- усі замовлення потрапляють до електронної бази, у якій за допомогою людини або навіть штучного інтелекту (боту) можна перевірити

кількість доступного товару, побудувати маршрути доставки, необхідний транспорт та ін.

- можливість здійснювати оплату різними способами. Це може бути як оплата банківськими картами з дотриманням рівня безпеки та ідентифікації, оплата кур'єру, оплата накладним платежем або системами типу Visa.

Ні для кого не є таємницею, що кількість людей, які створюють покупки через електронні засоби збільшується, як і зростає рівень продаж у великих та маленьких інтернет-магазинах (рівень покупок у обласних центрах вже перевершує чверть, кожен десятий українець робить усі свої покупки через мережу).

Відома продуктова українська ІТ-компанія EVO, яка є власником найбільших у країні Інтернет-магазинів, зазначає, що за 2022 рік в українському сегменті e-commerce здійснено угоди на чотири мільярди \$, при цьому прес-служба визначає зростання більше ніж на третину у порівнянні з минулим роком. Кількість замовлень зростає (див. Рис.1), і не будучи навіть оптимістом, можна зробити прогноз про зростання світових покупок у Інтернеті до 2040 року приблизно до ста відсотків.

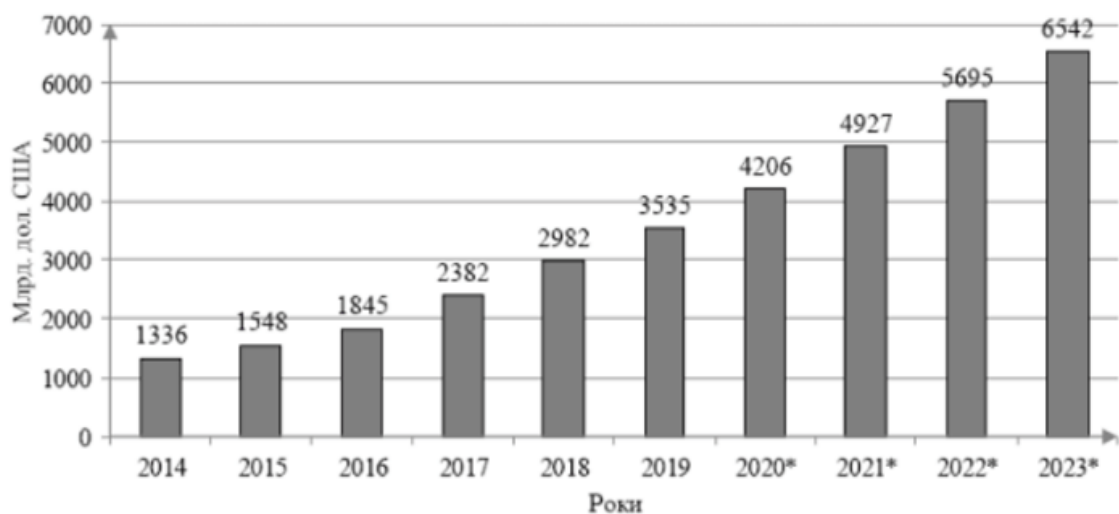


Рисунок 1 — Динаміка світових обсягів електронного роздрібного продажу станом на період 2020-2023р.

Інтернет-торгівля дозволяє постачальникам та споживачам у рівній мері використовувати нові можливості, до яких можна віднести наступне:

- Глобальний вибір та присутність: Оскільки інтернет глобальний, інтернет-торгівля дозволяє навіть найдрібнішим постачальникам досягати глобальної присутності та займатися бізнесом у світовому масштабі. Також споживачі отримують глобальний вибір із усіх потенційних постачальників, які пропонують свої товари чи послуги, незалежно від їх географічне розташування.

- Підвищення конкуренції та якість обслуговування: Інтернет-торгівля дозволяє постачальникам конкурувати між собою, стаючи ближче до споживачеві. Відповідно, споживач отримує збільшену якість обслуговування.

- Більш детальна інформація про попит споживачів, персоналізація товарів та послуг: Використовуючи електронну взаємодію, компанії можуть отримувати докладну інформацію про запити кожного споживача та автоматично надавати послуги та продукти, що відповідають їх вимогам.

- Скорочення доставки товару до споживача, швидка реакція на попит: Інтернет-торгівля суттєво дозволяє покращити товарні потоки. Товари доставляються безпосередньо від постачальника до споживача.

- Зменшення витрат, зниження цін.

Підкреслимо загальні проблеми розробки. Коли людина фізично відвідує торговельний комплекс, вона має можливість оцінити товар візуально та визначити його якісні характеристики. В електронній торгівлі він не має такої можливості. Найчастіше візуальної інформації достатньо, однак є проблема з доставкою товарів.

Щорічно кількість Інтернет-магазинів збільшується, оскільки це дійсно прибуткове та зручно для покупця, а також економить час на покупку та бюджет клієнта. Безперечною перевагою електронної комерції є можливість виконувати продажі у будь-який час та економити не тільки на продавцях, а й на обслуговуючому персоналі (прибиральниці, охорона та ін). Крім того, можна продавати ще не завезений товар, тобто раціонально планувати закупки

та зекономити на логістиці та базах зберігання. Клієнти можуть підключатися з будь якої точки країни, а можливо й із за кордону, якщо зробити якісну рекламу, підібрати правильну цільову аудиторію і, звісно, надавати товари чи послуги високої якості. Крім того, великі існуючі торгівельні мережі (Розетка, Епіцентр) працюють з іншими продавцями, що дозволяє як користуватися їхніми рекламними площами, так і спрощує логістику та доставку.

Існує кілька загально визнаних категорій, на які підрозділяється електронна комерція. Як правило, таке розмежування проводиться за цільовою групою споживачів [26]:

- B2B («business-to-business») — бізнес для бізнесу [26];
- B2C («business-to-consumer») — бізнес, орієнтований на кінцевого фізичного споживача; [26]
- C2C («consumer-to-consumer») — бізнес, що забезпечує взаємодію між великою кількістю фізичних споживачів; [26]
- C2B («consumer-to-business») — системи обробки цінових заявок, за якими споживачі хотіли б придбати товари та послуги; [26]
- B2A («business-to-administration») — бізнес, що оперує угодами між приватними компаніями та урядовими організаціями (Адміністрацією); [26]
- C2A («consumer-to-administration») — бізнес, побудований на організації взаємовідносин фізичних осіб та державних служб. [26]

У нашому випадку використовується модель B2C (business-to-consumer).

Як говорять автори роботи [1]: «Електронний бізнес, і зокрема інтернет-магазин як його складова, стає більш конкурентним, що у свою чергу позитивно позначається на рівні якості послуг і пропонованих товарів» [1].

1.2 Аналіз основних типів інтернет-магазинів

Створюючи мультибреднові магазини користувач переслідує різні цілі.

На сайті таких інтернет магазинів можна побачити одразу кілька різних напрямів. Відповідаючи на питання чому з'являються такі мультибрендові магазини, акцентується увага на тому що потрібно уникнути ризики, пов'язані з тим, що не всі види товарів можуть бути корисними для майбутніх покупців.

Плануючи створити такого типу інтернет магазини базуючись при цьому тільки на власні інтереси та вподобання, замовники допускають частіше за все допускають помилку. Проведені маркетингові дослідження показали, що такий підхід є не найкращою ідеєю, та може зашкодити подальшому просуванню продукту на ринку. Окрім того, майбутні користувачі інтернет магазину можуть мати власну думку стосовно того, щоб вони хотіли побачити на сторінках такого додатку.

Серед значної кількості різноманітних торгових напрямків можна виділити два основних типи торгового майданчика:

1. Єдиний сайт із однаковим інтерфейсом для різних груп товарів.
2. Сайт із кількох розділів, кожен з яких є орієнтованим на конкретного покупця послуг, а також має інтерфейс, який адаптований під конкретні завдання.

Тип 1. Єдиний («шаблонний») не є орієнтованим на конкретну користувацьку аудиторію, в ньому не має виділених індивідуальних особливостей та характеристик. Шаблонними сайтами користуються найчастіше оптові компанії, оскільки вони працюють з досвідченими закупниками, які і так добре знаються на товарі. Головним для них є зручність швидкого вибору товару з наявного асортименту та легкого заповнення відповідних форм для подальшої відправки менеджерам інтернет-магазину.

Тип 2. Будучи орієнтованим на конкретну цільову аудиторію спеціалізований інтернет-магазин, архітектура якого спроектована під певні групи товарів. У разі створення сайту інтернет-магазину, що є орієнтованим на роздрібну торгівлю, обирають саме такий тип.

На початку ініціації ідеї розробки проєкту, бізнес-аналітику або власнику продукту треба визначити, саме для кого буде призначено цей

продукт. На етапі складання технічного завдання або специфікації формується портрет майбутнього користувача, який складається з певних відомостей про нього. Також не менш важливим є бачення повної картини, щодо всіх класів користувачів додатку, що розробляється.

1. Характеристики користувача сайту. Описуючи клієнта треба врахувати такі параметри, як: стать, вік, орієнтований дохід, професія, ресурси через які клієнт зможе дізнатися про можливості сайту та буде в подальшому з ним взаємодіяти.

2. Мотиви відвідувача. Оскільки у кожного користувача є не тільки свої потреби але і власна мета щодо відвідування того чи іншого сайту треба зробити так, щоб він здійснив покупку саме в даному інтернет-магазині.

1.3 Огляд видів веб-сайтів

Структура будь якого сайту складається з набору веб-сторінок, логічно пов'язаних між собою та доступних через головну сторінку, за допомогою будь якого браузеру. В представленій літературі класифікації веб-сайтів розрізняють перш за все за їх розміром. На невеликих сайтах міститься коротка інформація про саму компанію та її послуги, в той час як на великих подається детальний опис товарів, характеристики, зображення або фото та ціна.

Виділяють наступні види веб-сайтів:

Бізнес-сайт

Бізнес-сайт — це цифрова присутність компанії. Він представляє бізнес для відвідувачів. Головною метою бізнес сайтів, незалежно від їх форм та розміру, є представлення бізнесу нових продуктів або послуг, так щоб залучити нових партнерів та клієнтів. На сторінках бізнес сайту подається опис пропозицій та способів, які спонукають зробити одразу покупку через веб-сайт або зв'язатися з командою, щоб ініціювати продаж або партнерство.

Веб-сайт електронної комерції

Веб-сайти електронної комерції продають товари, або фізичні товари або цифровий контент. Користувачі таких сайтів є потенційними покупцями представлених товарів, вони можуть не тільки передивлятися перелік та знайомитися з інформацією про продукти, але і здійснювати покупку безпосередньо на самому сайті. Також як і на бізнес-сайтах, блогах та інших типів веб-сайтів, веб-сайти електронної комерції може розміщуватись інтернет магазин для продажу товарів або послуг. Подача інформації на сайтах електронної комерції відповідає загальній моделі, а саме: продукти класифікуються та подаються у форматі списку категорій, сайт перенаправляє користувача на сторінку обраного продукту. Окрім того відвідувачам доступна панель пошуку. На кожній сторінці продукту користувачі мають можливість додати товар до «кошика» та продовжити подальший пошук. Після закінчення формування свого кошику користувачі переходять до процесу оформлення замовлення, додаючи інформацію про доставку та спосіб оплати.

Блог

Формат блогу (скорочення від «веб-блог») є способом, за допомогою якого кожен міг публікувати випадковий, докладний письмовий контент про свої інтереси. Майже кожна компанія та підприємство починає вести свій власний блог щоб заявити про себе в інтернеті, тому зараз прийнято вважати, що блог це будь-який веб-сайт, що публікує письмовий контент у вигляді статей за певною тематикою.

Сайт новин

Новинні веб-сайти схожі на розважальні сайти, але в основному містять новини. Таким чином, ці сайти більше інформують, ніж розважають. Новинні веб-сайти також, як правило, мають помітно іншу естетику, ніж розважальні веб-сайти, часто з більш чистим макетом та естетикою.

Веб-сайт некомерційної організації

Веб-сайти є одним з кращих способів встановити легітимність бізнесу. Ці типи веб-сайтів служать для просування організації, інформування про цілі організації, а також часто запитують і надсилають пожертвування. Веб-сайт некомерційної організації повинен чітко передавати свою місію та цілі з головної сторінки з емоційною вагою, а додаткові сторінки повинні містити більш докладну інформацію про окремі проекти та ініціативи.

Портфоліо

Онлайн-портфоліо — відмінний спосіб для креативних фрілансерів та агентств представити свою роботу. Незалежно від спеціалізації (живопис, ілюстрація, кіно, фотографія, графічний дизайн, скульптура, проза або поезія) веб-сайт-портфоліо демонструє всі творчі здібності. Портфоліо також підходить для професіоналів, не пов'язаних з мистецтвом: наприклад, програмісти можуть створювати портфоліо для своїх проектів з програмування, а маркетологи можуть використовувати портфоліо для опису своїх найбільш ефективних компаній.

Персональний сайт

Персональний сайт — представлення особистості. Він використовується як засіб для вираження себе та своїх думок у письмовій формі, проєктах, візуальних матеріалах або засобах масової інформації, якщо вони відображають особисту точку зору людини. Мета особистого сайту залежить від особистості — якщо він використовується для просування своєї кар'єри, то на ньому публікується резюме з детальним описом власного досвіду роботи.

Форум

Форуми — це онлайн-майданчики, де люди можуть обговорювати те, що їх цікавить. Вони складаються з дошок оголошень на різні теми та модераторів, щоб все залишалось цивілізовано.

База знань

У разі, якщо займатися бізнесом, можливо посперечатися, що у клієнтів виникнуть питання. Щоб вийти вперед, можливо створити базу знань, веб-

сайт, який містить документацію та відповіді на поширені питання про ваш продукт або послугу. Бази знань сортують свої ресурси підтримки в ієрархії, що спрощує навігацію та пошук того, що потрібно користувачам. Бази знань, як правило, також доступні для пошуку — чим менше перешкод між вами та бажаним ресурсом, тим краще, крім того, ви можете аналізувати пошукові запити ваших відвідувачів та змінювати свій продукт відповідно до їх потреб. Все це працює на зменшення кількості запитів у службу підтримки та забезпечує безперебійну роботу.

1.4 Огляд основних методів створення сайтів

Існує кілька способів, які використовують для розробки веб-сайтів. Кожен з яких має свої переваги та недоліки. Розглянемо найбільш поширені з них.

Конструктор сайтів

Конструктор сайтів надає набір інструментів, що дозволяє навіть не маючи спеціальні знання створити та адмініструвати сайт. Конструктор дозволить легко обрати як тип сайту (сайт візитівка, інтернет-магазин), так і його шаблон (готовий дизайн сайту), окрім того це можуть бути модулі які дозволять розширити функціональність сайту. Одні з самих популярних конструкторів сайтів це: Wix, HOSTiQ та багато інших.

Основні переваги конструкторів сайтів:

- Простота у використанні. Не потрібно певних знань у веб-розробках, все робиться покроково і легко.
- Готові шаблони дизайну. Шаблон можна легко і легко замінити у пару нажатій.
- Можливість зекономити як час так і кошти.

До недоліків конструкторів сайтів можна віднести наступне:

- Наявність реклами. Зайва реклама на таких сайтах заважає і як наслідок це відштовхує користувача додатку та зменшує прибуток сайту, тому що це може бути сама реклама конкурентів.
- Обмежено дисковий простір. Багато безкоштовних конструктори виділяють дуже мало місця на жорсткому диску, яке, згодом закінчується, і доводиться обмежувати себе у поповненні будь-яких файлів.
- Наявність обмеженого функціонала. Набір функціональних можливостей конструктора є обмеженим та стандартним.
- Відсутність індивідуальних рішень. Якщо потрібно внести які небудь не індивідуальні рішення на сайт, то цього зробити не вийде.

Готові CMS системи

CMS (Content management system) — Система управління вмістом. Коли йде мова про готові CMS, це означає, що вона є готовою до релізу, та можна додати до неї різні модулі та шаблони дизайну. Розробник повинен тільки встановити цю систему на веб-сервер, обрати тему оформлення, а в подальшому наповнити даними і ми будемо мати вже готовий сайт. Плануючи дизайн сайту використовують готові шаблони, їх можна редагувати під свої потреби змінюючи HTML та стилі. У кожній CMS системи є панель Адміністрування з відповідним доступом. Така панель дозволяє змінити структуру сайту використати вбудовані модулі. Найбільш популярними є наступні системи управління вмістом: Joomla, Wordpress, Drupal, Бітрікс.

До переваг готових CMS систем можна віднести:

- Готовий продукт. Найбільшою перевагою готових CMS систем: є те, що не потрібно нічого програмувати з нуля і все вже готове до використання. Наприклад, якщо треба швидко запустити свій власний сайт, або інтернет магазин то вже є готове рішення.
- Простота встановлення. Готову CMS систему дуже легко та швидко встановити на веб-сервер. Процес установки виглядає як розміщення готової CMS на веб-сервер, після якого потрібно інстальювати програмне

забезпечення, до якого відносять підключення до бази даних, ім'я сервера, логін, пароль (у разі потреби). Далі CMS сама створить відповідні таблиці у базі даних та заповнить їх необхідними даними. Завершивши цей процес можна приступити до наповнення та оформлення нового сайту.

- Функціональність. У готових CMS системах доволі великий функціонал та просунута адміністративна панель. Така різноманітність функціоналу пояснюється необхідністю вирішення багатої кількості завдань.
- Популярність. Оскільки користувач може знайти в інтернеті рішення доволі широкого кола проблем, готові CMS системи користуються великою популярністю. Розробляючи нові теми існуючі програмісти працюють саме з популярними CMS системами.

До недоліків готових CMS систем можна віднести наступне:

- Надмірна функціональність. Зазвичай готові CMS системи реалізують доволі велику кількість функціоналу, що зазвичай приводить до збільшення часу для формування веб-сторінки та використання невиправдано великої бази даних. За даними статистичних досліджень показано, що для реалізації поставлених завдань, використовується лише 30% функціоналу готової CMS, а інші 70% не використовується взагалі.
- Низька безпека. Популярність готових CMS систем робить доступним для кожного вихідний код. Наявність в інтернет просторі публікацій щодо того, як «зламати» Joomla або Wordpress створює суттєві проблеми з безпекою.
- Складність. Використання готові CMS системи ускладнюється наявністю великої кількості налаштувань, частина яких не завжди є потрібною. містять багато потрібних та непотрібних налаштувань, які ускладнюють використання готової CMS системи.

Самописна CMS система

Самописні CMS системи створюються командою розробників або веб студією. Серед причин розробки такої CMS системи є бажання долучити зацікавлених осіб до конкретної веб студії або розробити власний продукт, який да деякими показниками буде кращий ніж готові CMS системи.

Виділяють два типи самописних CMS систем:

- повністю розроблені з нуля (не враховуючи деякі деталі та інструменти);
- розроблені на основі готових фреймворків та інших веб-інструментів.

До переваг самописних CMS систем можна віднести:

- Швидкість роботи. Оскільки є відсутнім непотрібний функціонал та немає потреби створювати системи під конкретні умови, самописні CMS системи зазвичай відрізняються високою швидкістю роботи.
- Вирішення нестандартних завдань. Оригінальна ідея клієнтів значно швидше та простіше реалізується за допомогою CMS власної розробки.
- Безпека. Вихідним кодом володіють власне лише розробники, що забезпечує високий рівень безпеки.

До недоліків самописних CMS можна віднести:

- Прив'язка до програміста. Якщо рівень розробника доволі низький то йому буде складно розібратися у коді, що був створений певним програмістом. Однак, якщо при розробці системи використовувався не власний фреймворк та програміст дотримувався стандартів розробки програмного забезпечення з подальшим використанням не повинно бути проблем.
- Тривала розробка. Розробка власних модулів та функцій потребує значно більше ресурсів.

Інтернет-магазин вантажних шин розроблений на самописній CMS системі, яка створена на основі фреймворку Django.

1.5 Аналіз ресурсів аналогічної тематики

Аналіз успішних інтернет-магазинів показав, що їх успіх залежить від таких показників [5]:

- наскільки якісно інтернет-магазин відповідає вимогам користувача та технічним характеристикам;
- наскільки успішно ведеться бізнес.

Розглянемо приклади інтернет-магазинів, що займається реалізацією шин та комплектуючих. В ході проведеного аналізу було обрано декілька сайтів зі схожим направленням. Тир Шина вважається магазином шин №1 в Україні. Головна сторінка сайту вантажних шин «ТИР-шина» представлена на рисунку 2.

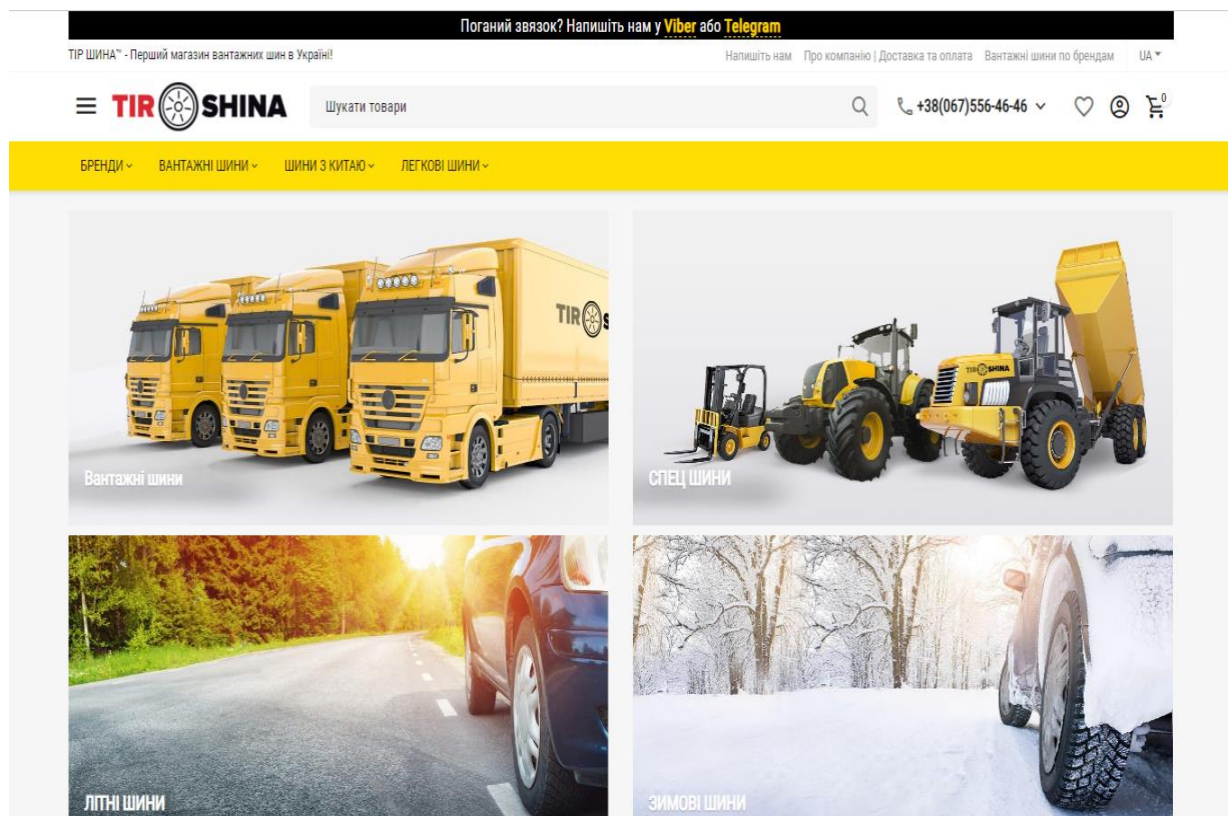


Рисунок 2 — Головна сторінка сайту вантажних шин «ТИР-шина»

Додаток надає допомогу у підборі та купівлі вантажної гуми, проводить професійну консультацію щодо вибору таких товарів, як шини для вантажних

автомобілів та автобусів. До недоліків цього сайту можна віднести те, що у цього сайту повністю відсутній функціонал. Відвідувач сайту ТИР-шина не може обрати за характеристиками шини, може тільки скачати прайс, що вважається дуже незручним у пошуку необхідної шини за конкретними характеристиками. Для покупців відсутня система покупки через корзину, також вони не мають можливості переглянути рисунки протекторів, які їх цікавлять.

Сайт інтернет-магазину «Shiny & diski» надає користувачеві повний спектр послуг, функціонал даного сайту реалізовано на кваліфікаційному рівні. Головну сторінку сайту вантажних шин «Shiny & diski» представлено на рисунку 3.

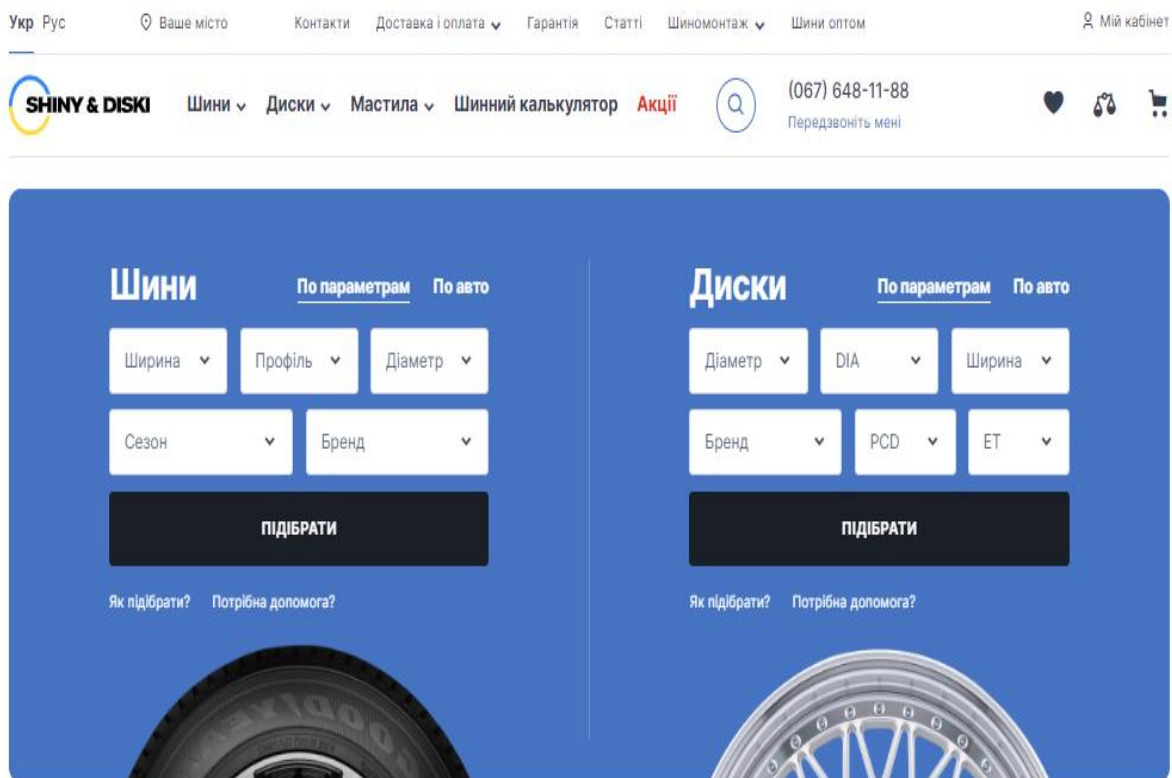


Рисунок 3 — Головна сторінка сайту вантажних шин «Shiny & diski»

Перевагою даного сайту є досить зручний інтерфейс, на головній сторінці одразу можна обрати необхідні характеристики які шукає клієнт

(розмір, сезонність, бренди); реалізовано мій кабінет та корзина; користувач має можливість порівнювати товари; реалізовано опис товарів з усіма характеристиками; користувач може порівняти протектори шин.

Однією з особливостей даного сайту є наявність шинного калькулятора, що надає можливість самостійно розрахувати нові шини в залежності від моделі та модифікації авто. Сторінка «Шинний калькулятор» сайту «Shiny & diskі» представлена на рисунку 4.

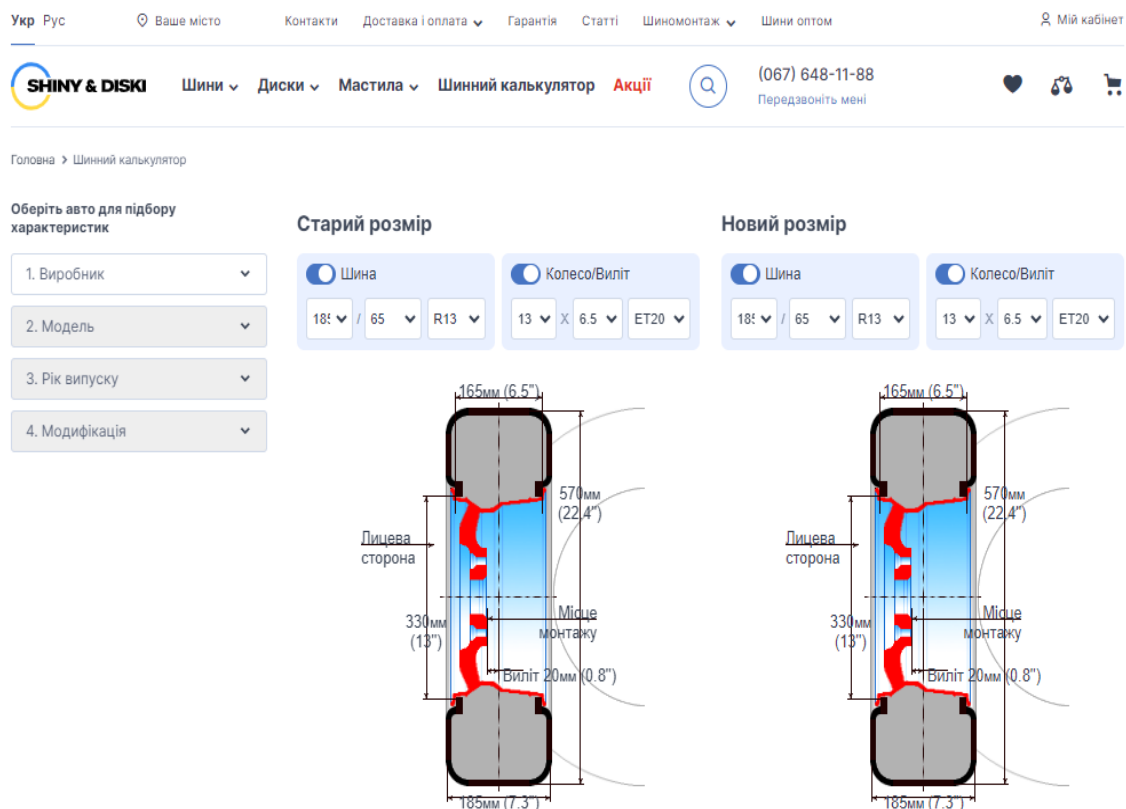


Рисунок 4 — Сторінка «Шинний калькулятор» сайту «Shiny & diskі»

Інтернет-магазин Технооптторг трейд пропонує широкий вибір шин для всіх видів техніки. Головну сторінку сайту інтернет-магазину «Технооптторг трейд» представлено на рисунку 5. У даному сайті непоганий дизайн, але є недоліки, такі як неможливість купувати з корзини, тому що вона не передбачена серед функціоналу. Окрім того, відвідувач сайту не бачить цін на сайті, що вважається дуже незручним для користувача.

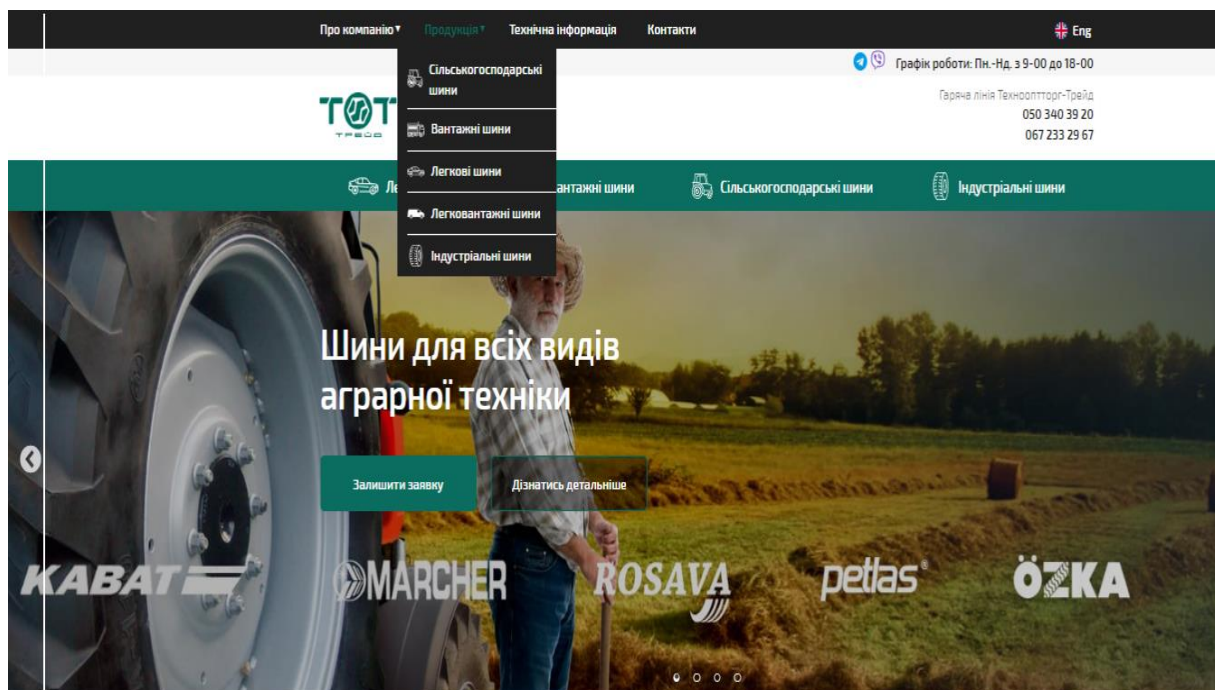


Рисунок 5 — Головна сторінка сайту «Технооптторг трейд»

Розглядаючи додатки схожої тематики виділено повнофункціональний сайт компанії Tyres International Group, що веде свою діяльність на території всього Європейського Союзу. Головна сторінка сайту представлена на рисунку 6.

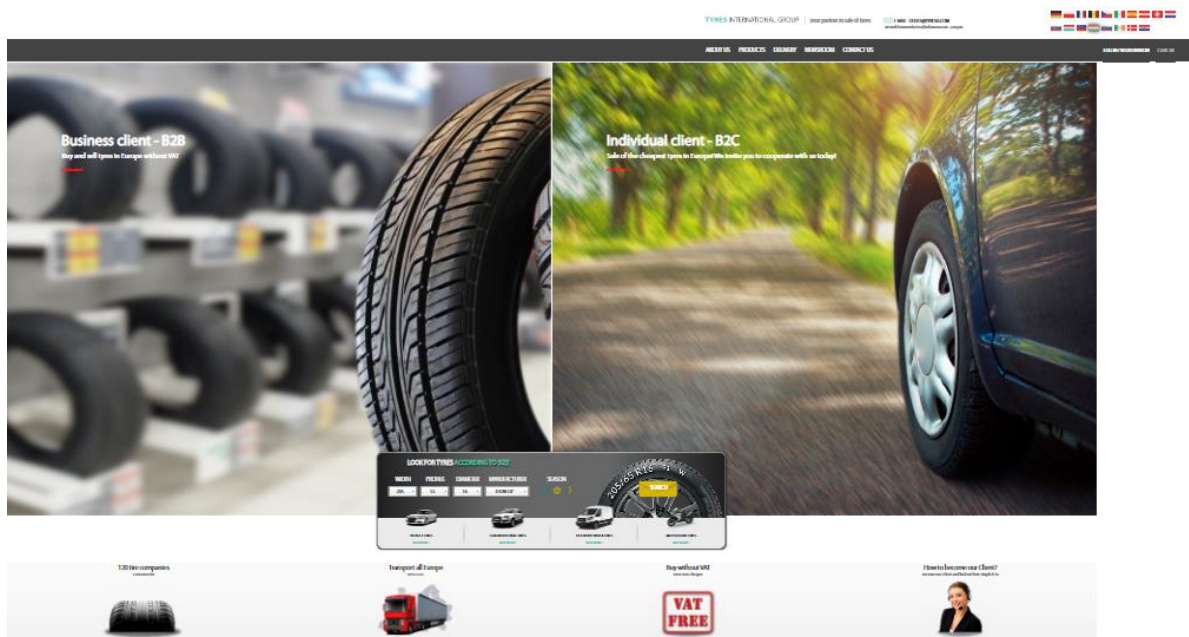


Рисунок 6 — Головна сторінка сайту «Tyres International Group»

Пропонуючи комплексні рішення як для індивідуальних так і для бізнес-клієнтів користувачам одразу надається інформація про товари, що є найбільш затребуваними на ринку на даний період часу. Сторінка сайту «Tyres International Group» з інформацією про актуальні товари представлена на рисунку 7.

RECOMMENDED PRODUCTS

Brand	Model	Size	Fuel Economy	Wet Grip	Noise	Price	Availability	Quantity
GOODYEAR	EFFICIENTGRIP ROF SUMMERTIRE	255/40 R18 95 W	E	B	69	210.38€	✓	▼
NOKIAN	ILINE SUMMERTIRE	185/60 R14 82 T	C	C	67	56.70€	✓	▼
BRIDGESTONE	DUELER HP SPORT SUMMERTIRE	255/50 R19 103 W	E	C	72	178.16€	✓	▼
MICHELIN	LATITUDE SPORT 3 SUMMERTIRE	295/35 R21 107 Y	C	A	72	250.25€	✓	▼

Рисунок 7 — Сторінка сайту «Tyres International Group» з інформацією про актуальні товари

Обрані для розгляду та описані інтернет-магазини мають одну суттєву перевагу. Вони є максимально орієнтовані на свою цільову аудиторію (про що свідчить і навігація, і дизайн, і функціонал), а також мають вони мають структуру зручну для SEO-просування, що дуже важливо для подальшого просування додатку на ринку.

1.6 Висновки до розділу 1

В першому розділі кваліфікаційної роботи було проведено аналіз основної термінології об'єкта дослідження. Наведено основні елементи програмного забезпечення для створення сайту. Наведено основні приклади веб-сайтів в інтернеті та їх особливості. Приведено приклади реальних сайтів, розглянуто їх архітектури на прикладі інтернет-магазинів: ТИР-шина, Shiny & diskі, Технооптторг трейд.

2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ЗАСТОСУНКІВ

2.1 Вибір інструментів реалізації

З аналізу специфікації додатку який розробляється зрозуміло, що сайт буде мати клієнт-серверну архітектуру. Огляд існуючих веб-сайтів схожої тематики показав, що в цілому виділяють два типи сайтів, а саме статичні та динамічні. Розробляючи статичні сайти використовують зазвичай зв'язок HTML-сторінок та CSS-таблиці стилів. При цьому не враховується поведінка користувача та немає можливості змінювати вміст сторінки в залежності від зміни умов.

HTML

HTML — це засіб розмітки тексту, яка включає кількість елементів за допомогою яких можна задовільнити вподобання користувача.

CSS

CSS — мова таблиць каскадних стилів, яка дозволяє розширити можливості оформлення веб-сторінок додатку. Розробники використовують таблиці стилів для того, щоб змінити зовнішній вигляд веб сторінок. Це дозволить підкреслити індивідуальність застосунку.

До переваг використання розмітки тексту та таблиці каскадних стилів відносять наступне:

- Форматування тексту логічне за допомогою HTML коду, тобто, він задає структури веб-сторінки: розташування та порядок проходження абзаців, графічних зображень та рядків [18].
- Таблиці стилів CSS формують тексти фізично, вони представляють веб-сторінки, виділяючи шрифти для звичайного тексту абзаців, колір для заголовків, наявність рамки у таблицях.

При організації структури проєкту враховується те, що визначення стилів CSS виносяться в окремі файли. В HTML одночасно присутні теги

логічного та фізичного форматування веб-сторінок. Каскадні таблиці стилів за своєю суттю не динамічні. Вони дозволяють визначати, як виглядатиме документ при завантаженні.

CMS

Як зазначено авторами публікації [27]: «Система керування вмістом (контентом) (англ. Content management system, CMS) — інформаційна система або комп'ютерна програма, що використовується для забезпечення та організації спільного процесу створення, редагування та управління контентом (тобто вмістом)» [27].

Серед основних функцій CMS виділяють:

- доступ до інструментів для створення вмісту, можливість організувати спільну роботу з вмістом;
- керування вмістом: зберігання, контроль версій, дотримання режиму доступу;
- можливість публікації вмісту;
- зручне для навігації та пошуку даних подання інформації.

Процес створення динамічних сайтів поділяють на два підходи. Використання графічних конструкторів сайтів дозволяє створювати сайти людям, які не мають будь-яких знань та умінь у веб-розробці, але вони мають функціонал з обмеженим набором шаблонів, які надає конструктор. Отже, сайт створений за допомогою конструктора втрачає такий важливий аспект як унікальність продукту. Цього можна уникнути, використавши другий підхід — розробка сайту з використанням бібліотек динамічного вмісту. Для створення таких сайтів використовуються спеціальні мови програмування, такі як: PHP, Ruby, Python, JavaScript.

Ruby

Ruby це динамічна, рефлексивна, інтерпретована високорівнева мова програмування. Він має незалежну від операційної системи реалізацію багатопоточності, сувору динамічну типізацію і просунутий збирач сміття [6]. Ruby як правило використовується як скриптова мова. Для створення сайтів та

веб-застосунків за його допомогою використовується програмна платформа Ruby on Rails. Одним з головних недоліків цієї мови програмування є невелика кількість готових, вільно розповсюджуваних бібліотек розширення, що суттєво збільшує терміни розробки типових рішень.

PHP

Як зазначено авторами роботи [7]: «PHP — скриптова мова загального призначення, що інтенсивно застосовується для розробки сайтів та веб-додатків». Специфіка PHP дозволяє поєднувати його та HTML розмітку сторінки. Завдяки простому синтаксису, він є лідером на ринку рішень для розробки веб-додатків [7]. Однак, PHP має ряд серйозних недоліків. Одним із головних недоліків є синтаксична нагромадженість. При порівняно простому синтаксисі, в екосистемі мови присутня велика кількість функцій, які виконують одну і ту ж дію, при цьому мають різні назви та різний порядок параметрів, що передаються.

JavaScript

Мова програмування JavaScript підтримує декілька стилей, а саме: об'єктно-орієнтований, імперативний та функціональний. Популярність мови JavaScript пов'язана з її широкими можливостями щодо взаємодії з елементами веб-сторінки без додаткового перезавантаження. Це дасть можливість приховувати та показувати фрагменти дизайну, переміщувати їх та змінювати оформлення. Шляхом таких дій можна створювати презентаційні ефекти, меню, невеликі ігри, обробляти дані форм та керувати вмістом.

- Підтримка шарів. Шари є фрагментами HTML, які можна розміщувати на веб-сторінці шляхом накладання їх один на одного з точністю до пікселя. Шари все більше набувають визнання як засіб верстки веб-сторінок та створення різних ефектів. Частково це пов'язано з тим, що їх параметри легко змінюються динамічно через скрипти, що дає можливість створювати меню та інші виразні засоби дизайну.

- Робота із формами. Через скрипти зручно отримувати та обробляти будь-які дані форм, це дозволяє перевірити інформацію на правильність введення перед її надсиланням на сервер.
- Робота із зображеннями. Через скрипти можна робити попереднє завантаження зображень. Ще до прямого звернення до зображень, браузер поміщає їх у свою пам'ять, щоб у міру необхідності швидше відобразити у документі.
- Створення нових вікон. Скрипти дозволяють створювати нові вікна, задавати у цих вікон вид відображення та формувати їх вміст.
- Математичні функції. JavaScript містить усі необхідні арифметичні операції, підтримує всі стандартні математичні функції, як із цілими числами, і з плаваючою точкою.

Автори публікації [8] говорячи про практичне застосування JavaScript акцентують увагу на тому, що: «JavaScript використовується як вбудована мова для програмного доступу до об'єктів додатків». «Найбільш широке застосування вона знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам» [8]. Однак завдяки таким фреймворкам, як NodeJS, AngularJS, BackboneJS є можливість створювати цією мовою високопродуктивні веб-додатки. Однак, для створення проектів з декількома видами представлення інформації JavaScript не підходить через складність архітектури проекту, що отримується.

Python

В основній документації до фреймворку Django вказано: «Python — високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника та читання коду» [9]. Маючи мінимальний синтаксис ядра, Python має досить потужну стандартну бібліотеку, яка включає великий обсяг корисних функцій [9]. Для створення веб-додатків та сайтів на Python використовуються спеціальні програмні платформи.

Як мову програмування для створення серверної частини сайту обрано Python, тому вибір програмної платформи обмежений продуктами, які використовують цю мову.

2.2 Аналіз Python-фреймворків

Фреймворк — інструмент, який полегшує процес написання та використання веб-додатків. Змістом Python фреймворків є набори інструментів, бібліотек, що без сумніву будуть корисні для розробки логічно розподілених модулів, в яких представлена реалізація завдань, з якими найчастіше придется зустрітися при розробці, що полегшує роботу. Розробник буде мати можливість зосередитися на логіці самого проєкту, не витрачаючи при цьому час на однотипні дії.

Існуючи на теперішній час фреймворки для Python поділяються за типами на фулстекові, мікрофреймворки та асинхронні. До універсальних монолітних платформ, які містять інструменти для повного життєвого циклу розробки програмного забезпечення відноситься фулстек. Оскільки, асинхронні та мікрофреймворки не мають достатньої кількості додаткових шаблонів і модулів, то їх використання вимагає більше зусиль для структурування коду.

Фреймворк Web2py

Фулстек веб-платформа для розробки веб-додатків, яка базується на шаблоні проєктування MVC. Наряду з тим, що він містить всі необхідні компоненти для реалізації проєкту згідно фазам життєвого циклу проєкту, також самостійно формує SQL-запити до баз даних та не вимагає від розробника знань SQL. На відміну від інших цей фреймворк для мови Python не потрібно встановлювати і конфігурувати, він може використовуватися на будь-яких архітектурах, сумісних з Python.

Фреймворк Bottle

Згідно офіційної документації: «Мікрофреймворк розроблявся для створення програмних інтерфейсів, підходить для прототипів, невеликих

додатків, які можна реалізувати одним вихідним файлом» [28]. Даний фреймворк залежить від стандартної бібліотеки Python.

Фреймворк CherryPy

Згідно офіційної документації [29]: «CherryPy поставляється з мінімальною кількістю готових компонентів і призначений для виконання основних завдань фреймворку» [29]. У веб-середовище вбудовано власний багатопотоковий високошвидкісний сервер WSGI, що дозволяє здійснювати кешування, обробку сесій, завантаження файлів. До основних особливостей CherryPy відносять: «здатність працювати поверх інших фреймворків, можливість одночасного запуску декількох HTTP-серверів, потужні інструменти для кодування, аутентифікацію» [29]. Фреймворк характеризується високою швидкістю роботи, простотою розгортання, продуктивністю.

Найпопулярнішими платформами з підтримкою Python є Django Framework та Flask.

Flask

Програмна платформа для створення сайтів мовою програмування Python Відноситься до мінімалістичних рішень, являє собою каркас веб-додатку, що надає лише базові можливості. Він містить наступні можливості: сервер обробки і налагодження додатків, систему модульного тестування, шаблонізатор, обробку RESTful запитів, підтримку безпечних cookie. Flask може бути інтегрований в будь-яку ORM. Підходить для створення нетривіальних проєктів, у яких необхідно відходити від базових принципів побудови веб-додатків. На цій програмній платформі створено такі проєкти, як Twillo, LinkedIn [10].

Django Framework

Вільно розповсюджувана програмна платформа для веб-застосунків мовою програмування Python, що використовує шаблон проектування Model-View-Controller. Структура сайту на Django складається з одного або декількох додатків, які або відокремлені, або підключаються. Це одна із суттєвих

архітектурних відмінностей цієї програмної платформи від Flask. Він підходить для створення проєктів будь-якої складності.

Виділяючи можливості ядра Django, можна сказати, що воно використовує ORM та API, крім того має систему кешування, а також простий менеджер URL, вбудовану систему аутентифікації та бібліотеки, що дозволяють працювати з формами. На цій програмній платформі створено такі проєкти, як Instagram, Pinterest, YouTube, Google [11].

2.3 Аналіз систем управління базами даних

Оскільки в рамках кваліфікаційної роботи розробляється динамічний сайт розглянемо засоби зберігання даних, а саме бази даних.

Система управління базами даних (СУБД) забезпечує реляційну модель роботи з даними. Сама модель представляє зв'язок між сутностями із різних таблиць. Для зберігання та подальшої роботи з даними такий тип СУБД має певні таблиці (сутності). Кожен стовпець такої таблиці містить дані різного типу, кожен запис в базі даних має значну кількість атрибутів (стовпців) та обов'язково має унікальний ключ.

Обираючи базу даних та відповідно СУБД аналізують заплановану кількість даних, що будуть зберігатися. Веб-сайт інтернет магазину з продажу вантажних шин не потребує великої кількості даних тому під його потреби підходить деякі засоби управління даних, а саме: MySQL, Microsoft SQL, PostgreSQL, SQLite.

Microsoft SQL Server

Згідно основної документації: «Microsoft SQL Server — система управління реляційними базами даних, розроблена корпорацією Microsoft» [15].. Істотним обмеженням цією СУБД для проєкту є обмеження безкоштовної версії MS SQL Express. Такі самі можливості, як і MS SQL, представляє СУБД MySQL. MySQL є рішенням для малих та середніх додатків. Гнучкість СУБД забезпечується підтримкою великої кількості типів таблиць. Продукт

розповсюджується під ліцензією GNU General Public License, яка передбачає передачу програмного забезпечення у громадську власність. Виходячи з цієї ліцензії, використовувати СУБД MySQL можна безкоштовно.

SQLite

SQLite — це база даних, що базується на файлах. Завдяки технологіям бібліотек SQLite є швидкою та потужною. На відміну від інших СУБД при роботі з цією системою звернення відбувається не до мережної інфраструктури у мережеских базах даних а безпосередньо до файлів.

До переваг SQLite можна віднести наступне :

- легке перенесення на різні типи машин, оскільки файлова структура (база даних) складається з одного файлу;

- відмінна при розробці та тестуванні — оскільки SQLite складається всього з одного файлу та бібліотеки що написана мовою C це дає змогу легко масштабувати додатки.

До недоліків SQLite можна віднести наступне:

- відсутність системи користувачів — дана СУБД зазвичай використовується у невеликих додатках, тому відсутність системи управління правами доступу користувачів не є критичною на відміну від більших за розміром додатків;

- відсутність можливості збільшення продуктивності — під час проектування складно зорієнтуватися як можна ще підвищити продуктивність.

Враховуючи переваги та недоліки використання даної СУБД можна зробити висновок, що робота з даними в ній є обмеженою, не повним є синтаксис запитів SQL відсутня система управління правами користувачів.

PostgreSQL

Згідно з офіційною документацією: «PostgreSQL — це вільно поширювана об'єктно-реляційна система управління базами даних» [16]. На відміну від інших СУБД розробники використовують PostgreSQL у проєктах з високим навантаженням на базу даних та необхідністю зберігати значну кількість даних. Але ця система управління має суттєвий недолік такий як продуктивність.

Виконуючи прості операції читання PostgreSQL зазвичай уповільнює сервер особливо в порівнянні з MySQL [16].

Аналіз представлених систем управління базами даних дозволив зробити висновок, що оптимальним рішенням для веб-сайту інтернет магазину з продажу вантажних шин буде саме SQLite. Виділені недоліки цієї СУБД не є критичними для даного продукту, оскільки для розробки використовується фреймворк Django, за допомогою якого контролюється система користувачів. Окрім того для функціонування будь якого інтернет магазину швидкість роботи не є головним критерієм. Визначені переваги SQLite а саме те, що вся база даних складається з одного файлу, не потребує встановлення різних сервісів та служб, що дасть змогу заощадити місце на хостингу.

2.4 Аналіз архітектури сучасних веб-додатків

Головною метою розробки будь яких інтернет-магазинів є задовільнити потреби бізнесу, тобто створений програмний продукт повинен бути конкуренто спроможним на ринку та відповідати потребам користувачів.

Для розробки інтернет-сайту обрали фреймворк Django який має архітектуру MVT (Model-View-Template). Використання Django дало можливість швидкої розробки а обрана архітектура MVT розподілила код на моделі, представлення та шаблони, що спростило подальше управління та підтримку проєкту.

Порівняльний аналіз MVT з іншими архітектурами, такими як MVC (Model-View-Controller) представлено на рисунку 8:

1. MVT vs MVC:

- MVT: У MVT в Django, вид (View) відповідає за обробку запитів та взаємодію з моделлю та шаблонами.
- MVC: У класичній архітектурі MVC, контролер (Controller) відповідає за обробку запитів та взаємодію з моделлю та

представленням. В MVT вид та контролер функціонально подібні, але вони розділені.

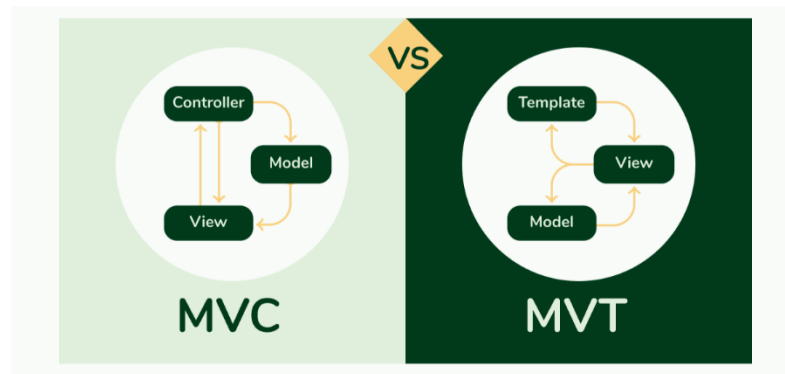


Рисунок 8 — Порівняльний аналіз MVT vs MVC

2. Простота та Продуктивність:

- MVT: MVT може бути сприйняті як спрощена версія MVC, що полегшує розробку та управління проектом.
- MVC: Деякі вважають, що MVT є простішою та ефективною архітектурою, особливо для веб-розробки.

3. Використання Шаблонів:

- MVT: Використання шаблонів дозволяє легко відокремити представлення від бізнес-логіки.
- MVC: У деяких варіаціях MVC, таких як ті, що використовуються в фронтенд-фреймворках, шаблонізація може бути менш прямолінійною.

4. Робота з даними:

- MVT: Django надає ORM для легкої роботи з базою даних та моделями.
- MVC: Робота з даними може вимагати більше налаштувань та коду у деяких фреймворках MVC.

5. Інтеграція та Готові Рішення:

- MVT: Django має вбудовані компоненти, такі як адміністративна панель, що полегшує інтеграцію та розробку.

- MVC: Залежно від конкретної реалізації MVC, може бути потрібно вибирати готові рішення окремо.

2.5 Висновки до розділу 2

З метою вибору інструментів реалізації були розглянуті різні платформи та технології для розробки сайтів інтернет-магазинів. При вивченні переваг та недоліків було вирішено, що у якості мови програмування для створення серверної частини сайту обрано мову програмування Python. В ході дослідження було проаналізовано велике коло Python-фреймворків та систем управління базами даних, які використовуються в аналогічних застосунках. У якості платформи для розробки додатку обрано Django Framework.

Розглянувши подані СУБД, можна зробити висновок, що оптимальним рішенням для програми, що розробляється, є SQLite.

За результатами дослідження архітектури зроблено висновок, що для проекту з розробки інтернет-магазину з продажу вантажних шин доцільно обрати архітектуру MVT. Загалом, дослідження архітектури MVT в Django підтверджує її ефективність, зручність та готовність до використання в різних типах веб-додатків.

3 РОЗРОБКА ВЕБ-САЙТУ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ ВАНТАЖНИХ ШИН

3.1 Постановка завдання

Отже, темою кваліфікаційної роботи є розробка Інтернет-магазину вантажних шин, який має на меті надати користувачам можливість здійснювати покупки онлайн та надавати послуги в сфері обслуговування та продажу вантажних шин. Додаток буде розроблено на основі одного з фреймворків мови Python, Django.

З метою досягнення поставленої задачі буде реалізовано ряд, серед яких можна виділити:

- 1) огляд понять веб-сайту;
- 2) визначити принцип роботи інтернет-магазинів;
- 3) порівняти системи та платформи для створення веб-сайтів різноманітної тематики;
- 4) розробити інтернет — магазин з продажу вантажних шин.

3.2 Вимоги до інтернет-магазину

Для забезпечення управління інтернет-магазином доцільно розмежувати права користувачів. Інтернет-магазин має два типи користувачів:

- 1) адміністратор;
- 2) користувач.

Адміністратор має можливість додавати, видаляти, змінювати товар і категорії товарів, вносити зміни в дизайн сайту, здійснювати перегляд замовлень, зокрема з частково відсутніми даними для доставки. Також адміністратор виконує управління контентом сайту.

Користувач переглядає сторінки інтернет-магазину за допомогою веб-браузера, обирає категорії товарів, переглядає інформацію про конкретний

товар, поміщає товар у кошик, переглядає кошик, здійснює пошук товару за назвою, оформлює замовлення на купівлю одного або декількох товарів.

У рамках процесу проектування розглянемо основні вимоги, що пред'являються до автоматизованої системи магазину. Функціонал інтернет-магазину буде включати наступні основні можливості:

1. Реєстрація та авторизація користувачів: користувачі зможуть створювати облікові записи, виконувати вхід у систему та управляти своїми особистими даними.

2. Перегляд каталогу товарів: користувачі зможуть переглядати доступні товари, переглядати деталі кожного товару та отримувати інформацію про ціни та наявність.

3. Додавання товарів до кошика: користувачі зможуть додавати товари до свого кошика для подальшої покупки.

4. Оформлення замовлення: користувачі зможуть заповнювати форму замовлення, вибирати способи доставки та оплати, а також підтверджувати свої замовлення.

5. Управління замовленнями: адміністратори магазину матимуть доступ до панелі адміністрування, де вони зможуть переглядати, редагувати та видаляти замовлення.

6. Корзина покупок: користувачі зможуть переглядати та редагувати вміст своєї кошика, видаляти товари або змінювати їх кількість перед оформленням замовлення.

Наведені функціональні вимоги описують усі ключові можливості, які включає веб-додаток Інтернет — магазину вантажних шин. Ці вимоги враховують всі ключові аспекти та деталі, необхідні для успішної роботи та задоволення потреб користувачів сайту.

До нефункціональних вимог можна віднести

1. Продуктивність:

а. Система повинна мати високу швидкодію в обробці запитів та відгуках на дії користувачів.

b. Час завантаження сторінок та відображення результатів пошуку повинен бути мінімальним.

2. Безпека:

a. Система повинна забезпечувати конфіденційність особистих даних користувачів, таких як інформація про їх пароль та контактні дані.

b. Механізми аутентифікації та авторизації повинні бути належним чином реалізовані, щоб унеможливити несанкціонованого доступу до системи.

3. Інтерфейс користувача:

a. Користувацький інтерфейс повинен бути зручним, інтуїтивно зрозумілим та привабливим для користувачів.

b. Вимоги до дизайну, навігації та взаємодії повинні бути враховані, забезпечуючи зручне використання системи.

Вище наведені основні вимоги до системи визначають ключові характеристики та функціональність, які мають бути вбудовані у програмний продукт. Ці вимоги створюють фундаментальну основу для розробки та реалізації системи, забезпечуючи її коректну роботу.

3.3 Проектування веб-сайту інтернет магазину з продажу вантажних шин

За існуючим стандартом визначенням проектування є процес визначення архітектури, компонентів, інтерфейсу та інших показників системи. Як говорять автори роботи [23]: «Результатом проектування є проект — цілісна сукупність моделей, властивостей або характеристик, описаних у формі, придатної для реалізації системи». В рамках проведеного дослідження з метою коректності проведення проектування зостосунку використовувалась методологія IDEF0. На відміну від інших, акцент робиться на підпорядкованості об'єктів. Використовуючи для моделювання нотації IDEF0

не тимчасову послідовність, або потік робіт, розглядаються логічні відносини між визначеними роботами. Отже, «функціональна модель IDEF0 — це набір блоків, кожен з яких є «чорною скринькою» з входами та виходами, управлінням та механізмами, які деталізуються (декомпозиуються) до необхідного рівня» [23]. Контекстна діаграма «Замовлення товару в Інтернет-магазині» представлена на рисунку 9.



Рисунок 9 — Контекстна діаграма «Замовлення товару в Інтернет-магазині»

В основі діаграм нотації IDEF0 лежить контекстна діаграма А-0, яка представляє систему, що розробляється, як цілісний процес, а потім «відображає взаємозв'язок системи з іншими зовнішніми об'єктами (системами, організаційними групами, зовнішніми сховищами даних тощо)» [23].

Аналіз проблеми та потреба в спрощенні взаємодії з клієнтами та майбутніми користувачами системи, а також для визначення необхідних характеристик системи сформовані вимоги до системи (бізнес-вимоги, вимоги користувачів, функціональні та нефункціональні). Функціональні вимоги досліджуються та формуються у вигляді моделі варіантів використання, складання яких здійснюється на основі аналізу вимог до інтернет-магазинів.

На рисунку 10 зображено діаграму варіантів використання у нотації UML [14].

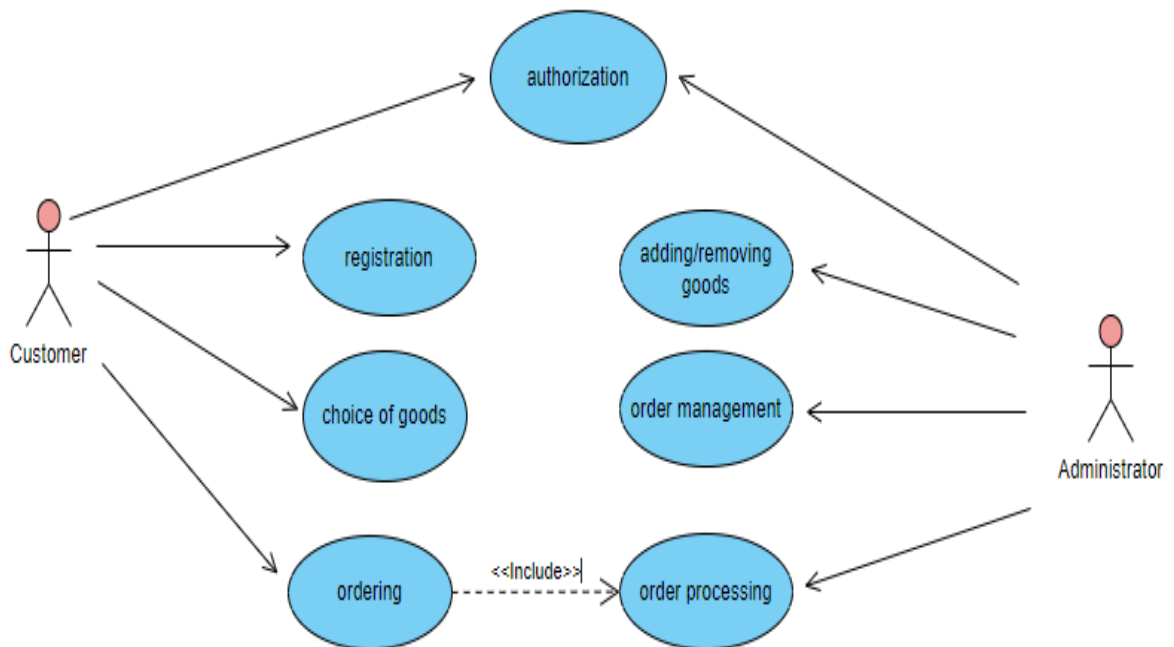


Рисунок 10 — Діаграма варіантів використання

Цей проєкт має на меті надати зручний та функціональний інтернет-магазин вантажних шин, який задовольнить потреби користувачів у покупках онлайн.

Діаграми послідовностей використовуються з метою уточнення діаграм прецедентів та можливості детального опису логіки сценаріїв використання. Вони дозволяють документувати проєкт з погляду сценаріїв використання,

визначають поведінкові аспекти системи. Діаграма послідовності відображає взаємодію об'єктів у динаміці, у часі. На діаграмі інформація має вигляд повідомлень, а взаємодія об'єктів передбачає обмін цими повідомленнями в рамках сценарію. На рисунку 11 зображено діаграму послідовності інтернет-магазину у нотації UML.

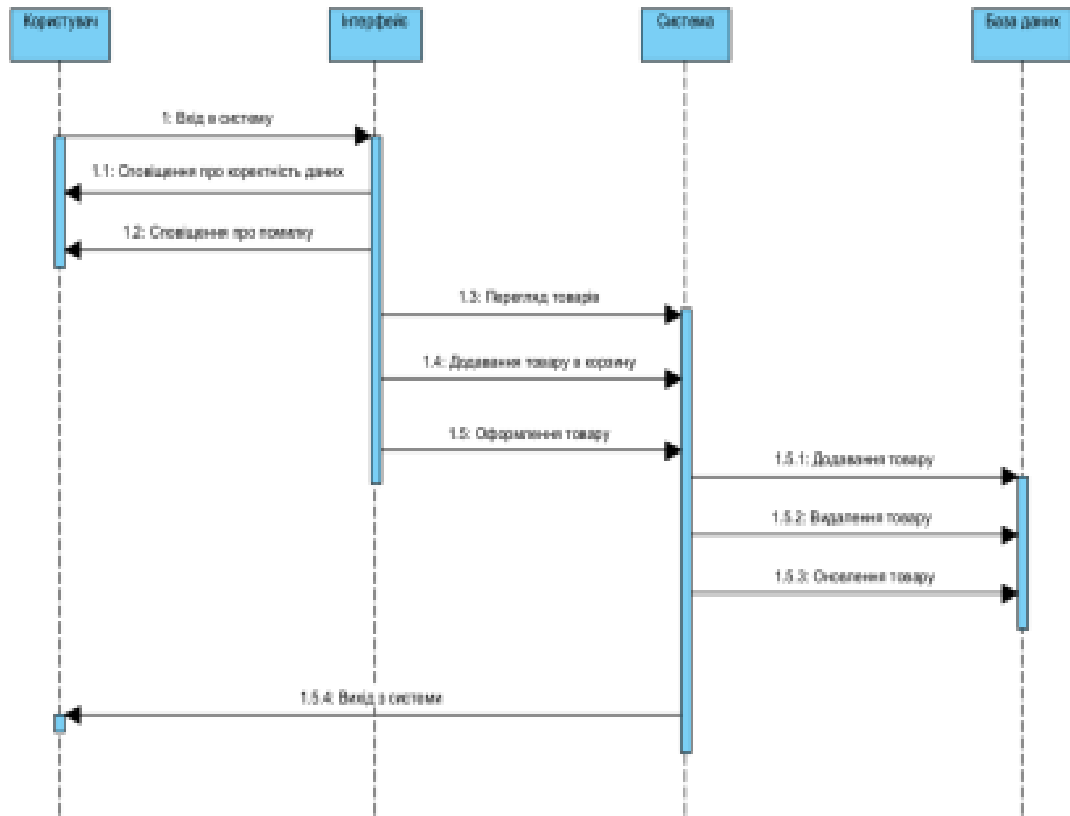


Рисунок 11 — Діаграма послідовності інтернет-магазину вантажних шин

3.4 Архітектура інтернет-магазину вантажних шин

Архітектура інтернет-магазину вантажних шин основана на трирівневій архітектурі, яка є однією з архітектури веб-додатків та має доволі широку популярність. Архітектура «Клієнт-сервер» будь-якого інтернет-магазину представлена на рисунку 12. Як видно, вона містить три рівня: Клієнт, Сервер та База даних.

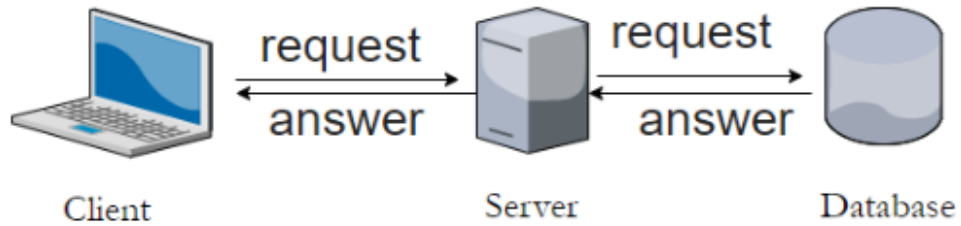


Рисунок 12 — Архітектура «Клієнт-сервер» інтернет-магазину

Застосування методу, що лежить в основі організації відносин між клієнтом і сервером, коли клієнт отримує лише результати своєї роботи, а сам веб-додаток розташований і виконується на сервері, дозволяє клієнтам не залежати від операційної системи конкретного користувача, роблячи тим самим веб-додатки міжплатформенними послугами [17].

Здійснивши аналіз предметної області та розглянувши всі можливі проблеми розроблено концептуальну модель бази даних. Це дозволить організувати чітку роботу з даними, проектуючи базу даних.

На етапі концептуального проектування проводиться збір, аналіз та формування вимог до даних. Здійснивши аналіз предметної області, визначено найважливіші сутності і атрибути, а також зв'язки між ними, одержано концептуальну модель, яка є симетрична до структури бази даних.

Предметна область інтернет-магазину ватажних шин є комерційним порталом для продажу шин та комплектуючих для вантажної техніки, мета якої спряти тому, щоб користувач міг легко знайти на сторінках сайту потрібний йому товар.

Згідно з поставленим завданням в рамках дипломної роботи було виділено сутності та спроектована ER-модель, на якій представлено опис логічного представлення концептуальної схеми бази даних інтернет-магазину вантажних шин.

База даних складається з чотирьох таблиць, які містять в собі відповідні сутності, між якими визначено взаємозв'язки. Кожна таблиця має первинний

ключ. На підставі опису предметної області та встановлених вимог, виділено наступні сутності.

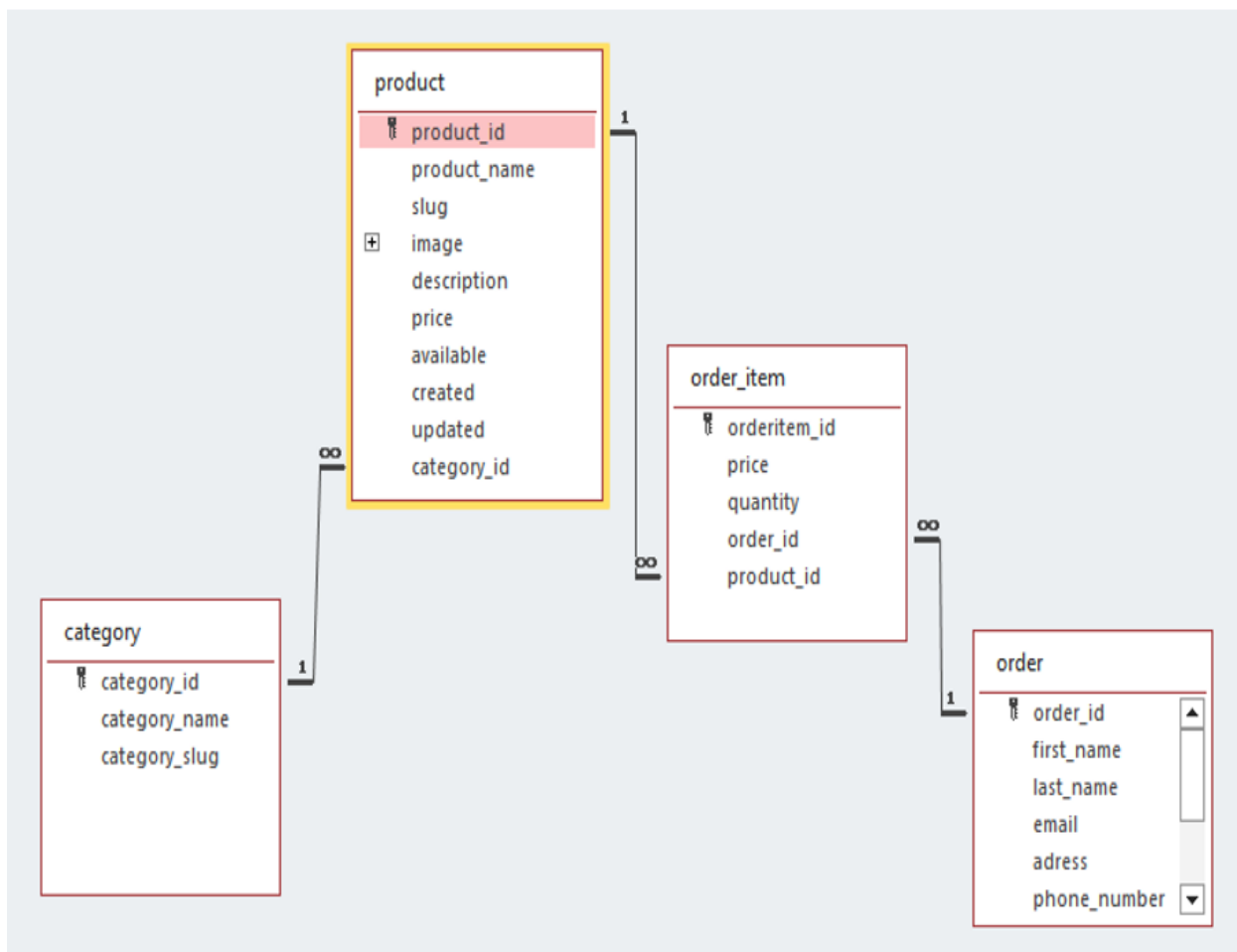


Рисунок 13 — Проектування ER — діаграми інтернет-магазину вантажних шин

Сутність «Category» визначає перелік категорій товарів, які пропонує інтернет магазин, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 1.

Сутність «Product» визначає товар, що поширює інтернет-магазин, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 2.

Таблиця 1 — Атрибути сутності «Category»

Назва	Тип	Зв'язок	Опис
Id		-	Первинний ключ
Category_name	varchar(200)	-	Назва категорії товарів
Category_slug	varchar(200)	-	Ідентифікатор, унікальність передбачає створення індексу

Таблиця 2 — Атрибути сутності «Product»

Назва	Тип	Зв'язок	Опис
Id		-	Первинний ключ
Category_id		Category	
Product_name	varchar(200)	-	Назва товару
Product_slug	varchar(200)	-	Склад товару для створення красивих URL-адрес
Image	varchar(250)	-	Опціональне зображення товару
Description	varchar(254)		Опціональний опис товару
Price	decimal(10)	-	Ціна за одиницю обраного товару
Available	bool	-	Значення, що вказує на наявність або відсутність товару
Created	date	-	Інформація про дату/час створення об'єкта
Updated	date	-	Дата/час оновлення інформації

Сутність «Product», в свою чергу, з'єднана із сутністю «Category» по типу «один до багатьох», оскільки товар належить одній категорії, а категорія містить кілька товарів.

Сутність «Order» містить інформацію про оформлення замовлення, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 3.

Таблиця 3 — Атрибути сутності «Order»

Назва	Тип	Зв'язок	Опис
Id		-	Первинний ключ
Last_name	varchar(50)	-	Прізвище
First_name	varchar(50)	-	Ім'я
Address	varchar(250)	-	Адреса замовника
Email	varchar(254)		Пошта замовника
Phone_number	varchar(10)	-	Телефон замовника
City	varchar(100)	-	Місто замовника
Created	date	-	Дата створення замовлення
Updated	date	-	Дата оновлення замовлення
Paid	bool	-	Статус сплати замовлення

Сутність «OrderItem» містить інформацію про товари, які користувач обрав та додав до корзини замовлення, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 4.

Таблиця 4 — Атрибути сутності «OrderItem»

Назва	Тип	Зв'язок	Опис
Id		-	Первинний ключ
Order_id		Order	
Product_id		Product	
Price	decimal(10)	-	Ціна за одиницю обраного товару
Quantity		-	Кількість одиниць товару в замовленні

Сутність «OrderItem», в свою чергу, з'єднана із сутністю «Order» по типу «один до багатьох», оскільки одне замовлення може містити декілька товарів.

Сутність «OrderItem», також, з'єднана із сутністю «Product» по типу «один до багатьох», оскільки в одному замовленні можуть бути різні товари.

3.5 Програмна реалізація застосунку

Результатом кваліфікаційної роботи є створення інтернет-магазину для продажу ватажних шин. Сайт повинен мати зрозумілий та зручний інтерфейс, а також для зручності подальшої підтримки та наповнення його даними мати достатній набір функціональних можливостей.

Користувачі сайту мають можливість переглядати каталог товарів і додавати товари у кошик. Також, вони можуть оформляти та розміщувати замовлення.

Каталог магазину складається із товарів, розділених на різні категорії. Кожен товар має ім'я, опціональний опис, опціональне зображення, ціну та наявність.

У разі успішного виконання створених сутностей в рамках структури додатка та згідно з логіки обраної архітектури створюється відповідний файл міграцій.

Процес успішного створення та заповнення сутностей бази даних проекту представлено на рисунку 14.

При реалізації подання додавання товарів до кошика або оновлення кількості існуючих товарів використовується декоратор `require_POST`, щоб дозволяти запити лише методом POST. Якщо форма валідна, то товар у кошику або додається, або оновлюється. Подання перенаправляє на URL-адресу `cart_detail`, яка відображає вміст кошика.

id	first name	last name	email	address	phone number	city	created	updated	paid
1	Костянтин	Лещенко	leshenkokos@gmail.com	Проспект Соборний	+38098000000	Запоріжжя, Запорізька обл.	2023-10-14 13:04:03.414866	2023-10-14 13:04:03.414866	0
2	Костянтин	Лещенко	ztirshinaservice@gmail.com	Софіївська	+38098000002	Запоріжжя, Запорізька обл.	2023-10-14 13:05:42.545402	2023-10-14 13:05:42.545402	0
3	Костянтин	Лещенко	ztirshinaservice@gmail.com	Проспект Соборний	+38098000002	Запоріжжя	2023-10-20 15:06:53.729252	2023-10-20 15:06:53.729252	0
4	Костянтин	Лещенко	lizakostyakostya@gmail.com	Задніпровська	+38098000000	Запоріжжя, Запорізька обл.	2023-11-05 21:51:23.893879	2023-11-05 21:51:23.893879	0
5	Костянтин	Лещенко	leshenkokos@gmail.com	Проспект Соборний	+380971112233	Запоріжжя, Запорізька обл.	2023-11-26 10:31:17.363969	2023-11-26 10:31:17.363969	0

Рисунок 14 — Заповнення сутностей бази даних проекту

Для того, щоб відобразити каталог товарів на сторінці, необхідно створити подання переліку списку всіх товарів або фільтрації товарів за заданою категорією. Опціональний параметр `category_slug` використовується для додаткової фільтрації товарів за заданою категорією. Також подано вилучення та відображення одного товару. У лістингу 1 представлено фрагмент коду щодо реалізації формування уявлень каталогу товарів.

Лістинг 1 Формування уявлень каталогу

```
def about(request):
    return render(request, 'tyres/about.html',
{'title': 'Про нас'})

def montag(request):
    return render(request, 'tyres/montag.html',
{'title': 'Вантажний шиномонтаж'})

def payment(request):
```

```

        return render(request, 'tyres/payment.html',
{'title': 'Оплата та доставка'})

def contact(request):
    return render(request, 'tyres/contact.html',
{'title': 'Зворотній зв'язок'})

```

Після розробки подань списку товарів та детальної інформації про товар для них визначено шаблони URL-адрес, що наведено в лістингу 2.

Лістинг 2 Шаблони URL-адрес для каталогу товарів

```

from django.urls import path
from . import views
app_name = 'tyres'
urlpatterns = [
    path('', views.product_list, name='product_list'),
    path('about/', views.about, name='about'),
    path('montag/', views.montag, name='montag'),
    path('payment/', views.payment, name='payment'),
    path('contact/', views.contact, name='contact'),
]

```

Це шаблони URL-адрес для каталогу товарів. Для представлення *product_list* визначено два різні шаблони URL-адреси: шаблон з ім'ям *product_list*, який викликає подання *product_list* без будь-яких параметрів, та шаблон з ім'ям *product_list_by_category*, який передає подання параметр *category_slug*, щоб фільтрувати товари відповідно до заданої категорії. Для представлення *product_detail* додано шаблон, який передає до нього параметри *id* та *slug*, щоб витягувати конкретний товар.

Створення форм за допомогою фреймворку Django дозволяє визначати їх функціональність в одному місці, а використовувати ці форми багаторазово в різних місцях. Це суттєво спрощує валідацію даних та допомагає пов'язувати


```
class Meta:
    model = User
```

В структурі проекту Django є окремий пакет з шаблонами кожного додатку окремо, що дозволить легко переносити ці додатку в інші схожі проекти. Приклад заповнення шаблону для здійснення оплати наведено у лістингу 4.

Лістинг 4 Заповнення шаблону для здійснення оплати

```
<h1 class="Oplata"></h1>
  <h1 class="Title">Оплата</h1>
  <hr>
  <h3 class="title title-small">Готівкою</h3>
  <p>Оплата готівкою можлива при отриманні товару в
м.Київ, Харків, Запоріжжя у пунктах
самовивозу</p>
  <h3 class="title title-small">Безготівковим
розрахунком</h3>
  <p>Оплата за безготівковим розрахунком здійснюється
наступним чином: після оформлення замовлення, менеджер
магазину
      електронною поштою висилає Вам рахунок-фактуру
(з ПДВ або без ПДВ), який Ви зможете оплатити в касі
відділення будь-якого банку або
      з розрахункового рахунку Вашої фірми.</p>
  <h3 class="title title-small">Оплата карткою</h3>
  <p>Оплата на карту приват банку без процентів.</p>
```

Приклад заповнення базового шаблону наведено у лістингу 5.

Лістинг 5 Заповнення базового шаблону

```

<title>{% block title %}Шиномаркет{% endblock
%}</title>
    <link href="{% static "css/base.css" %}"
rel="stylesheet">
    <link href="{% static "css/bootstrap.css" %}"
rel="stylesheet">
</head>
<body>

<div id="header">
    <div class="headermenu">

        <a href="/" class="logo">Шиномаркет</a>

        <a href="{% url 'tyres:montag' %}"
class="menu">Вантажний шиномонтаж</a>
        <a href="{% url 'tyres:about' %}"
class="menu">Про нас</a>
        <a href="{% url 'tyres:payment' %}"
class="menu">Оплата та доставка</a>
        <a href="{% url 'tyres:contact' %}"
class="menu">Звороній зв`язок</a>

        {% if not user.is_authenticated %}

        <a href="{% url 'reg' %}" class="reg"><button
class="btn btn-outline-primary">Реєстрація</button></a>
        <a href="{% url 'user' %}" class="log"><button
class="btn btn-outline-primary">Увійти</button></a>

```

```

    {% else %}
    <a href="{% url 'profile' %}"
class="reg"><button class="btn btn-outline-
primary">Профіль</button></a>
    <a href="{% url 'exit' %}" class="log"><button
class="btn btn-outline-primary">Вийти</button></a>

    {% endif %}

```

3.6 Проектування інтерфейсу користувача

Взаємодія користувача додатка з комп'ютером здійснюється через інтерфейс. Якщо його розроблено у відповідності до загальних вимог до інтерфейсів додатків користувач не буде приділяти йому багато уваги, а зосередиться на основному функціоналі. Відвідувачі, як правило, приходять на сайт з певною метою, та їх цікавить швидкий та зручний доступ до необхідної інформації або товарів, тому важливо організувати простір так, щоб врахувати всі аспекти сприйняття інформації та потреби користувачів. Розглянемо найбільш важливі принципи, що слід враховувати, розробляючи інтерфейс користувача додатку.

Ясність — це найголовніша мета будь-якого інтерфейсу. Для того, щоб ефективно використати інтерфейс додатку, який розробляється, треба мати повне розуміння як саме буде здійснюватися взаємодія з іншими об'єктами.

Функціональність. При тривалому спілкуванні з інтернет-простором у користувачів складається якийсь стереотип, що є певною напрацьованою системою умовних інстинктів. Тому зовнішній вид елементів дизайну має відповідати уявленням користувача. Заходячи на сайт, користувач вже заздалегідь припускає, в якій частині сторінки буде знаходитись та інформація, яка йому потрібна, або відповідний елемент меню. Тобто, розробляючи інтерфейс додатку треба роботи його інтуїтивно зрозумілим для користувача.

Розглянуті особливості та принципи побудови користувацьких інтерфейсів враховані при розробці дизайну автоматизованої системи з продажу вантажних шин. Головна сторінка інтернет-магазину має зацікавлювати (викликати бажання залишитися на сайті) та спонукати до покупки.

Потрапляючи на головну сторінку сайту користувач додатку має можливість ознайомитися з переліком категорій. Інтерфейс головної сторінки сайту представлено на рисунку 15.

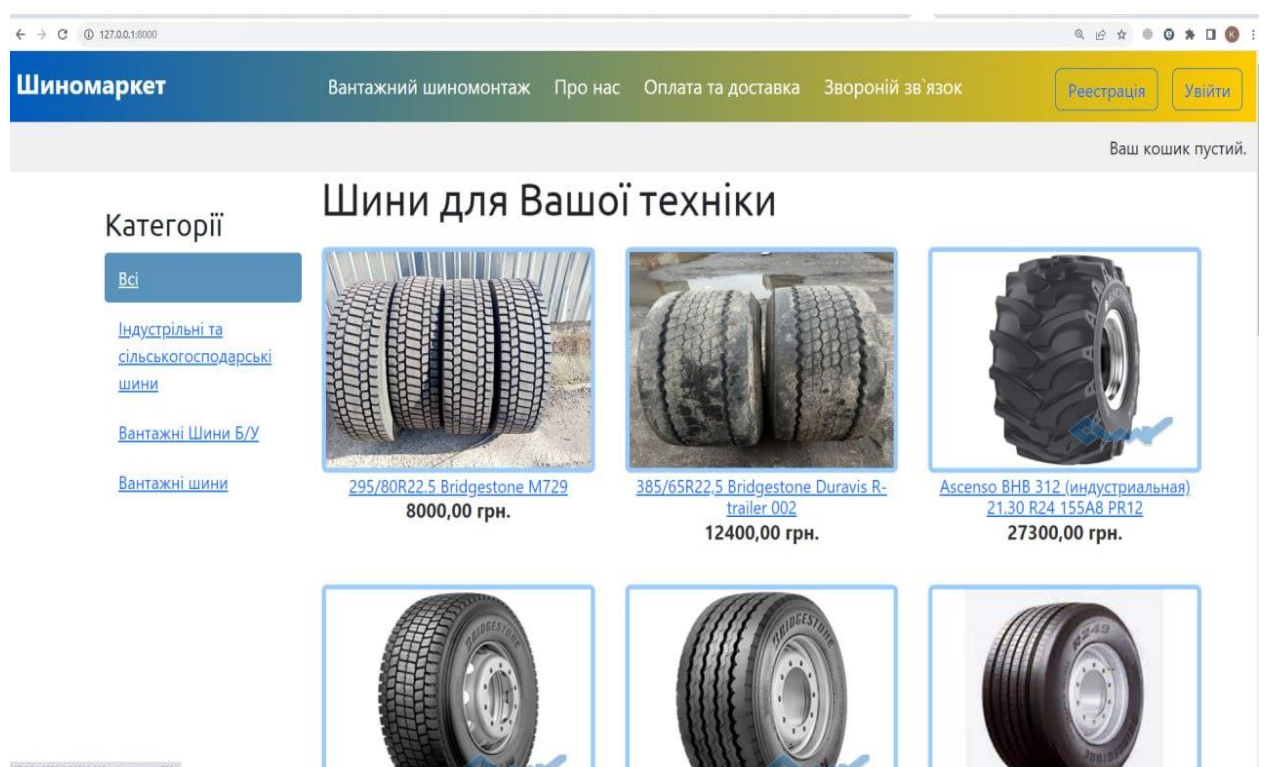


Рисунок 15 — Інтерфейс головної сторінки

Обравши ту чи іншу категорію товару, користувач має можливість обрати товар та отримати його опис та характеристики. Інтерфейс сторінки сайту з описом товарів, доступних для покупки за обраною категорією представлено на рисунках 16-17.

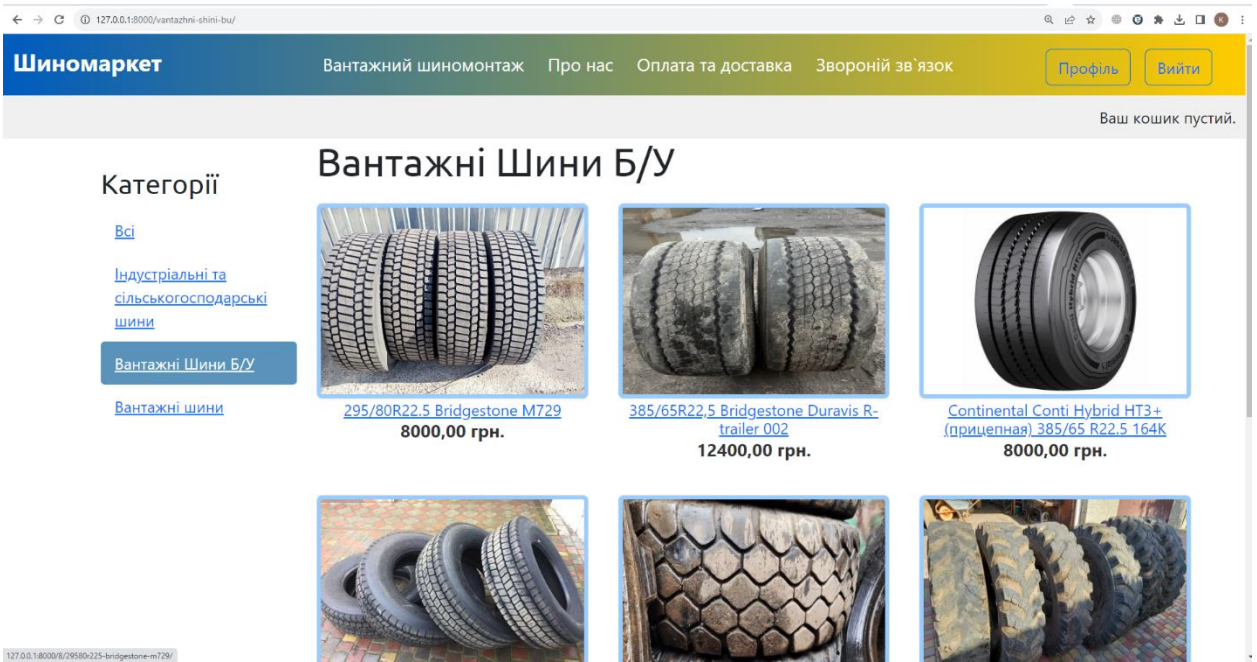


Рисунок 16 — Доступні для покупки товари інтернет магазину вантажних шин

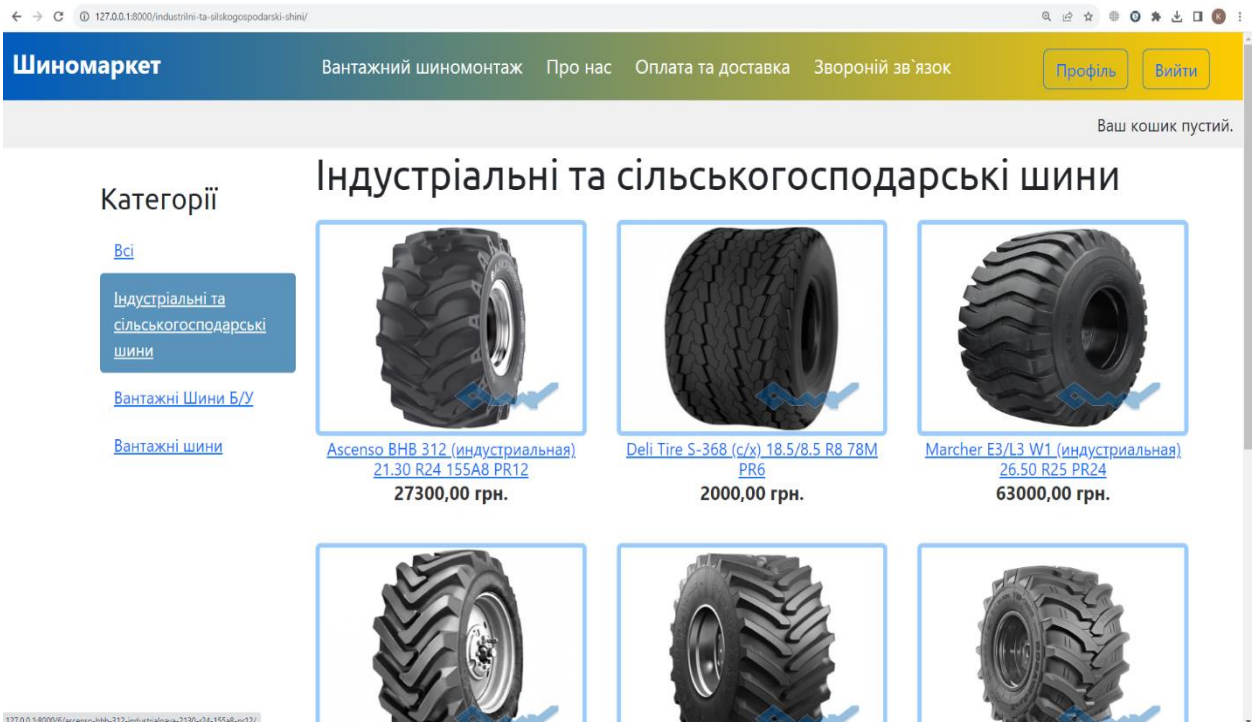


Рисунок 17 — Доступні для покупки товари інтернет магазину за обраною категорією товарів

Представлення сторінки сайту з рекомендованими товарами подано на рисунку 18.

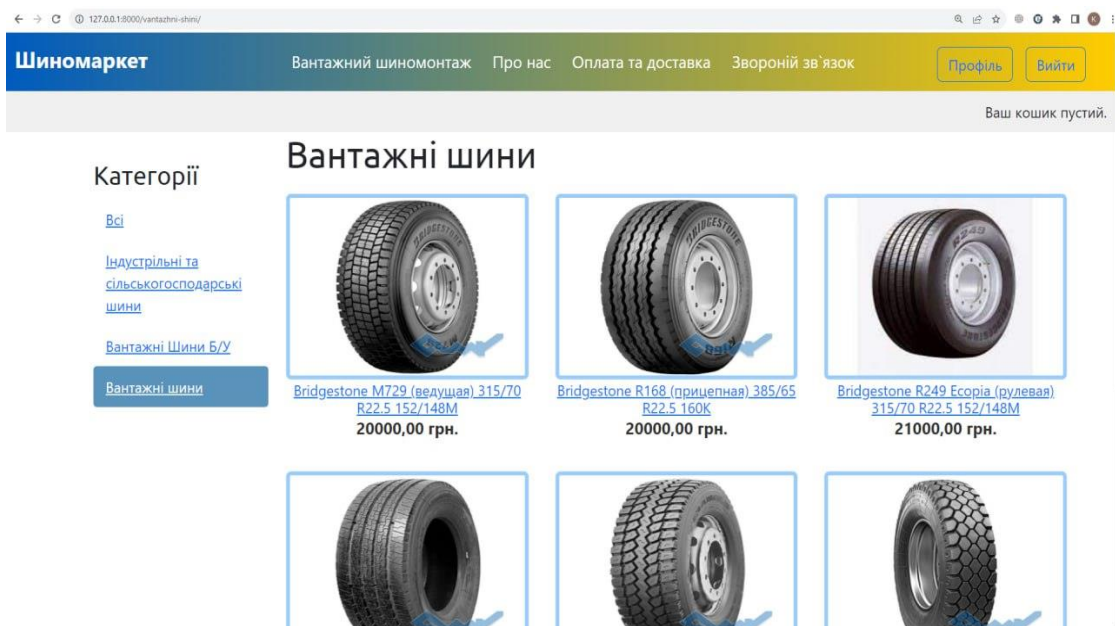


Рисунок 18 — Рекомендовані товари на головній сторінці

У разі, якщо користувач хоче здійснити покупку він повинен зареєструватися на сайті, вказав мінімальну інформацію про себе. Інтерфейс сторінки реєстрації користувача на сайті інтернет-магазину вантажних шин представлено на рисунку 19.

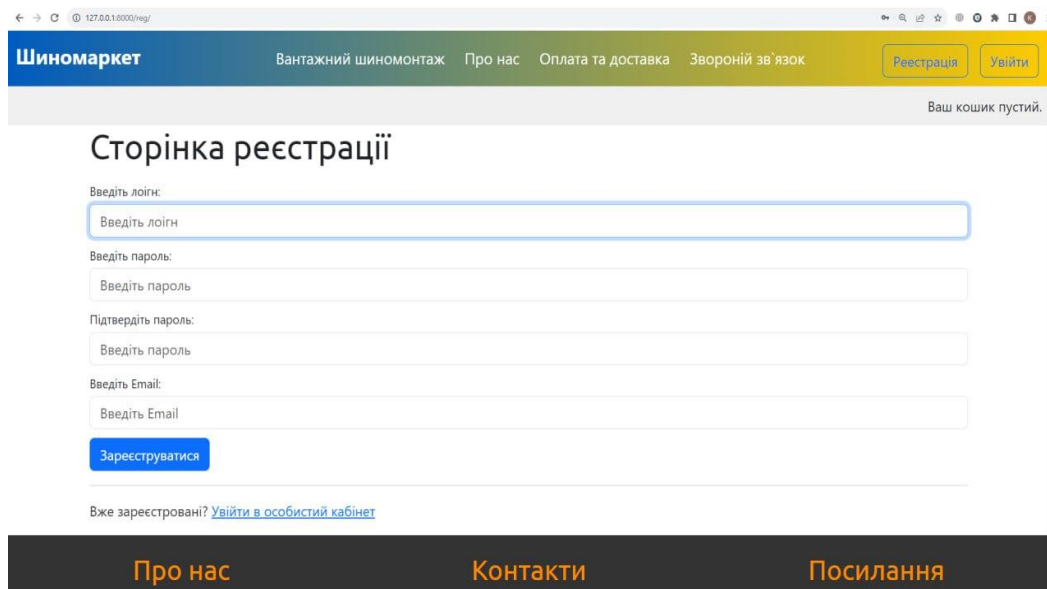


Рисунок 19 — Інтерфейс сторінки реєстрації

Якщо цей користувач вже є в базі, він потрапляє на сторінку авторизації.

Інтерфейс сторінки авторизації зареєстрованого користувача на сайті інтернет-магазину вантажних шин представлено на рисунку 20.

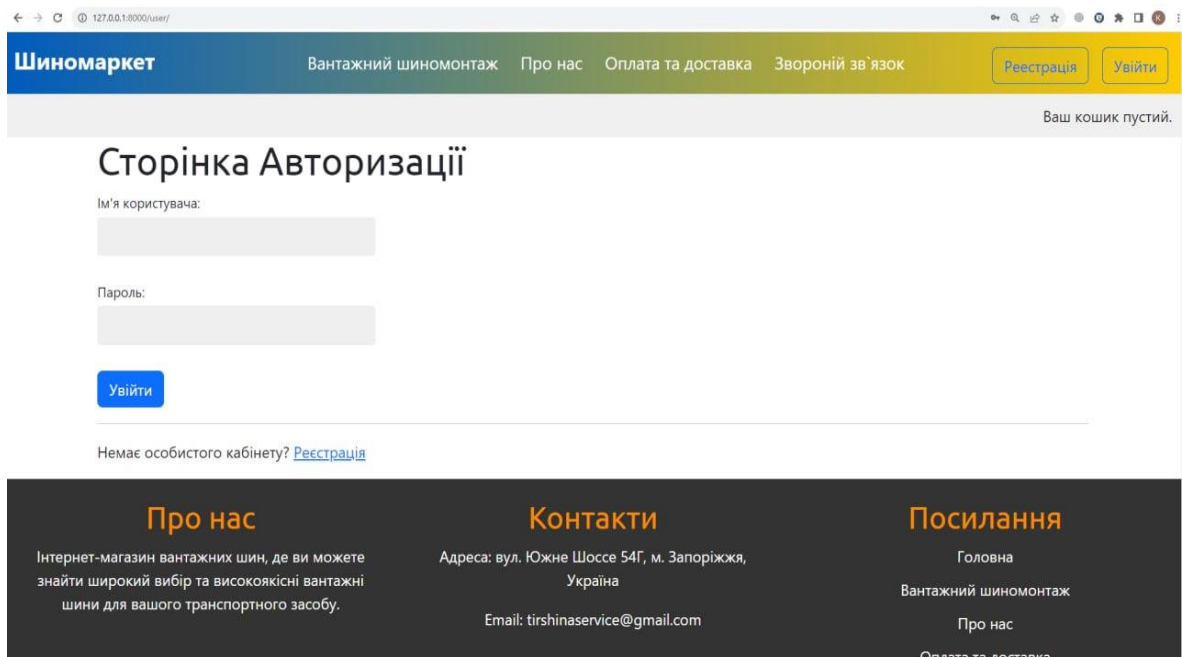


Рисунок 20— Інтерфейс сторінки авторизації

У разі успішної реєстрації та авторизації користувач потрапляє в особистий кабінет, інтерфейс відповідної сторінки представлено на рисунку 21.

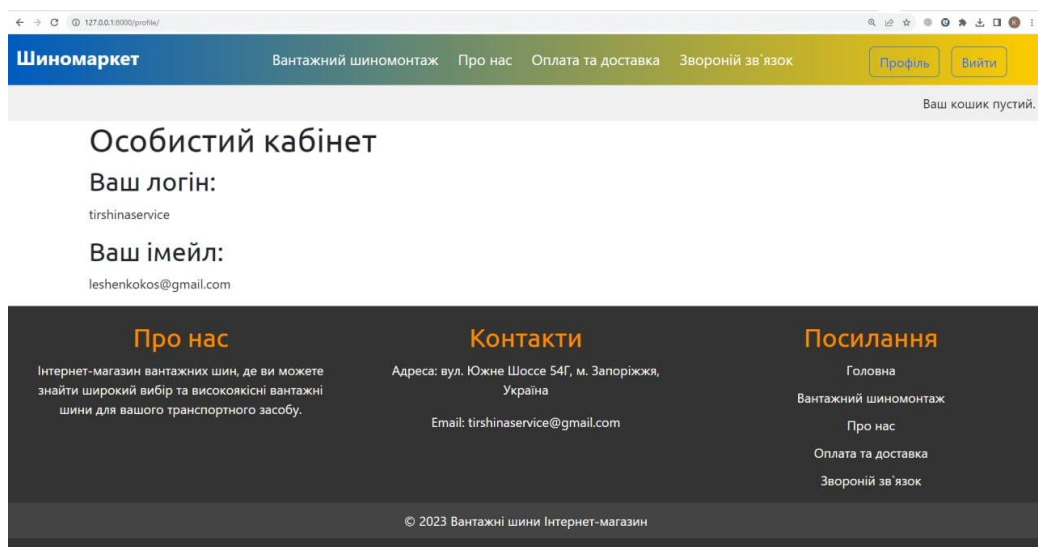


Рисунок 21 — Інтерфейс сторінки особистого кабінету користувача

Користувач сайту може отримати більш детальну інформацію про магазин та його місце розташування. Інтерфейс сторінки «Про нас» наведено на рисунку 22.

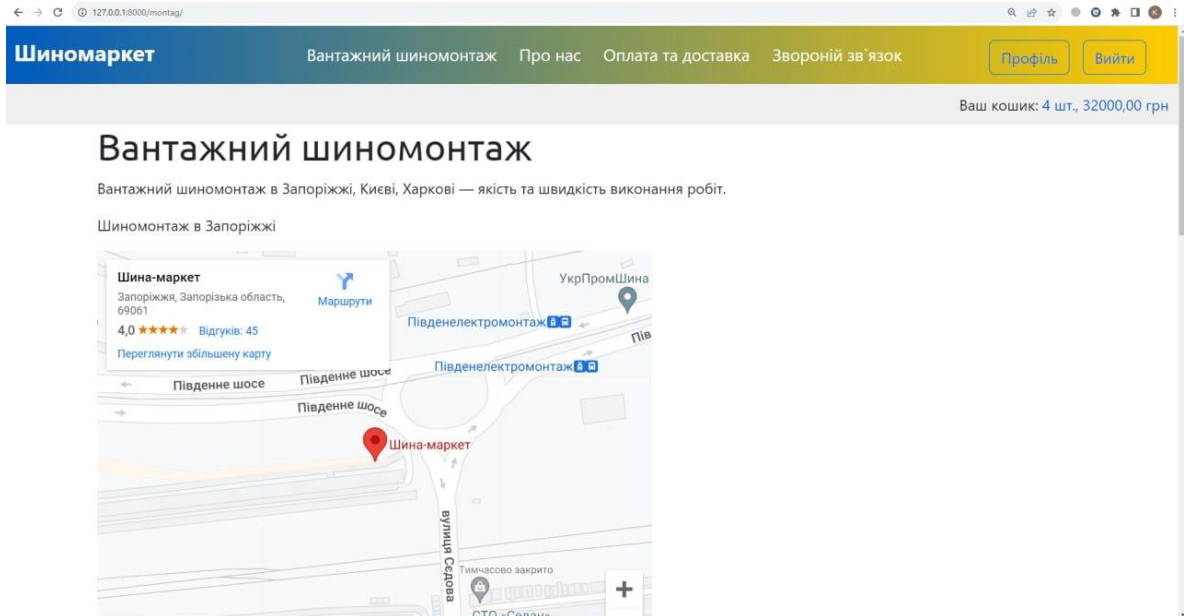


Рисунок 22— Інтерфейс сторінки «Про нас»

Головна сторінка завершується підвалом, де знаходяться контакти (рисунок 23).

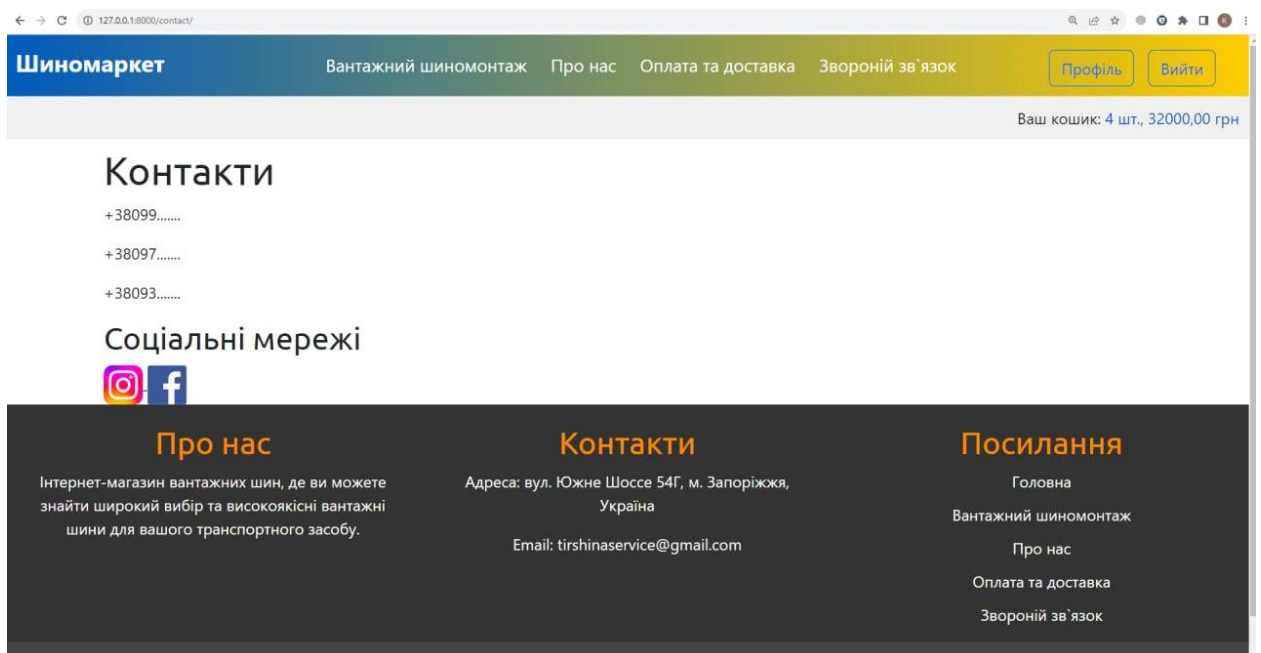


Рисунок 23 — Інтерфейс сторінки контактної інформації

Кошик у модальному вікні з'являється на сторінці сайту, одразу після додавання товару до кошика та дозволяє швидко перейти до оформлення замовлення. Користувач обирає товар та додає його до кошика. Інтерфейс сторінки вибору товару на сайті інтернет-магазину вантажних шин представлено на рисунку 24.

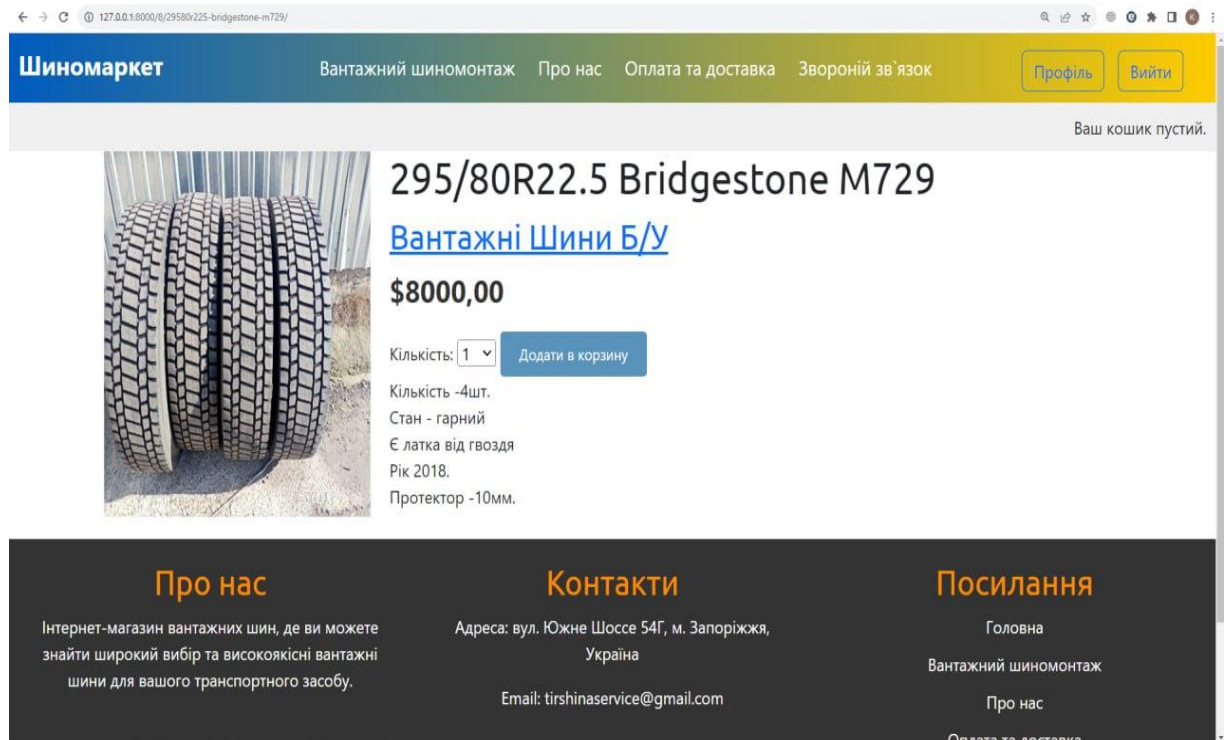


Рисунок 24— Інтерфейс сторінки вибору товару

Користувач заповнює всю необхідну інформацію для оформлення замовлення. Інтерфейс сторінки оформлення замовлення на сайті інтернет-магазину вантажних шин представлено на рисунку 25.

Після підтвердження замовлення користувач може перейти в свій кошик та переглянути перелік доданих туди товарів та або продовжити подальший пошук і додавання товарів або обрати спосіб оплати. Інтерфейс сторінки кошику замовлень на сайті інтернет-магазину вантажних шин представлено на рисунку 26.

Шиномаркет Вантажний шиномонтаж Про нас Оплата та доставка Звороній зв'язок Профіль Вийти

Ваш кошик: 4 шт., 32000,00 грн

Завершення замовлення

First name:

Last name:

Email:

Address:

Phone number:

City:

[Розмістити замовлення](#)

Ваше замовлення

- 4x 295/80R22.5 Bridgestone M729 32000,00 грн


Усього: 32000,00 грн

Рисунок 25 — Інтерфейс оформлення замовлення

Шиномаркет Вантажний шиномонтаж Про нас Оплата та доставка Звороній зв'язок Профіль Вийти

Ваш кошик: 4 шт., 32000,00 грн

Ваш кошик

Зображення	Продукт	Кількість	Видалити	Ціна за одиницю	Ціна
	295/80R22.5 Bridgestone M729	4 <input type="button" value="Оновити"/>	<input type="button" value="Видалити"/>	8000,00 грн	32000,00 грн
Усього					32000,00 грн

[Продовжити покупки](#) [Оформити замовлення](#)

Про нас

Інтернет-магазин вантажних шин, де ви можете знайти широкий вибір та високоякісні вантажні шини для вашого транспортного засобу.

Контакти

Адреса: вул. Южне Шоссе 54Г, м. Запоріжжя, Україна
Email: tirshinaservice@gmail.com

Посилання

Головна
Вантажний шиномонтаж
Про нас

Рисунок 26— Сторінка кошику замовлень

Для завершення замовлення користувач обирає спосіб оплати. Інтерфейс сторінки обрання користувачем способу оплати на сайті інтернет-магазину вантажних шин представлено на рисунку 27.

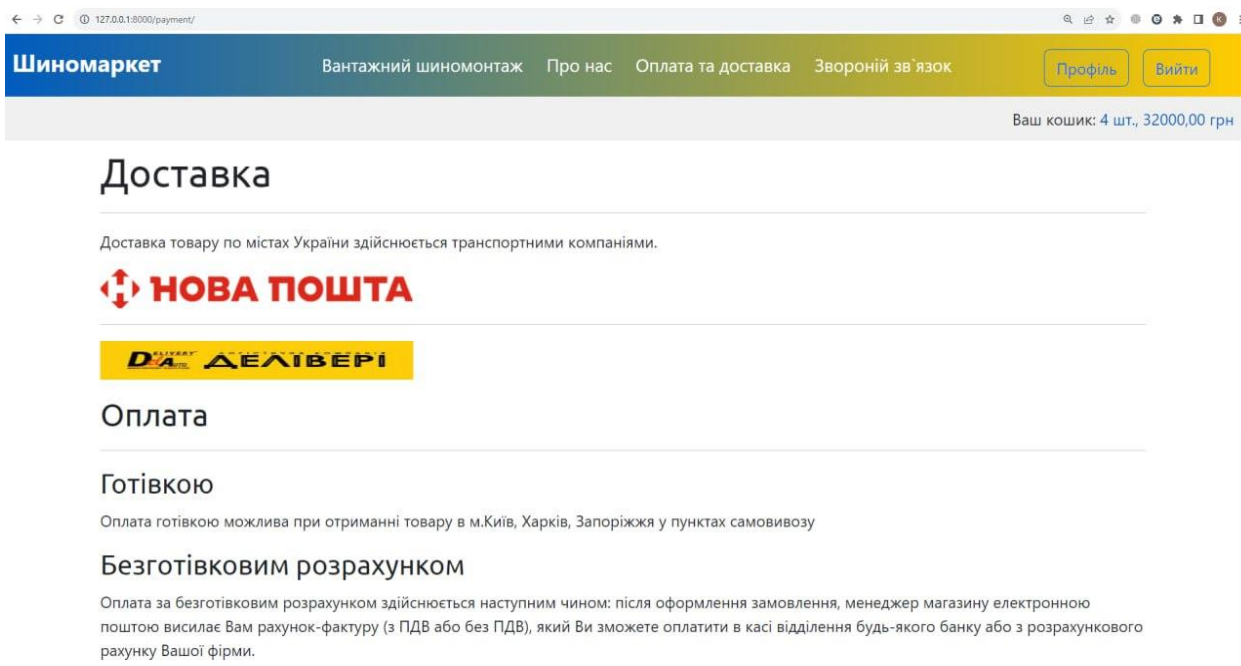


Рисунок 27 — Обрання способу оплати

Вбудована адмін панель дозволяє легко працювати з базою даних, а саме змінювати, додавати інформацію про доступні для користувачів товари. Сторінка адміністратора сайту інтернет-магазину вантажних шин представлена на рисунку 28.

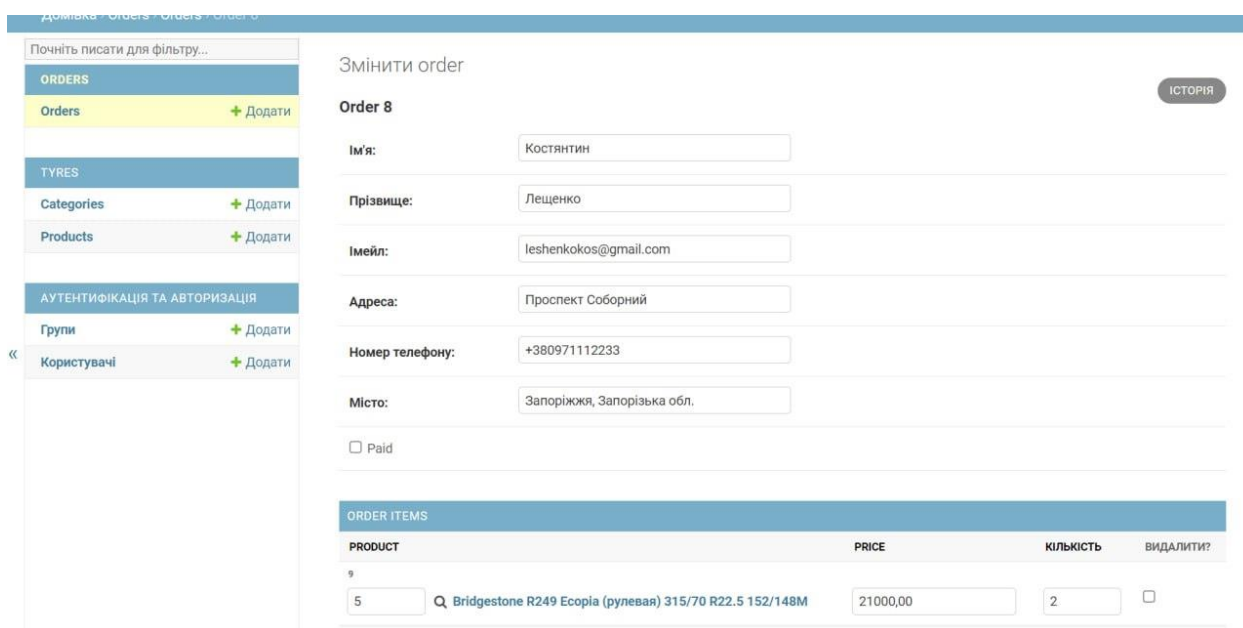


Рисунок 28 — Сторінка адмін панелі

3.7 Висновки до розділу 3

В третьому розділі було сформовано концепцію та розроблено дизайн веб-сайту інтернет-магазину вантажних шин та їх комплектуючих. Було сформульовано технічне завдання до кваліфікаційної роботи, в якому описується призначення сайту, мета створення, зацікавлені в проєкті особи. Згідно з сформованим технічним завданням та побажаннями замовника продукту було визначено основні складові інтернет-магазину. Відповідно спроектовано UML-діаграми: діаграму варіантів використання, діаграму послідовності та діаграму класів. Також було спроектовано відповідну базу даних. Спроектовано інтерфейс користувача сайту інтернет-магазину з продажу вантажних шин

4 ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ ТА АНАЛІЗ ХМАРНИХ ПЛАТФОРМ ДЛЯ ЙОГО РОЗГОРТАННЯ

4.1 Тестування функціональних можливостей додатку

Функціональне тестування — це тестування програмного забезпечення з метою перевірки реалізованості функціональних вимог, тобто можливості програмного забезпечення у певних умовах вирішувати завдання, необхідні користувачам. Функціональні вимоги визначають, як саме буде вести себе програмне забезпечення, і які задачі воно буде вирішувати. Окрім того, його використовують для перевірки працездатності посилань.

Тест №1. Перехід на посилання меню.

Вхідні дані: користувач, що тільки-но зайшов на сайт. Користувач перевіряє каталог усіх товарів із категорії «Вантажні шини», обирає довільний товар та дивиться про нього інформацію. Сторінка перегляду каталогу «Вантажні шини» сайту інтернет-магазину вантажних шин представлена на рисунку 29.

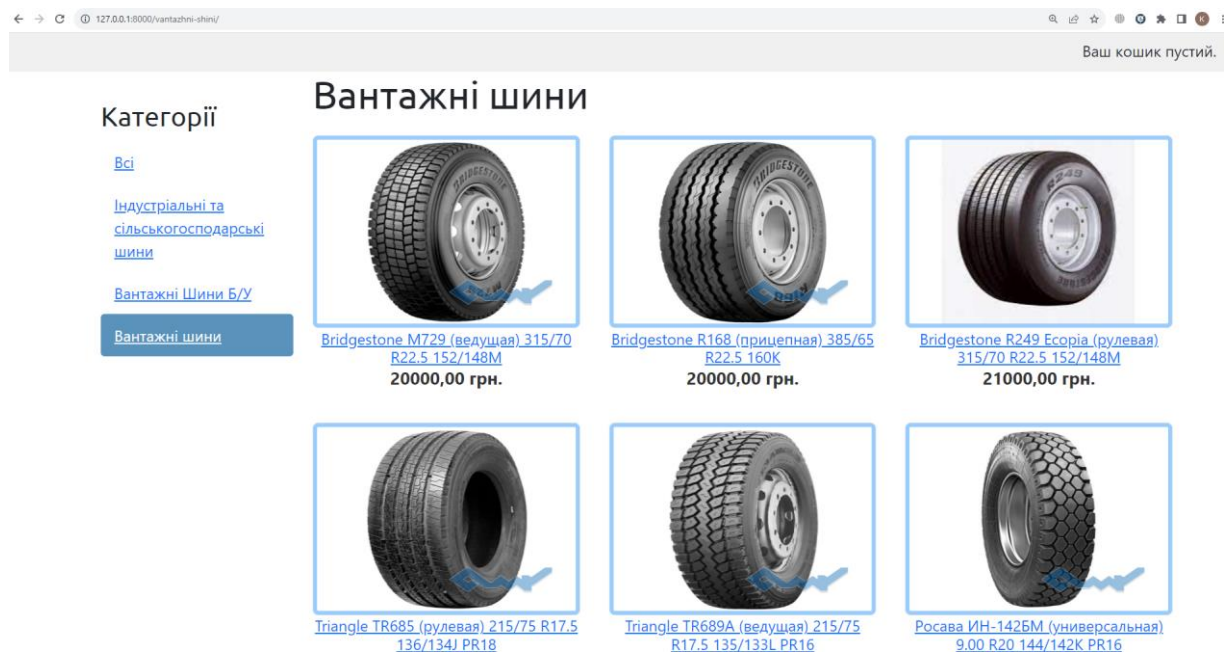


Рисунок 29 — Перегляд каталогу «Вантажні шини»

Сторінка перегляду інформації про обраний товар сайту інтернет-магазину вантажних шин представлена на рисунку 30.

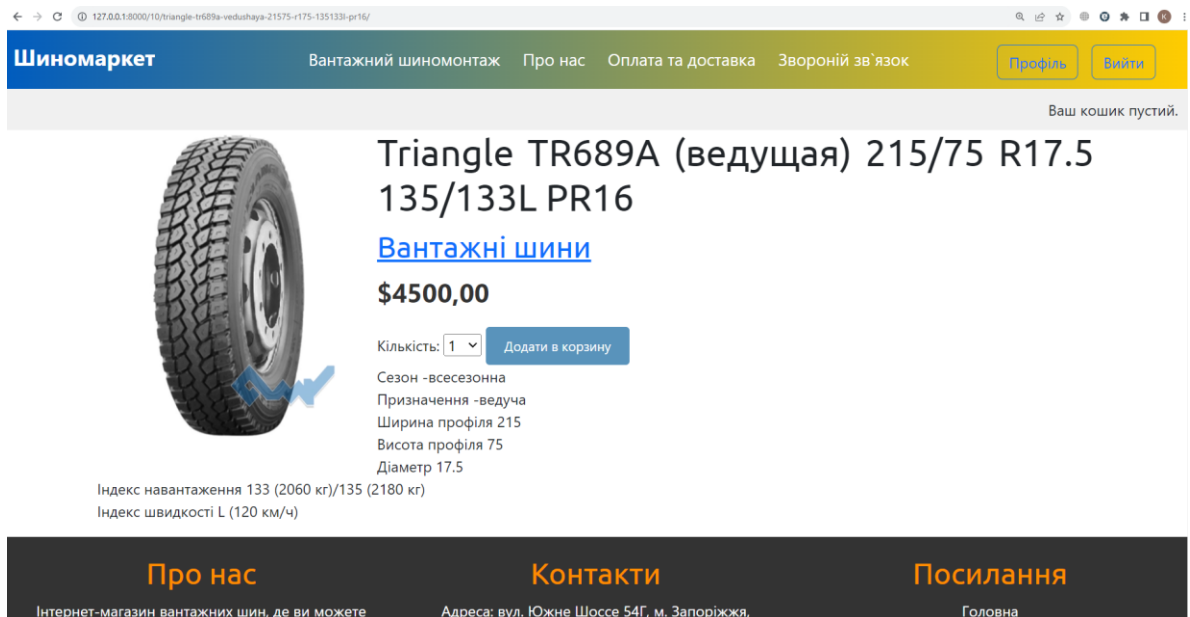


Рисунок 30 — Перегляд інформації про обраний товар

Очікувані дані: у процесі виконання цих дій не повинно виникнути труднощів.

Отримані результати: користувач не зіткнувся з проблемами.

Тест пройдено успішно.

Тест №2. Реєстрація на сайті.

Вхідні дані: користувач обирає пункт реєстрація та вводить свої дані у формі реєстрації. Сторінка введення даних для реєстрації нового користувача сайту інтернет-магазину вантажних шин представлена на рисунку 31.

Очікуваний результат: повідомлення про успішну реєстрацію.

Отримані результати: успішна реєстрація нового користувача.

Тест пройдено успішно.

Тест №3. Вхід в особистий кабінет та перегляд профілю користувача.

Вхідні дані: користувач вводить на сторінці авторизації дані за якими він реєструвався «Ім'я користувача та пароль».

Шиномаркет Вантажний шиномонтаж Про нас Оплата та доставка Звороній зв'язок [Реєстрація](#) [Увійти](#)

Ваш кошик пустий.

Сторінка реєстрації

Введіть логін:

Введіть пароль:

Підтвердіть пароль:

Введіть Email:

[Зареєструватися](#)

Вже зареєстровані? [Увійти в особистий кабінет](#)

[Про нас](#) [Контакти](#) [Посилання](#)

Рисунок 31 — Введення даних для реєстрації

Сторінка введення у формі авторизації ім'я користувача та паролю на сайту інтернет-магазину вантажних шин представлена на рисунку 32.

Очікуваний результат: після успішної реєстрації користувач потрапить на сторінку свого профілю.

Шиномаркет Вантажний шиномонтаж Про нас Оплата та доставка Звороній зв'язок [Реєстрація](#) [Увійти](#)

Ваш кошик пустий.

Сторінка Авторизації

Ім'я користувача:

Пароль:

[Увійти](#)

Немає особистого кабінету? [Реєстрація](#)

[Про нас](#) [Контакти](#) [Посилання](#)

Інтернет-магазин вантажних шин, де ви можете знайти широкий вибір та високоякісні вантажні шини для вашого транспортного засобу.

Адреса: вул. Южне Шоссе 54Г, м. Запоріжжя, Україна
 Email: tirshinaservice@gmail.com

Головна
 Вантажний шиномонтаж
 Про нас
 Оплата та доставка

Рисунок 32 — Введення у формі авторизації ім'я користувача та паролю

Отримані результати: у результаті успішної авторизації користувач потрапив на сторінку свого профілю. Сторінка успішної реєстрації та переходу на сторінку профіля на сайту інтернет-магазину вантажних шин представлена на рисунку 33.

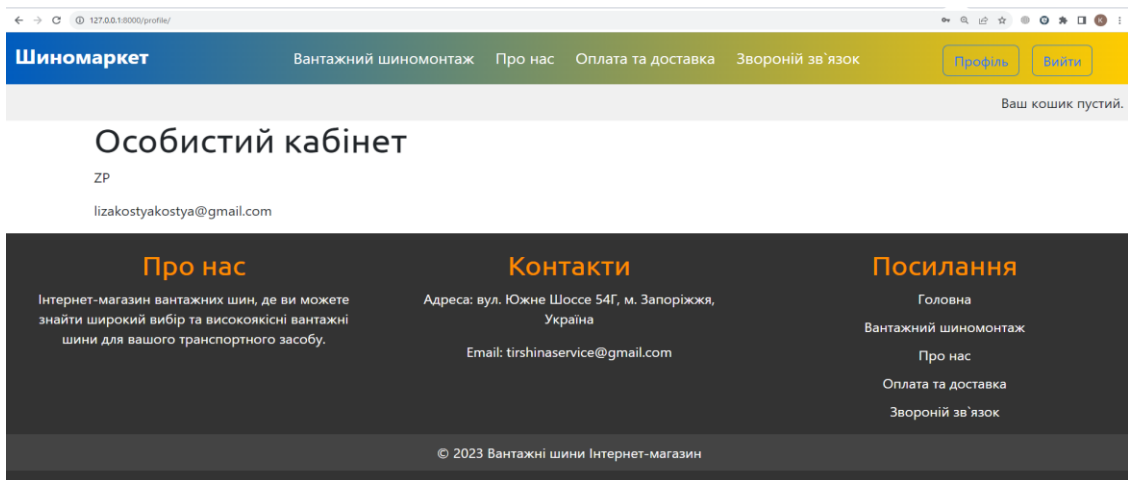


Рисунок 33 — Успішна реєстрація та перехід на сторінку профілю

Тест пройдено успішно.

Тест №4. Додавання товару до кошика.

Вхідні дані: користувач обирає товар та кількість товару, натискає кнопку «додати у кошик», після чого товар додається в кошик, при цьому не виникають помилки (див. Рис.34).

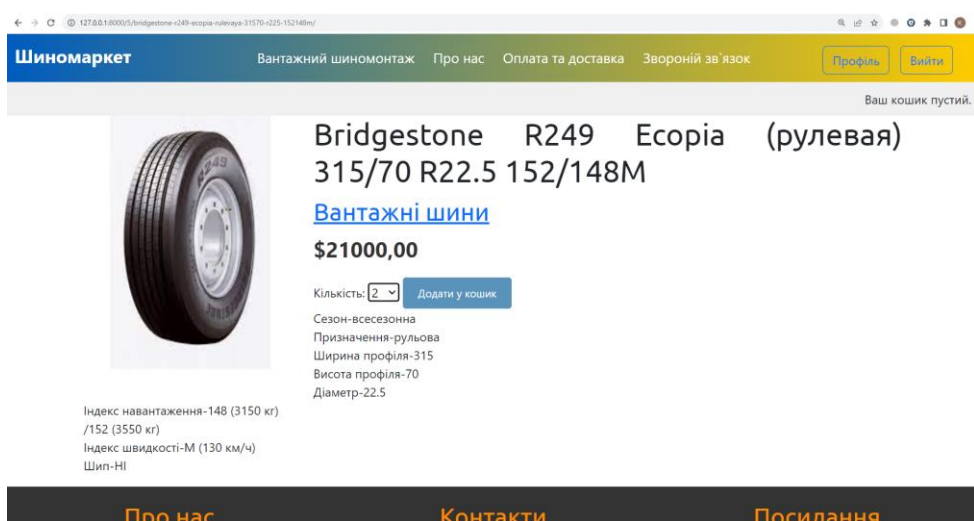


Рисунок 34 — Додавання товару до кошика

Отриманий результат: після натискання «Додати у кошик» товар додався до кошика без помилок (див. Рис.35).

Тест пройдено успішно.

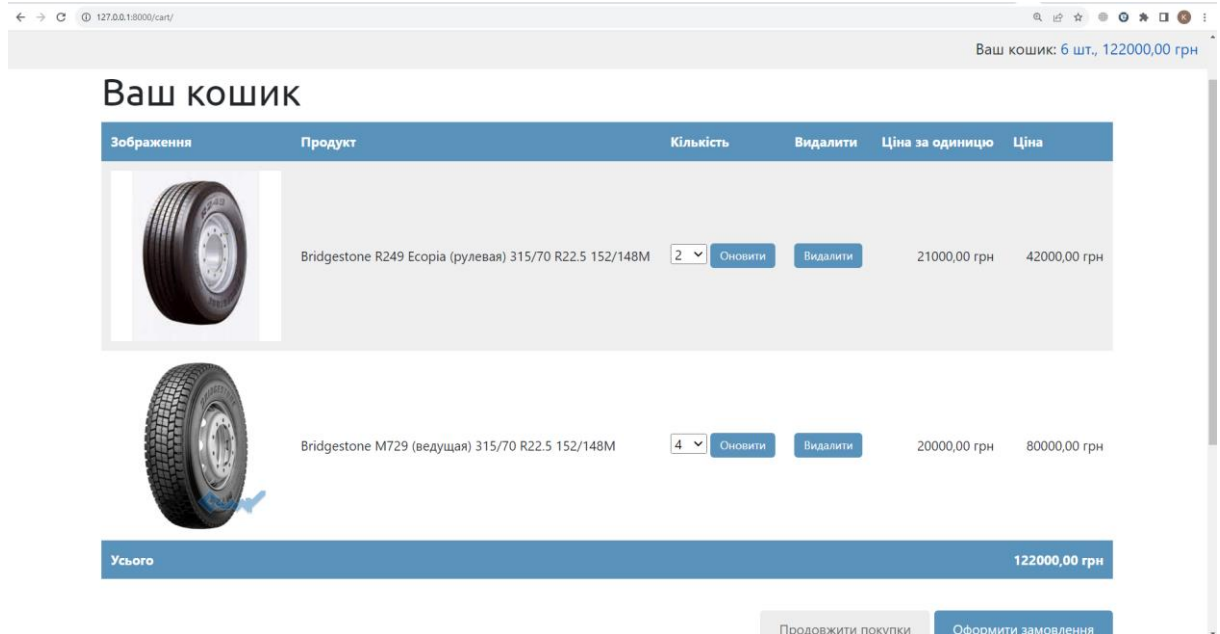


Рисунок 35 — Перевірка кошика на наявність товарів

Тест №5 Оформлення замовлення.

Вхідні дані: користувач, який додав до кошика один або кілька товарів, натискає на кнопку «Оформлення замовлення» (див. Рис. 36).

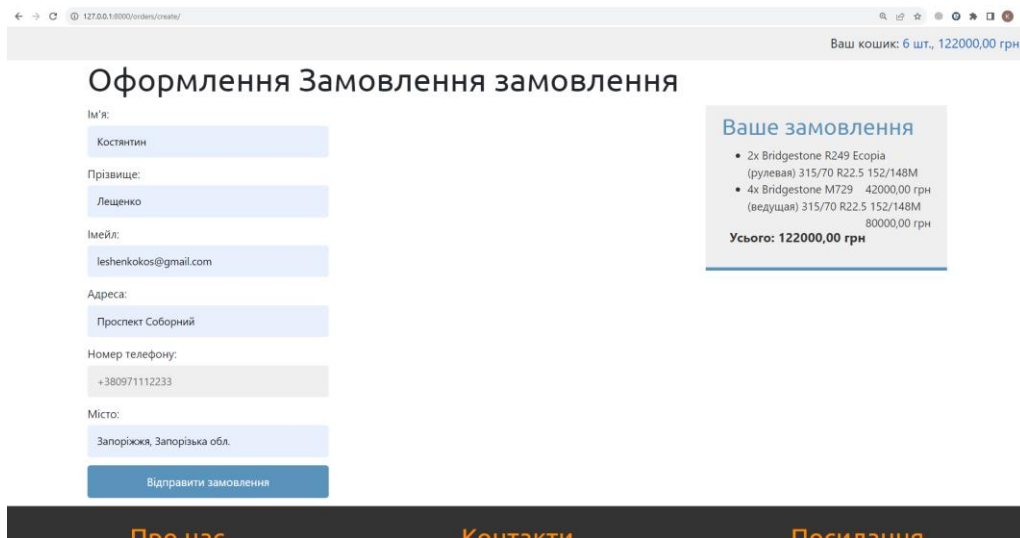


Рисунок 36 — Сторінка оформлення замовлення

Очікуваний результат: замовлення приймається в обробку, система видає повідомлення користувачеві «Ваше замовлення успішно завершено. Номер Вашого замовлення..» (див. Рис. 37).

Отриманий результат: одержаний результат збігається з очікуваним.

Тест пройдено успішно.

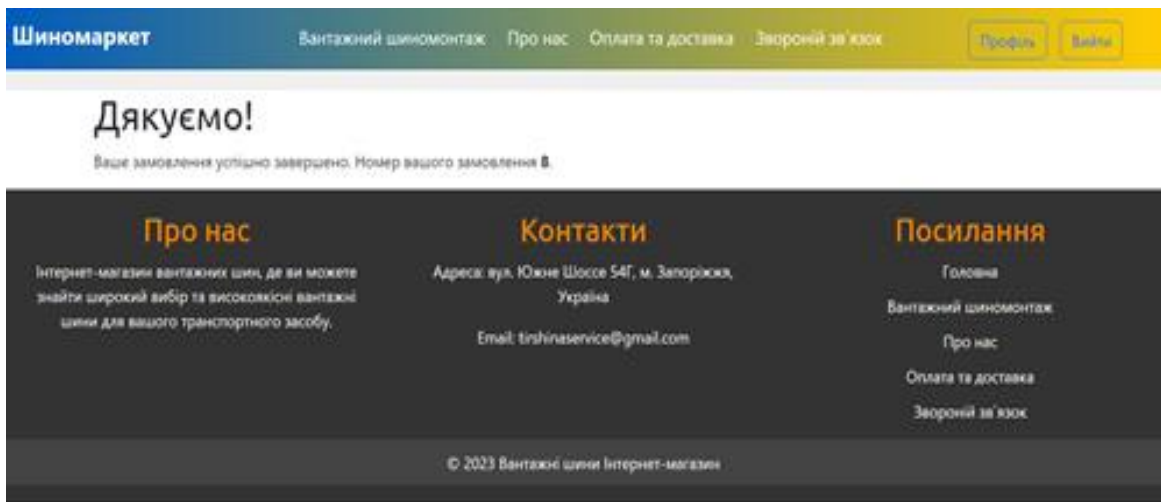


Рисунок 37 — Сторінка підтвердження прийняття замовлення

Внаслідок тестування не було виявлено функціональних помилок. Усі функції працюють стабільно та швидко.

4.2 Аналіз платформ, що надають послуги хмарних сервісів

Завершивши етап розробки та тестування програми з продажу вантажних шин, необхідно розмістити її на публічному веб-сервері, щоб вона стала доступною через інтернет відвідувачам магазину. З метою вибору хостингу для розміщення сайту було проведено аналіз платформ, що надають послуги хмарних сервісів.

Heroku

Це PaaS (Platform as a Service) (платформа як послуга), до появи якої розробники стикалися з рядом проблем. Технологія, яку представила

платформа Heroku, дозволяє створювати, масштабувати та розгорнути програми швидше та простіше.

Недоліками використання цієї платформи є таке:

Використання контейнерів для запуску та масштабування (як вертикального, так і горизонтального) додатку збільшує його вартість. Багатьом власникам додатків важко окупити витрати на масштабування додатків на Heroku.

Використання безкоштовного тарифного плану, який пропонує Heroku, має свої плюси та мінуси. Як любителі, так і професійні розробники використовують його для розміщення тестових проектів.

Незважаючи на те, що Heroku працює на AWS, він доступний лише у двох регіонах — США та Європі. Використання програми з клієнтами з країн Азії, збільшується час виконання запитів на відміну від користувачів із США чи Європи.

Back4app

Back4app надає повністю керований серверний інтерфейс з автоматизованим контролем використання та масштабуванням додатків, резервним копіюванням та відновленням, цілодобовим моніторингом та оповіщенням, веб-інструментами керування, технічною підтримкою. Має зручну панель моніторингу, можливість плавного масштабування додатків та підтримку клієнтів. Це надійна безсерверна платформа баз даних, що сприяє швидкій розробці програм, розробці серверної частини без написання власного коду, розміщення програм без вирішення проблем інфраструктури та простого масштабування програм. Його безсерверна архітектура дозволяє зосередитись виключно на розробці додатків замість вирішення нескінченних проблем з інфраструктурою.

Elastic Beanstalk (AWS)

Elastic Beanstalk (EB) — це інструмент DevOps для розгортання та масштабування веб-додатків та сервісів. Він підтримує безліч мов, таких як Java, .NET, PHP, Node.js, Python, Ruby, Go, а також веб-додатки Docker. EB

дозволяє швидко розгортати велику кількість серверів. Elastic Beanstalk (AWS) дозволяє вибрати бажаний рівень безпеки. Ви можете використовувати його розширені мережеві можливості для створення більш строгої та безпечної топології у розробленому додатку.

Google App Engine

Google App Engine — це платформа PaaS, розміщена на хостингу і що належить компанії Google, це дозволяє створювати, розробляти та розгортати мобільні бекенди та масштабовані веб-програми. Підтримує багато мов програмування, таких як Python, Java та PHP. Він поставляється з безкоштовним рівнем, що відмінно підходить для новачків та розробників, які бажають швидко створювати невеликі програми.

Dokku on Digital Ocean

Dokku — це інструмент, який дозволяє створювати та розгортати складні програми, просто розміщуючи їх у репозиторії Git. Міні-Heroku на базі Docker - найменша з існуючих реалізацій PaaS. Він безпосередньо підтримує вбудовані пакети Heroku, тому ви можете без проблем перенести існуючі програми Heroku. Він також пропонує безліч плагінів для сховищ даних та інших компонентів.

Firebase

Firebase — це платформа хмарних сервісів для роботи з бекенд від Google. Вона призначена для створення та підтримки додатків у реальному часі, надаючи такі функції, як сповіщення, автентифікація та інші послуги для роботи з фронтендом.

Firebase — відмінний інструмент для автентифікації; він надає протокол OAuth, який за допомогою надійного API полегшує автентифікацію через Facebook, Twitter, Google та інші платформи.

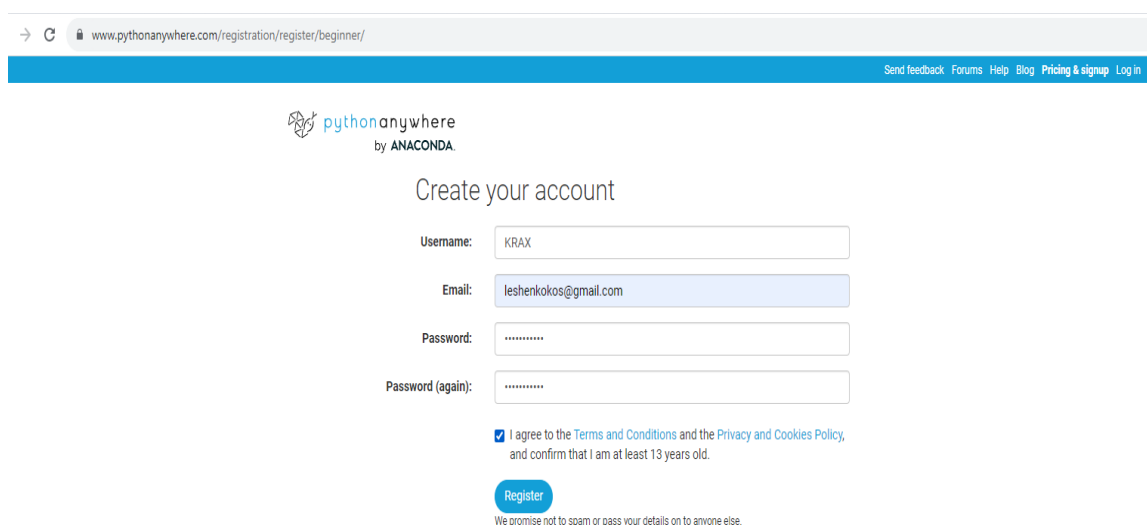
Платформа бази даних використовує JSON, що підходить для веб- або мобільного додатка. Вона підтримує Android, iOS та веб-додатки, що робить її чудовим вибором для створення нативних програм та веб-компонентів адміністратора.

Docker

Docker — це набір пропозицій платформи як послуги, які використовують технологію віртуалізації на рівні ОС для надання програмного забезпечення у вигляді пакетів чи контейнерів. Контейнери відокремлені один від одного та мають своє програмне забезпечення, файли конфігурації та бібліотеки. В результаті контейнери можуть взаємодіяти один з одним заздалегідь створеними каналами. Для всіх контейнерів використовується єдине ядро ОС, тому контейнери Docker використовують менше ресурсів, ніж віртуальні машини. Docker доступний у безкоштовній та преміум-версії для своїх користувачів. Він постачається разом із Docker Engine — програмним забезпеченням, що використовується для розміщення контейнерів.

Проаналізувавши платформи, що надають послуги хмарних сервісів, було вирішено використати для розгортання створеного додатку PythonAnywhere, оскільки його можливостей достатньо, щоб протестувати роботу сайту.

Процес розміщення сайту на хостингу представлено на рисунках 38 — 41.



The screenshot shows a web browser window with the URL www.pythonanywhere.com/registration/register/beginner/. The page header includes navigation links: [Send feedback](#), [Forums](#), [Help](#), [Blog](#), [Pricing & signup](#), and [Log in](#). The main content area features the PythonAnywhere logo (by ANACONDA) and the heading "Create your account". The registration form consists of the following fields:

- Username:** KRAX
- Email:** leshenkoks@gmail.com
- Password:** (masked with dots)
- Password (again):** (masked with dots)

Below the form, there is a checked checkbox for the agreement: I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old. A blue "Register" button is positioned below the checkbox. At the bottom of the form, a small disclaimer reads: "We promise not to spam or pass your details on to anyone else."

Рисунок 38 — Створення акаунта

```

1 |
2 |
3 |
4 | # ***** DJANGO *****
5 | # To use your own django app use code like this:
6 | import os
7 | import sys
8 |
9 | ## assuming your django settings file is at "/home/KRAK/mysite/mysite/settings.py"
10 | ## and your manage.py is at "/home/KRAK/mysite/manage.py"
11 | path = "/home/KRAK/Django1/tyremarket"
12 | if path not in sys.path:
13 |     sys.path.append(path)
14 |
15 | os.environ['DJANGO_SETTINGS_MODULE'] = 'tyremarket.settings'
16 |
17 | # then:
18 | from django.core.wsgi import get_wsgi_application
19 | application = get_wsgi_application()
20 |
21 |
22 |

```

Рисунок 39 — Пропис шляху

Log files:
The first place to look if something goes wrong.

- Access log: [krax.pythonanywhere.com.access.log](#)
- Error log: [krax.pythonanywhere.com.error.log](#)
- Server log: [krax.pythonanywhere.com.server.log](#)

Log files are periodically rotated. You can find old logs here: [/var/log](#)

Static files:
Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
/static/	/home/KRAK/Django1/tyremarket/tyres/static	🗑️
/media/	/home/KRAK/Django1/tyremarket/media	🗑️
Enter URL	Enter path	

Security:
An HTTPS certificate is needed so that people can access your site securely. We automatically provide a certificate for [krax.pythonanywhere.com](#).

HTTPS certificate: **Automatically provided for this hostname**

You need to **Reload your web app** to activate any changes made below.

Forcing HTTPS means that anyone who goes to your site using the insecure http:// URL will immediately be redirected to the secure https:// one. [More information here.](#)

Force HTTPS: Enabled

Password protection is ideal for sites that are under development when you don't want anyone to see them yet.

Password protection: Disabled

Username: [Enter a username](#)

Password: [Enter a password](#)

Disable:

Рисунок 40 — Підключення статичних та медіа файлів

Для налаштування статичних файлів треба конкретизувати місце, де саме знаходяться статичні файли того проєкту, що розробляється. Важливим є активація секції Static files, оскільки вона містить таблицю відповідності адрес

статичних файлів та реального їх місце розташування на диску сервера. Підключення статичних та медіа файлів веб-сайту інтернет-магазину вантажних шин представлено на рисунку 38.

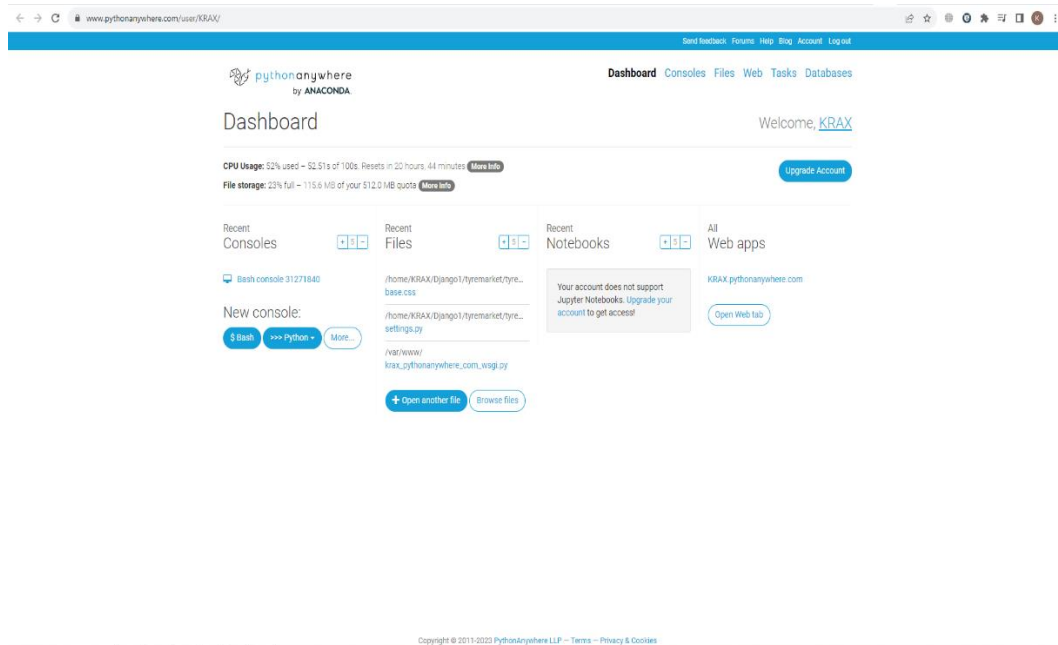


Рисунок 41 — Реєстрація на PythonAnywhere

Отже, є багато компаній, які надають сервери в Інтернеті. Для розміщення розробленого додатка скористалися послугами одного з них, а саме PythonAnywhere, який має досить простий процес публікації. Головною його перевагою є те, що він безкоштовний для маленьких програм з невеликою кількістю відвідувачів. Для тестування розробленого додатка з продажу вантажних шин це буде більш ніж достатньо.

4.3 Висновки до розділу 4

У процесі використання розроблений код проекту знаходиться у трьох місцях. На локальному комп'ютері було розроблено та протестовано проект. Завершивши роботу, проект розмістили на GitHub. Сам сайт знаходиться на PythonAnywhere, що дозволяє оновлювати його, просто завантажуючи нову версію коду з GitHub.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було поставлено та виконано наступні задачі:

1. Визначена предметна область та проаналізована існуюча проблематика.

В ході аналізу існуючих сайтів інтернет магазинів було з'ясовано, що у разі використання стандартних платформ і прийомів знижується ефективність як демонстрації характеристик і переваг конкретної групи товарів, так і донесення пропозицій до цільової аудиторії покупців. Було вивчено західні та вітчизняні аналоги, в ході чого стало зрозуміло, що на ринку відсутні застосунки, які відповідають всім потребам сучасного менеджменту та маркетингу.

2. Були розглянуті різні платформи та технології для розробки інтернет магазинів. При вивченні переваг та недоліків перевага була віддана мові Python з використанням фреймворку Django. В ході дослідження було проаналізовано велике коло Python-фреймворків та систем управління базами даних, які використовуються в аналогічних застосунках.

3. Спроектовано і реалізовано інтернет-магазин вантажних шин виходячи з аналізу існуючих рішень та регіональних вимог.

4. Виконано тестування розробленого проекту, проаналізовано варіанти розгортання проекту в хмарних середовищах. Розгорнуто і протестовано проєкт на одному з хмарних сервісів.

Таким чином, розроблений спеціалізований Інтернет-магазин відповідає усім вимогам, які були висунуті до програмного застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Harris A. Boost E-commerce Sales and Make More Money: Three Hundred Tips to Increase Conversion Rates and Generate Leads : Kindle Edition, 2014. 114 с.
2. Данько Т., Яворська Н. Особливості розвитку Інтернет-торгівлі та порівняльна характеристика з традиційною торгівлею. *Економіка та суспільство*. 2021. № 33. URL : <https://doi.org/10.32782/2524-0072/2021-33-43> (дата звернення: 03.03.2023).
3. Pyl'nova V. P. Online trade: features of implementation of goods by using the internet. *Economy. Management. Business*. 2020. Vol. 31, no. 1. URL : <https://doi.org/10.31673/2415-8089.2020.012929> (дата звернення: 28.03.2023).
4. Синявська О. А. Електронна торгівля в Україні: тенденції та перспективи розвитку. *Вісник Харківського національного університету імені В.Н. Каразіна. Серія "Міжнародні відносини. Економіка. Країнознавство. Туризм"*. 2019. Вип. 9. С. 126—132.
5. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL. ДияСофтЮП, 2013. 672 с.
6. Flanagan D., Matsumoto Y. The Ruby Programming Language: O'Reilly Media, 2008. 404 p.
7. Zandstra M. PHP 8 Objects, Patterns, and Practice: Mastering OO Enhancements, Design Patterns, and Essential Development Tools 6th ed. Edition. 2021. 858 p.
8. Camden R. The Jamstack Book: Beyond static sites with JavaScript, APIs, and markup. Apress; 6th ed. edition. 2022. 280 p.
9. Головна документація до фреймворку Django : веб-сайт. URL: <https://www.djangoproject.com/> (дата звернення: 20.04.2023).
10. Miguel Grinberg. Flask Web Development: Developing Web Applications with Python 2nd Edition. O'Reilly Media, 2018. 312 p.

11. Pedamkar P. Django vs Flask | Top 7 Awesome Differences to Learn. EDUCBA. URL : <https://www.educba.com/django-vs-flask> (дата звернення: 02.05.2023).
12. MySQL Workbench : веб-сайт. URL: <https://www.mysql.com/products/workbench> (дата звернення: 17.05.2023).
13. Regina Obe (Author), Leo Hsu. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database 3rd Edition. O'Reilly Media, 2017. 312 p.
14. Класифікація шаблонів проектування Мартина Фаулера URL: <http://martinfowler.com/eaaCatalog/index.html> (дата звернення: 21.09.2023).
15. MySQL Workbench : веб-сайт. URL: <https://www.mysql.com/products/workbench> (дата звернення: 11.09.2023).
16. Regina Obe (Author), Leo Hsu. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database 3rd Edition. O'Reilly Media, 2017. 312 p.
17. Мельник Р.А. Програмування веб-застосунків (фронт-енд та бек-енд): навч. посіб. Львів: Львівська політехніка, 2018. 248 с.
18. Офіційний ресурс про HTML специфікацію W3C : електронний ресурс. URL: <https://www.w3.org/TR/> (дата звернення: 05.10.2023).
19. Офіційний ресурс про HTML специфікацію WHATWG : електронний ресурс. URL: <https://html.spec.whatwg.org/multipage/> (дата звернення: 05.10.2023).
20. Django. Django Project : електронний ресурс. URL: <https://www.djangoproject.com/> (дата звернення: 15.04.2023).
21. DjangoBook.com : електронний ресурс. URL: <https://djangobook.com/> (дата звернення: 09.09.2023).
22. Vincent W. S. Django for Beginners: Build Websites with Python and Django. Welcome to code, 2020. 292 p.
23. What is a Context Diagram and what are the benefits of creating one? : електронний ресурс. URL: <https://cutt.ly/0gSE5PF> (дата звернення: 17.09.2023).

24. Лещенко К. Особливості побудови сайту надання послуг з використанням Python і Django. Збірник наукових праць студентів, аспірантів і молодих вчених «Молода наука-2023» Запоріжжя: ЗНУ, 2023. Т. V С. 98-99.

25. Попівций В.І., Лещенко К.К. Особливості побудови сайту надання послуг з використанням Python і Django : матеріали III Всеукраїнської науково-практичної конференції за участю молодих науковців «Актуальні питання сталого науково-технічного та соціально-економічного розвитку регіонів України». Запоріжжя : ЗНУ, 2023. 106–161.

26. Моделі електронного бізнесу в Інтернет : електронний ресурс. URL: <http://um.co.ua/8/8-13/8-137771.html> (дата звернення: 12.03.2023).

27. СМС що це таке?: електронний ресурс. URL: <https://freehost.com.ua/ukr/faq/wiki/cms-что-eto-takoe> (дата звернення: 30.10.2023).

28. Bottle: Python Web Framework: електронний ресурс. URL: <https://bottlepy.org/docs/dev/> (дата звернення: 30.10.2023).

29. CherryPy. A Minimalist Python Web Framework: електронний ресурс. URL: <https://cherrypy.dev/> (дата звернення: 30.10.2023).

**Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ**

Я, Лазарєв Вадим Андрійович, студент 2 курсу, денної форми навчання,
Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ,
спеціальність 121 Інженерія програмного забезпечення, адреса електронної
пошти p314154265@gmail.com,

- підтверджую, що написана мною кваліфікаційна робота на тему:
«Особливості побудови сайту надання послуг з використанням Python і Django» відповідає вимогам академічної доброчесності та не містить порушень,
що визначені у ст. 42 Закону України «Про освіту», зі змістом яких
ознайомлений;

- заявляю, що надана мною для перевірки електронна версія роботи є
ідентичною її друкованій версії;

- згоден на перевірку моєї роботи на відповідність критеріям академічної
доброчесності у будь-який спосіб, у тому числі за допомогою Інтернет - системи,
в також архівування моєї роботи у базі даних цієї системи.

Дата 30.11.2023 Підпис _____ Лещенко Костянтин Костянтинович
(студент)

Дата 30.11.2023 Підпис _____ Попівщій Василь Іванович
(науковий керівник)