

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

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р. техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____ 2023 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
ВЕБЗАСТОСУНОК З ПРОДАЖУ ТКАНИН

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.21910211

Виконав студент 4-го курсу, групи 402
_____ *А. Ю. Карпенко*
«19» червня 2023 р.

Керівник: PhD, ст. викладач
_____ *К. О. Антіпова*
«19» червня 2023 р.

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

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

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

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Видано студенту групи 402 факультету комп'ютерних наук Карпенку Андрію
Юрійовичу.

1. Тема кваліфікаційної роботи «Вебзастосунок з продажу тканин».

Керівник роботи Антіпова Катерина Олександрівна, PhD, ст. викладач.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «__» ____ 20__ р. № ____

2. Строк представлення кваліфікаційної роботи студентом «__» ____ 20__ р.

3. Вхідні (початкові) дані до роботи: складність у орієнтуванні на вебсторінках у покупців, вікової категорії від 30 до 60 років.

Очікуваний результат: інтернет-магазин з продажу тканин з ергономічним інтерфейсом.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз сучасних стандартів ергономічного інтерфейсу вебзастосунку;
- проведення аналізу інтернет-магазинів конкурентів;
- вибір технології для розробки вебсайту;
- створення моделей та структури БД;
- налаштування проєкту;

- розробка структури вебзастосування;
- розробка системи авторизації та аутентифікації;
- розробка адміністративної та клієнтської частини вебзастосування.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Захист органів зору під час роботи за ЕОМ».

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Боженко А. Л., викладач	

Керівник роботи PhD, ст. викладач Антіпова К. О.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Карпенко А. Ю.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 23 » _____ листопада _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: Вебзастосунок з продажу тканин

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	27.10.2022	27.10.2022	виконано
2	Отримання завдання на виконання БКР	23.11.2022	23.11.2022	виконано
3	Складання календарного плану роботи на весь період виконання БКР	08.12.2022	08.12.2022	виконано
4	Отримання завдання на переддипломну практику	01.05.2023	01.05.2023	виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	01.05.2023	14.05.2022	виконано
6	Розробка звіту з переддипломної практики	15.05.2023	17.05.2023	виконано
7	Виконання БКР: аналіз сучасних стандартів інтерфейсу інтернет-магазинів, огляд існуючих технологій розробки, розробка ПЗ	15.05.2023	19.06.2023	виконано
8	Попередній захист БКР на засіданні комісії кафедри	29.05.2023	30.05.2023	виконано
9	Доробка та остаточне оформлення БКР	02.06.2023	19.06.2023	виконано
10	Подання БКР рецензенту	15.06.2023	17.06.2023	виконано
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.06.2023	22.06.2023	виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	26.06.2023	29.06.2023	виконано

Розробив студент Карпенко А. Ю. _____
(прізвище, ім'я, по батькові студента) (підпис)

Керівник роботи PhD, ст. викладач Антіпова К. О. _____
(посада, прізвище, ім'я, по батькові) (підпис)

«_08_» _____ 12 _____ 2022 р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студента групи 402 ЧНУ ім. Петра Могили

Карпенка Андрія Юрійовича

Тема: «Вебзастосунок з продажу тканин»

Актуальність: У зв'язку з карантинними обмеженнями та повномасштабним вторгненням збільшилась потреба у інтернет-магазинах. Існує потреба у якісному вебзастосунку для того, щоб не втратити постійних і потенційних клієнтів.

Об'єкт роботи – вебзастосунок з продажу тканин.

Предмет роботи – технології для розробки вебзастосунку з продажу тканин.

Метою кваліфікаційної роботи є автоматизація процесу продажу тканин за рахунок розробки надійного вебзастосунку з ергономічним інтерфейсом.

Відповідно до мети було визначено такі завдання:

- аналіз сучасних стандартів ергономічного інтерфейсу вебзастосунку;
- проведення аналізу інтернет-магазинів конкурентів;
- вибір технології для розробки вебзастосунку;
- створення моделей та структури БД;
- налаштування проєкту;
- розробка структури вебзастосунку;
- розробка системи авторизації та аутентифікації;
- розробка адміністративної та клієнтської частини вебзастосунку.

Пояснювальна записка складається зі вступу, чотирьох розділів та висновків.

У першому розділі розглядаються наявні стандарти ергономічного інтерфейсу інтернет-магазинів.

У другому розділі досліджено технології розробки, фреймворки, бази даних та інструменти для створення інтерфейсу.

У третьому розділі розроблено та створено базу даних та її моделі. Підключено БД до проєкту. Налаштування структури проєкту та підключення залежностей Maven для подальшої розробки вебзастосунку.

У четвертому розділі розроблено систему авторизації та аутентифікації, адміністративну та клієнтську частину вебзастосунку.

В результаті було отримано надійний вебзастосунок з ергономічним інтерфейсом і високим рівнем захисту даних.

Бакалаврська кваліфікаційна робота містить 77 сторінок, 39 рисунків, 31 використаних джерел.

Ключові слова: інтернет-магазин, ергономічний інтерфейс, технології, вебзастосунок, розробка, аналіз, інтернет-торгівля.

ABSTRACT

bachelor's qualification work of a student of group 402 of Petro Mohyla Black Sea National University

Karpenko Andrii Yuriyovych

Topic: «Fabric sales web application»

Relevance: In connection with quarantine restrictions and a full-scale invasion, the need for online stores has increased. There is a need for a quality web application in order not to lose regular and potential customers.

The object of the work is a web application for the sale of fabrics.

The subject of the work is technologies for the development of a web application for the sale of fabrics.

The purpose of the qualification work is to automate the fabric sales process by developing a reliable web application with an ergonomic interface.

In accordance with the goal, the following tasks were defined:

- analysis of modern standards of ergonomic web application interface;
- analysis of competitors' online stores;
- choice of technology for website development;
- creation of models and database structure;
- project settings;
- development of the structure of the web application;
- development of the authorization and authentication system;
- development of the administrative and client part of the web application.

The explanatory note consists of an introduction, four chapters and conclusions.

The first section examines the existing standards of the ergonomic interface of online stores.

The second chapter explores development technologies, frameworks, databases, and tools for creating an interface.

In the third chapter, the database and its models are developed and created. The database is connected to the project. Setting up the project structure and connecting Maven dependencies for further development of the web application.

In the fourth chapter, the authorization and authentication system, administrative and client part of the web application are developed.

As a result, a reliable web application with an ergonomic interface and a high level of data protection was obtained.

Bachelor thesis contains 77 pages, 39 figures, 31 used sources.

Key words: online store, ergonomic interface, technologies, web application, analysis, online trade.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП.....	4
1 ПРОБЛЕМАТИКА СТВОРЕННЯ САЙТІВ ДЛЯ МАЛОГО БІЗНЕСУ	5
1.1 Розбір поняття «інтернет-торгівля».....	6
1.2 Дослідження успіху в інтернет-торгівлі.....	8
1.3 Дослідження проблематики малого бізнесу	9
1.4 Аналіз інтернет-магазинів конкурентів.....	10
1.5 Специфікація проекту	15
Висновки до розділу 1	19
2 МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКУ З ПРОДАЖУ ТКАНИН.....	20
2.1 Вебсервери.....	20
2.2 Протоколи передавання даних.....	21
2.3 Підходи та технології	22
2.4 Технології для розробки.....	26
2.5 Вибір фреймворку	30
2.6 Бази даних.....	32
2.7 Інтерфейс	35
2.8 Тимчасові файли.....	36
Висновки до розділу 2	37
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	38
3.1 Моделі бази даних	38
3.2 Підключення БД	41
3.3 Структура вебзастосунок	44
3.4 Конфігурація проєкту.....	47
Висновки до розділу 3	52
4 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ	53
4.1 Реалізація аутентифікація та авторизації	53
4.2 Адміністративна панель сайту.....	57
4.3 Клієнтська частина сайту.....	62
Висновки до розділу 4	67
ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70

ПЕРЕЛІК СКОРОЧЕНЬ

B2B	–	business-to-business (бізнес-до-бізнесу)
B2C	–	business-to-consumer (бізнес-до-споживача)
C2C	–	consumer-to-consumer (споживач-до-споживача)
B2G	–	business-to-government (бізнес-до-уряду)
C2B	–	consumer-to-business (споживач-до-бізнесу)
G2C	–	government-to-consumer (уряд-до-споживача)
БД	–	база даних

ВСТУП

Під час виконання кваліфікаційної роботи було поставлено за ціль визначити основні переваги та стандарти для предмету дослідження - інтернет магазину.

Об'єктом кваліфікаційної роботи є вебзастосунок з продажу тканин.

Предметом кваліфікаційної роботи є технології, необхідні для розробки вебзастосунку з продажу тканин.

Метою кваліфікаційної роботи є автоматизація процесу продажу тканин за рахунок розробки надійного вебзастосунку з ергономічним інтерфейсом.

Для реалізації поставленої мети потрібно вирішити такі завдання:

- проаналізувати сучасні стандарти ергономічного інтерфейсу вебзастосунку;
- провести аналіз інтернет-магазинів конкурентів;
- обрати технології для розробки вебсайту;
- створити моделі та структуру БД;
- налаштувати проєкт;
- розробити структуру вебзастосунку;
- розробити системи авторизації та аутентифікації;
- розробити адміністративну та клієнтську частину вебзастосунку.

Також робота містить спеціальну частину з охорони праці, де буде досліджено вплив несприятливих чинників на оператора, який працює з інформаційними системами. Буде прораховано кількість світла на робочій поверхні працівника.

1 ПРОБЛЕМАТИКА СТВОРЕННЯ САЙТІВ ДЛЯ МАЛОГО БІЗНЕСУ

В наш час люди хочуть більше знати, та менше витратити часу на базові речі. На допомогу прийшли сайти та мобільні пристрої. Ви можете замовити холодильник або телевізор з іншого кінця світу за декілька хвилин, знаючи про нього все: від технічних характеристик до інформації в експлуатації, завдяки відгукам. Тому даний напрям так швидко розвивається.

Сфера застосування онлайн-торгівлі майже нічим не обмежена. Сайти потрібні всюди, багато яких компаній використовують web-технології та мають гарний досвід і статистику. Для керівників компаній дуже важливі робітники у цій сфері. Використовуючи web-розробку компанії можуть отримати переваги в конкурентоспроможності.

Кожного дня все більше і більше людей передають перевагу інтернет замовленням, замовляючи мільйони різноманітних товарів з різних куточків світу. Їх популярність обумовлена багатьма факторами:

- можливість перевірити товар та приміряти до оплати;
- можливість переглянути відгуки інших покупців;
- можливість порівняти товари з різних магазинів;
- можливість замовити товар з інших міст та країн;
- відсутність можливості покинути дім.

Таким чином, через ряд причин, покупці все частіше надають перевагу покупкам в інтернеті, а не походам в магазини.

Під час пандемії багато людей сидять вдома, але щоб придбати товари треба вийти до магазину. Це дуже небезпечно для організму та ставить людину під ризик захворіти [1]. Тому усі компанії по продажу товарів вирішили цю проблему створенням інтернет-магазинів.

З початку карантину і обмеженнями в торгівлі, підприємці почали використовувати або розвивати торгівлю в мережі інтернет. Після чого деякі підприємці зменшили кількість магазинів або взагалі займаються лише продажами в мережі.

1.1 Розбір поняття «інтернет-торгівля»

Інтернет-торгівля, також відома як електронна комерція або e-commerce [2], - це процес купівлі та продажу товарів і послуг через Інтернет. Цей вид торгівлі дозволяє клієнтам замовляти товари та послуги з будь-якої точки світу за допомогою комп'ютера або мобільного пристрою, з підтримкою Інтернету.

Основними перевагами інтернет-торгівлі є доступність, швидкість та зручність. Замовлення можуть бути зроблені у будь-який час доби, 7 днів на тиждень, тому що магазини в Інтернеті ніколи не закриваються. Крім того, клієнти можуть швидко знайти та порівняти товари із різних магазинів, що дозволяє знайти найкращі пропозиції і знижки.

Інтернет-торгівля також забезпечує можливість підвищити ефективність торгової діяльності за рахунок автоматизації бізнес-процесів та оптимізації логістики. Такі переваги, як можливість швидко змінювати ціни, пропонувати індивідуальні пропозиції та застосовувати цільовий маркетинг, дозволяють залучати нових клієнтів та зберігати лояльність існуючих. Інтернет-магазин викликає фізичну аналогію купівлі товарів чи послуг у звичайного роздрібного торгового центру чи торгового центру; цей процес називається інтернет-магазинами бізнес-споживач (B2C). Коли вебмагазин створений для того, щоб він міг купувати інший бізнес, цей процес називається інтернет-магазинами бізнес-бізнес (B2B). Типовий інтернет-магазин дозволяє замовнику переглядати асортимент товарів і послуг фірми, переглядати фотографії або зображення товарів, а також інформацію про технічні характеристики, характеристики та ціни.

Інтернет-магазини зазвичай дозволяють покупцям використовувати функції "пошуку" для пошуку конкретних моделей, марок або предметів. Клієнти в Інтернеті повинні мати доступ до Інтернету та дійсний спосіб оплати, щоб завершити транзакцію, наприклад, кредитну карту, дебетову карту з підтримкою Interac або послугу, наприклад PayPal. Для фізичних продуктів (наприклад, книжок у м'якій обкладинці чи одягу) електронний пристрій відправляє продукцію замовнику; для цифрових продуктів, таких як цифрові аудіофайли пісень або файли

програмного забезпечення, зазвичай файл надсилається замовнику через Інтернет, за допомогою електронної пошти.

Інтернет-торгівля може бути різних типів, залежно від того, хто купує та продає товари і послуги. Найбільш поширені типи інтернет-торгівлі включають [3]:

- B2B: у цьому типі торгівлі компанії продають свої товари та послуги іншим компаніям. Такий тип торгівлі зазвичай характеризується великими обсягами закупівель та довгостроковими контрактами;

- B2C: у цьому типі торгівлі компанії продають свої товари та послуги безпосередньо споживачам. Це може бути як крупний онлайн-ритейлер, так і малий інтернет-магазин;

- C2C: у цьому типі торгівлі споживачі продають товари один одному. Найбільш відомими платформами цього типу є eBay та Craigslist;

- B2G: у цьому типі торгівлі компанії продають товари та послуги урядовим організаціям та державним установам;

- C2B: у цьому типі торгівлі споживачі надають свої товари та послуги компаніям. Найбільш відомими платформами цього типу є фриланс-біржі та платформи краудсорсингу;

- G2C: у цьому типі торгівлі урядові організації продають товари та послуги безпосередньо споживачам. Це може бути, наприклад, державний інтернет-магазин з продажу ліцензій та сертифікатів.

Поява інтернет-магазинів, як ми знаємо сьогодні, розвивалася з появою Інтернету. Спочатку ця платформа функціонувала лише як рекламний інструмент для компаній, надаючи інформацію про свою продукцію. Це швидко перейшло від цієї простої утиліти до фактичної транзакції по магазинах в Інтернеті завдяки розробці інтерактивних вебсторінок і безпечних передач. Зокрема, зростання Інтернету як захищеного торгового каналу розвивалося з 1994 року, з першими продажами альбому Стінга «Ten Summoner's Tales». Вино, шоколадки та квіти незабаром пішли в трійку піонерських роздрібних категорій, що сприяло зростанню інтернет-магазинів. Дослідники встановили, що наявність продуктів, відповідних

для електронної комерції, є ключовим показником успіху в Інтернеті [4]. Багато з цих продуктів справились добре, оскільки це загальна продукція, яку покупцям не потрібно було чіпати та відчувати, щоб купувати. Але також важливо, що в перші дні покупців в Інтернеті було мало, і вони були з вузького сегмента: заможні, чоловіки, старше 30 років. Інтернет-покупки почали розвиватися з цих перших днів, і у Великобританії припадає значна частка відсотків (залежно від категорії товару, оскільки відсотки можуть змінюватися).

1.2 Дослідження успіху в інтернет-торгівлі

Успішна робота в інтернет-торгівлі залежить від багатьох факторів, зокрема від типу торгівлі, конкуренції, маркетингових стратегій, якості продукту та обслуговування клієнтів. Розглянемо декілька основних чинників успішної інтернет-торгівлі:

- розуміння аудиторії: треба вивчати, хто є цільовою аудиторією та її потреби, щоб забезпечити якісне обслуговування та продукти, які зможуть задовольнити їх вимоги;
- розробка ефективної маркетингової стратегії: бажано використовувати ефективні канали маркетингу, такі як соціальні медіа, контент-маркетинг, реклама в пошукових системах та інші, для того, щоб залучити нових клієнтів та зберегти існуючих;
- якісний продукт та обслуговування клієнтів: забезпечення високої якості продукту та швидке та якісне обслуговування клієнтів принесе лояльність та збільшення ймовірностей повторної покупки;
- зручний та простий інтерфейс для користувачів, забезпечить легкість навігації та замовлення товарів;
- безпеку та захист персональних даних: безпека платіжних транзакцій та захист персональних даних клієнтів, допоможе зберегти їх довіру;
- вивчення конкуренції: вивчення конкурентів, їхні стратегії та ціни, допоможуть бути в курсі тенденцій ринку та забезпечити конкуренцію.

Інтернет-торгівля постійно розвивається, тому важливо бути в курсі останніх тенденцій і відповідним чином адаптувати свої стратегії. Регулярне аналізування власної продуктивності і відгуки клієнтів, допоможе внести необхідні покращення та зберегти конкурентну перевагу.

1.3 Дослідження проблематики малого бізнесу

Основна проблематика малого бізнесу полягає у тому, що більшість підприємців використовує для продажу лише соціальні мережі і месенджери, такі як Instagram та Viber. Відсутність сайту, в першу чергу, викликає недовіру, так як створити сторінку в соціальній мережі Instagram займає лише декілька хвилин і досить багато аферистів [5].

Наявність сайту це перш за все структурована інформація, з якою може ознайомитись покупець, а саме контакти компанії, інформацію про виробника. Також можна навести наступні переваги вебсайту:

- підвищення досяжності: Сайт дозволяє бізнесу бути доступним для клієнтів з будь-якого місця та в будь-який час, що дозволяє підвищити досяжність бізнесу та залучити більше клієнтів;
- збільшення продажів: Вебсайт дозволяє збільшити продажі, оскільки є можливість пропонувати товари або послуги в будь-який час та з будь-якого місця, незалежно від географічного місця знаходження клієнта;
- зниження витрат: Онлайн-торгівля дозволяє знизити витрати на зберігання та транспортування товарів, оскільки товари можуть бути зберіганні в одному місці та доставлятися відразу до клієнта;
- покращення обслуговування клієнтів: Сайт дозволяє клієнтам швидко та легко знайти інформацію про бізнес, товари та послуги, що покращує їхнє задоволення від обслуговування та збільшує шанси на повторні покупки;
- можливість використання маркетингових інструментів: За допомогою вебсайту є можливість використовувати різні маркетингові інструменти, такі як

SEO, контент-маркетинг, електронна розсилка та соціальні мережі, для залучення нових клієнтів та збільшення продажів;

– збереження часу: Створення вебсайту дозволяє заощадити час на обробці замовлень та виконанні платежів, оскільки всі ці процеси можуть бути автоматизовані.

Розглянемо більш детально стандарти для інтернет магазину з продажі тканин. Основна категорія людей, які купують тканини від 30 до 60 років. Виходячи з цього можемо сформуванати наступний перелік стандартів, які можуть бути корисними для сайту з продажу тканин[6]:

– якісні фотографії: Візуальний елемент є дуже важливим для продажу тканин, тому мають бути якісні фотографії, які показують кольори, текстуру та інші деталі тканини;

– опис товарів: Детальний опис кожного виду тканин. Це може включати склад, країну виробника, ширину тканини та інші характеристики;

– каталог товарів: Зручний каталог товарів, який дозволить клієнтам швидко знайти ту тканину, яку вони шукають;

– корзина покупок: Можливість додавання товарів у корзину покупок, а також зручний процес оформлення замовлення;

– зв'язок з клієнтами: Можливість зв'язку з клієнтами, наприклад, через електронну пошту, телефон або чат;

– ергономічний інтерфейс: Зручний інтерфейс, для того, щоб клієнти не губились у великій кількості елементів.

1.4 Аналіз інтернет-магазинів конкурентів

Знайдемо найпопулярніші інтернет-магазини тканин за допомогою пошукової системи Google.

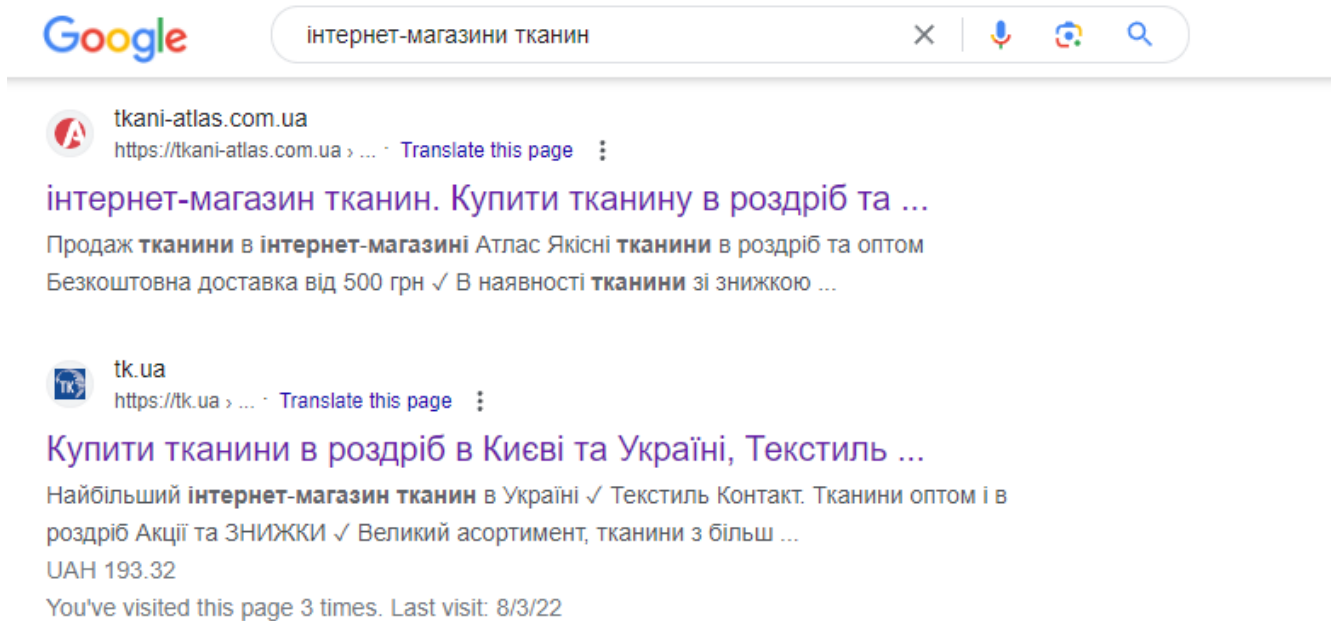


Рисунок 1.1 – Результат пошуку інтернет-магазинів

Проаналізуємо інтернет-магазини конкурентів. Першим розглянемо інтернет-магазин «ТК» [7]. На рисунку 1.2 наведено головну сторінку.

По-перше, згідно зі стандартами ергономічного інтерфейсу, варто підмітити накопичення інформації на головній сторінці. Для непросунутих користувачів, це великий недолік, так як виникає купа питань, а саме: «Куди натискати?» - і зазвичай у таких випадках кличуть більш просунутих користувачів, щоб допомогли їм розібратись [8].

По-друге, відсутність товарів на головні сторінці, один із можливих планів дій, які прийме користувач від 30 до 60 років, це покинути даний інтернет-магазин і знайти інший більш простий.

В результаті інтернет «ТК» втрачає потенційних клієнтів, лише через відсутність ергономічного дизайну.

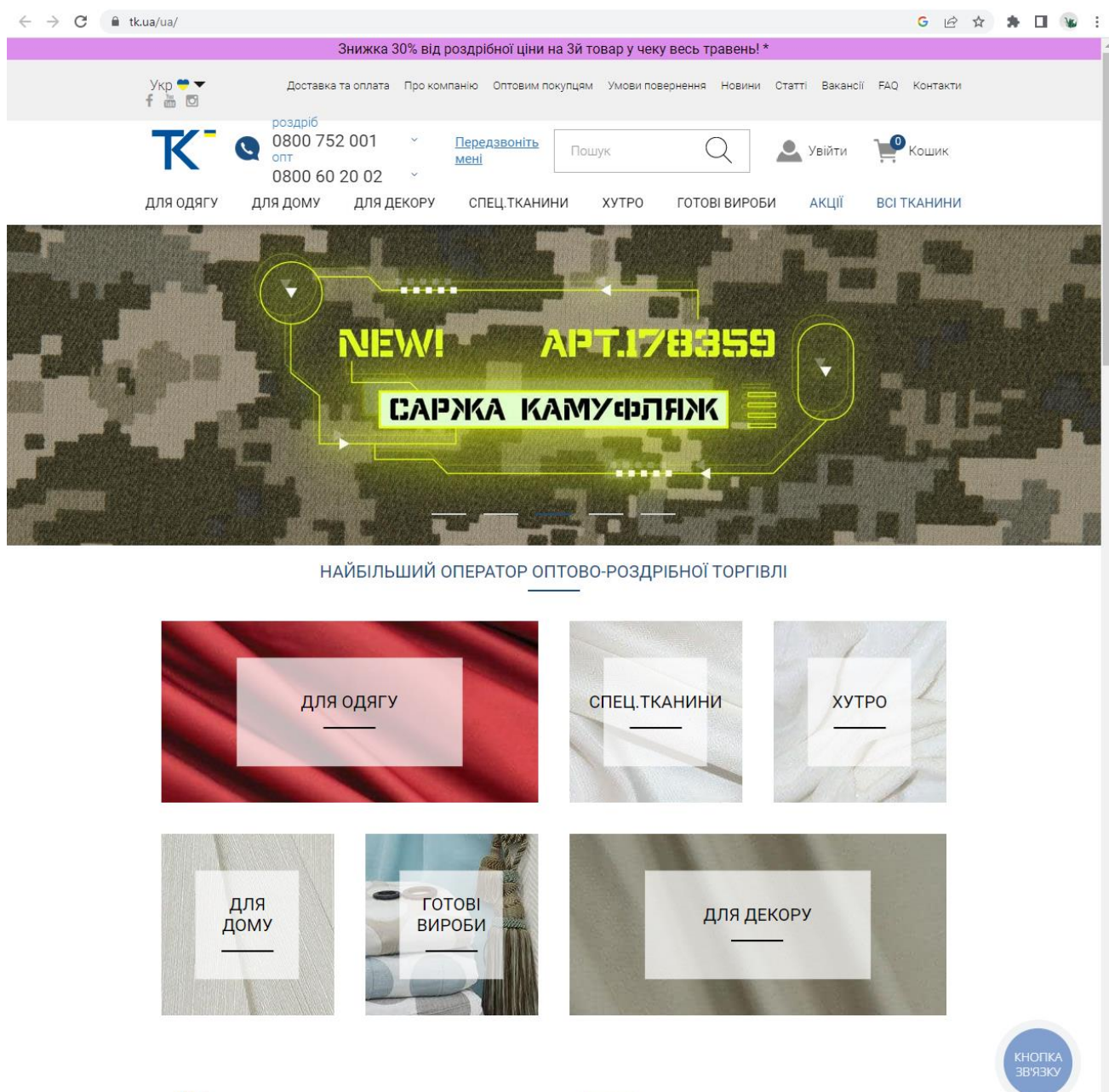


Рисунок 1.2 – Інтернет магазин «ТК»

Також розглянемо мобільну версію даного інтернет магазину на прикладі смартфона iPhone XR, так як більшість потенційних покупців переходять на веб-ресурс з мережі Instagram чи Viber. На рисунку 1.3 наведено фрагмент головної сторінки сайту.



Рисунок 1.3 – Мобільна версія магазину «ТК»

Перше, що варто відмітити це нерівномірно велику картинку, з логотипом компанії, яка займає майже весь простір на екрані, хоча не несе жодної корисної інформації, щоб зацікавити потенційного клієнта. Відсутність товару, що також суперечить стандартам ергономічного дизайну, так як потенційному покупцю треба зробити не відому кількість натисків, щоб отримати доступ до перегляду товару.

Розглянемо інтернет-магазин «АТЛАС» [9]. На рисунку 1.4 наведено фрагмент головної сторінки сайту. Згідно зі стандартами ергономічного інтерфейсу, він є зразковим. Хочу відмітити гарне рішення винести категорії тканин у ліву

частину екрану, так як на ній не акцентується увага, але в будь-який момент користувач може до нього звернутись, без зайвих рухів.

На головні сторінці, потенційний покупець одразу може розглянути товари, які його зацікавлять, що є неодмінною перевагою, серед інших інтернет магазинів.

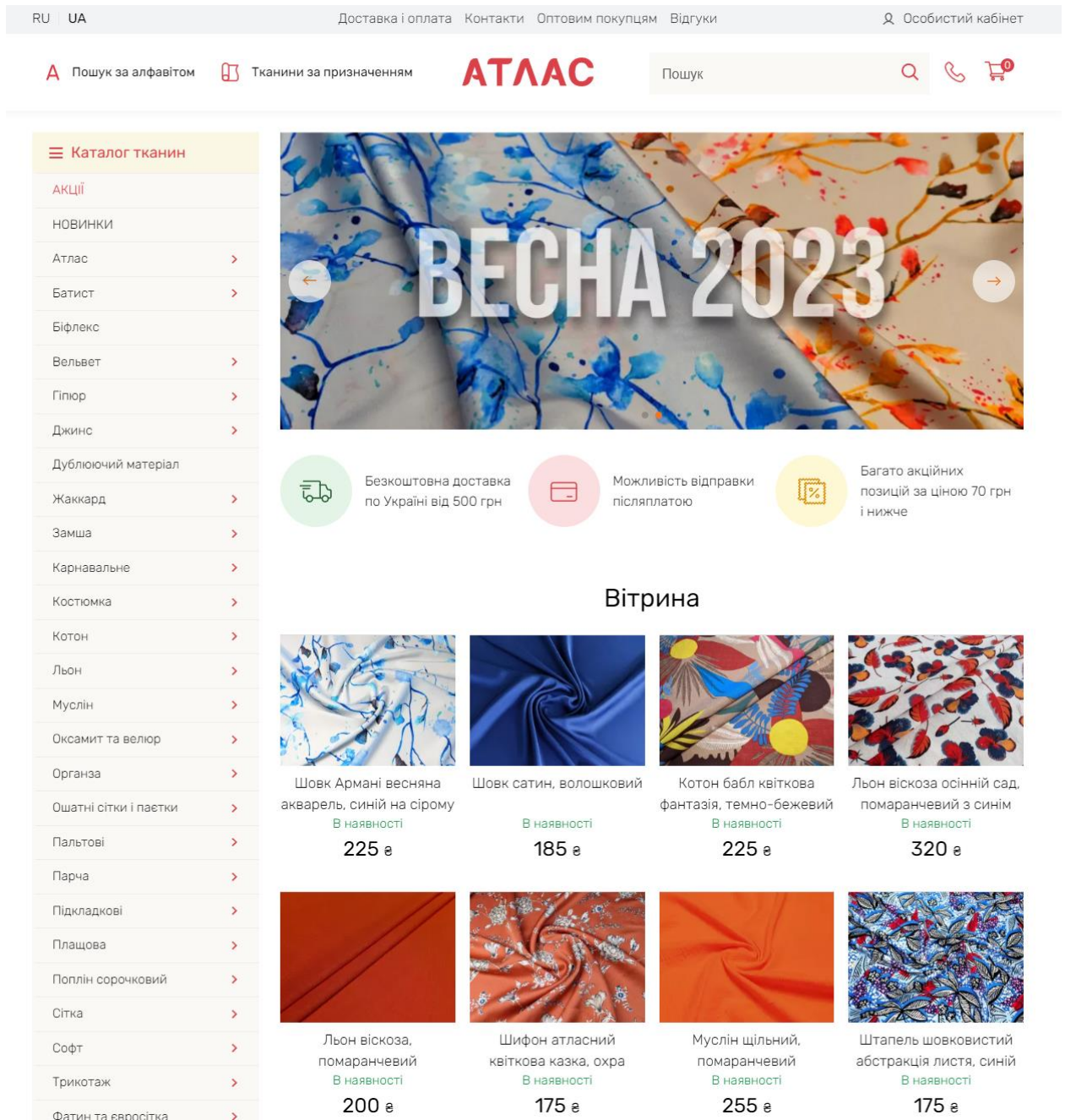


Рисунок 1.4 – Інтернет-магазин «АТЛАС»

Додатково розглянемо мобільну версію інтернет магазину «АТЛАС». На рисунку 1.5 наведено фрагмент головної сторінки сайту для смартфонів. Як і на версії для персональних комп'ютерів, ми отримаємо одразу товари для перегляду.

Хоча варто підмітити, що відступ між вітриною і акціями, варто зробити меншим, так як він надто великий.

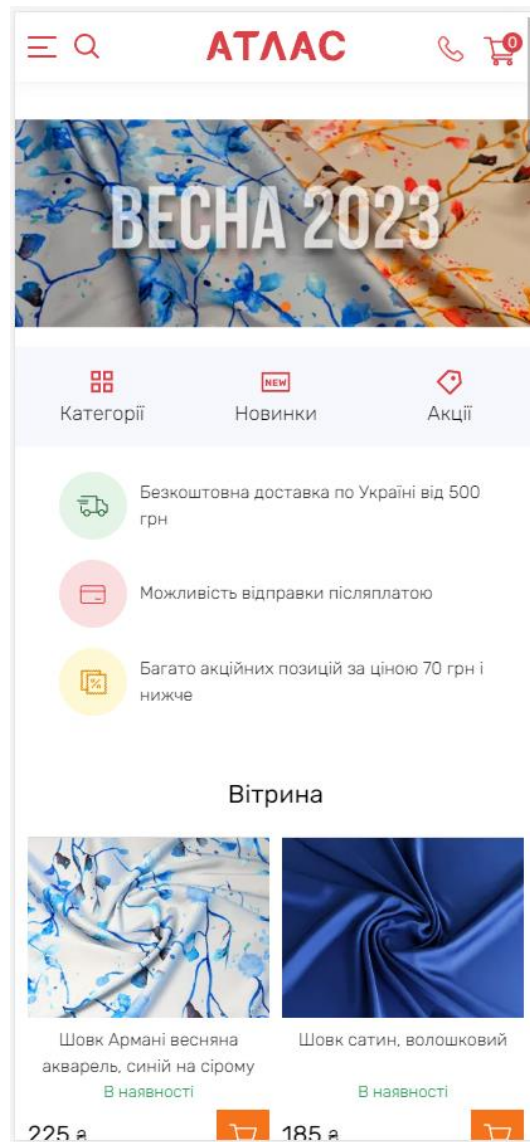


Рисунок 1.5 – Мобільна версія інтернет-магазину «АТЛАС»

Як результат, варто відмітити гарний ергономічний інтерфейс, який буде досить простим для потенційних покупців, що в майбутньому збільшить вірогідність повернення покупців, до даного інтернет магазину.

1.5 Специфікація проєкту

1.5.1 Вступ

1.5.1.1 Мета

Метою цього документа є визначення функціональних та нефункціональних вимог до інтернет-магазину тканин, розробленого на мові програмування Java за допомогою фреймворку Spring Boot.

1.5.1.2 Глядачі

Цей документ призначений для команди розробників, менеджерів проєктів, тестувальників, дизайнерів інтерфейсів та замовників системи.

1.5.1.3 Словник термінів

Товар: Тканина, що продається в інтернет-магазині.

Користувач: Особа, яка використовує інтернет-магазин.

Адміністратор: Користувач, який має права на управління інтернет-магазином.

Кошик: Список товарів, обрані користувачем для покупки.

1.5.2 Загальний опис

1.5.2.1 Контекст продукту

Інтернет-магазин тканин розробляється для підприємства, що займається продажем тканин різних видів, кольорів та фактур. Магазин має бути доступним для користувачів через веббраузери на настільних та мобільних пристроях.

1.5.3 Функціональні вимоги

1.5.3.1 Каталог товарів

Система повинна мати каталог товарів, що включає:

- перегляд списку товарів з фільтрацією та сортуванням;
- перегляд деталей окремого товару;
- можливість додавання товарів до кошика.

1.5.3.2 Кошик

Користувачі повинні мати можливість:

- додавати товари до кошика;
- видаляти товари з кошика;
- змінювати кількість товарів в кошику;
- оформляти замовлення на покупку товарів у кошику.

1.5.3.3 Авторизація та реєстрація

Система повинна дозволяти:

- реєстрацію нових користувачів;
- авторизацію користувачів за допомогою електронної пошти та пароля;
- відновлення пароля користувачів.

1.5.3.4 Особистий кабінет користувача

Користувачі повинні мати можливість:

- переглядати свої особисті дані та змінювати їх;
- переглядати історію замовлень та статус поточних замовлень.

1.5.3.5 Панель адміністратора

Адміністратори повинні мати можливість:

- керувати каталогом товарів (додавати, редагувати, видаляти товари);
- обробляти замовлення користувачів (перегляд, оновлення статусу, відміна);

1.5.4 Нефункціональні вимоги

1.5.4.1 Продуктивність

Система повинна підтримувати високу продуктивність та швидкість завантаження сторінок для користувачів.

1.5.4.2 Масштабованість

Система повинна бути масштабованою, щоб підтримувати збільшення кількості користувачів та замовлень.

1.5.4.3 Безпека

Система повинна захищати персональні дані користувачів та забезпечувати безпечні платежі.

1.5.4.4 Сумісність з браузером

Система повинна бути сумісною з різними сучасними веббраузерами, такими як Google Chrome, Mozilla Firefox, Safari та Microsoft Edge.

1.5.4.5 Адаптивність дизайну

Система повинна мати адаптивний дизайн, щоб коректно відображатися на різних пристроях та роздільних здатностях екранів.

1.5.5 Технічні вимоги

1.5.5.1 Мова програмування

Система повинна бути розроблена на мові програмування Java.

1.5.5.2 Фреймворк

Система повинна використовувати фреймворк Spring Boot.

1.5.5.3 База даних

Система повинна використовувати реляційну базу даних, сумісну з Java та Spring Boot, наприклад, MySQL, PostgreSQL або MariaDB для зберігання інформації про товари, замовлення та користувачів.

1.5.5.4 Інтеграція з платіжними системами

Система повинна інтегруватися з платіжними системами, такими як PayPal, Stripe або іншими, для безпечної обробки платежів користувачів.

1.5.6 Етапи розробки та впровадження

1.5.6.1 Розробка та тестування

- розробка основних компонентів системи;
- розробка вебінтерфейсу для користувачів та адміністраторів;
- тестування функціоналу та безпеки системи.

1.5.6.2 Розгортання та підтримка

- розгортання системи на віддаленому сервері;
- налаштування бази даних та підключення до платіжних систем;
- моніторинг та підтримка системи після запуску.

1.5.7 Висновок

Специфікація проєкту програмного забезпечення визначає основні функціональні та нефункціональні вимоги до інтернет-магазину тканин, розробленого на мові програмування Java за допомогою фреймворку Spring Boot. Вона служить основою для спілкування між розробниками, менеджерами проєктів,

тестувальниками, дизайнерами інтерфейсів та замовниками, забезпечуючи спільне розуміння вимог до продукту [10].

Висновки до розділу 1

Під час написання першого розділу, було досліджено проблематику створення сайтів для малого бізнесу. Проведення дослідження для формування стандартів інтернет-магазинів та в результаті було сформовано чіткі стандарти, які впливають на кількість потенційних покупців, а саме на основну вікову категорію людей віком від 30 до 60 років.

Було проаналізовано інтернет-магазини конкурентів, та виявлено основні недоліки, через які, частіше за все, магазин втрачає потенційних покупців.

Отже, на основі отриманих знань та сформованих знань буде розроблено інтернет-магазин з продажу тканин.

2 МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКУ З ПРОДАЖУ ТКАНИН

2.1 Вебсервери

Вебсервер - це програмне забезпечення або комп'ютер, яке надає послуги обробки запитів HTTP (Hypertext Transfer Protocol) [11]. Він відповідає на запити вебклієнтів, такі як веббраузери, і доставляє їм вебсторінки, файлів, мультимедійних даних та іншого контенту через мережу Інтернет. Сервер може приймати, опрацьовувати та зберігати файли(данні), надіслані від клієнта.

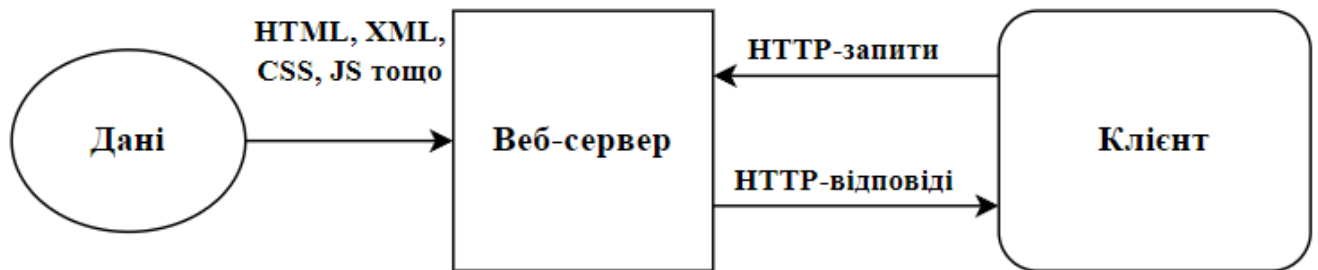


Рисунок 2.1 – Схематичне зображення роботи вебсервера

Глобальний рейтинг вебсерверів на 2023 рік [12]:

- Apache – 40.03%;
- Nginx – 28.35%;
- IIS – 10.49%;
- LiteSpeed – 2.75%;
- Apache Traffic Server – 0.51%.

Всі інші мають менше 0.5%.

Основні характеристики та функції вебсервера:

1) обробка HTTP-запитів: Вебсервер отримує HTTP-запити від клієнтів, такі як GET, POST, PUT, DELETE і т. д. Він обробляє ці запити та повертає відповідь клієнту;

2) доставка вебсторінок та контенту: Вебсервер може надавати статичні вебсторінки, що зберігаються на сервері, клієнтам безпосередньо. Він також може

виконувати обробку динамічних вебсторінок, які генеруються на льоту з використанням програмного коду, такого як PHP, Java, Python тощо;

3) управління сеансами та станом: Вебсервер може підтримувати механізми управління сеансами, які дозволяють зберігати стан клієнта між послідовними запитами. Це дозволяє зберігати інформацію про авторизацію, персоналізацію або інші деталі, що пов'язані з конкретним користувачем;

4) обробка безпеки: Вебсервери можуть забезпечувати захист від небажаного доступу та зберігання даних у безпечному режимі. Вони можуть використовувати протоколи шифрування, такі як HTTPS, для забезпечення конфіденційності та цілісності даних, а також механізми аутентифікації та авторизації для контролю доступу до ресурсів;

5) логування та моніторинг: Вебсервери можуть здійснювати запис журналів подій, який включає інформацію про запити, помилки, відповіді та іншу діагностичну інформацію. Це дозволяє відстежувати та аналізувати роботу сервера, а також здійснювати моніторинг його продуктивності та доступності.

Загалом, вебсервер є ключовим елементом вебінфраструктури та дозволяє користувачам отримувати доступ до вебсторінок та ресурсів через мережу Інтернет.

2.2 Протоколи передавання даних

Майже всі нові вебдодатки використовують більш безпечну, модернізовану версію HTTPS, який прийшов на заміну протоколу HTTP. Він дозволяє більш безпечно обмінюватись інформацією комп'ютерною мережею, так як майже всі його елементи шифруються. Дане шифрування гарантує помірний захист від підслуховування та від нападу «людина посередині», за умови це коректних налаштувань та підпису сертифікату авторизованим центром сертифікації [13].

Основні відмінності між HTTP і HTTPS:

1) безпека: Основна різниця полягає в безпеці передачі даних. У HTTPS дані шифруються, що ускладнює перехоплення та розшифрування третіми особами;

2) сертифікати: Для використання HTTPS потрібен SSL-сертифікат, який підтверджує, що вебсервер, до якого здійснюється з'єднання, є довіреним. Це допомагає уникнути атак типу "Man-in-the-Middle" (коли зловмисник стає проміжним ланцюжком між клієнтом і сервером);

3) URL: У HTTP URL починається з "http://", тоді як у HTTPS - з "https://". Це показує, який протокол використовується для передачі даних;

4) порти: HTTP працює на порту 80, тоді як HTTPS використовує порт 443. Це дозволяє серверу розрізнити, який протокол використовується клієнтом.

Узагальнюючи, HTTPS є безпечнішою версією протоколу HTTP, яка забезпечує шифрування та захист даних. Він широко використовується для передачі конфіденційної інформації, такої як особисті дані, паролі, фінансові та банківські відомості, з метою забезпечення приватності та запобігання несанкціонованому доступу до цих даних.

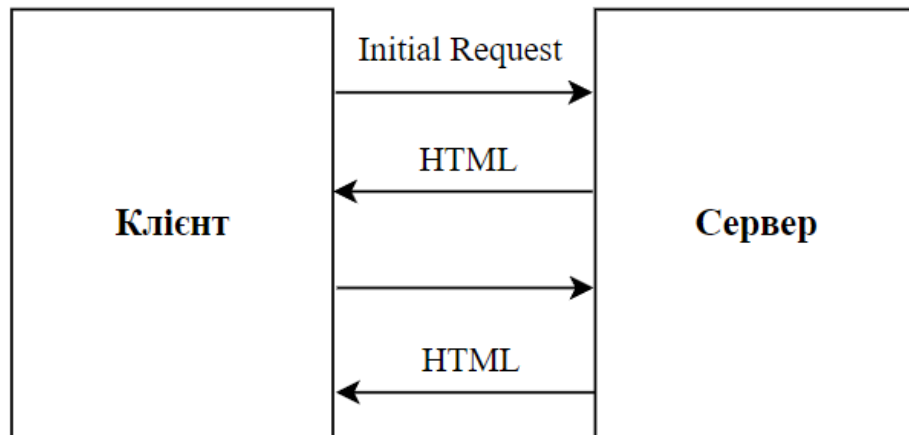
2.3 Підходи та технології

2.3.1 AJAX

AJAX (Asynchronous JavaScript and XML) - це технологія, що дозволяє вебсторінкам взаємодіяти з сервером асинхронно без необхідності перезавантаження сторінки [14]. Вона базується на комбінації різних технологій, таких як JavaScript, XML, HTML і CSS.

Основна ідея AJAX полягає в тому, що вебсторінка може відправляти запити до сервера, отримувати дані та оновлювати лише частини сторінки, які потребують оновлення, без необхідності перезавантаження всієї сторінки. Це дозволяє створювати більш інтерактивні та зручні вебдодатки з більш плавною і швидкою взаємодією з користувачем.

Принцип роботи багатосторінкового застосунку



Принцип роботи односторінкового застосунку

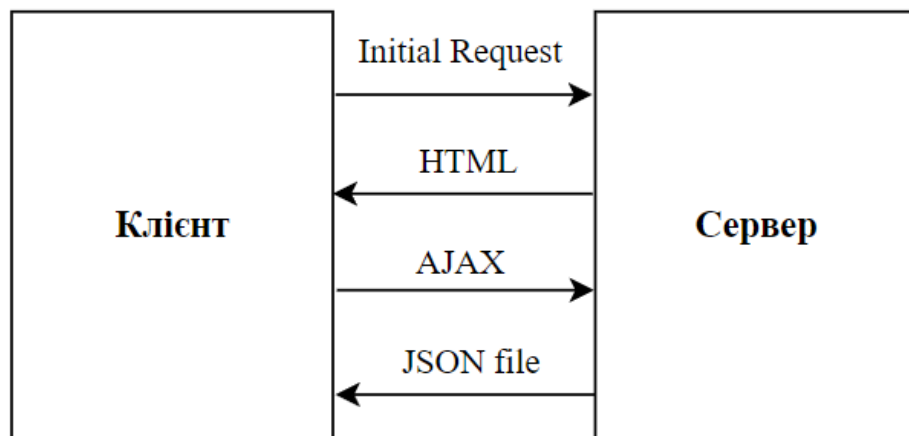


Рисунок 2.2 – Схематичне зображення роботи звичайних сайтів та односторінкових застосунків

Основні компоненти AJAX:

1) JavaScript: AJAX використовує JavaScript для взаємодії зі сторінкою та обробки подій. Він дозволяє виконувати асинхронні запити до сервера та маніпулювати вмістом сторінки без перезавантаження;

2) XMLHttpRequest: Це об'єкт JavaScript, що дозволяє вебсторінці взаємодіяти з сервером асинхронно. Він використовується для відправки запитів до сервера та отримання відповідей;

3) серверні технології: На серверній стороні використовуються різні технології, такі як PHP, Java, Python або .NET, для обробки запитів, взаємодії з базою даних та формування відповідей у форматі, зручному для передачі (наприклад, JSON або XML);

4) асинхронність: AJAX використовує асинхронний підхід, що означає, що він може відправляти запити до сервера та продовжувати виконувати інші дії без очікування відповіді. Коли відповідь отримана, вона може бути оброблена та відображена на сторінці;

5) обробка даних: Отримані відповіді від сервера можуть бути оброблені та використані для оновлення сторінки або взаємодії з іншими елементами сторінки. Це може включати оновлення тексту, відображення списку даних, динамічне валідування форми та інше.

Застосування AJAX включають завантаження даних з сервера без перезавантаження сторінки, відправку форм без оновлення сторінки, автозаповнення полів форми, реалізацію пошуку в режимі реального часу та багато іншого [15].

AJAX дозволяє створювати більш динамічні та ефективні вебдодатки, які забезпечують покращену користувацьку взаємодію та швидку відповідь на дії користувача.

2.3.2 MVC

MVC (Model-View-Controller) - це архітектурний шаблон проектування програмного забезпечення, що використовується для розділення компонентів додатку на три основні частини: модель (Model), представлення (View) та контролер (Controller)[16]. Цей шаблон сприяє покращенню організації коду, забезпеченню його модульності та підтримці відокремленості логіки, що дозволяє покращити розширюваність та обслуговування додатку [17-18].

Модель (Model): Модель представляє дані та бізнес-логіку додатку. Вона відповідає за обробку, збереження та валідацію даних. Модель може включати

класи, об'єкти бази даних, сервіси та інші компоненти, необхідні для роботи з даними.

Представлення (View): Представлення відповідає за відображення даних користувачеві. Це може бути вебсторінка, графічний інтерфейс або будь-який інший спосіб відображення інформації. Представлення отримує дані з моделі та відображає їх у зрозумілій формі для користувача.

Контролер (Controller): Контролер відповідає за обробку вхідних подій та взаємодію з моделлю та представленням. Він приймає запити від користувача, виконує відповідні дії, оновлює модель та вибирає відповідне представлення для відображення оновлених даних користувачеві. Контролер виступає в якості посередника між моделлю та представленням.

Шаблону MVC розділяє різні аспекти програмного забезпечення, такі як логіка, візуалізація та взаємодія з користувачем. Це полегшує розробку, тестування і підтримку програмного забезпечення, оскільки зміни в одному компоненті не впливають на інші компоненти безпосередньо.

Майже усі популярні мови програмування та фреймворки надають підтримку шаблону MVC або його варіацій, що дозволяє розробникам ефективно використовувати цей шаблон для побудови структурованих та легко змінюваних додатків.



Рисунок 2.3 – Схема роботи MVC

Основна ідея MVC полягає в тому, що модель, представлення та контролер мають різні відповідальності та повинні бути відокремлені один від одного.

2.4 Технології для розробки

Малий бізнес може замовляти сайти на різних мовах програмування, проте деякі мови програмування є більш поширеними серед цільової аудиторії малого бізнесу. Серед мов програмування, які часто використовуються для розробки сайтів малим бізнесом, популярними є: PHP, Python, Java, JavaScript. Оглянемо деякі з них.

2.4.1 PHP

PHP (Hypertext Preprocessor) - це популярна мова програмування загального призначення, яка використовується для розробки вебдодатків та динамічних вебсторінок. Вона була створена Расмусом Лердорфом у 1994 році і вперше називалася "Personal Home Page" (особиста домашня сторінка). З того часу PHP

значно розширилася і стала однією з найпопулярніших мов програмування для веброзробки. Вона має широку підтримку, багато ресурсів та фреймворків, які полегшують створення сайтів. PHP часто використовується для розробки простих інформаційних сайтів, блогів або малих електронних магазинів [19].

Варто підмітити, що PHP надає багатий набір функцій і бібліотек для роботи з веброзробкою. Вона може генерувати HTML-код, працювати з формами, встановлювати з'єднання з базами даних і здійснювати інші операції, необхідні для розробки вебдодатків.

Синтаксис PHP схожий на C, що робить його відносно простим для вивчення і розуміння. Багато розробників, які мають досвід роботи з C, C++ або Java, легко адаптуються до PHP.

PHP має вбудовану підтримку багатьох популярних систем управління базами даних, таких як MySQL, PostgreSQL, SQLite і багато інших. Це дозволяє зручно працювати з базами даних і здійснювати розширені операції над ними.

PHP можна легко розширити за допомогою додаткових модулів і бібліотек. Існує велика кількість розширень PHP, які додають нові функції і можливості.

2.4.2 Python

Django – як і всі описані до нього є опен-сорсом(відкритим кодом). Як і Ruby on rails, потребує малої кількості коду для опису логіки, іншу роботу автоматизує. Реалізує дещо змінену модель MVC – MTV(model-template-view), вона використовує більшу модульність. Один з основних принципів фреймворка - DRY (Do not repeat yourself – не повторюй себе). Завдяки популярності мови на якій працює, може містити додаткові утиліти такі як, наприклад, wkhtmltopdf, що реалізує просту та ефективну конвертацію вебсторінок у PDF-файли [20].

Додає адміністрацію сайту за замовчуванням, оскільки початково розроблювався для використання у новинних сайтах.

Може реалізовувати великі проекти з високим навантаженням та великим об'ємом.

Python є потужною мовою програмування загального призначення, яка також добре підходить для веброзробки. Хоча Python не був спеціально розроблений для веброзробки, він набув великої популярності в цій сфері завдяки своїм перевагам. Ось деякі ключові аспекти Python у веброзробці:

- простота вивчення і читабельність;
- широкий вибір вебфреймворків;
- багаті бібліотеки і модулі;
- розширюваність і підтримка сторонніх інструментів;
- бібліотека для роботи форм;
- інтернаціоналізація.

Python пропонує багато інших можливостей для веброзробки, таких як вбудована підтримка JSON, високий рівень абстракції, простота розгортання та гнучкість. Завдяки цим перевагам Python є популярним вибором для розробки вебдодатків різного рівня складності [21].

2.4.3 Java

Java є потужною мовою програмування, яка також використовується для веброзробки. Хоча Java спочатку була розроблена для створення незалежних від платформи додатків, вона також має широке застосування у веброзробці. Ось деякі ключові аспекти Java у веброзробці [22]:

- Java Enterprise Edition (Java EE) і фреймворки;
- велика кількість бібліотек і пакетів;
- безпека і надійність;
- платформна незалежність;
- підтримка вебсерверів і контейнерів.

Java веброзробка відома своєю масштабованістю, надійністю та широким вибором фреймворків та бібліотек. Вона використовується для створення великих корпоративних систем, електронних комерційних платформ, вебслужб та багато іншого.

2.4.4 JavaScript

JavaScript є однією з найбільш важливих мов програмування для веброзробки. Вона використовується для розробки інтерактивних вебсторінок, вебдодатків та іншого фронтенд функціоналу. Ось деякі ключові аспекти JavaScript у веброзробці [23]:

- клієнтський скриптинг;
- фреймворки та бібліотеки;
- взаємодія з сервером;
- розробка мобільних додатків;
- розширення можливостей браузера.

JavaScript є незамінним інструментом у веброзробці, що дозволяє створювати динамічні та інтерактивні вебдодатки. Його поширена підтримка в браузерах, широкий вибір фреймворків та бібліотек, а також швидкий розвиток технологій роблять JavaScript одним з основних інструментів у веброзробці.

Серед усіх цих мов програмування варто виділити Java, так як відома своєю надійністю та стабільністю. Вона має потужну вбудовану підтримку для обробки великих обсягів даних та паралельного виконання, що може бути корисним для масштабування сайту в майбутньому. Це допоможе легко розширити мережу інтернет-магазинів, та зробити мобільний застосунок, бо Java є багатоплатформовою мовою, що означає, що програми, написані на Java, можуть працювати на різних операційних системах без необхідності переписування коду. Це може бути важливим фактором для малого бізнесу, який може мати різні типи серверів або платформ.

Ключовим фактором інтернет-торгівлі є безпека клієнтських, тільки так можна тримати високий рівень довіри. Для цього мова програмування Java має вбудовані механізми безпеки, що робить її популярною для розробки сайтів, які потребують захисту конфіденційної інформації, такої як особисті дані клієнтів або фінансові транзакції.

З точки зору розробника вибір Java є наймовірною перевагою, так як велика спільнота та екосистема допоможе у вирішенні певних проблем, з якими розробник

неодмінно зіштовхнеться. Це означає, що малому бізнесу можна легше знайти ресурси та підтримку при розробці та підтримці вебсайту на Java.

2.5 Вибір фреймворку

Мова програмування Java має різноманітні фреймворки, які спрощують розробку вебдодатків та прискорюють процес програмування. Ось кілька популярних фреймворків Java [24].

Spring Framework є одним з найпопулярніших фреймворків Java. Він надає широкий спектр інструментів і модулів для розробки вебдодатків, включаючи вебсайти. Spring дозволяє легко управляти залежностями, реалізувати інверсію керування (IoC) та аспектно-орієнтоване програмування (AOP). Крім того, в Spring є модулі, такі як Spring MVC і Spring Boot, які спрощують розробку вебдодатків і створення самостійних довантажених додатків [25].

Java Server Faces (JSF) є стандартним фреймворком Java для розробки вебінтерфейсів користувача. Він забезпечує шаблони та компоненти для побудови вебсторінок та форм. JSF спрощує взаємодію з базою даних, обробку подій та валідацію даних. Він дозволяє розробникам зосередитися на вебінтерфейсі, не думаючи про деталі роботи з HTTP-протоколом або обробку форм [26].

Hibernate є фреймворком об'єктно-реляційного відображення (ORM), що дозволяє взаємодіяти з базами даних, використовуючи об'єктно-орієнтований підхід. Він дозволяє розробникам працювати з об'єктами Java, а не з SQL-запитами безпосередньо. Hibernate автоматично створює SQL-запити, виконує управління транзакціями та забезпечує рівень абстракції між базою даних і додатком [27].

Проаналізувавши фреймворки, варто виділити Spring Framework та Hibernate. Основними перевагами для самостійної розробки буде підпроект Spring Boot, який спрощує розробку самостійних, готових до використання додатків. Він забезпечує автоматичну конфігурацію та підтримку вбудованих серверів, що дозволяє розробникам швидко створювати додатки з меншими зусиллями.

Spring використовує принцип IoC, що дозволяє розробникам управляти залежностями між компонентами додатку. Замість того, щоб зручно створювати об'єкти та їх залежності, Spring надає контейнер, який автоматично керує життєвим циклом об'єктів та їх взаємозв'язками.

Додатково варто виділити Spring MVC. Це модуль Spring Framework, який надає підтримку для розробки вебдодатків на основі моделі-вид-контролер (MVC). Spring MVC забезпечує розділення бізнес-логіки, представлення та обробку запитів вебклієнтів. Він надає багато корисних функцій, таких як мапінг URL-адрес до контролерів, обробка форм, валідація та багато іншого.

Для роботи з базою даних, для зручності, використаємо фреймворк Hibernate. Це фреймворк об'єктно-реляційного відображення (ORM) для мови програмування Java. Він надає спрощений спосіб взаємодії з базами даних, де реляційна модель бази даних перетворюється на об'єктну модель, що використовується в програмі.

Нижче написані основні особливості та компоненти Hibernate [28]:

1) об'єктно-реляційне відображення (ORM): Hibernate дозволяє розробникам працювати з об'єктами Java, а не з SQL-запитами безпосередньо. Він автоматично виконує відображення між об'єктами та таблицями бази даних, а також між відношеннями між об'єктами та зв'язками між таблицями;

2) керування транзакціями: Hibernate надає підтримку для керування транзакціями у додатках. Він дозволяє використовувати механізми транзакційної безпеки, такі як ACID (атомарність, консистентність, ізольованість, довершеність), для забезпечення цілісності даних та виконання операцій у безпечному режимі;

3) мови запитів: Hibernate підтримує різні мови запитів, такі як Hibernate Query Language (HQL) та Native SQL. HQL - це об'єктно-орієнтована мова запитів, схожа на SQL, але використовує об'єктні імена та відношення між об'єктами замість таблиць та стовпців;

4) Hibernate підтримує механізми кешування для покращення продуктивності додатків. Він може кешувати часті запити та результати, що дозволяє зменшити навантаження на базу даних та прискорити виконання запитів;

5) міграції баз даних. Hibernate надає зручний спосіб оновлення схеми бази даних з використанням механізму міграцій. Він автоматично створює або змінює таблиці та структури бази даних відповідно до змін в об'єктній моделі без необхідності вручну писати SQL-запити.

Hibernate дозволяє розробникам зосередитися на бізнес-логіці додатків, а не на деталях взаємодії з базою даних, спрощуючи розробку та підтримку додатків на мові Java. Він є потужним інструментом для роботи з базами даних у Java-додатках.

2.6 Бази даних

При веброботі використовуються різні типи баз даних залежно від потреб проєкту. В основному їх поділяють на два основних типа: реляційні (Relational databases) та нереляційні (NoSQL databases) бази даних. Приклади реляційних баз даних включають MySQL, PostgreSQL, Oracle та Microsoft SQL Server. Вони забезпечують надійність, цілісність даних та можливості для складних операцій, таких як з'єднання таблиць, індексування та транзакції [29].

Нереляційні бази даних стають все більш популярними у веброботі. Вони пропонують неструктуроване зберігання даних та гнучкі моделі. Нереляційні бази даних підтримують такі типи баз даних, як документні бази даних (наприклад, MongoDB), ключ-значення (наприклад, Redis), стовпчикові (наприклад, Apache Cassandra) та графові (наприклад, Neo4j). Вони особливо корисні для зберігання та обробки великих обсягів неструктурованих даних, таких як соціальні мережі, логи, сесії та інші.

Слід більш детально розглянути найбільш поширений тип баз даних.

Реляційні бази даних (RDBMS - Relational Database Management System) - це тип баз даних, який використовує структурований підхід до зберігання даних. Вони базуються на реляційній моделі даних, запропонованій Едгаром Коддом у 1970 році. Основною одиницею даних в реляційних базах даних є таблиці, які складаються з рядків (записів) та стовпців (полів).

Основні поняття реляційних баз даних:

– таблиці: Дані в реляційній базі даних зберігаються у вигляді таблиць. Кожна таблиця має назву та набір стовпців, які визначають типи даних, які можуть бути збережені в цих стовпцях. Кожен рядок таблиці представляє окремий запис зі значеннями полів для цього запису;

– ключі: Ключі використовуються для ідентифікації унікальних записів в таблиці. Основний ключ (Primary Key) є унікальним ідентифікатором для кожного запису в таблиці. Зовнішні ключі (Foreign Key) використовуються для зв'язку двох таблиць шляхом посилання на значення основного ключа однієї таблиці в полі іншої таблиці;

– запити: Запити SQL (Structured Query Language) використовуються для взаємодії з реляційною базою даних. SQL надає мову запитів, яка дозволяє виконувати операції вибірки, вставки, оновлення та видалення даних з таблиць. Запити SQL можуть бути складними і включати умови, з'єднання таблиць, групування, сортування та інші операції;

– нормалізація: Нормалізація є процесом організації даних в реляційних таблицях для забезпечення ефективності, цілісності та оптимальності бази даних. Вона включає розбиття таблиць на менші логічні частини та усунення повторюваних даних, щоб забезпечити кращу структуру бази даних;

– транзакції: Транзакції використовуються для забезпечення цілісності даних та виконання операцій як атомарних, консистентних, ізольованих та довіреної (ACID) одиниць. Транзакція - це група запитів, які виконуються як єдина одиниця. Якщо будь-який запит у транзакції не може бути виконаний, то всі зміни відміняються (rollback), інакше вони фіксуються (commit).

Це лише кілька ключових аспектів реляційних баз даних. Реляційні бази даних мають широке застосування у веброзробці та дозволяють зберігати, управляти та отримувати доступ до даних ефективно та надійно.

Структурована мова запитів (SQL) — це мова програмування для зберігання та обробки інформації в реляційній базі даних. Реляційна база даних зберігає інформацію в табличній формі, у якій рядки та стовпці представляють різні

атрибути даних і різні зв'язки між значеннями даних. Ви можете використовувати оператори SQL для зберігання, оновлення, видалення, пошуку та отримання інформації з бази даних. Ви також можете використовувати SQL для підтримки та оптимізації продуктивності бази даних.

SQLite – це легка, вбудовувана база даних, яка не вимагає окремого сервера та налаштувань. Вона зберігає всю базу даних у єдиному файлі, що дозволяє легко переміщувати та розповсюджувати її. Однак, SQLite підтримує багато з основних можливостей реляційних баз даних, включаючи транзакції, індекси, підзапити та багато інших.

Основні переваги SQLite:

- легкість використання і налаштування;
- відсутність необхідності в окремому сервері;
- швидкість роботи завдяки простоті структури та використанню локального файлу бази даних;
- портативність, оскільки база даних зберігається в одному файлі.

SQLite найчастіше використовується у вбудованих системах, мобільних додатках та невеликих проєктах, де не потрібні великі обсяги даних або багатопотоковий доступ до них.

MySQL - це потужна, відкрита реляційна база даних, яка забезпечує широкі можливості для зберігання, управління та отримання доступу до даних. Вона працює на основі клієнт-серверної архітектури, де сервер MySQL відповідає за обробку запитів та збереження даних, а клієнтські програми взаємодіють з базою даних через мову запитів SQL.

Основні переваги MySQL:

- висока продуктивність та швидкодія, підтримка оптимізації запитів та індексів;
- розширюваність та масштабованість, дозволяє обробляти великі обсяги даних та велику кількість одночасних підключень;

- надійність та безпека, має можливості резервного копіювання, відновлення даних та захисту від несанкціонованого доступу;
- багатий функціонал, включаючи транзакції, підтримку зовнішніх ключів, операції з об'єднаннями та багато інших.

MySQL є однією з найпоширеніших баз даних у веброзробці. Вона використовується для розробки вебсайтів, вебдодатків, систем управління вмістом (CMS), електронної комерції та багатьох інших проєктів.

2.7 Інтерфейс

Для фронтенду використовуються стандартні сторінки написані на HTML та CSS та з додаванням JS. Для написання сторінки потрібно знати певний перелік тегів та структуру її створення.

Для часткової автоматизації під час створення сторінки існують утиліти, наприклад Bootstrap, що надає інструменти для створення зовнішньої частини сайту. Bootstrap - це потужний та популярний фреймворк для розробки вебінтерфейсів. Він забезпечує швидку та ефективну розробку вебсторінок, що адаптуються до різних пристроїв і розмірів екранів, таких як комп'ютери, планшети та мобільні пристрої. Bootstrap генерує HTML та CSS код.

Bootstrap містить велику кількість готових стилів, які можна застосовувати до різних елементів інтерфейсу. Це включає в себе кнопки, форми, таблиці, навігаційні панелі, картки, модальні вікна та багато інших. Готові стилі спрощують розробку і забезпечують єдиноформний вигляд вебсторінок.

Одним з головних переваг Bootstrap є його адаптивність. Він надає гнучкі класи та компоненти, які допомагають сторінкам коректно відображатися на різних пристроях та екранах різних розмірів. Це спрощує розробку мобільних версій вебсайтів та підвищує користувацький комфорт.

Bootstrap має багато вбудованих JavaScript-компонентів, які розширюють функціональність вебсторінок. Це включає в себе переключення вкладок,

спливаючі повідомлення, слайдери, акордеони, каруселі, валідацію форм та інші. Вони дозволяють додати інтерактивність та динаміку до вебінтерфейсу.

Також існує шаблонізатор Jinja, що використовується у Flask, та Django. Jinja є шаблонізатором для мови програмування Python. Він дозволяє розділити логіку програми від представлення, що є важливим принципом проєктування MVC (Model-View-Controller). Jinja надає можливість створювати шаблони HTML-сторінок, де можна вставляти змінні, виконувати умови та цикли, робити наслідування шаблонів та багато іншого.

Django є повноцінним вебфреймворком, який базується на мові програмування Python. Він надає набір інструментів та конструкцій, що допомагають швидко розробляти складні вебдодатки. Django включає в себе багато функціональних можливостей, включаючи роботу з базами даних, автентифікацію, URL-маршрутизацію, шаблонізацію та багато іншого.

Django є потужним інструментом для розробки вебдодатків з використанням Python. Він широко використовується для будівництва вебсайтів, соціальних мереж, електронної комерції та багатьох інших типів вебдодатків.

2.8 Тимчасові файли

Тимчасові файли – це одна з технологій, з якою користувачі інформаційних систем стикаються щоденно. Вони присутні в додатках, програмному забезпеченні і на рівні комплектуючих.

Кешування - це процес зберігання тимчасової копії даних з метою їх подальшого швидкого доступу. Кеш може бути використаний для зберігання різних типів даних, включаючи HTML-сторінки, зображення, стилі, скрипти та інші ресурси, які використовуються вебсайтом або додатком.

Основна ідея кешування полягає в тому, що коли клієнт (браузер або інший клієнтський програмний засіб) запитує ресурс, сервер спочатку перевіряє, чи є відповідна копія цього ресурсу у кеші. Якщо копія є, сервер повертає цю копію клієнту, що дозволяє уникнути повторного обчислення або завантаження ресурсу

знову. Це приводить до покращення продуктивності, зменшення навантаження на сервер та скорочення часу відгуку для клієнтів.

Для кешування статичних файлів використовуються сервери доставки контенту «CDN» (content delivery network). Вони, розташовані по всьому світу і оптимізовані для роздачі таких статичних файлів. Іноді буває ефективніше покласти картинки, відео, JS-скрипти на CDN.

Використання кешування має свої переваги і викликає ряд викликів. Переваги включають покращення продуктивності, зменшення навантаження на сервер, поліпшення швидкодії вебсайту та поліпшення враження користувачів. Однак, необхідно бути уважним при використанні кешування, оскільки некоректна робота кешу може призвести до неправильного відображення даних або застарілих результатів.

Висновки до розділу 2

Проаналізувавши вище описане, можна дійти висновку, що для різних типів задач існують різні фреймворки та мови програмування і неможливо під всі задачі підібрати один підхід.

Під час аналізу технологій розробки було досліджено, що найкращим варіантом буде обрати мову програмування Java, так як вона має вбудовані механізми безпеки, що робить її популярною для розробки сайтів, які потребують захисту конфіденційної інформації, такої як особисті дані клієнтів або фінансові транзакції. В результаті інтернет-магазин матиме бездоганну репутацію з точки зору безпеки даних користувачів.

Для розробки функціоналу серверної частини сайту, було обрано Spring фреймворк. Основними перевагами є підпроект Spring Boot, який спрощує розробку самостійних, готових до використання додатків.

Для бази даних було обрано MySQL.

Для зручності оперування даними було обрано фреймворк Hibernate, який дозволяє звертатись до даних, як до об'єктів, уникаючи написання SQL-запитів.

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Моделі бази даних

Для роботи з даними було створено реляційну базу даних в MySQL та створено декілька моделей.

База даних складається з кількох таблиць, кожна з яких має унікальне ім'я. Кожна таблиця містить набір рядків, які представляють записи, і стовпців, які визначають атрибути цих записів.

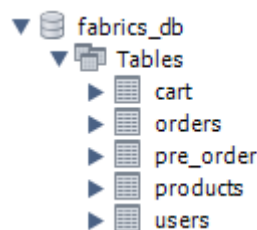


Рисунок 3.1 – База даних fabrics_db

Розглянемо моделі, які нам знадобляться для функціонування інтернет-магазину.

Модель cart(товари в кошику) має наступні поля:

- id – первинний ключ (нумерація);
- user_id – первинний ключ моделі users;
- id_prod – первинний ключ моделі products;
- quantity – кількість одиниць товару;
- add_date – дата додавання товару.

Модель orders(замовлення) має наступні поля:

- id – первинний ключ (нумерація);
- user_id – первинний ключ моделі users;
- total_sum – загальна вартість;
- first_name – ім'я замовника;
- last_name – прізвище замовника;
- middle_name – по-батькові;

- address – адреса доставки;
- phone_num – номер телефону замовника;
- comment – коментар до замовлення;
- status – статус замовлення;
- delivery – назва служби доставки;
- user_ac_id – електронна пошта.

Модель pre_order(зберігає кожен товар та ціни окремо) має наступні поля:

- id – первинний ключ (нумерація);
- user_email – електронна пошта користувача;
- id_prod - первинний ключ моделі products;
- count – кількість одиниць товару;
- price – ціна за одну позицію товару;
- order_id – первинний ключ моделі orders.

Модель products (тканини):

- id – первинний ключ (нумерація);
- p_type – вид тканини;
- title – назва тканини;
- description – опис тканини;
- price – ціна;
- p_w_sale – ціна зі знижкою;
- stocks – кількість тканини на складі;
- stock_status – статус наявності тканини (в наявності, закінчується, немає в наявності);

- date_posting – час додавання тканини;
- country – країна виробника;
- composition – вміст тканини;
- width – ширина тканини;
- density – щільність тканини;
- image1 – перша фотографія тканини;

- image2 – друга фотографія тканини;
- image3 – третя фотографія тканини;

Модель users (користувачі):

- id – первинний ключ (нумерація);
- first_name – ім'я користувача;
- last_name – прізвище користувача;
- middle_name – по-батькові;
- email – електронна пошта;
- phone – номер телефону;
- status – статус користувача (BANNED, ACTIVE);
- role – роль користувача (USER, ADMIN);
- password – пароль;
- reset_password_token – токен для відновлення паролю.

Для того, щоб зберегти унікальність акаунтів, було створено умову унікальності для колонок email та reset_password_token моделі users.

Для моделей pre_order, orders та cart були встановлені обмеження щодо пустих рядків. Якщо відсутнє хоча б одне значення, запис додано не буде.

Задано вторинні та первинні ключі, вони використовуються для звернення до даних з різних таблиць за допомогою ключів. Наприклад, модель "orders" має зовнішній ключ, який посилається на головний ключ моделі "users", щоб встановити зв'язок між замовленнями і користувачами.

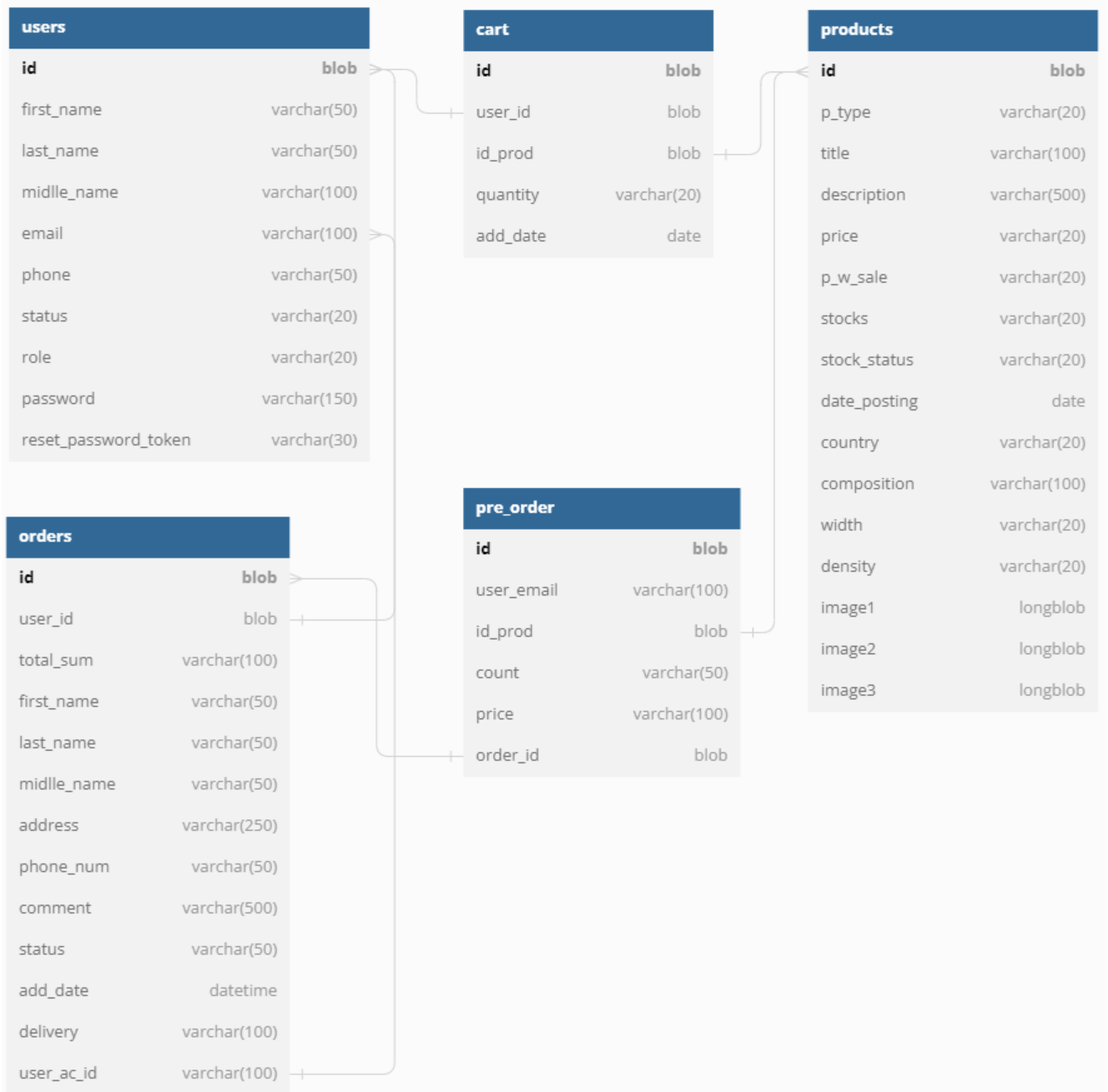


Рисунок 3.2 – Структура реляційної бази даних fabrics_db

В результаті отримали структуру бази даних.

3.2 Підключення БД

Для майбутніх операцій треба підв'язати дану БД fabrics_db до проєкту. В файлі application.properties задамо наступні параметри:

- spring.jpa.hibernate.ddl-auto=update – доступ до даних лише для читання та оновлення даних;

- `spring.datasource.url=jdbc:mysql://localhost:3306/fabrics_db` – адреса бази даних;
- `spring.datasource.username=root` – логін від БД;
- `spring.datasource.password=root` – пароль від БД;
- `spring.datasource.driver-class-name=com.mysql.jdbc.Driver` – вказує клас Driver;
- `spring.servlet.multipart.max-file-size=6MB` – обмеження розміру для завантаження зображення;
- `spring.servlet.multipart.max-request-size=6MB` – обмеження розміру для отримання зображення.

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/fabrics_db
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.servlet.multipart.max-file-size=6MB
spring.servlet.multipart.max-request-size=6MB
```

Рисунок 3.3 – Файл application.properties

В результаті отримуємо підключену БД до нашого проєкту. Для того щоб оперувати даними треба створити java класи для кожної моделі.

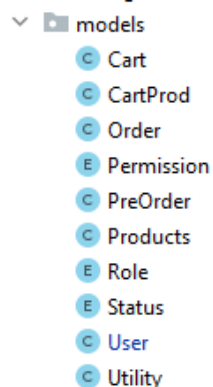


Рисунок 3.4 – Моделі

Для того, щоб користуватись фреймворком Hibernate, треба створити для кожної моделі репозиторій (java інтерфейс).

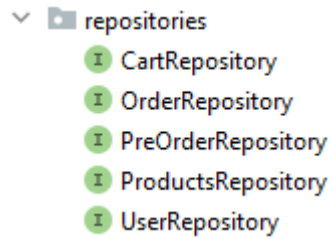


Рисунок 3.5 – Репозиторії для кожної моделі

Щоб вказати, які саме типи SQL запитів створювати, треба розширити інтерфейсами `JpaRepository` або `CrudRepository`. Вони є інтерфейсами, які надають спрощений спосіб доступу до операцій зберігання та витягування даних з бази даних у фреймворку Spring Data.

Отже, основна різниця між ними полягає у вмісті методів та можливостях, які вони надають.

`CrudRepository` є базовим інтерфейсом, який надає основні CRUD (створення, читання, оновлення, видалення) операції для роботи з базою даних. Він містить методи, такі як `save()`, `findOne()`, `findAll()`, `delete()`, тощо. Цей інтерфейс є простим у використанні та підходить для більшості випадків, коли потрібно виконати базові операції зберігання даних [30].

`JpaRepository` розширює `CrudRepository` і додає додаткові методи для роботи з базою даних. Цей інтерфейс надає розширені можливості, такі як сортування, пагінація, виконання замовних запитів тощо. Крім того, він також має вбудовані методи для виконання типових операцій, таких як пошук за умовою, розрахунок кількості записів, видалення за умовою та інші [30].

Отже, для роботи краще обрати `JpaRepository`, так як він має розширений функціонал.


```
import ...
8 usages  ↗ TENOVID
@Repository
public interface UserRepository extends JpaRepository<User, Long> {
    12 usages  ↗ TENOVID
    Optional<User> findByEmail(String email);

    1 usage  ↗ TENOVID
    public User findByResetPasswordToken(String password);
}
```

Рисунок 3.6 – Підключення інтерфейсу JpaRepository для моделі User

При наслідуванні в атрибутах треба обов'язково вказати модель та тип параметру первинного ключа моделі.

3.3 Структура вебзастосунку

Фреймворк Spring Web MVC розроблений на основі моделі-представлення-контролера (MVC) і побудований навколо диспетчера запитів DispatcherServlet, який маршрутизує запити до обробників. Він надає можливість налаштування мапування обробників, вирішення представлень, визначення локалі і теми, а також підтримку завантаження файлів. Основний обробник за замовчуванням базується на анотаціях @Controller та @RequestMapping і пропонує широкий спектр гнучких методів обробки. З появою Spring 3.0 механізм @Controller також дозволяє створювати вебсайти та додатки, що використовують REST, за допомогою анотації @PathVariable та інших функцій.

Вебфреймворк Spring MVC, подібно до багатьох інших вебфреймворків MVC, працює за принципом обробки запитів, побудовано навколо центрального сервлету, який маршрутизує запити до контролерів та надає інші функціональні можливості для полегшення розробки вебдодатків. Проте, сервлет DispatcherServlet у Spring робить більше, ніж просто це. Він повністю інтегрований з контейнером IoC Spring і, таким чином, дозволяє використовувати всі інші функції, які має Spring.

Процес обробки запиту в Spring Web MVC з використанням DispatcherServlet ілюструється наступною діаграмою.

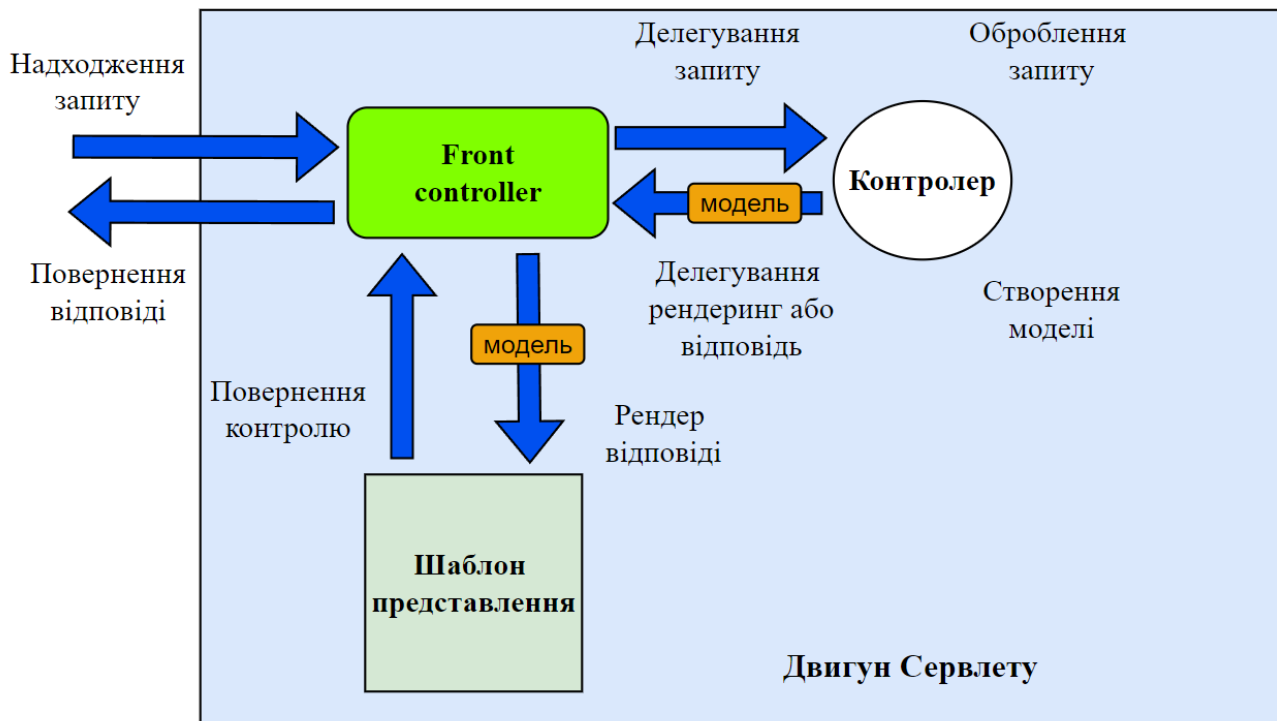


Рисунок 3.7 – Обробка запиту Spring Web MVC

DispatcherServlet є виразом шаблону проектування "Front Controller" (цей шаблон є загальним для багатьох провідних вебфреймворків, включаючи Spring Web MVC).

Контролери надають доступ до поведінки додатка, яку зазвичай визначають через інтерфейс сервісу. Контролери інтерпретують введення користувача і перетворюють його на модель, яка відображається користувачу через представлення. Spring реалізує контролер у дуже абстрактний спосіб, що дозволяє створювати різноманітні контролери.

У версії Spring 2.5 був впроваджений модель програмування контролерів на основі анотацій, таких як `@RequestMapping`, `@RequestParam`, `@ModelAttribute` та інших. Підтримка цих анотацій доступна як для Servlet MVC, так і для Portlet MVC. Контролери, реалізовані в цьому стилі, не обов'язково повинні успадковувати

конкретні базові класи або реалізовувати конкретні інтерфейси. Крім того, вони зазвичай не мають прямих залежностей від API Servlet або Portlet.

Розглянемо на прикладі контролеру авторизації. Для початку треба створити модель User, яка буде містити всі необхідні поля. Після цього необхідно створити контролер, який буде оброблювати запити. Для цього створимо контролер AuthController з анотацією @RequestMapping з параметром «/auth» для структуризації вебзастосунку. Всі сторінки, які відповідають за авторизацію будуть мати адресу типу «/auth/login».

```
@Controller
@RequestMapping("/auth")
public class AuthController {
    2 usages
    @Autowired
    JavaMailSender mailSender;
    1 usage
    @Autowired
    UserServices userServices;
    3 usages
    @Autowired
    private UserRepository userRepository;
    2 usages
    private String prod_count_on_cart;
    1 usage
    @Autowired
    private CartRepository cartRepository;
    1 usage
    @Autowired
    private PasswordEncoder passwordEncoder;

    no usages TENOVO
    @GetMapping("/login")
    public String getLoginPage(@CookieValue(value = "userid", defaultValue = "newus") String usid, Model model) {
        int cntProdInCart = Integer.parseInt(cartRepository.countProd(usid));
        prod_count_on_cart = String.valueOf(cntProdInCart);
        model.addAttribute(attributeName: "countProd", cntProdInCart);
        return "login";
    }
}
```

Рисунок 3.8 – Контролер авторизації

Метод getLoginPage з анотацією @GetMapping("/login") відповідає за обробку даних та відображення сторінки логіну. Також створена модель, яка передає дані за ключем до html-файлу login.

3.4 Конфігурація проєкту

Збірником проєкту є Apache Maven. Файл `pom.xml` (Project Object Model) є файлом конфігурації для проєктів, які використовують Apache Maven в якості системи управління залежностями та збірки проєкту. Spring Boot, будучи фреймворком для розробки додатків на Java, використовує Maven для управління залежностями та збірки проєкту, тому `pom.xml` є важливим файлом у структурі Spring Boot-проєкту.

У файлі `pom.xml` визначаються всі залежності, необхідні для проєкту. Це можуть бути бібліотеки, фреймворки або інші модулі, які використовуються в проєкті. Maven використовує цю інформацію для автоматичного завантаження необхідних залежностей з централізованого репозиторію та включення їх до проєкту. Це спрощує управління залежностями та дозволяє легко додавати, оновлювати або вилучати залежності з проєкту. Також у файлі `pom.xml` визначаються налаштування збірки проєкту. Наприклад, можна вказати, які файли потрібно включити до пакету, які ресурси скопіювати, які тести виконати та інше. Maven використовує цю інформацію для автоматичної збірки проєкту відповідно до визначених налаштувань.

Отже, `pom.xml` у Spring Boot-проєкті дозволяє легко управляти залежностями, конфігурувати збірку проєкту та визначати налаштування життєвого циклу проєкту за допомогою Maven.

Задамо основні параметри:

- кодування тексту;
- посилання на Maven;
- версію Maven;
- версію Spring Framework;
- назву та опис проєкту;
- версію Java;
- плагін `springframework.boot`.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.1</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.stapelok</groupId>
  <artifactId>stapelok</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>stapelok</name>
  <description>Stapelok Project</description>
  <properties>
    <java.version>17</java.version>
  </properties>
```

Рисунок 3.9 – Налаштування проєкту

Підключимо необхідні залежності для повноцінного функціонування проєкту. У секції `<dependencies>` файлу `pom.xml` треба додати блоки `<dependency>`, які визначають необхідні залежності для проєкту. Кожен блок `<dependency>` містить інформацію про артефакт, таку як ім'я групи, ідентифікатор та версію.

Після визначення залежностей в файлі `pom.xml` треба виконати команду `mvn install` або `mvn package`, щоб Maven завантажив необхідні залежності з віддаленого репозиторію. Maven знайде вказані артефакти залежностей, завантажить їх та збереже в локальному репозиторії.

Після завантаження залежностей вже можна використовувати їх у проєкті. Maven автоматично вирішує залежності та додає їх до шляху компіляції та виконання проєкту. Також можна імпортувати класи з встановлених залежностей і використовувати їх у коді.

Maven також надає можливості для роботи з репозиторіями, кешування залежностей, управління версіями та багато іншого. Він автоматизує процес підключення залежностей та полегшує управління проєктами Java.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-mail</artifactId>
  </dependency>

  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>

  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
  </dependency>
</dependencies>
```

Рисунок 3.10 – Залежності в файлі pom.xml

Першу залежність, яку треба підключити це `spring-boot-starter` - це Maven залежність, яка включає основні модулі Spring Boot, необхідні для розробки додатків на базі Spring Boot.

Залежність `spring-boot-starter` включає в себе наступні модулі:

- `spring-boot-starter-actuator`: Надає моніторинг і керування додатком використовуючи HTTP-запити або JMX;
- `spring-boot-starter-aop`: Надає підтримку аспектно-орієнтованого програмування (Aspect-Oriented Programming - AOP);
- `spring-boot-starter-data-jpa`: Надає підтримку роботи з базами даних за допомогою Java Persistence API (JPA);
- `spring-boot-starter-security`: Надає підтримку механізмів аутентифікації і авторизації в додатку;
- `spring-boot-starter-web`: Надає підтримку для розробки вебдодатків, включаючи вбудований вебсервер (зазвичай Tomcat або Jetty) і Spring MVC.

Це лише декілька прикладів модулів, які можуть бути включені у `spring-boot-starter`. Залежність `spring-boot-starter` дозволяє швидко налаштувати базовий функціонал Spring Boot для розробки додатків із заданою функціональністю. Ви можете додавати інші модулі та залежності до свого проєкту в залежності від потреб вашого додатку.

Для полегшення створення підключено залежність `spring-boot-starter-test`. Це Maven залежність для розробки тестів у додатках, побудованих з використанням Spring Boot. Ця залежність надає різноманітні інструменти та бібліотеки для написання, виконання та перевірки тестів в Spring Boot додатках.

Залежність `spring-boot-starter-test` включає в себе наступні модулі та бібліотеки:

- JUnit: Фреймворк для написання тестів у Java. Включає анотації, класи і методи для визначення й виконання тестових сценаріїв;

- Spring Test: Підтримка тестування в Spring Framework. Надає анотації, класи та інші інструменти для створення тестових середовищ та інтеграційних тестів;
- AssertJ: Бібліотека для написання більш експресивних і читабельних тверджень (assertions) у тестах;
- Mockito: Фреймворк для створення імітованих об'єктів (mock objects) та виконання перевірок у тестах.

Залежність `spring-boot-starter-test` допомагає забезпечити зручне середовище для написання тестів у Spring Boot додатках. Вона містить ряд інструментів, які дозволяють легко створювати і виконувати тести, перевіряти різні аспекти додатка та переконуватися, що він працює відповідно до очікувань.

Maven залежність `spring-boot-starter-thymeleaf` надає підтримку шаблонування на базі Thymeleaf для додатків, побудованих з використанням Spring Boot.

Thymeleaf є потужним шаблонним двигуном, який дозволяє вбудовувати динамічний контент у HTML-сторінки. Залежність `spring-boot-starter-thymeleaf` додає необхідні компоненти та конфігурацію для використання Thymeleaf у Spring Boot додатках.

Thymeleaf дозволяє використовувати вбудовані шаблони та фрагменти для полегшення створення спільних частин HTML-сторінок.

Залежність `spring-boot-starter-thymeleaf` робить розробку динамічних HTML-сторінок у Spring Boot додатках простішою та зручнішою за допомогою потужних можливостей Thymeleaf [31].

Maven залежність `spring-boot-starter-mail` надає підтримку для надсилання електронних листів в додатках, побудованих з використанням Spring Boot.

Залежність `spring-boot-starter-mail` включає в себе необхідні компоненти та конфігурацію для використання поштової функціональності в Spring Boot додатках. Вона дозволяє легко налаштувати та використовувати розсилку електронних листів з використанням простого та зручного API.

Деякі з можливостей, які надаються залежністю `spring-boot-starter-mail`, включають:

- надсилання електронних листів: Залежність додає необхідні компоненти для надсилання електронних листів з використанням протоколів SMTP, SMTPS або Sendmail;
- підтримка поштових серверів: Spring Boot дозволяє легко налаштувати параметри підключення до поштового сервера, такі як адреса сервера, порт, аутентифікація тощо;
- шаблони електронної пошти: Залежність підтримує використання шаблонів для створення електронних листів. Можна використовувати шаблонізатори, такі як Thymeleaf або Freemarker, для генерації динамічного контенту в електронних листах;
- перевірка стану доставки: Залежність надає можливість перевіряти статус доставки електронних листів та отримувати сповіщення про повернення або помилки доставки.

Залежність `spring-boot-starter-mail` робить роботу з електронною поштою простішою у Spring Boot додатках, дозволяючи надсилати електронні листи з легкістю та налаштовувати різні аспекти електронної комунікації.

Висновки до розділу 3

Проаналізувавши вище описане, можна дійти висновку, що була створена структура проєкту та попередні налаштування. Було створено моделі та інформаційну діаграму по первинним ключам.

В файлах проєкту було підключено та налаштовано права користування БД.

Було підключено репозиторій Apache Maven. Створено файл `pom.xml` для конфігурації проєкту та управління залежностями. Вказано версію Java та додано залежності, які необхідні для розробки безпечного та якісного вебзастосунку.

Отже, проведені всі підготовчі роботи для розробки та тестування вебзастосунку.

4 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

4.1 Реалізація аутентифікація та авторизації

Перш за все, варто почати з форм авторизації та адміністративної частини вебзастосунок, для того щоб додавати товари і оброблювати замовлення.

Для того щоб реалізувати аутентифікацію треба реалізувати декілька ролей, а саме користувачі та адміністратори. Це дозволить обмежити користувачів за правами доступу.

Створимо клас SecurityConfig та сконфігуруємо його за допомогою методу configure.

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .csrf().disable()
        .authorizeRequests()
        .antMatchers("/").permitAll()
        .antMatchers("/auth/registration").permitAll()
        .antMatchers("/admin/**").hasAnyAuthority("admin:all")
        .antMatchers("/user/**").hasAnyAuthority("users:all")
        .and()
        .formLogin()
        .loginPage("/auth/login").permitAll()
        .defaultSuccessUrl("/")
        .and()
        .logout()
        .logoutRequestMatcher(new AntPathRequestMatcher("/auth/logout", "POST"))
        .invalidateHttpSession(true)
        .clearAuthentication(true)
        .deleteCookies("JSESSIONID")
        .logoutSuccessUrl("/");
}
```

Рисунок 4.1 – Налаштування прав доступу користувачів

Ми можемо побачити що до головної сторінки мають доступ всі користувачі, незалежно від ролі. До сторінок, по адресі, яка починається з “/admin/” мають доступ лише адміністратори. Відповідно до сторінок, по адресі, яка починається з “/user/” мають доступ авторизовані покупці, а саме до особистого кабінету.

Після успішної авторизації всі користувачі будуть перенаправленні до головної сторінки.

Зберігання даних про успішну авторизацію і тип прав зберігаються у локальних файлах.

Невід’ємною частиною вебзастосунку є зовнішня частина сайту, а саме всі візуальні елементи: кнопки, списки, розділи.

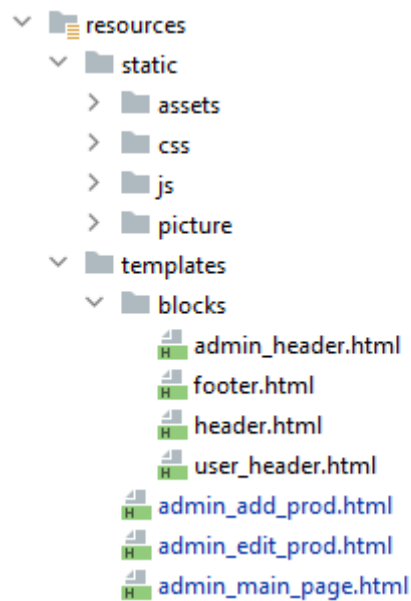
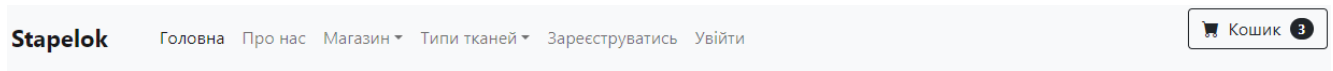


Рисунок 4.2 – Структура розташування ресурсних файлів

Структуризація ресурсних файлів згідно з правилами чистого коду є важливим аспектом організації проєкту. Це допомагає забезпечити легкість розуміння, підтримки та розширення проєкту. Згідно зі стандартами чистого коду, для файлів html треба виділити окрему папку в ресурсах, з назвою templates.

Папка static створена для js, css файлів та картинок.

В папці blocks зберігаються блоки сайту, які постійно повторюються, а саме header та footer. Це зроблено для того, щоб при зміні шапки сайту або номера телефону в описі, не треба було змінювати кожен сторінку сайту, бо потужні сайти можуть налічувати декілька десятків або сотню сторінок.



Вхід

Рисунок 4.3 – Сторінка авторизації

На сторінці авторизації ми можемо спостерігати по центру екрану два поля, для введення інформації та кнопку. Цим самим основна концентрація уваги користувача припадає на потрібну інформацію.

На верхній частині вебзастосунку знаходиться header, який вміщує в собі декілька кнопок. Всі неавторизовані користувачі будуть бачити даний header на будь-якій сторінці вебсайту. Він може перейти на головну сторінку, прочитати інформацію про компанію, зареєструватись, увійти, переглянути вміст кошику або перейти по спискам, які випадають.

Всі паролі користувачів будемо кодувати за допомогою hash-функції base64. Завдяки цьому при потраплянні БД у руки зловмисників, вони не зможуть зайти у особисті кабінети покупців чи адміністраторів.

email	phone	status	role	password
admin@gmail.com	023213	ACTIVE	ADMIN	\$2a\$12\$LaUJXmqN6dqmyE2qtaMhruPTVMlqRyr...
karpenko2002in42@gmail.com	0667517681	ACTIVE	USER	\$2a\$12\$XLdgrEPpMo5yhF0bNUtb8eGfMzxSq/T...
user@gmail.com	22213	ACTIVE	USER	\$2a\$12\$B/7K6JHE60q.1cBoE/ymfeEuDZt2rQTc...
NULL	NULL	NULL	NULL	NULL

Рисунок 4.4 – Дані моделі users

Таким чином, при вводі даних, пароль хешується hash-функцією та порівнюється результат з даними в БД. Якщо хеш-суми однакові то відбувається авторизація.

Реєстрація

Ім'я:

Прізвище:

По-батькові (за бажанням):

Електронна пошта:

Номер телефону:

Пароль:

Повторіть пароль:

Рисунок 4.5 – Форма реєстрації

Для реєстрації потрібно ввести всі необхідні данні, які знадобляться для замовлення товару. При замовленні всі дані будуть підтягуватись автоматично.

При реєстрації є ряд перевірок. Перевіряється пошта, щоб не створювали декілька аккаунтів на одну поштову адресу. Користувачу обов'язково треба повторити пароль, для того щоб запобігти натисканню випадкового символу на клавіатурі. Як результат користувач не зіштовхнеться з труднощами при першій авторизації і матиме позитивний досвід при роботі з сайтом.

4.2 Адміністративна панель сайту

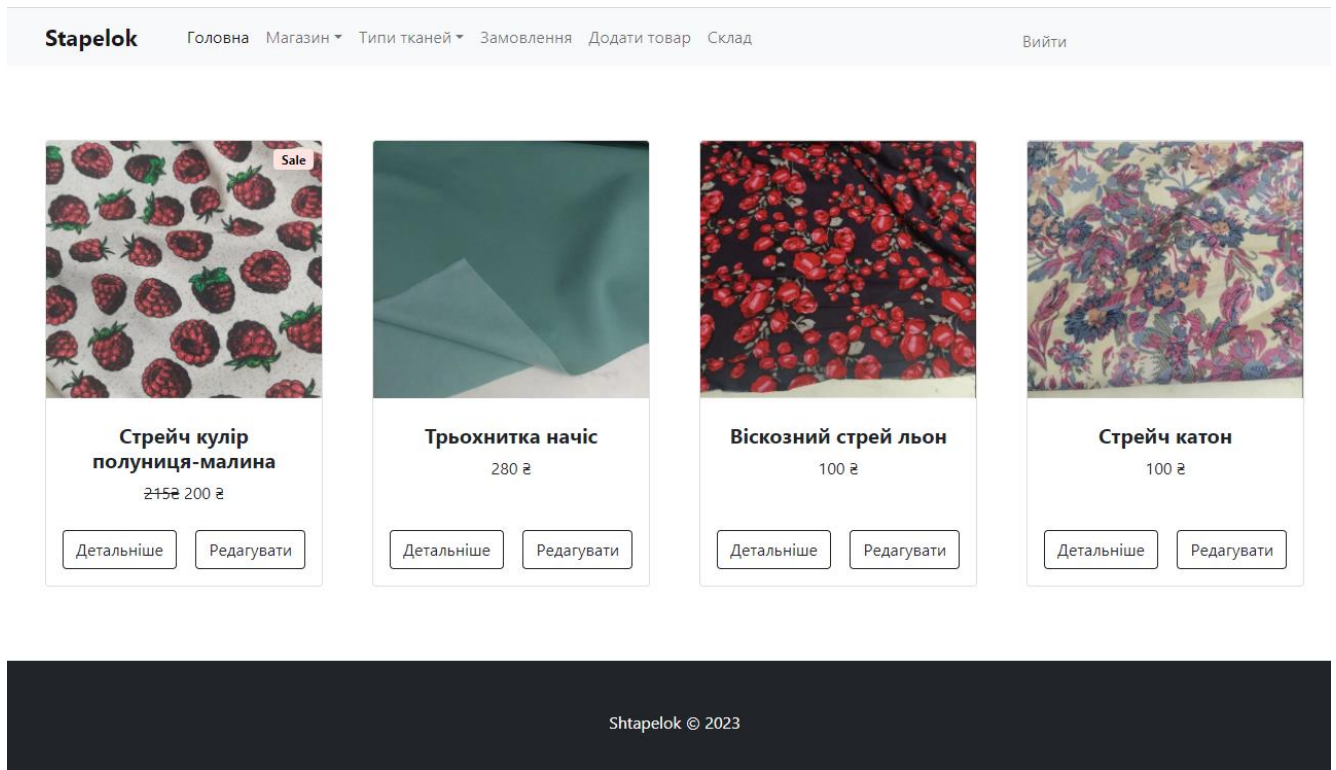


Рисунок 4.6 – Головна сторінка адміністратора

На головній сторінці адміністратора знаходяться декілька кнопок: головна сторінка, навігаційні списки, які випадають, замовлення, додавання товарів та склад.

По центру екрану знаходяться товари, які можна переглянути та редагувати. Дана секція є аналогічною до клієнтської, але має інший функціонал.

При переході за посиланням з правами покупця, користувач отримує помилку 403. Помилка 403, також відома як "Помилка заборони доступу" (Forbidden Error), вказує на те, що користувач намагається отримати доступ до ресурсу або сторінки, до якої у нього немає дозволу.

Тип тканини:

Шоек Льон Шифон Костюмні

Мереживо,Гіпюр Трикотаж Біфлекс Пальтові

Плащівка, шкіра, хутро Джинс Паєтки Вельвет, велюр

Підкладкові Для будинку Фатін, сітка Бавовняні

Атлас, сатин

Назва:

Котон бабл квітова фантазія, темно-бежевий

Опис:

Коттон бабл – натуральна, тонка, легка, повітряна тканина. Фактура гладка і одночасно є легкий ефект хвилястого стиску. Матеріал нееластичний, нег

Загальна довжина тканини (см):

2000

Країна:

Китай

Склад тканини:

60% котон, 40% віскоза

Ширина (см):

150

Щільність:


85

Ціна за погонний метр:

225

Завантажити фото №1

Катон баб...нтазія 1.jpg



Завантажити фото №2

Катон баб...антазія.jpg




Рисунок 4.7 – Форма додавання товару

Форма додавання містить в собі поля для заповнення та можна завантажити до трьох фотографій. Також можна не вказувати опис, країну виробника та склад тканини. При відсутності, вони не будуть відображатись.

Тип тканини:
 Шовк Льон Шифон Костюмні
 Мереживо,Гіпюр Трикотаж Біфлекс Пальтові
 Плащівка, шкіра, хутро Джинс Паєтки Вельвет, велюр
 Підкладкові Для будинку Фатін, сітка Бавовняні
 Атлас, сатин

Назва:

Опис:

Загальна довжина тканини на складі (метри):

Країна:

Склад тканини:

Ширина (см):

Щільність:

Ціна за погонний метр:

Ціна зі знижкою (якщо немає знижки ставити 0):

Завантажити фото №1
 Файл не вибрано




Рисунок 4.8 – Форма редагування товару

Дана форма дозволяє видаляти фотографії, редагувати всю інформацію про товар і поставити знижку. Для цього значення 0 змінимо на акційну ціну.

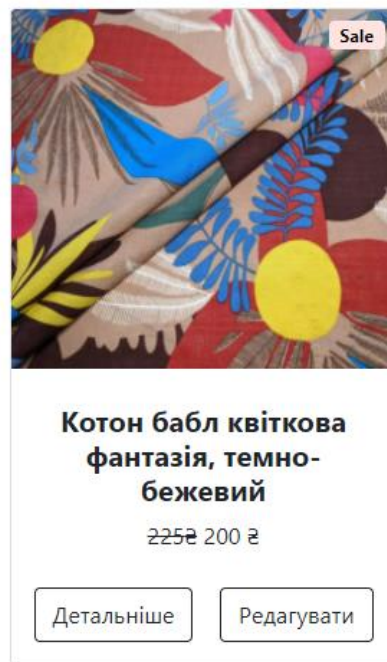


Рисунок 4.9 – Відображення товару зі знижкою

Акційні товари мають маркування “Sale” та перекреслену стару ціну.

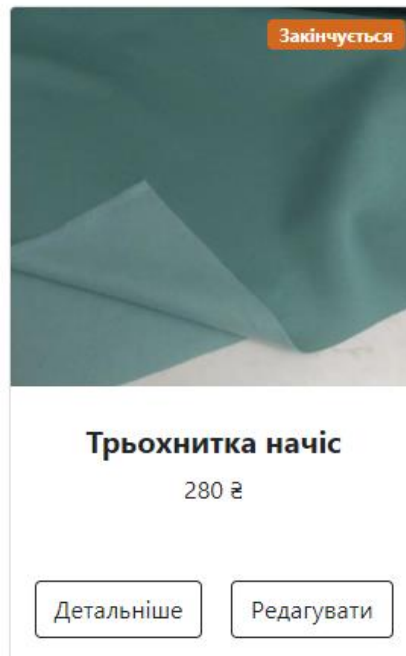


Рисунок 4.10 – Відображення товару, який закінчується

Якщо довжина тканини на складі менше 10 метрів, то товар маркується написом “Закінчується”.

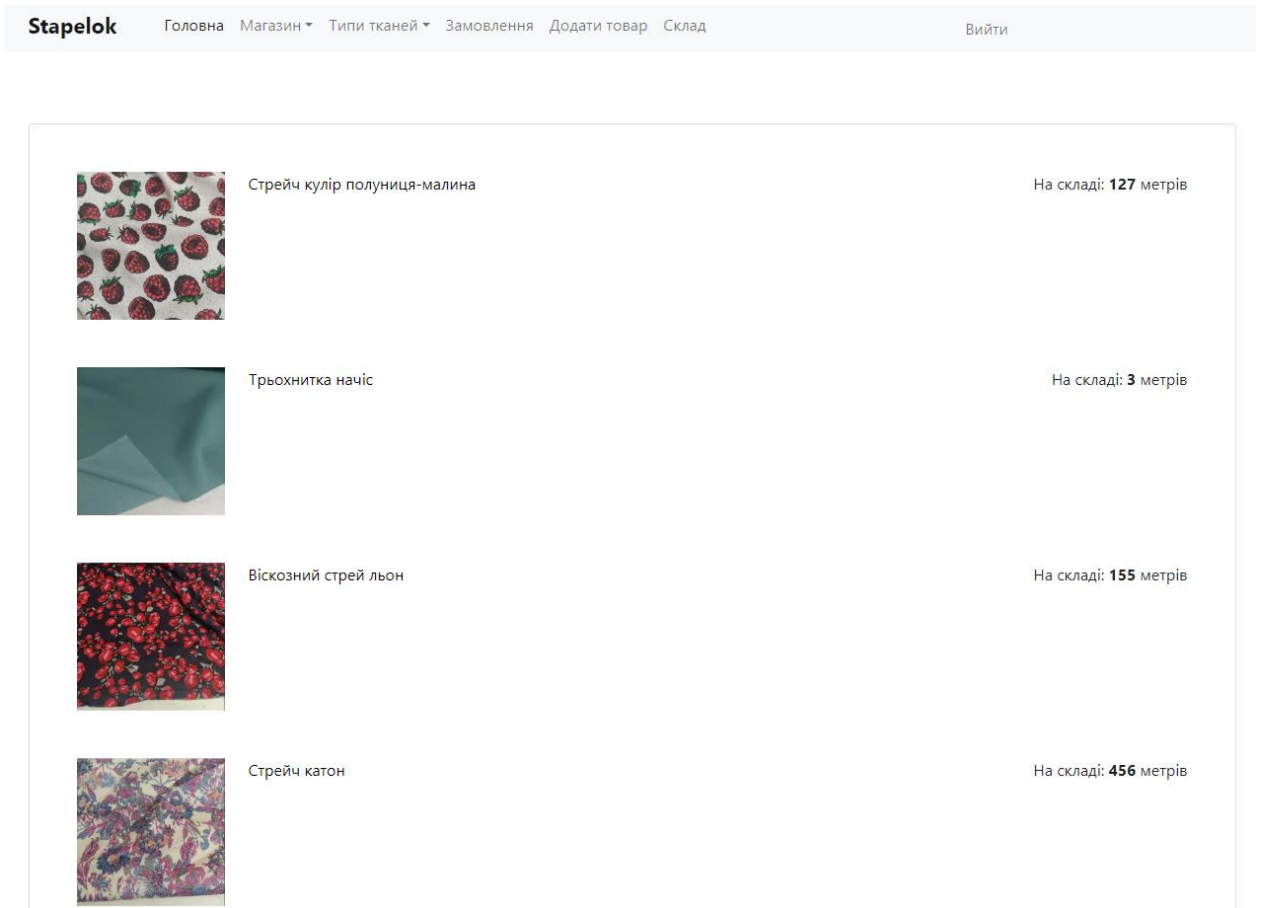


Рисунок 4.11 – Склад тканин

Є розділ складу товарів, де можна переглянути залишок на складі та додати довжину при надходженні товару. Натиснувши на назву тканини, адміністратора буде перенаправлено на сторінку редагування товару.

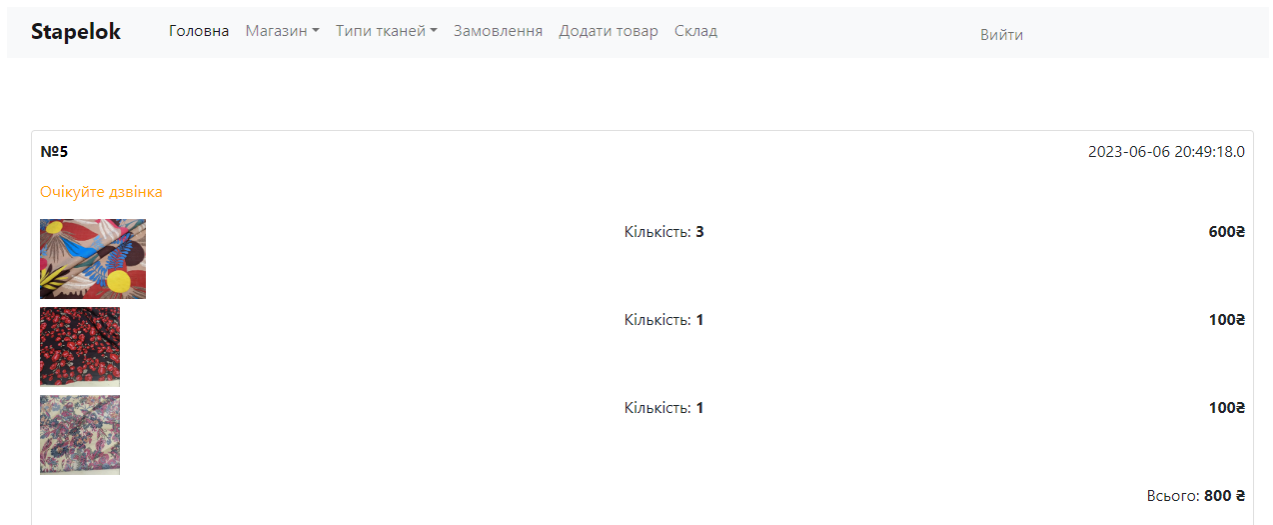





Рисунок 4.12 – Сторінка замовлень

На сторінці замовлень відображені всі замовлення клієнтів. Сортування по статусу.

Замовлення: №5

	Котон бабл квіткова фантазія, темно-бежевий	Кількість: 3	600€
	Віскозний стрей льон	Кількість: 1	100€
	Стрейч катон	Кількість: 1	100€
Всього			800€

Статус замовлення:
 Очікуйте дзвінка Підтверджено Відправлено Відміна

Ім'я:

Прізвище:

Рисунок 4.13 – Сторінка редагування замовлення

При оформленні замовлення, йому автоматично ставиться статус “очікуйте дзвінка”. Після дзвінку, залежно від відповіді, оператор може змінити статус, інформацію про отримувача та адресу доставки.

4.3 Клієнтська частина сайту

Згідно зі стандартами інтернет-торгівлі клієнт повинен одразу бачити товар, який його потенційно може зацікавити. Для цього на першій сторінці варто розмістити популярні, акційні або підібрані для покупця товари, з категорій, які йому цікаві.

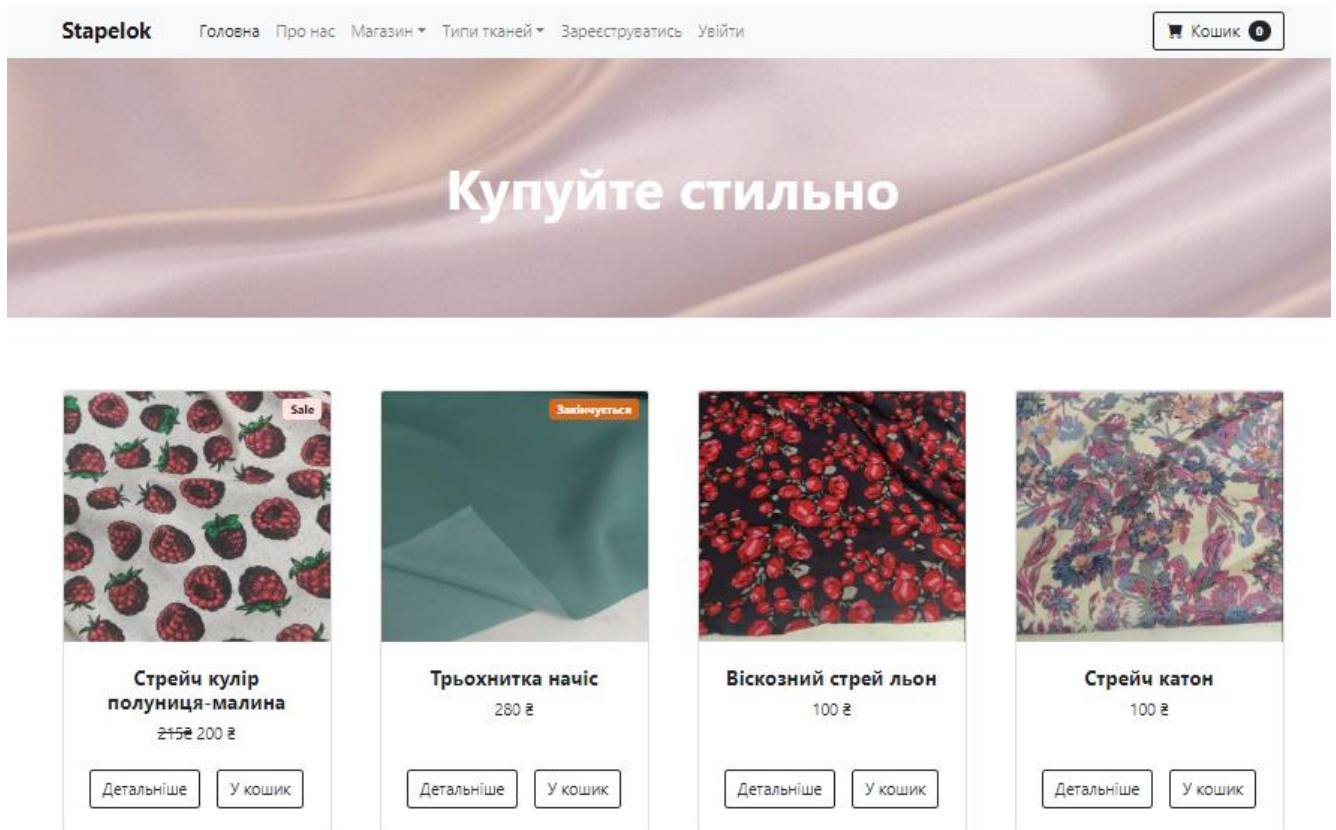


Рисунок 4.14 – Головна сторінка інтернет-магазину

На головній сторінці відображені товари та кнопки навігації по сайту. По центру можемо спостерігати товари та кнопки, які перенаправлять користувача на сторінку товару або перенесуть тканину у кошик. Усталено тканина додається довжиною в один метр. У кошику це можна змінити.

Для того, щоб переглянути типи тканин або розділ популярних товарів, варто скористуватись списками, що випадають.

В шапці сайту знаходяться кнопки для реєстрації та авторизації. Реалізовано кнопка кошику та лічильник товарів, які у ньому знаходяться. Для зручності вибору та покупок була реалізована система збереження доданих товарів у кошику. Це реалізовано за допомогою файлів локального збереження. При першому заході на сайт користувачу дається id, який прив'язаний до його браузера. Це зроблено для того, щоб не втрачати обрані товари при перезавантаженні браузера.

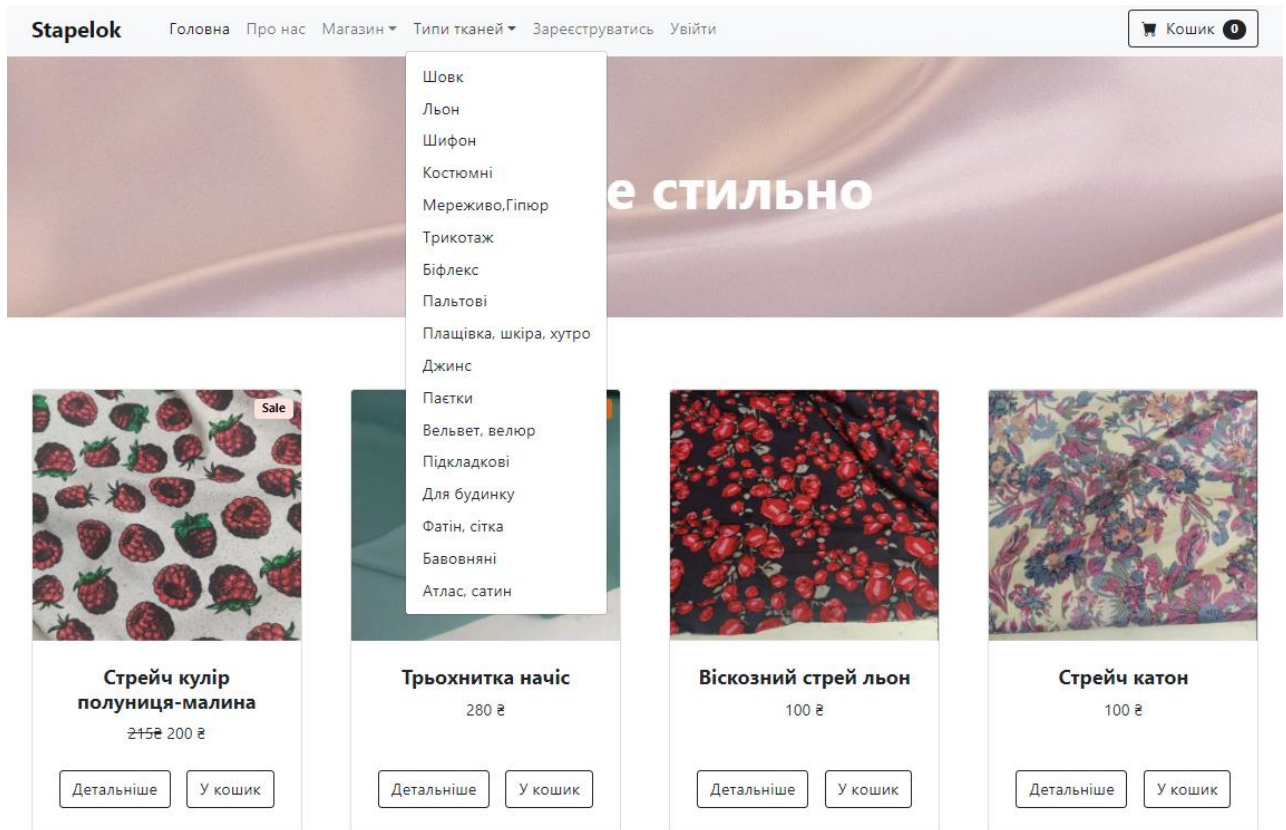


Рисунок 4.15 – Навігаційний список типів тканин

У списку магазин можна відсортувати товари за популярністю та новизні.

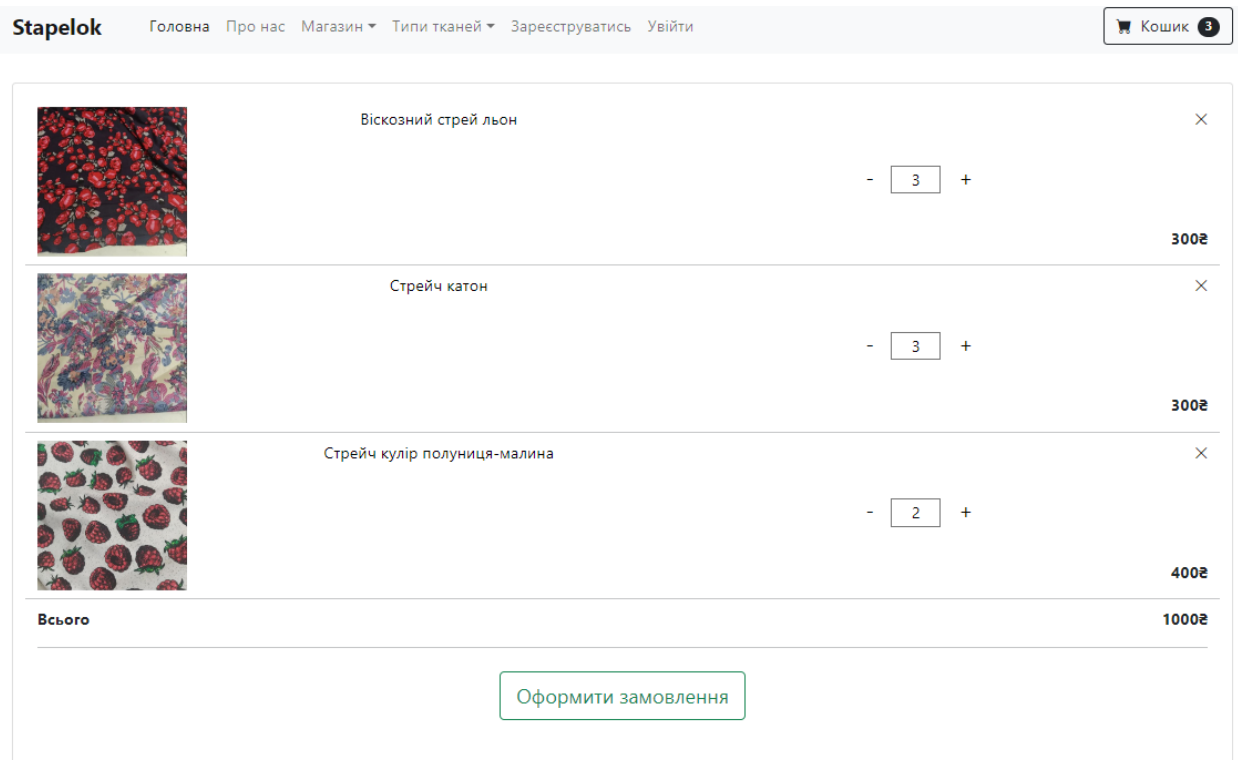
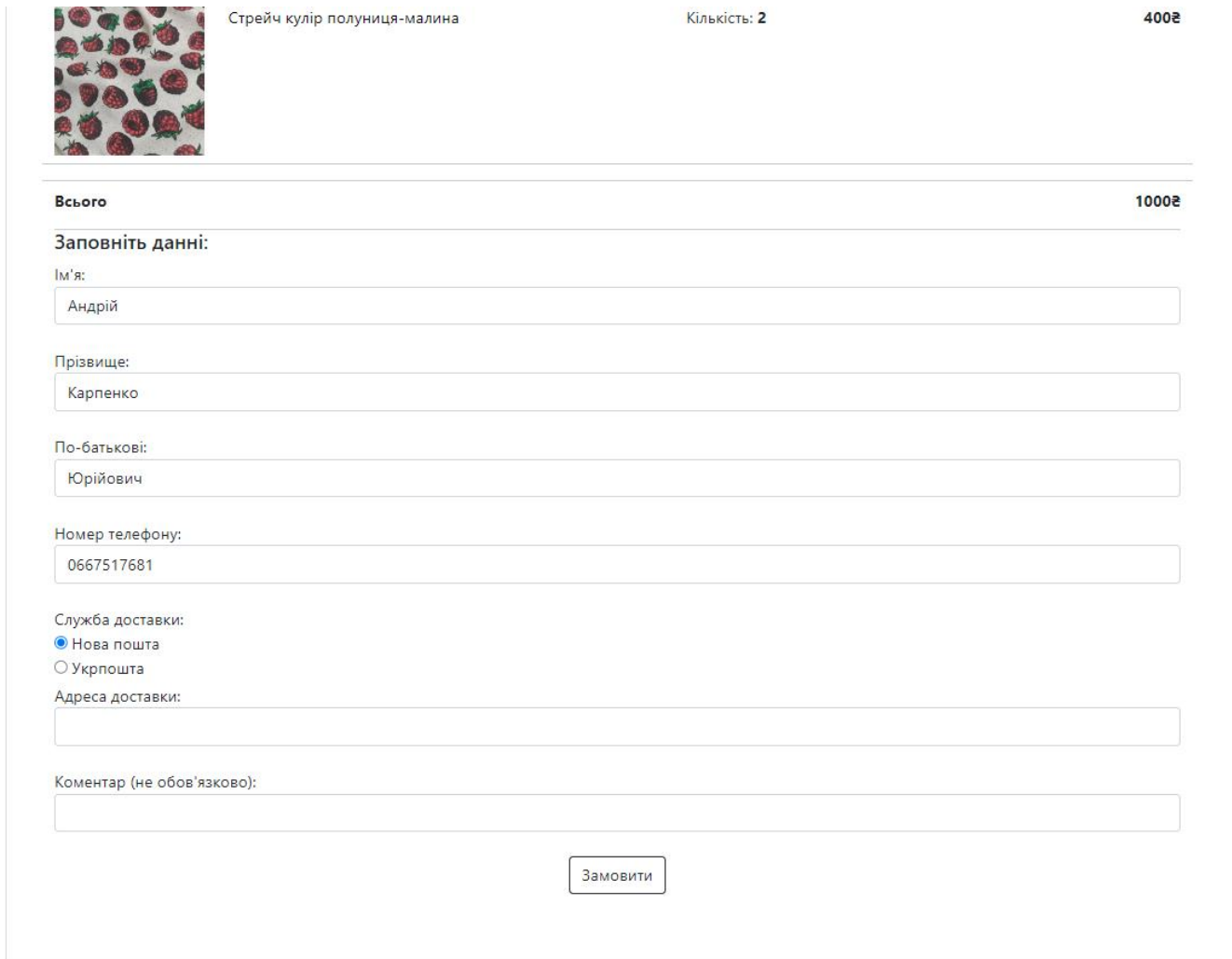


Рисунок 4.16 – Сторінка кошику

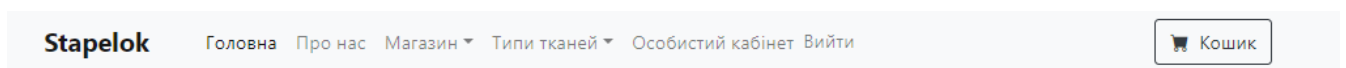
На сторінці кошику крім шапки є ще декілька кнопок за допомогою яких можна видаляти товар, збільшувати або зменшувати кількість, прочитати детальну інформацію і оформити замовлення.



The screenshot shows a shopping cart item and a checkout form. The cart item is a fabric with a strawberry pattern, labeled "Стрейч кулір полуниця-малина" with a quantity of 2 and a price of 4000₴. Below the cart item, a summary shows a total of 10000₴. The checkout form is titled "Заповніть данні:" and includes fields for name (Andriy), surname (Karpenko), patronymic (Yuriyovych), phone number (0667517681), and delivery service (Nova Poshta selected). There is also a field for the delivery address and a comment field. A "Замовити" button is at the bottom.

Рисунок 4.17 – Сторінка оформлення замовлення

Якщо користувач авторизований, то данні підтягуються автоматично. При необхідності ви можете змінити дані.



Дякую за замовлення, Андрій

Номер вашого замовлення: №13

Найближчим часом з Вами зв'яжуться!

Рисунок 4.18 – Результат успішного замовлення

Stapelok Головна Про нас Магазин Типи тканин Особистий кабінет Вийти Кошик 0

Вітаємо, Андрій Карпенко !

[Змінити персональні дані](#)

Ваші замовлення:




№ 13	2023-06-07 15:44:43.0
Очікуйте дзвінка	
	Кількість: 3 300€
	Кількість: 3 300€
	Кількість: 2 400€
Всього: 1000 €	

Рисунок 4.19 – Особистий кабінет користувача

У кожного зареєстрованого користувача є особистий кабінет, де він може переглянути свої замовлення та змінити персональні дані.

Змінити основні дані:

Ім'я:

Прізвище:

По-батькові:

Номер телефону:

Змінити пароль:

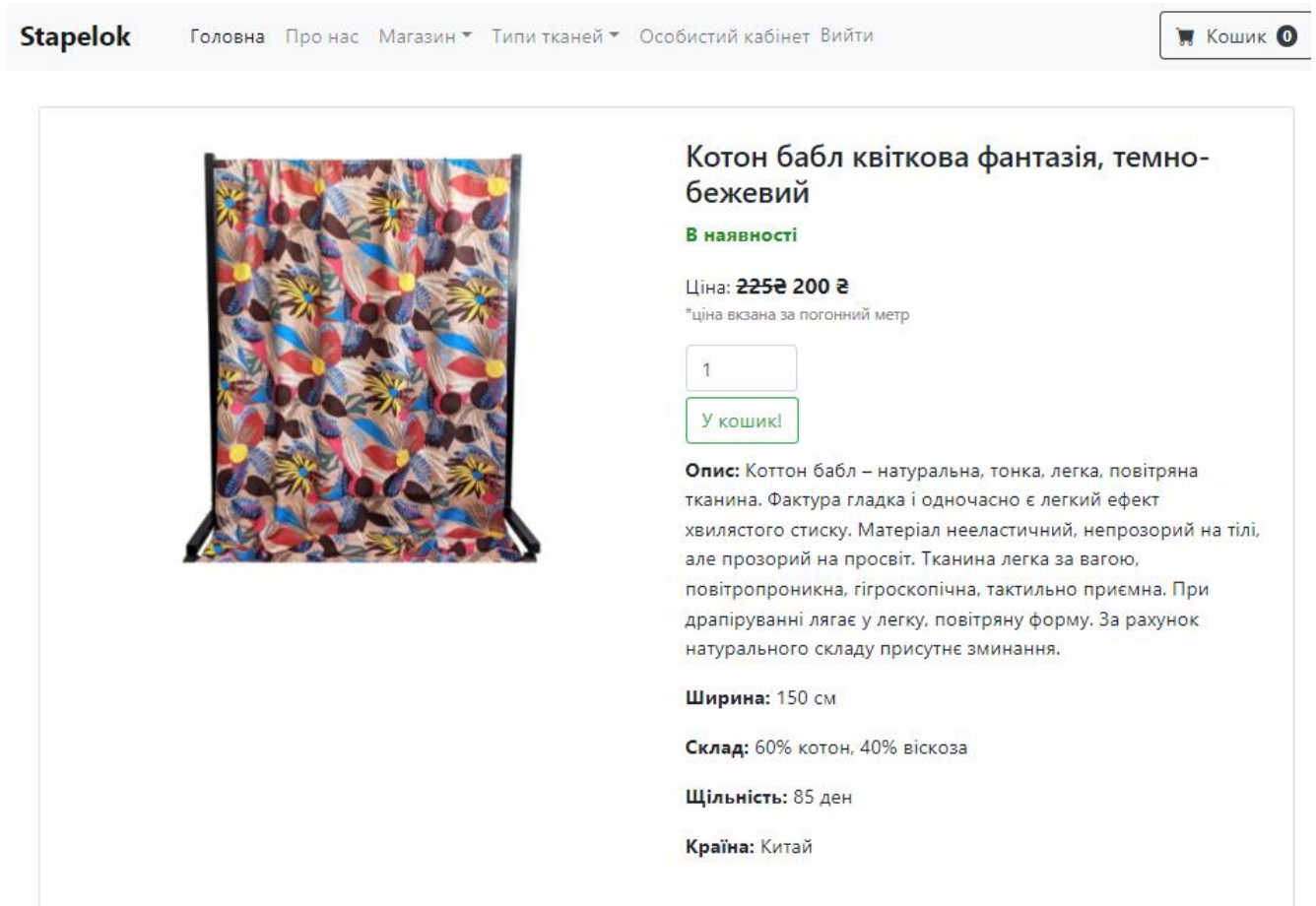
Введіть старий пароль:

Введіть новий пароль:

Повторіть новий пароль:

Рисунок 4.20 – Сторінка зміни персональних даних

Сторінка зміни персональних даних має перевірки та підказки, якщо у користувача виникнуть проблеми під час зміни даних або паролю.



The screenshot shows a product page for 'Котон бабл квіткова фантазія, темно-бежевий' (Cotton bubble floral fantasy, dark beige). The page includes a navigation bar with 'Stapelok', 'Головна', 'Про нас', 'Магазин', 'Типи тканин', and 'Особистий кабінет Вийти', along with a shopping cart icon. The product image shows a vibrant floral pattern on a dark beige background. The price is listed as 225€ 200 € (with a strikethrough on 225€) and is noted as 'ціна вказана за погонний метр'. There is a quantity selector set to 1 and a 'У кошик!' button. The description states it is a natural, thin, light, airy fabric with a wavy texture, non-elastic, and non-transparent. It is breathable, hygroscopic, and soft. The width is 150 cm, the composition is 60% cotton and 40% viscose, the density is 85 denier, and it is made in China.

Рисунок 4.21 – Сторінка товару

Товар виглядає наступним чином. Автоматично перемикаються фото товару. При відсутності опису чи країни виробника зміст буде автоматично перерозподілено. Продаж в погонних метрах.

Висновки до розділу 4

Проаналізувавши вище описане, можна дійти висновку, що під час написання розділу було розроблений вебзастосунок згідно зі стандартами дизайну інтернет-магазинів. Проект завантажено на репозиторій GitHub. Знайти його можна за посиланням: <https://github.com/TENOVOD/OnlineStoreStapelok>.

Було розроблено форми авторизації та систему аутентифікації, яка працює з декількома ролями. Також реалізовано зберігання паролів користувачів за допомогою хеш-функцій, що в результаті запобігає можливість доступу третіх осіб до особистих кабінетів покупців та адміністраторів.

Була проведена робота з розробки ергономічного дизайну сторінок для покупців та для адміністраторів. Реалізовано роботу кошика за допомогою локальних файлів.

Отже, було розроблено цілісну систем інтернет-магазину з продажу тканин.

ВИСНОВКИ

У цій роботі було доведено потребу у якісному вебзастосунку для малого бізнесу та розроблено інтернет-магазин з продажу тканин з ергономічним дизайном.

Задачі, поставлені у вступній частині були повністю виконані. Було проведено аналіз сучасних стандартів інтернет-магазинів та інтернет-магазини конкурентів. Також у другому розділі обрано технології для розробки вебзастосунку.

У третьому розділі було розроблено БД та її моделі. Після чого були проведенні базові налаштування проекту та підключено базу даних MySQL. Додатково було підключено репозиторій Apache Maven і його залежності для подальшої розробки вебзастосунку. Розглянуто структуру сайту на основі Spring Boot MVC.

Останнім кроком було розроблено вебзастосунок. В першу чергу було створено систему авторизації та аутентифікації, для того, щоб розділити покупців та користувачів за правами доступу. Зберігання паролів було реалізовано за допомогою хеш-функцій, для того, щоб в результаті можливого витоку інформації зловмисник не зміг отримати доступ до акаунту адміністратора чи покупця. Потім було розроблено адміністративну частину сайту, щоб додавати товари та редагувати зовнішній вигляд сторінок вебзастосунку. Після чого створено клієнтську частину з розширеним функціоналом, а саме реалізовано можливість перегляду власних замовлень та збереження товарів у кошику довгий час, за допомогою локальних файлів.

Отже, всі поставлені задачі виконані, відповідно, мета кваліфікаційної роботи досягнута.

Робота надала досвід розробки безпечних, надійних та ергономічних застосунків на основі мови програмування Java і фреймворку Spring Boot MVC.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрощук Г. COVID-19: вплив на електронну комерцію. URL: <https://yur-gazeta.com/publications/practice/medichne-pravo-farmaceutika/covid19-vpliv-na-elektronnu-kommerciyu.html> (дата звернення: 01.05.2023).
2. Стольницький І. Е-commerce живий? Що відбувається з онлайн-продажами в Україні під час війни. URL: <https://biz.nv.ua/ukr/experts/shcho-stalosya-z-internet-torgivleyu-v-ukrajini-pid-chas-viyuni-chi-zapracyuvali-internet-magazini-50251494.html> (дата звернення: 02.05.2023).
3. МастХев для сучасного інтернет магазину. URL: <https://web-systems.solutions/blog/musthave-dlya-suchasnogo-internet-magazynu/> (дата звернення: 02.05.2023).
4. 10 STANDARDS EVERY ECOMMERCE STORE SHOULD MEET. URL: <https://d5creation.com/ecommerce-standards/> (Last accessed: 03.05.2023).
5. Основні проблеми і помилки в роботі інтернет магазину, які ведуть до зниження продажів. URL: <https://ua-retail.com/2021/03/osnovni-problemi-i-pomilki-v-roboti-internet-magazynu-yaki-vedut-do-znizhennya-prodazhiv/> (дата звернення: 05.05.2023).
6. Simplify online shopping for senior citizens: 11 UX ideas. URL: <https://www.convertcart.com/blog/online-shopping-for-seniors> (Last accessed: 06.05.2023).
7. Інтернет-магазин тканин «ТК». URL: <https://tk.ua/> (дата звернення: 07.05.2023).
8. Older shoppers feel ignored online by retailers. URL: <https://newsroom.pricerunner.com/posts/pressreleases/older-shoppers-feel-ignored-online-by-retailer> (Last accessed: 07.05.2023).
9. Інтернет-магазин тканин «АТЛАС». URL: <https://tkani-atlas.com.ua/ua/> (дата звернення: 08.05.2023).
10. Специфікація вимог до програмного забезпечення. URL: <http://surl.li/hasga> (дата звернення: 09.05.2023).

11. What is web-server? URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (Last accessed: 16.05.2023).
12. Глобальний рейтинг вебсерверів на 2023. URL: <https://uk.hostadvice.com/marketshare/server/> (дата звернення: 16.05.2023).
13. Комп'ютерні мережі / Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Львів: «Магнолія 2006», 2013. 256 с.
14. AJAX - розробка. URL: <https://armedsoft.com/ua/blog/ajax-rozrobka> (дата звернення: 18.05.2023).
15. AJAX. URL: <https://uk.wikipedia.org/wiki/AJAX> (дата звернення: 18.05.2023).
16. Goldberg A. Smalltalk-80: The Interactive Programming Environment. 1984. P. 218-233. URL: <http://stephane.ducasse.free.fr/FreeBooks/TheInteractiveProgrammingEnv/TheInteractiveProgrammingEnv.pdf> (Last accessed: 18.05.2023).
17. The Principles of Clean Architecture by Uncle Bob Martin. URL: https://www.youtube.com/watch?v=o_TH-Y78tt4&t=1667s (дата звернення: 19.05.2023).
18. Inside Smalltalk / LaLonde, Wilf R., Pugh, John R. U.S.A.: Prentice-Hall, 1991. P. 8.
19. What is PHP? The PHP Programming Language Meaning Explained. URL: <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/#:~:text=The%20abbreviation%20PHP%20initially%20stood,recursive%20acronym%20for%20Hypertext%20Preprocessor> (Last accessed: 23.05.2023).
20. Django introduction. URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> (Last accessed: 23.05.2023).
21. What Is Python Used For? A Beginner's Guide. URL: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> (Last accessed: 23.05.2023).

22. Java (programming language). URL: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)) (Last accessed: 28.05.2023).
23. What is JavaScript? URL: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (Last accessed: 28.05.2023).
24. Most Popular Java Web Frameworks in 2023. URL: <https://rollbar.com/blog/most-popular-java-web-frameworks/> (Last accessed: 28.05.2023).
25. The Spring Framework Beginner's Guide: Features, Architecture & Getting Started. URL: [https://www.bmc.com/blogs/spring-framework/#:~:text=The%20Spring%20Framework%20is%20an,Dependency%20injection%20\(DI\)](https://www.bmc.com/blogs/spring-framework/#:~:text=The%20Spring%20Framework%20is%20an,Dependency%20injection%20(DI)) (Last accessed: 29.05.2023).
26. Chris Schalk. Introduction to Javasever Faces - What is JSF? Oracle Corporation. 2005.
27. Beginning Hibernate (Second ed.) / Linwood, Ananda, Minter, Dave. Apress, 2010. P. 400.
28. Pro Hibernate 3 (First ed.) / Minter, Dave, Linwood, Jeff. Apress, 2005. P. 242.
29. Apeksha Mehta; Top 15 best database for web applications to Use in 2022. URL: <https://appinventiv.com/blog/top-web-app-database-list/> (Last accessed: 03.06.2023).
30. Difference Between CrudRepository and JpaRepository in Java. URL: <https://www.tutorialspoint.com/difference-between-crudrepository-and-jparepository-in-java> (Last accessed: 04.06.2023).
31. Introduction to Using Thymeleaf in Spring. URL: <https://www.baeldung.com/thymeleaf-in-spring-mvc> (Last accessed: 04.06.2023).