

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО


_____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-наукової програми «Інформаційні технології проєктування»
на тему: «Веборієнтована інформаційна система підтримки діяльності байєра одягу»

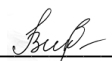
Здобувача (ки) групи ІТ.м-32 Сидоренко Валерії Олександрівни
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.


_____ (підпис)

Валерія СИДОРЕНКО
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник к. т. н., доц. Вікторія АНТИПЕНКО
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)


_____ (підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«_____» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Сидоренко Валерії Олександрівні

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Веборієнтована інформаційна система підтримки діяльності байєра одягу

затверджена наказом по університету від «11» жовтня 2024 р. № 1044-VI

2 Термін здачі студентом кваліфікаційної роботи «06» грудня 2024 р.

3 Вхідні дані до кваліфікаційної роботи процеси ведення бізнесу байєра одягу

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі, методи дослідження, проектування та моделювання веборієнтованої інформаційної системи підтримки діяльності байєра одягу, розробка даного програмного продукту, тестування створеної веборієнтованої інформаційної системи

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) актуальність, мета та задачі дослідження, огляд існуючих аналогів, результати проведеного аналізу аналогів, функціональні вимоги, засоби реалізації, структурно-функціональне моделювання, моделювання варіантів використання, архітектура веборієнтованої інформаційної системи, схема даних реалізованої бази даних, програмна реалізація, демонстрація роботи програмного продукту, тестування, апробація, висновки

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник



(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

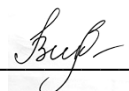
№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Підготовка специфікації	20.09.2024	
2	Розробка веборієнтованої інформаційної системи підтримки діяльності байера одягу	20.11.2024	
3	Тестування програмного продукту	16.11.2024	
4	Впровадження в дію веборієнтовану інформаційну систему	01.12.2024	
5	Налаштування правильної роботи веборієнтованої інформаційної системи підтримки діяльності байера одягу	01.12.2024	

Магістрант



Валерія СИДОРЕНКО

Керівник роботи



к.т.н., доц. Вікторія АНТИПЕНКО

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Веборієнтована інформаційна система підтримки діяльності байєра одягу».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 41 найменувань, 3 додатки. Загальний обсяг роботи – 104 сторінок, у тому числі 68 сторінок основного тексту, 5 сторінок списку використаних джерел, 31 сторінок додатків.

Актуальність роботи обумовлена зростаючою популярністю онлайн-інструментів для удосконалення виконання бізнес-процесів у різних сферах діяльності, включаючи fashion-індустрію. У сучасних умовах, зокрема в Україні, через економічні виклики та вихід багатьох закордонних брендів, роль байєра одягу стає все більш актуальною. Однак, вони часто стикаються зі складнощами при управлінні інформацією про постачальників, асортимент, замовлення та закупівлі, яка надходить із різних джерел. Усе це потребує централізованого підходу для обробки й аналізу даних. Саме в такій ситуації веборієнтована інформаційна система підтримки діяльності байєра одягу може значно спростити їхню роботу, підвищити ефективність виконання процесів та зменшити навантаження на продавця. Таким чином, створення запропонованого програмного продукту є вкрай актуальною.

Мета роботи: розробка веборієнтованої інформаційної системи підтримки діяльності байєра одягу для автоматизації процесу придбання відповідних товарів модних закордонних магазинів зі сторони клієнта та удосконалення механізму ведення бізнесу для продавця. Для досягнення цієї мети виконано аналіз предметної області, розглянуто існуючі аналоги програмних продуктів, сформульовано функціональні вимоги, обрано відповідні інструменти для реалізації, проведено структурно-функціональне моделювання, спроектовано базу даних, а також програмно реалізовано основні модулі даного програмного продукту. Завершальним етапом стало тестування системи для забезпечення її належної роботи.

Результатом дослідження є розроблена веборієнтована інформаційна система, яка надає байєру інструменти для зручного управління асортиментом, дозволяє швидко опрацьовувати замовлення, аналізувати статистику продажів та прибутковість, а також централізувати всю необхідну інформацію в одному місці. Для клієнтів даний програмний продукт пропонує додаткові переваги. Це можливість швидкого доступу до актуального асортименту товарів, прозорість умов співпраці та зручність у комунікації з байєром. Як наслідок розроблена система забезпечить належне управління відповідними бізнес-процесами, що стане вигідним як для продавця, так і для його клієнтів.

Результати представленого дослідження були апробовані на Всеукраїнській науково-практичній студентській конференції у Харківському національному університеті імені В. Н. Каразіна в м. Харків.

Ключові слова: ВЕБОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА, БАЙЄР, ОДЯГ, ПРОГРАМНИЙ ПРОДУКТ, ОНЛАЙН ЗАМОВЛЕННЯ, ПАРСИНГ, LARAVEL, БАЗА ДАНИХ.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Огляд останніх досліджень і публікацій	10
1.2 Аналіз продуктів-аналогів.....	12
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	22
2.1 Мета та задачі дослідження	22
2.2 Методи дослідження.....	23
2.3 Вибір технологій	25
3 ПРОЄКТУВАННЯ ВЕБОРІЄНТОВАНОЇ СИСТЕМИ	27
3.1 Діаграми нотації IDEF0	27
3.2 Діаграма Use Case	33
3.3 Проєктування моделі бази даних	36
4 РОЗРОБКА ВЕБОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ДІЯЛЬНОСТІ БАЙЄРА ОДЯГУ.....	38
4.1 Архітектура веборієнтованої інформаційної системи	38
4.2 Розробка бази даних.....	39
4.3 Програмна реалізація веборієнтованої інформаційної системи	41
4.4 Демонстрація роботи веборієнтованої інформаційної системи.....	45
4.5 Тестування веборієнтованої інформаційної системи.....	62
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	69
ДОДАТОК А.....	74
ДОДАТОК Б.....	87
ДОДАТОК В	89

ВСТУП

Актуальність. Комерційні вебдодатки стали невід’ємною частиною сучасного бізнесу, забезпечуючи відповідним організаціям можливість не лише розширювати свою присутність на ринку, але й вдосконалювати процеси взаємодії з клієнтами, постачальниками та партнерами. Розвиток інтернет-технологій сприяє зростанню кількості таких розробок у різних сферах: від електронної торгівлі до банківських послуг і логістики. Завдяки комерційним вебдодаткам підприємства можуть оперативно реагувати на зміни в потребах споживачів, збільшувати продажі, автоматизувати процеси обробки замовлень і знижувати операційні витрати. І сфера збуту модних речей й аксесуарів із-за кордону не є виключенням.

Тому, у такій ситуації корисним рішенням може стати розробка та застосування веборієнтованої інформаційної системи підтримки діяльності байера одягу. Даний інструмент дозволить удосконалити виконання певних бізнес-процесів, забезпечуючи зручний доступ до закордонного асортименту для потенційних клієнтів. В умовах економічної нестабільності України та відходу іноземних брендів використання такої системи сприятиме не лише збільшенню прибутків зазначених приватних підприємців, але й задоволенню потреб споживачів щодо якісного одягу. Таким чином, дане дослідження є важливим і актуальним сьогодні, адже воно відповідає сучасним запитам ринку та вимогам бізнесу.

Тема. Веборієнтована інформаційна система підтримки діяльності байера одягу.

Мета. Розробка веборієнтованої інформаційної системи підтримки діяльності байера одягу для автоматизації процесу придбання відповідних товарів модних закордонних магазинів зі сторони клієнта та удосконалення механізму ведення бізнесу для продавця.

Для досягнення мети даного проєкту необхідно виконати наступні задачі:

- визначити актуальність дослідження та цільову аудиторію;

- провести аналіз предметної області, здійснивши огляд останніх публікацій за відповідною проблематикою;
- виконати дослідження існуючих аналогічних рішень, виділити їхні переваги та недоліки й сформувавши основні функціональні вимоги до системи;
- обрати методи та програмні засоби реалізації;
- виконати структурно-функціональне моделювання процесів, які забезпечуються з використанням представленого програмного продукту;
- розробити та налаштувати модулі для стабільної роботи даної веборієнтованої інформаційної системи;
- провести тестування для забезпечення якості та надійності функціонування створеного програмного продукту.

Об'єкт дослідження. Процес аналізу та підтримки діяльності роботи байєра одягу.

Предмет дослідження. Веборієнтована інформаційна система підтримки діяльності байєра одягу.

Практична новизна. Використання запропонованої веборієнтованої інформаційної системи забезпечить автоматизацію процесів оформлення замовлень, їх оперативного оброблення, а також ведення детальної статистики та аналітики по даному бізнесу. Це є критичним для успішного ведення власної справи. Автоматизація виконання всіх ключових бізнес-процесів, таких як управління закупівлями, оновлення товарного асортименту, моніторинг наявності речей та їх статусу, сприятиме значному зменшенню часових витрат байєра через виключення його ручного втручання в реалізацію даних дій. Це є значною перевагою сьогодні. Використання представленої веборієнтованої інформаційної системи підтримки діяльності байєра одягу дозволить скоротити час на виконання рутинних операцій, мінімізувати ризик виникнення помилок, що можуть мати місце через людський фактор, і підвищити загальну ефективність управління відповідними бізнес-процесами. Крім того, застосування даного програмного продукту забезпечить актуальний доступ до певних аналітичних даних. Це сприятиме швидкому прийняттю байєром обґрунтованих рішень щодо закупівель, управління запасами та удосконаленню асортименту відповідно до змін ринку.

Результати роботи були апробовані на Всеукраїнській науково-практичній студентській конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку» в Харківському національному університеті імені В. Н. Каразіна.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Зростання електронної комерції має важливе значення для розвитку сучасного бізнесу, особливо в умовах глобальної економічної нестабільності. Сучасні підприємства зустрічають численні виклики, серед яких конкуренція на міжнародних ринках, зміна поведінки споживачів та необхідність швидкої адаптації до нових технологій. У статтях [1-3] розглянуто основні тенденції в електронній комерції та підкреслено важливість використання вебдодатків для спрощення виконання ряду бізнес-процесів, управління ланцюгами постачання та оперативного реагування на зміни в попиті. Застосування таких технологій дозволяє підприємствам удосконалити рівень обслуговування клієнтів, що є критично важливим для підвищення конкурентоспроможності сьогодні.

Дослідження [4] наголошує на важливості користувацького досвіду як основного чинника успіху в електронній комерції. Відповідно до висновків авторів, зручна навігація, висока швидкість роботи та персоналізація є ваговими аспектами, які впливають на утримання клієнтів та їхню лояльність до бренду. Ці фактори набувають особливої ваги в умовах глобальної конкуренції, де доступність та комфорт для користувача відіграють вирішальну роль. Впровадження засобів персоналізації та швидкого доступу до товарів і послуг сьогодні стає необхідністю для підприємств, які прагнуть закріпити свою позицію на ринку.

Застосування аналітичних інструментів у вебдодатках також є важливим аспектом сучасного бізнесу. У роботі [5] підкреслено значення таких засобів для дослідження ринкових тенденцій та прийняття обґрунтованих рішень. Інформація, отримана завдяки системам аналітики, дозволяє бізнесу своєчасно коригувати стратегії. А також оптимізувати процеси закупівель та підвищувати ефективність роботи. Крім

того, впровадження аналітичних інструментів сприяє зниженню витрат і зростанню загальної продуктивності компаній, які працюють у галузі електронної комерції.

У статті [6] наголошується, що правильне ведення статистики значно підвищує здатність бізнесу прогнозувати продажі та знижує ризик втрат. Наприклад, завдяки аналізу даних про попередні збути байєр зможе оцінити попит клієнтів на певні групи товарів, бренди та інше. Це допоможе влучно формувати каталог речей на майбутні сезони.

Для українського ринку, особливо в умовах війни, яка суттєво змінила економічні умови, зникнення багатьох міжнародних брендів створило нові виклики та можливості. У результаті вітчизняні підприємці опинилися в ситуації, де робота з закордонними постачальниками стала не лише вигідною, але й стратегічно необхідною. Придбання товарів за кордоном дозволяє пропонувати клієнтам оригінальні продукти, підтримуючи їх доступність і цінову конкурентоспроможність. Саме про зміни на ринку України з початком повномасштабної війни зазначається у джерелі [7].

Зважаючи на представлені дослідження, розробка веборієнтованої інформаційної системи підтримки діяльності байєра одягу є актуальною та важливою сьогодні. Застосування такого програмного продукту допоможе автоматизувати виконання процесів закупівель, надати власнику можливості для ефективного управління асортиментом і підвищити задоволеність клієнтів за рахунок зручності його використання.

Результати цього дослідження були апробовані в рамках конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку» [8].

1.2 Аналіз продуктів-аналогів

Перед розробкою продукту даного проєкту необхідно здійснити пошук і провести аналіз подібних вебресурсів, що орієнтовані на підтримку діяльності байерів у сфері моди та продажу одягу.

Основними критеріями аналізу є такі:

- оформлення (дизайн, підбір кольорів, фонових зображень, читабельність тексту);
- дотримання стилю;
- зручність навігації;
- автоматизований процес оформлення замовлень;
- повнота інформації.

І оскільки діяльність байерів тільки набирає обертів в Україні, то більшість таких підприємців не мають своїх власних онлайн платформ для продажу відповідних товарів, а базуються в соціальних мережах Telegram [9] або Instagram [10].

Першим було розглянуто Telegram-канал «MY Shopping» [11]. Його автор публікує акційні товари з закордонних сайтів, які можна придбати з її допомогою. Важко оцінити оформлення та стиль даного аналогу, оскільки це не є власним ресурсом, розробленим для управління саме цим бізнесом. Тоді увагу було сконцентровано на зручності такого способу організації взаємодії з клієнтами. Автор публікує як окремі пости про наявний асортимент, де надає посилання на іноземний магазин (рис. 1.1), так і оголошення про знижки в конкретному онлайн-сторі (рис. 1.2). Проте відсутня детальна інформація про товари, вартість послуг байера та безпосередньо умови та алгоритм оформлення замовлення. Проте, канал має ряд функціональних переваг. Наприклад, це зручна система навігації через хештеги. Вона дозволяє швидко знаходити потрібні категорії товарів. Однак, додатковим ускладненням є те, що доступ до деяких закордонних ресурсів потребує застосування Virtual Private Network (VPN) [12]. Це відповідно створює певні незручності

для клієнтів. Уся комунікація між байером і клієнтом відбувається через особисті повідомлення. Для більшості користувачів такий спосіб є зручним, враховуючи популярність і доступність платформи Telegram.

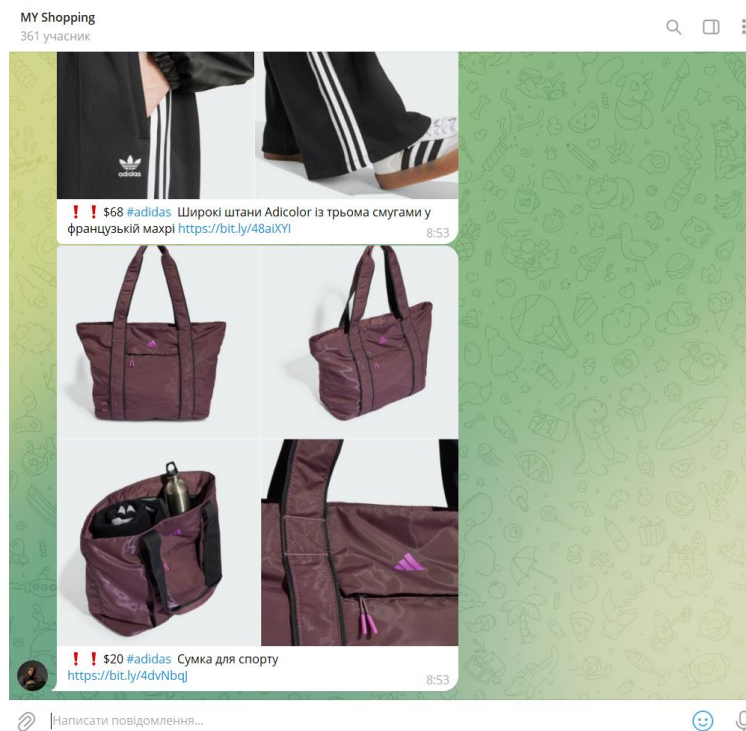


Рисунок 1.1 – Стрічка постів у Telegram-каналі «MY Shopping»

Джерело: [10]

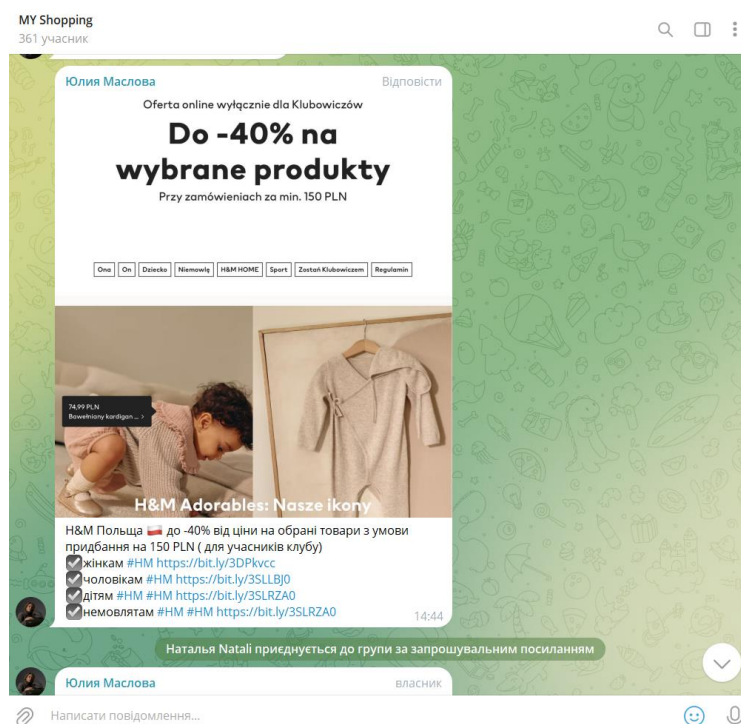


Рисунок 1.2 – Оголошення про знижки в Telegram-каналі «MY Shopping»
Джерело: [10]

Наступним було проаналізовано Telegram-канал «Eduard Store» [13]. Він пропонує клієнтам доступ до широкого асортименту брендovих товарів із закордонних магазинів, використовуючи відповідні сайти. Канал містить публікації з описом товарів, їхньою актуальною ціною, посиланням на оригінальне джерело та додатковими характеристиками (рис. 1.3). Даний підхід забезпечує покупців необхідною інформацією про певні речі. У свою чергу це полегшує прийняття рішення щодо їх придбання. Додатково байєр розміщує новини та дані про наявність товару в Україні (рис. 1.4), який можна купити та отримати в найшвидші терміни. Основним каналом зв'язку для оформлення замовлення є особисті повідомлення. Такий спосіб дозволяє покупцям отримувати індивідуальні консультації. Проте для деяких клієнтів обмеженість можливостей Telegram, як торговельного майданчика, може стати незручністю. Насамперед через відсутність автоматизації виконання процесу купівлі товару або інтеграції з платіжними системами. Незважаючи на це, «Eduard Store» забезпечує високий рівень

комфорту для клієнтів, оскільки Telegram є популярною платформою, яку зручно використовувати на смартфоні.

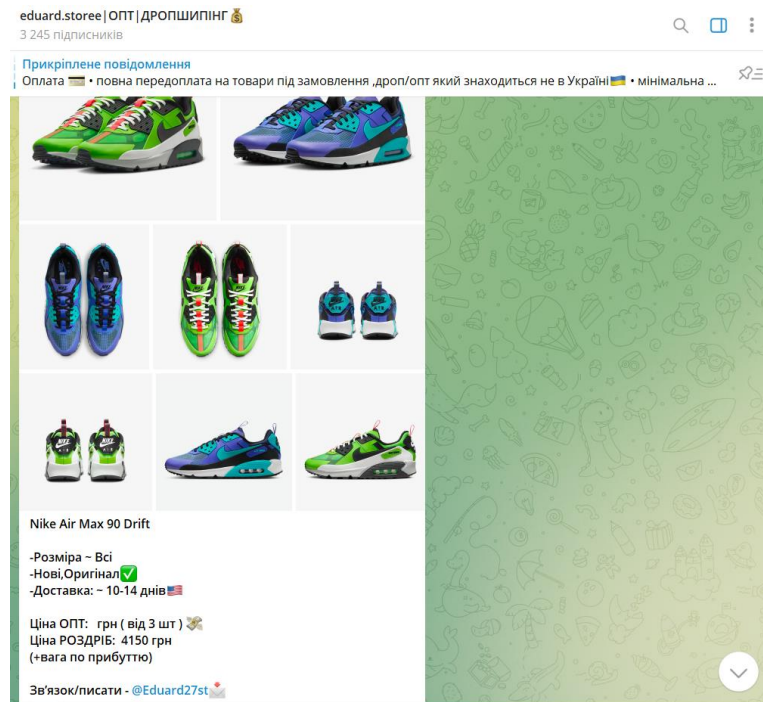


Рисунок 1.3 – Стрічка постів Telegram-каналу «Eduard Store»

Джерело: [12]

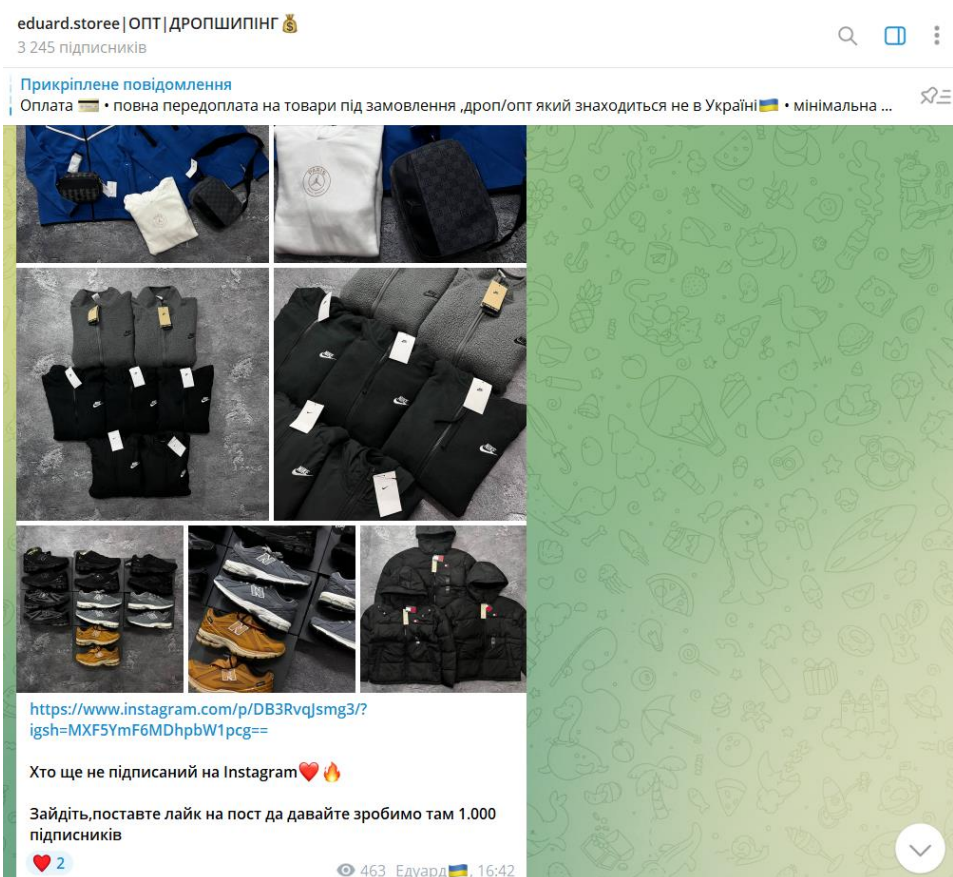


Рисунок 1.4 – Новинний пост з Telegram-каналу «Eduard Store»

Джерело: [12]

Далі було досліджено третій аналог – це Instagram-акаунт «Vinogradova Shop» [14]. Він також спеціалізується на продажі закордонних товарів. Основні переваги цього ресурсу полягають у зручності використання соціальної мережі Instagram як платформи для пошуку та перегляду відповідних предметів одягу, взуття й аксесуарів, а також у якісному візуальному оформленні. Кожен пост містить фотографію товару, короткий опис, ціну та інформацію про наявність в Україні або терміни доставки. Усе це допомагає користувачам швидко зорієнтуватися щодо власних вподобань і бажання придбати відповідну річ (рис. 1.5). Крім того, наявність функції «Stories» дозволяє оперативнo інформувати клієнтів про нові надходження, акції та актуальні знижки (рис. 1.6). Загалом, «Vinogradova Shop» є візуально привабливим і зручним для використання ресурсом. Авторка велику увагу приділяє публікації відео оглядів товарів та відгуків, що підвищує довіру клієнтів до її послуг. Проте

обмежена функціональність Instagram, як торгової платформи, не завжди задовольняє потреби клієнтів, які очікують швидкого й автоматизованого процесу замовлення.

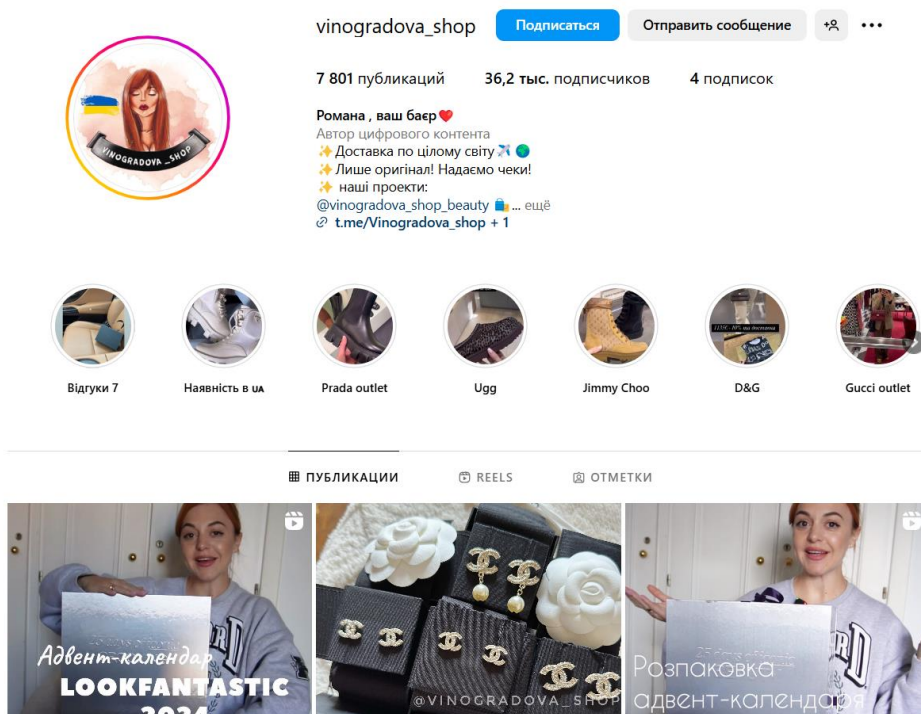


Рисунок 1.5 – Instagram-сторінка «Vinogradova Shop»

Джерело: [13]

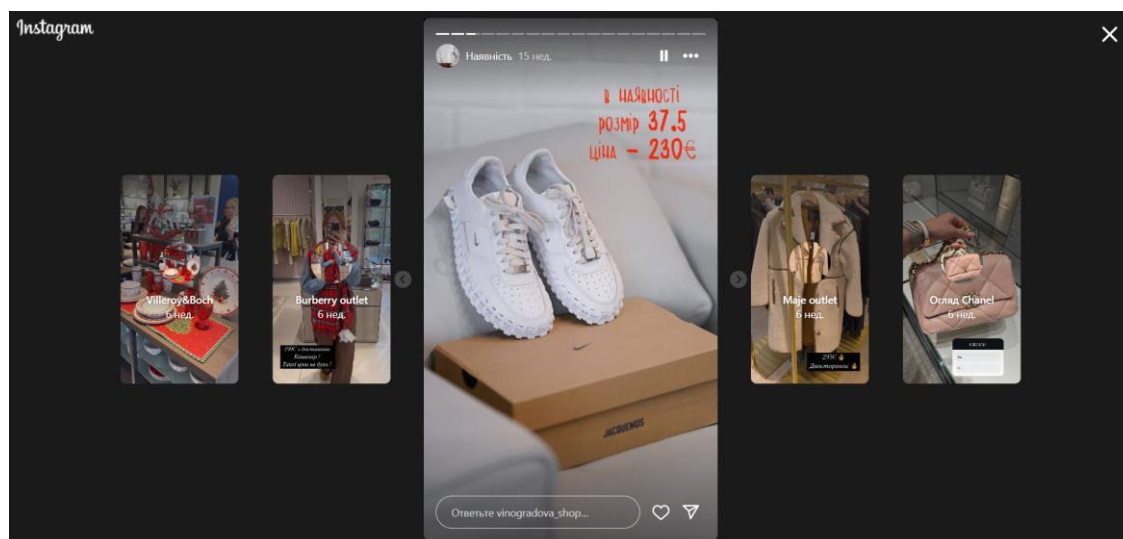


Рисунок 1.6 – Stories від «Vinogradova Shop»

Джерело: [13]

Четвертий аналог, яку було розглянуто, це вебдодаток «Brandshunter» [15]. Останній надає послуги щодо придбання жіночих та чоловічих товарів із Італії (рис. 1.7-1.9). Клієнт отримує повну інформацію про процес шопінгу з даним байєром, відповіді на найпоширеніші питання та доступ до форми зворотнього зв'язку. Проте дизайн даного вебдодатку виглядає застарілим. Йому також бракує детальної інформації про товари. Для клієнтів це може ускладнити прийняття рішення щодо їх купівлі. Крім того, відсутня інтегрована система оформлення замовлень, через що даний процес стає менш зручним для користувачів і вимагає додаткових кроків. У результаті, представлений ресурс не створює відчуття довіри у клієнтів та не стимулює їхню зацікавленість, що є серйозним недоліком для платформи такого типу.

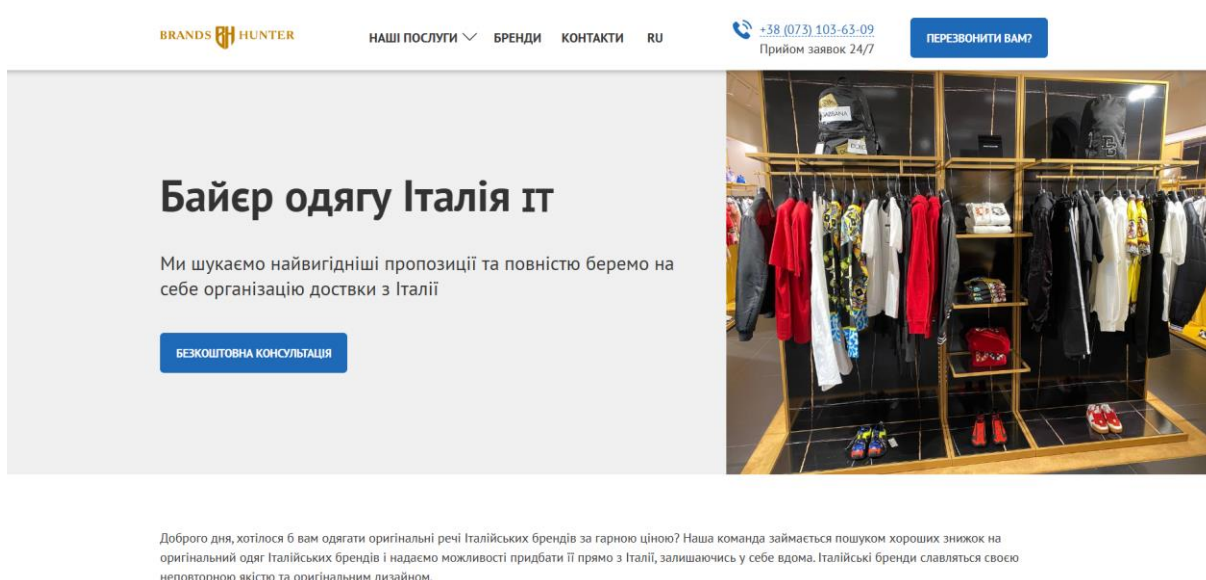


Рисунок 1.7 – Головна сторінка вебдодатку «Brandshunter»

Джерело: [14]

ЗАМОВИТИ ЗАРАЗ

- 02 Менеджер перевіряє та підтверджує наявність вашого розміру.
- 03 Оплата відбувається на реквізити моно або приватбанку. Працюємо по повній передплаті.
- 04 Вказати дані для доставки Нової пошти (область, місто, номер відділення, номер телефону, ПІБ). Також можемо надіслати іншим зручним для вас способом в інші країни.
- 05 Відправлення з Італії в Україну відбувається кожної п'ятниці. Після потрапляння товару на територію України товар відправляється Новою поштою на вказане відділення

Наші переваги

- Спеціалізуємось на гарячих знижках від преміальних Італійських брендів.
- Лише 100% оригінальні Італійські бренди.
- Повністю беремо на себе організацію швидкої доставки з Італії.

Приєм заявок

24/7

Телефонуйте
або залишайте заявку

+38 (073) 103-63-09

НАДІСЛАТИ

ЗАМОВЛЕННЯ ТА КОНСУЛЬТАЦІЯ ПО ТЕЛЕФОНУ:

+38 (073) 103-63-09

ТЕЛЕФОНУЙТЕ ПРЯМО ЗАРАЗ!

ЗАТЕЛЕФОНУВАТИ ВАМ?

Відповіді на популярні запитання

Рисунок 1.8 – Інформаційна блок вебдодатку «Brandshunter»

Джерело: [14]

НАШІ ПОСЛУГИ ▼ БРЕНДИ КОНТАКТИ RU

+38 (073) 103-63-09
Приєм заявок 24/7

ПЕРЕЗВОНИТИ ВАМ?

Головна [IT Бренди](#) [IT Coach](#)

Знижки на Coach

Наша команда займається пошуком гарячих знижок на оригінальний одяг, взуття та аксесуари Coach та повністю беремо на себе організацію швидкої доставки з Італії.

Для Вашої зручності ми викладемо всі гарячі пропозиції в наших телеграм каналах [для жінок](#) та [для чоловіків](#)

ЗАМОВИТИ ЗАРАЗ

Приєм заявок

24/7

Телефонуйте
або залишайте заявку

+38 (073) 103-63-09

НАДІСЛАТИ

Рисунок 1.9 – Сторінка окремого бренду вебдодатку «Brandshunter»

Джерело: [14]

Після детального аналізу наданих аналогів, було визначено їх основні переваги та недоліки. У таблиці 1.1 представлені його результати.

Таблиця 1.1 – Порівняльна таблиця характеристик ресурсів-аналогів

Характеристика/ Онлайн ресурс	MY Shopping	Eduard Store	Vinogradova Shop	Brandshunter
Зручний інтерфейс	Неможливо оцінити	Неможливо оцінити	Неможливо оцінити	+
Сучасне оформ- лення	+	+	+	-
Навігація	+	-	-	+
Наявність автома- тизованої системи замовлень	-	-	-	-
Повна інформація про товар	-	+	-	-
Інформативність	+	+	+	+
Наявність відгуків	-	-	+	-
Якісні зображення	+	+	+	+
Відслідковування замовлень	-	-	-	-
Зручна комунікація байера та клієнта	+	+	+	+

Джерело: побудовано автором

Після аналізу аналогів, можна зробити висновок, що жоден із розглянутих ресурсів не має автоматизованої системи оформлення замовлень. Також більшість зазначених онлайн ресурсів потребують активної залученості байера. Наприклад, при спілкуванні під час оформлення замовлення та отриманні відповідей на поширені питання. Усе це може уповільнювати процес придбання речей і викликати певні незручності для клієнтів.

Тому метою даного проєкту є розробка веборієнтованої інформаційної системи підтримки діяльності байєра одягу. Необхідно реалізувати такий функціонал, який би дозволив мінімізувати зусилля користувача для прямого контакту самого продавця з клієнтом на кожному етапі купівлі товару, зокрема при оформленні замовлення.

Для створення конкурентної та більш привабливої веборієнтованої інформаційної системи, яка б задовольнив визначені потреби потенційних споживачів, варто ретельно продумати та реалізувати сучасний, інтуїтивно зрозумілий дизайн. Власний програмний продукт (ПП) може успадкувати інформативний контент та ілюстративний матеріал із проаналізованих аналогів, але з додаванням певних унікальних функцій. Це спростить користування даною розробкою й дозволить подолати виявлені недоліки.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою даного дослідження є розробка веборієнтованої інформаційної системи підтримки діяльності байєра одягу для автоматизації процесу придбання відповідних товарів модних закордонних магазинів зі сторони клієнта та удосконалення механізму ведення бізнесу для продавця.

Основні вимоги до створюваного ПП наступні:

- забезпечення зручної навігації для швидкої орієнтації на сторінках веборієнтованої інформаційної системи;
- створення зручної форми реєстрації/авторизації для клієнтів;
- забезпечення пошуку товарів за назвою та фільтрації за категоріями, брендами та іншими характеристиками;
- розробка інтуїтивно зрозумілої форми оформлення замовлень;
- наявність функціоналу спільних покупок з друзями;
- створення адміністративної панелі для швидкого управління замовленнями, контролю закупівельних процесів та додавання асортименту;
- розробка блоку статистики по продажах для байєра;
- створення сучасного дизайну для залучення нових потенційних користувачів.

Для досягнення мети представленого проєкту необхідно виконати наступні задачі:

- визначити актуальність дослідження та цільову аудиторію: на цьому етапі буде проаналізовано потреби ринку, зокрема проблеми, із якими стикаються байєри одягу під час здійснення своєї діяльності; особливу увагу буде приділено вивченню ролі таких систем в умовах сучасної економічної ситуації, а також визначено ключові групи користувачів, для яких система буде найбільш корисною;

- провести аналіз предметної області, здійснивши огляд останніх публікацій за відповідною проблематикою;
- виконати дослідження існуючих аналогічних рішень, виділити їхні переваги та недоліки й визначити основні функціональні вимоги до системи: цей етап допоможе сформулювати чітке розуміння того, які можливості необхідно реалізувати в системі, а які аспекти потребують доопрацювання або покращення;
- обрати методи та засоби реалізації даного програмного продукту;
- виконати структурно-функціональне моделювання процесів, які забезпечуються з використанням представленого ПП: на цьому етапі будуть створені діаграми варіантів використання і проектування бази даних, що стане основою майбутньої розробки;
- створити прототип веборієнтованої інформаційної системи підтримки діяльності байєра одягу;
- розробити функціонал даного програмного продукту, що відповідатиме потребам користувачів;
- виконати тестування: це дозволить виявити та виправити можливі помилки, забезпечивши стабільність і якість програмного продукту.

Також на етапі визначення мети проєкту було здійснено планування робіт, детальний опис яких наведено в Додатку А.

2.2 Методи дослідження

Після визначення мети проєкту та вимог до розроблюваної веборієнтованої інформаційної системи, а також формулювання переліку завдань, які потребують виконання, важливо обрати відповідні методи для реалізації останніх. Саме вони будуть використані в рамках даного дослідження.

Загальнонаукові методи дослідження зазвичай поділяються на такі три основні групи:

- методи теоретичного дослідження [16];
- методи емпіричного дослідження [17];
- загальні методи [18].

Теоретичні методи застосовуються для систематизації й формалізації концепцій проєкту. Вони дозволяють сформулювати зв'язки між елементами та їх взаємодію. Це допомагає побудувати логічну структуру від загальних понять до конкретних рішень, які мають бути реалізовані в рамках веборієнтованої інформаційної системи. У процесі дослідження ці методи використовуються для того, щоб проаналізувати теоретичні основи, які стоять за технологіями та принципами роботи створюваного програмного продукту, а також для визначення загальних підходів до розробки.

Емпіричні методи включають різні види збору даних через спостереження, порівняння чи аналіз реальних ситуацій. У даному дослідженні цей метод застосовувався при аналізі ринку існуючих рішень, які можуть бути конкурентами майбутньої системи. Проведення порівняння між різними продуктами-аналогами дозволило виявити їхні сильні та слабкі сторони, а також знайти шляхи для вдосконалення розроблюваного програмного продукту. Емпіричний підхід використовується також під час вивчення публікацій, що висвітлюють сучасні тенденції та новітні досягнення в створенні подібних веборієнтованих інформаційних систем.

Загальні методи дослідження можуть застосовуватися на двох вищезазначених етапах. Як на теоретичному, так і на емпіричному. Одним із таких методів є моделювання, яке сприяє створенню структурних моделей системи. Вони дозволяють побачити взаємозв'язки між її елементами. Цей метод важливий для візуалізації та чіткого розуміння того, як повинен функціонувати розроблюваний програмний продукт. У процесі реалізації даного проєкту здійснюється моделювання для побудови діаграм. Це IDEF0 [19] (для зображення функціональних процесів). А також Use Case [20] (для опису сценаріїв використання системи кінцевими користувачами). Це дозволяє чітко

визначити функціональні можливості власної розробки й описати, як ці можливості будуть реалізовані в рамках даного програмного продукту.

Отже, для розробки веборієнтованої інформаційної системи підтримки діяльності байера одягу необхідно застосовувати комбінований підхід, використовуючи теоретичні, емпіричні та загальні методи. Це дозволить всебічно оцінити вимоги, можливості та обмеження створюваного програмного продукту.

2.3 Вибір технологій

Для створення бекенду було обрано PHP [21] як одну з найбільш поширених мов серверної розробки. Саме вони вирізняється високою сумісністю з вебсерверами та системами керування базами даних. PHP надає девелоперам широкий набір інструментів для створення динамічних вебзастосунків. Використання фреймворку Laravel [22] зумовлено його здатністю значно спростити процес розробки завдяки інтуїтивно зрозумілим механізмам маршрутизації та обробки запитів, розширеній бібліотеці готових модулів, ефективному управлінню базами даних за допомогою ORM-системи Eloquent [23], а також високим рівнем надійності. У фронтенд-розробці було застосовано Vue.js [24]. Завдяки реактивній архітектурі він забезпечує можливість створення динамічних і інтерактивних користувацьких інтерфейсів. Застосування бібліотеки PrimeVue [25] пришвидшило процес розробки за рахунок наявності готових компонентів і стилів. Для зберігання даних обрано MySQL [26] як надійну систему керування реляційними базами даних (БД). Вона забезпечує стабільність і високу продуктивність під час обробки запитів. Крім того, застосовано Redis [27] для кешування даних і зберігання запланованих задач, що підвищує швидкість обробки запитів та знижує навантаження на БД. Особливо за умов інтенсивного застосування застосунку. Такий набір технологій забезпечує ефективність, масштабованість і належну зручність

використання представленої веборієнтованої інформаційної системи, дозволяючи розробникам досягти високої продуктивності та надійності.

3 ПРОЄКТУВАННЯ ВЕБОРІЄНТОВАНОЇ СИСТЕМИ

Після завершення аналізу предметної області, уточнення бізнес-потреб, визначення цілей та формулювання завдань було розпочато проєктування структури веборієнтованої інформаційної системи та моделювання виконання її ключових процесів. У межах цього етапу створено діаграми нотації IDEF0. Вони детально відображають виконання основних процесів. А також їх взаємозв'язки. Ці процеси охоплюють здійснення автоматизованої підтримки діяльності байєра одягу, включаючи створення замовлень та їх опрацювання.

3.1 Діаграми нотації IDEF0

Для моделювання функціональних можливостей розроблюваної веборієнтованої системи застосовуються різні методологічні підходи. Одним із них є IDEF0. Він виступає як інструмент графічного моделювання зв'язків і формалізації процесів у вигляді сукупності взаємопов'язаних функцій. Застосування IDEF0 дозволяє дослідити функціональні аспекти об'єкта без прив'язки до конкретних технологій, зосереджуючись на загальній абстракції. Діаграма містить інформацію про різні типи даних. Це вхідні дані, управління, вихідні дані та ресурси.

Контекстна діаграма демонструє найвищий рівень декомпозиції системи. При цьому забезпечується максимально узагальнене уявлення про її структуру та основні дані, що використовуються. Було прийнято рішення розділити діаграму на дві частини, щоб показати основний процес із різних боків. Це з точки зору клієнта та адміністратора. На рисунку 3.1 зображено контекстну діаграму виконання процесу оформлення замовлення із перспективи користувача.



Рисунок 3.1 – Контекстна діаграма оформлення онлайн замовлення з точки зору гостя

Джерело: побудовано автором (знімок з екрану)

Для більш детального опису процесу було виконано декомпозицію першого рівня. На рисунку 3.2 представлена декомпозицію процесу оформлення замовлення з точки зору клієнта.

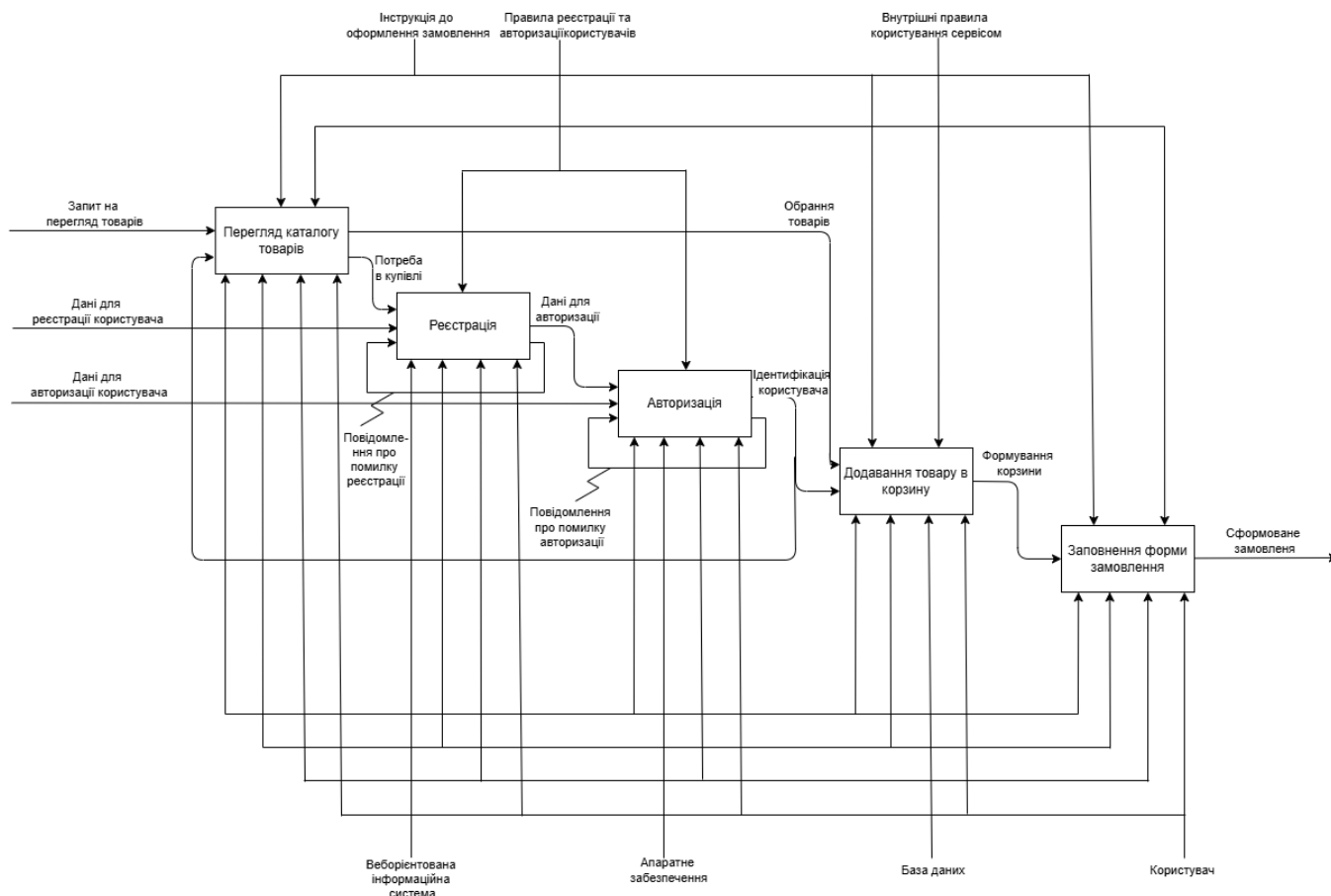


Рисунок 3.2 – Діаграма декомпозиції процесу оформлення онлайн замовлення з точки зору гостя

Джерело: побудовано автором (знімок з екрану)

Інформацію, представлену на діаграмі декомпозиції процесу оформлення онлайн замовлення з точки зору гостя, структуровано у таблиці 3.1.

Таблиця 3.1 – Дані з діаграми декомпозиції процесу оформлення онлайн замовлення з точки зору гостя

Стрілка/ Процес	Вхідні дані	Управління	Механізми	Вихідні дані
Перегляд каталогу товарів	Запит на перегляд товарів. Ідентифікація користувача	Інструкція до оформлення замовлення. Внутрішні правила користування сервісом	Веборієнтована інформаційна система. Апаратне забезпечення. База даних. Користувач	Обрання товарів. Потреба в купівлі
Реєстрація	Дані для реєстрації користувача. Потреба в купівлі	Правила реєстрації та авторизації користувачів		Дані для авторизації
Авторизація	Дані для авторизації користувача			Повний доступ до функціоналу користувача
Додавання товару в кошик	Повний доступ до функціоналу користувача. Обрання товарів	Інструкція до оформлення замовлення. Внутрішні правила користування сервісом		Формування кошика
Заповнення форми замовлення	Формування кошика			Сформоване замовлення

Джерело: побудовано автором

Контекстну діаграму процесу опрацювання замовлення з точки зору адміністратора зображено на рисунку 3.3.



Рисунок 3.3 – Контекстна діаграма процесу опрацювання замовлення з точки зору адміністратора

Джерело: побудовано автором (знімок з екрану)

Для деталізації процесів в даному компоненті було виконано декомпозицію першого рівня, щоб продемонструвати його роботу. На рисунку 3.4 зображено декомпозицію процесу опрацювання замовлення з точки зору адміністратора.

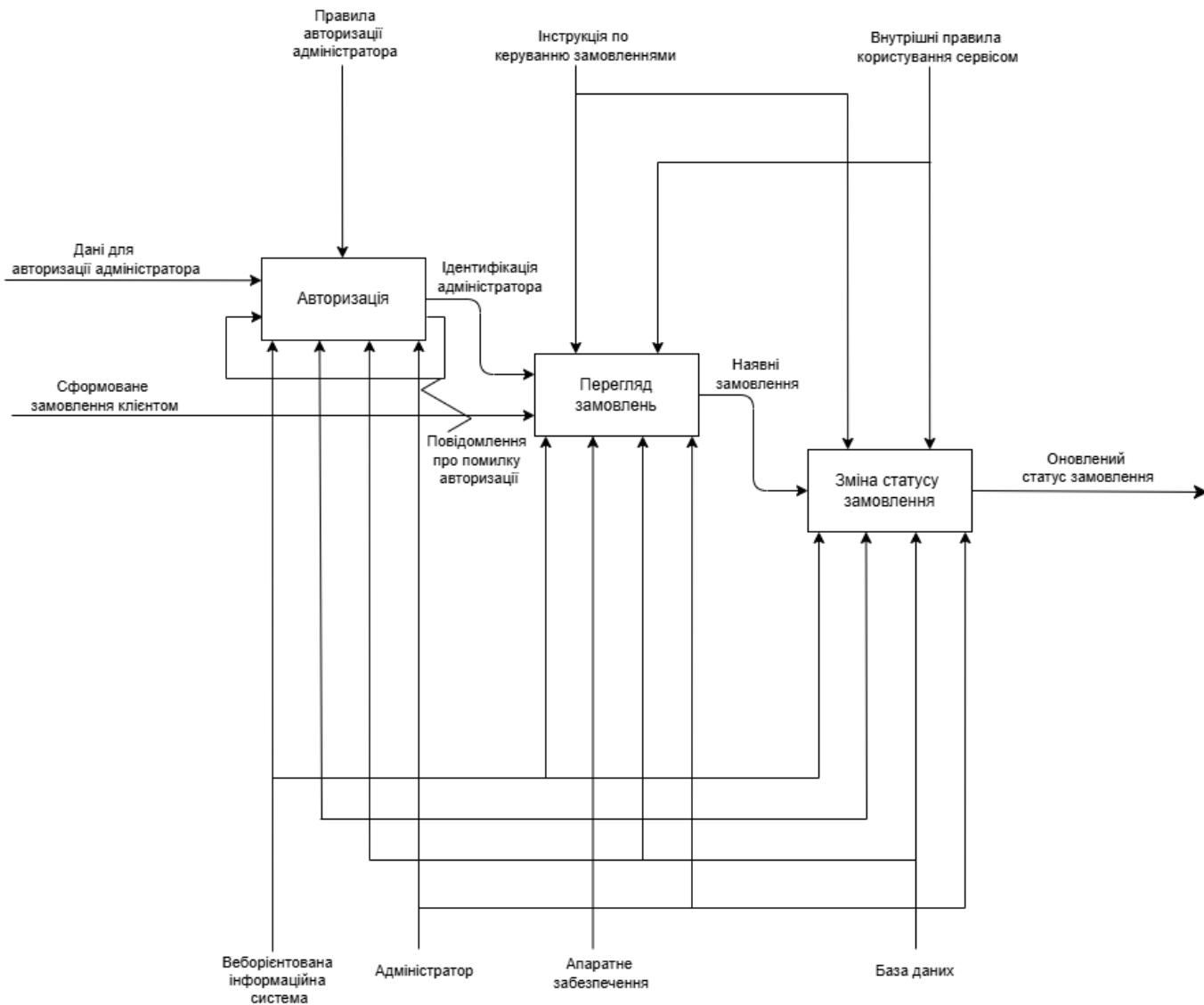


Рисунок 3.4 – Діаграма декомпозиції процесу опрацювання замовлення з точки зору адміністратора

Джерело: побудовано автором (знімок з екрану)

Для структуризації даних, зображених на діаграмі декомпозиції процесу опрацювання замовлення з боку адміністратора, було сформовано таблицю 3.2.

Таблиця 3.2 – Дані з декомпозиції процесу опрацювання замовлення з боку адміністратора

Стрілка/ Процес	Вхідні дані	Управління	Механізми	Вихідні дані
Авторизація	Дані для авторизації адміністратора	Правила авторизації адміністратора	Веборієнтована інформаційна система. Апаратне забезпечення. База даних. Адміністратор	Ідентифікація адміністратора
Перегляд замовлень	Ідентифікація адміністратора Сформоване замовлення клієнтом	Інструкція по керуванню замовленнями. Внутрішні правила користування сервісом		Наявні замовлення
Зміна статусу замовлення	Наявні замовлення			Оновлений статус замовлення

Джерело: побудовано автором

3.2 Діаграма Use Case

Наступним кроком є розробка діаграми варіантів використання даної веборієнтованої системи – Use Case. Вона призначена для візуалізації функціональних можливостей програмного продукту та взаємодії споживачів із ним. На діаграмі відображаються актори – користувачі. Вони взаємодіють із системою. А також їхні можливі сценарії роботи.

Для веборієнтованої інформаційної системи підтримки діяльності байєра одягу були визначені три ключові ролі користувачів: Гість, Користувач та Адміністратор.

Для користувача «Гість» доступні наступні функціональні можливості:

- реєстрація;
- авторизація;
- перегляд каталогу товарів;
- фільтрування товарів;
- додавання товарів в корзину.

Для «Користувача» доступні такі функції:

- авторизація;
- перегляд каталогу товарів;
- фільтрування товарів;
- додавання товарів в корзину;
- створення замовлення;
- запрошення користувачів до замовлення;
- перегляд замовлення;
- перегляд історії статусів.

Для користувача «Адміністратор» доступні наступні функції:

- зміна курсу валют;
- керування службами доставки;
- перегляд товарів;
- додавання товару;
- редагування товару;
- видалення товару;
- перегляд категорій;
- додавання категорії;
- редагування категорії;
- видалення категорії;
- перегляд замовлень;
- фільтрування замовлень за магазинами;
- оновлення статусу замовлення;

– перегляд статистики замовлень.

Діаграма Use Case зображена на рисунку 3.5.

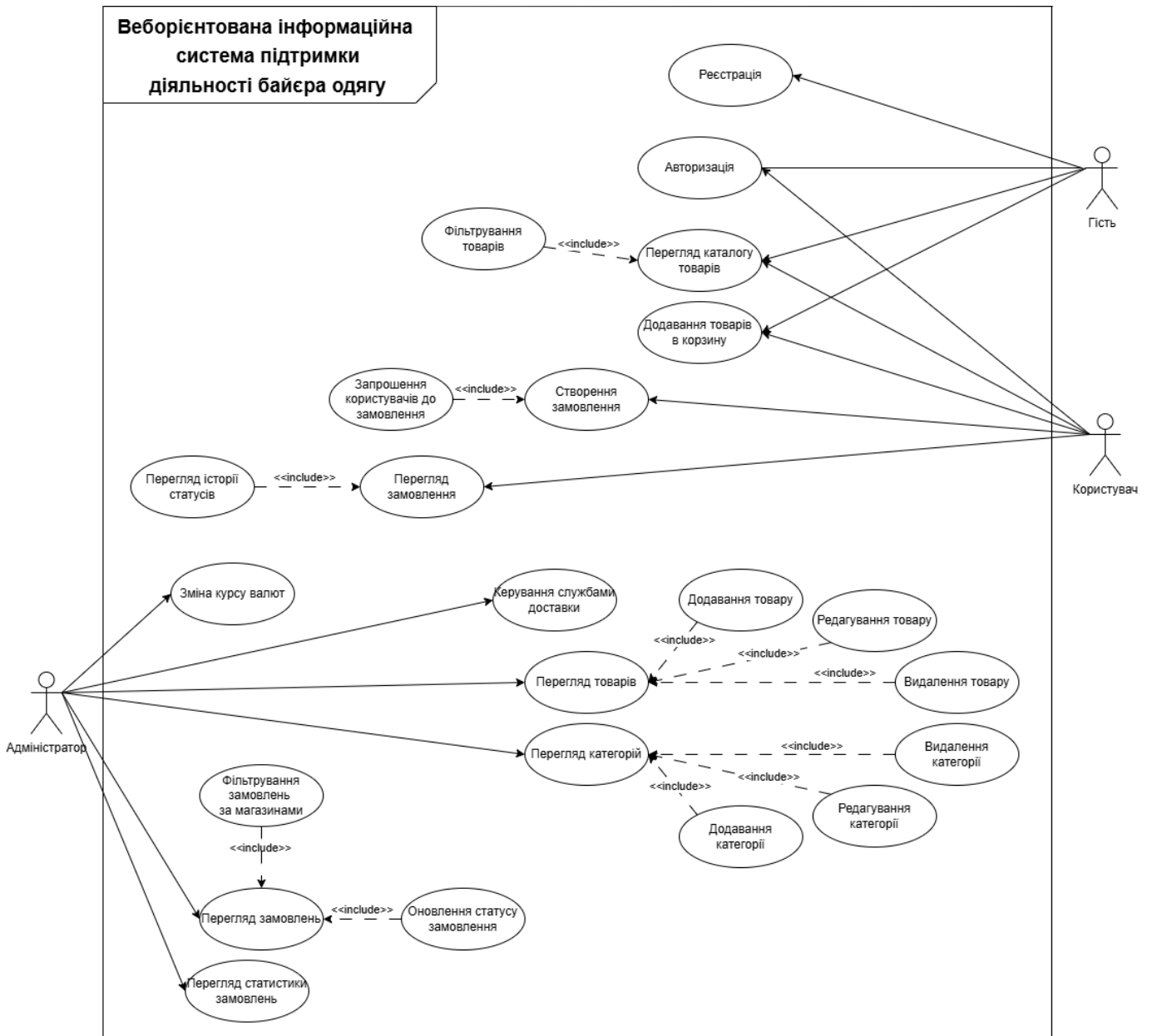


Рисунок 3.5 – Діаграма Use Case

Джерело: побудовано автором (знімок з екрану)

3.3 Проєктування моделі бази даних

Після створення основи проєкту необхідно розробити сховище даних, яке буде містити всю необхідну інформацію для забезпечення його коректної роботи. БД відіграє ключову роль при створенні будь-якої веборієнтованої системи. Це так, оскільки структуроване зберігання інформації в одному місці значно спрощує виконання процесів, пов'язаних із обробкою та маніпуляцією даними.

У процесі проєктування БД визначається кількість таблиць, їхній зміст. А також встановлюються зв'язки між ними. Кожна таблиця має власну структуру. Вона складається зі стовпців, які визначають тип, розмір і специфіку даних, що зберігаються.

У результаті проєктування та розробки відповідної бази даних було сформовано такі сутності:

- users (Користувачі): містить інформацію про користувачів вебзастосунку;
- personal_access_tokens (Токени доступу): містить токени для автентифікації користувачів;
- brands (Бренди): містить дані про бренди товару;
- categories (Категорії): зберігає категорії та підкатегорії товарів;
- products (Товари): зберігає дані про товари;
- attributes (Атрибути): зберігає атрибути товарів, як-от розмір, колір, тощо;
- product_attributes (Атрибути товарів): містить зв'язки товарів та їх атрибутів;
- deliveries (Служби доставки): містить дані про способи доставки товару;
- currencies (Курси валют): містить дані про курс валют до гривні;
- addresses (Адреси доставки): зберігає дані про адреси для доставки замовлення;
- orders (Замовлення): зберігає дані про замовлення користувачів;
- order_products (Замовлені товари): зберігає дані про товари, які належать замовленню;

- `order_products_attributes` (Атрибути замовлених товарів): зберігає атрибути вибраних для замовлення товарів;
- `order_statuses` (Статуси замовлення): містить дані про можливі статуси замовлення;
- `order_timelines` (Історія замовлення): зберігає дані про оновлення статусів замовлення, наприклад замовлення сформовано, доставлено, тощо;
- `cart_products` (Товари у корзині): зберігає дані про товари, які користувач додав у корзину;
- `cart_products_attributes` (Атрибути товарів у корзині): зберігає атрибути доданих у корзину товарів;
- `failed_jobs` (Невиконані задачі): зберігає задачі, які сервер не зміг виконати, як-от оновлення товару.

4 РОЗРОБКА ВЕБОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ДІЯЛЬНОСТІ БАЙЄРА ОДЯГУ

4.1 Архітектура веборієнтованої інформаційної системи

Програмні продукти, створені на основі Laravel, використовують архітектурний підхід Model-View-Controller (MVC) [28]. Ця структура забезпечує чітке розділення між логікою додатка та його користувацьким інтерфейсом. Завдяки даному підходу оновлювати їх стає простіше. Кожна сторінка вебдодатку відповідає окремому файлу View, кожна таблиця бази даних пов'язана з конкретною Model, а кожна функція, що стосується певного «об'єкта», реалізується через Controller. Така організація також спрощує написання коду, оскільки надає доступ до численних вбудованих інструментів.

Роботу запропонованої системи можна поділити на наступні декілька основних етапів, що виконуються під час завантаження сторінки:

- користувач переходить на сторінку за вказаним URL;
- відбувається виклик маршрутизатора, який аналізує URI і спрямовує запит до відповідного методу в контролері;
- контролер створює об'єкт моделі та виконує відповідні операції;
- модель або отримує дані з бази, або додає до неї нову інформацію;
- контролер обробляє шаблон сторінки (View) і передає в нього дані з моделі;
- користувачу доступна сторінка з візуальним оформленням і даними з бази.

Схема архітектури даної веборієнтованої інформаційної системи зображена на рисунку 4.1.

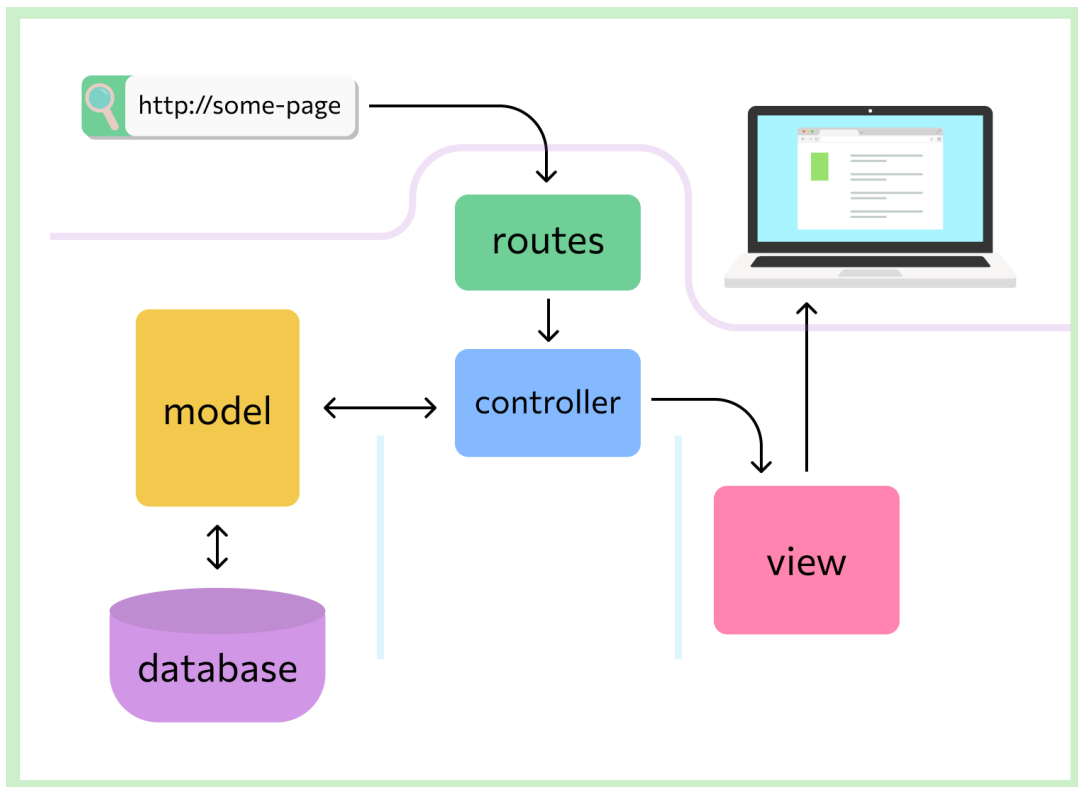


Рисунок 4.1 – Архітектура веборієнтованої інформаційної системи

Джерело: [27]

4.2 Розробка бази даних

База даних, структура якої була розроблена в розділі 3.3, пройшла етап реалізації. На рисунку 4.2 представлена схема реалізованої бази даних засобами DBDiagram.io [29], створеної для веборієнтованої інформаційної системи, яка підтримує роботу байера одягу. Система включає механізм синхронізації каталогу товарів з різних джерел, що забезпечує актуальність асортименту.



Рисунок 4.2 – Логічна модель бази даних

Джерело: побудовано автором (знімок з екрану)

4.3 Програмна реалізація веборієнтованої інформаційної системи

Перед тим, як розпочати безпосередньо реалізацію програмного продукту, необхідно налаштувати робоче середовище. Для цього спочатку створюється окрема папка на персональному комп'ютері. Вона буде слугувати основним сховищем для файлів проєкту. Як середовище розробки обрано PhpStorm [30], що надає розширені можливості для розробки РНР-додатків, зокрема зручний інтерфейс, інструменти для налагодження коду та підтримку популярних бібліотек і фреймворків.

У Laravel маршрути визначають зв'язок між URL-запитами та відповідними методами контролерів, що відповідають за обробку запитів. Було створено шляхи відповідно до функціональних вимог. Нижче наведено код, що задає маршрути для роботи з пошуком, корзиною, особистими та спільними замовленнями. Така їхня структура організована логічно й відповідає вимогам RESTful-дизайну [31]. У свою чергу це сприяє зрозумілості та підтримці проєкту.

Файл `api.php`:

```
// Товари
Route::get('/products', [ProductController::class, 'index']);
Route::get('/products/{product}', [ProductController::class, 'show']);

// Пошук
Route::get('/search', [SearchController::class, 'index']);

// Кошик
Route::get('/cart', [CartController::class, 'index']);
Route::post('/cart', [CartController::class, 'store']);
Route::delete('/cart/{product}', [CartController::class, 'destroy']);

// Замовлення
Route::apiResource('orders', OrderController::class)->only(['index',
'show', 'store']);
```

```
Route::get('/orders/{order}/timeline', [OrderController::class, 'time-
line']);

// Спільні замовлення
Route::apiResource('shared-orders', SharedOrderController::class);
Route::post('/shared-orders/{order}/invite', [SharedOrderControl-
ler::class, 'invite']);
```

Роль основного інструменту для отримання актуальних даних про асортимент, ціни та доступність товарів виконує парсер [32]. Цей компонент відповідає за автоматизацію збору інформації про товари з вебсайтів постачальників. Наприклад, таких як Puma [33]. Клас PumaShopParser відповідає за парсинг даних із сайту Puma. Він використовує бібліотеку Symfony DomCrawler [34] для отримання такої інформації, як назва продукту, опис, ціна, валюта, кольори, доступні розміри та зображення.

Метод parse реалізує процес обробки даних зі сторінки товару. Спочатку він завантажує HTML-контент. При цьому отримує структуру сторінки для подальшого аналізу. Потім із відповідних HTML-елементів витягується ключова інформація про продукт. Це назва, опис, характеристики, ціна та валюта. Отримані дані формуються в зручний для обробки вигляд. Зокрема ціна та валюта адаптуються для подальшого використання в системі, забезпечуючи їх коректне відображення та інтеграцію в загальну базу даних. Приклад реалізації парсера для магазину Puma представлено нижче.

Клас PumaShopParser:

```
public function parse(string $url): ParsedProduct
{
    $this->crawler->addHtmlContent($this->getContent($url));
    $name = $this->crawler->filter('#pdp-product-title')->text();
    $subName = $this->crawler->filter('[data-test-id="pdp-product-
subheader"]');

    if ($subName->count() > 0) {
        $subName = $subName->text();
    } else {
```

```

        $subName = '';
    }

    $price = $this->crawler->filter('[data-test-id="item-price-
pdp"]')->text();
    $color = $this->crawler->filter('.tw-6nh9ql')->text();

    ...
}

```

Клас `ProductTranslator`, код якого наведено нижче, автоматизує переклад назв і описів продуктів за допомогою бібліотеки `ChrisKonnertz\DeepLy` [35]. Вона працює з API `DeepL` [35]. Заголовок та опис продукту відправляється в одному запиті, щоб оптимізувати функцію перекладу.

Клас `ProductTranslator`:

```

public function translateProduct(ParsedProduct $product): ParsedProduct
{
    $translated = $this->translate($this->prepareText($product),
DeepLy::LANG_UK);
    $data = explode('{break}', $translated);

    $product->title = $data[0] ?? '';
    $product->description = $data[1] ?? '';

    return $product;
}

```

Для формування націнки було розроблено модуль конвертації курсу валют, код даного модулю продемонстровано нижче. Тут байєр самостійно встановлює певні дані. Це значення курсу, щоб отримати прибуток із продажу товару. Метод приймає запит. Останній містить дані про актуальні курси валют. Із отриманої інформації він витягує ID валют. А також їхні нові курси. Після цього відбувається масове оновлення відповідних записів у базі даних. Це дозволяє забезпечити їхню

актуальність. Після успішного завершення операції метод формує та повертає відповідь у форматі JSON [36], яка містить повідомлення про успішне оновлення.

Клас CurrencyService:

```
public function getCurrencyBySymbol(string $symbol): ?Currency
{
    return $this->currencies->first(fn (Currency $currency) =>
    $currency->symbol === $symbol);
}

public function convertToUAH(float $price, string $currencyCode):
float
{
    $currency = $this->getCurrencyBySymbol($currencyCode);

    if ($currency === null) {
        return $price;
    }

    return $price * $currency->rate;
}
```

Реалізація функціоналу оновлення статусів замовлення надано нижче:

Клас OrderController:

```
public function update(ChangeOrderStatusRequest $request, Order $order)
{
    try {
        DB::beginTransaction();
        $order->update([
            'updated_at' => now()
        ]);
        $order->timeline()->create([
            'status' => $request->status,
            'comment' => $request->comment
        ]);
    } catch (\Exception $e) {
        DB::rollBack();
        throw $e;
    }
}
```

```

    });
    DB::commit();
} catch (\Throwable) {
    DB::rollBack();
    return response()->json([
        'message' => 'Помилка при оновленні статусу замовлен-
ня '
    ], 500);
}

return response()->json(new OrderResource($order));
}

```

Лістинг коду основних компонентів веборієнтованої інформаційної системи на- дано в додатку В.

4.4 Демонстрація роботи веборієнтованої інформаційної системи

Веборієнтована інформаційна система підтримки діяльності байєра одягу роз- рахована на взаємодію з такими трьома типами користувачів, які мають різні рівні доступу до даних та певні обмеження до функціоналу системи:

- Адміністратор – це роль із найширшими правами в системі; він може до- давати, редагувати та видаляти товари в каталозі, управляти замовленнями, а також переглядати та аналізувати статистику продажів;
- Гість – неавторизований користувач, який має можливість переглядати каталог товарів, здійснювати пошук і фільтрацію, переглядати окремі сторінки това- рів та додавати їх у корзину, але не має доступу до оформлення замовлень або персо- налізованих функцій;

– Користувач – це авторизований користувач, який додатково до можливостей Гостя може створювати та оплачувати замовлення, брати участь у спільних замовленнях, а також відстежувати статус своїх замовлень у особистому кабінеті.

Спочатку розглянемо взаємодію клієнта з даною системою. Перше, що бачить відвідувач – це головна сторінка. Там представлено банер із закликом до покупок, карусель популярних товарів і перелік закордонних магазинів, із якими співпрацює байєр (рис. 4.3). У підвалі головної сторінки розміщено форму зворотного зв'язку для додаткової комунікації клієнта та продавця.

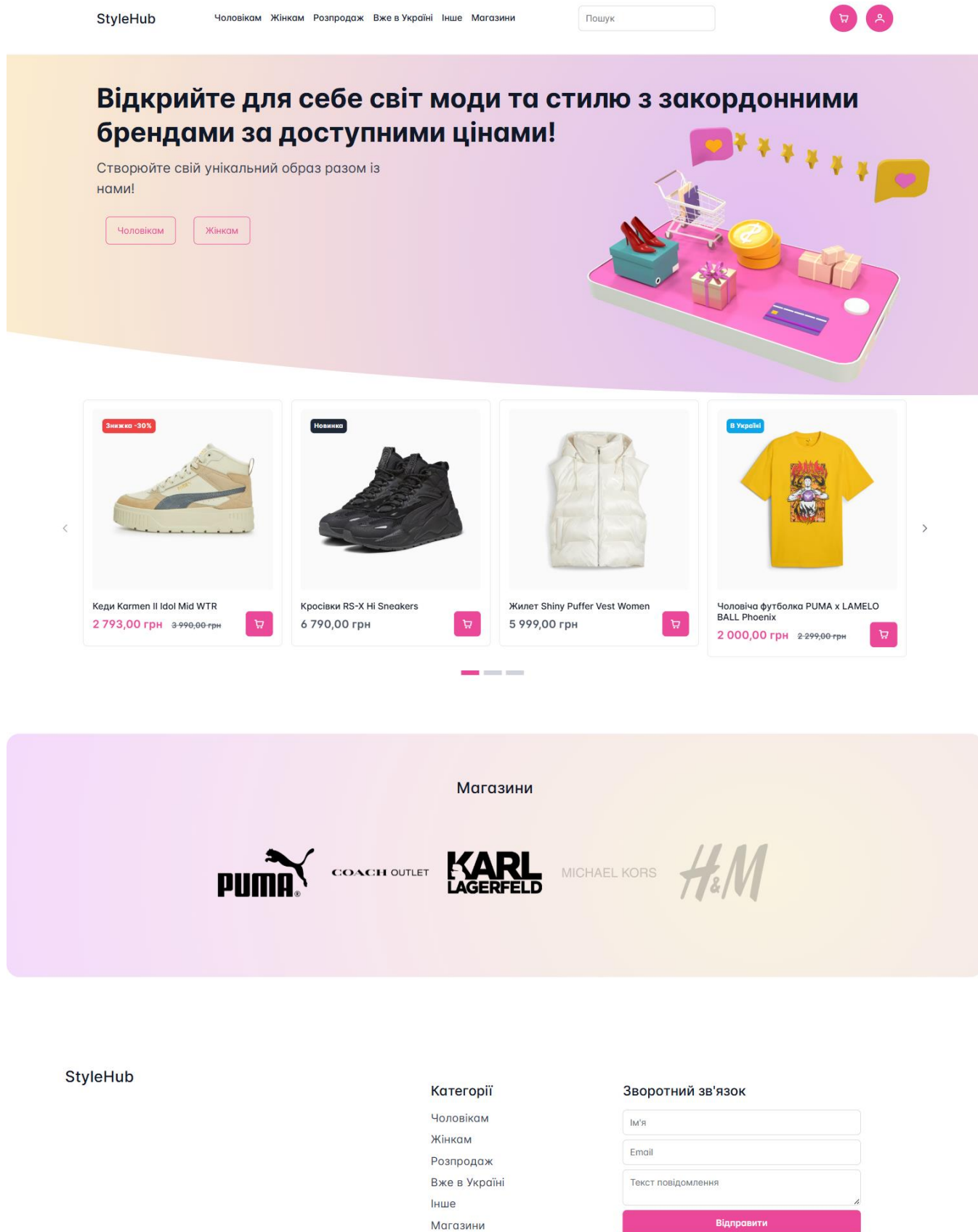


Рисунок 4.3 – Вигляд головної сторінки веборієнтованої інформаційної системи
Джерело: побудовано автором (знімок з екрану)

Із головної сторінки, скориставшись навігаційним меню (рис. 4.3), є можливість перейти до каталогу товарів (рис. 4.4). В останньому передбачено розмежування за такими категоріями, як «Чоловікам», «Жінкам», «Розпродаж», «Наявні товари в Україні» і «Товари відповідних закордонних магазинів».

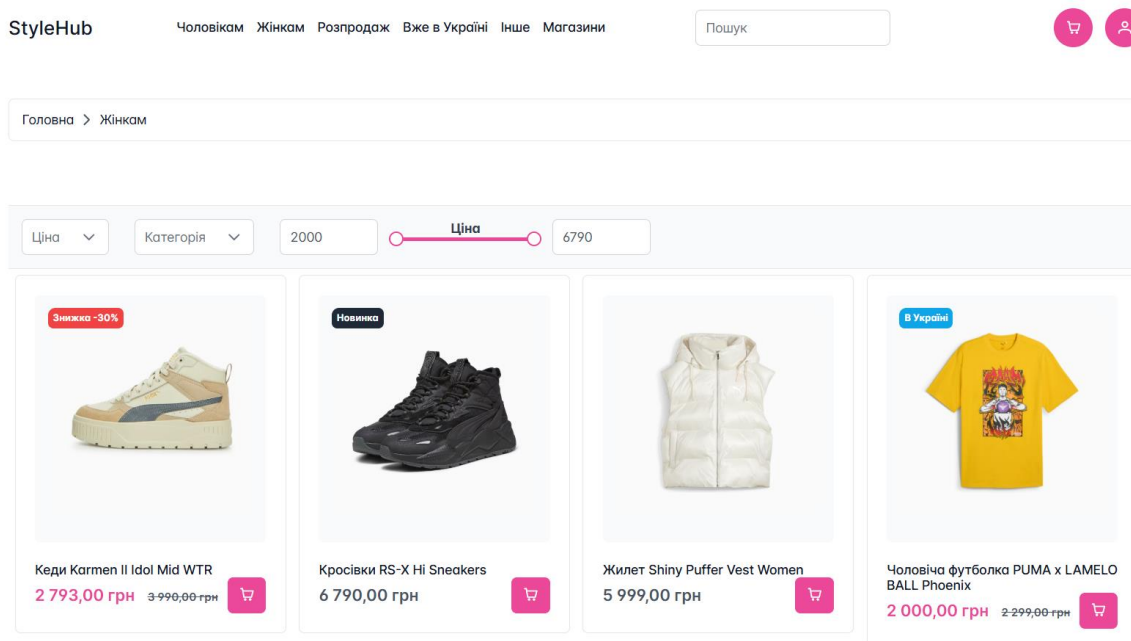


Рисунок 4.4 – Вигляд сторінки каталогу товарів

Джерело: побудовано автором (знімок з екрану)

До списку товарів із каталогу можна застосувати фільтр або сортування. Також доступним є здійснення пошуку за назвою продукції. На рисунку 4.5 продемонстровано застосування фільтру «Взуття» та «Кросівки».

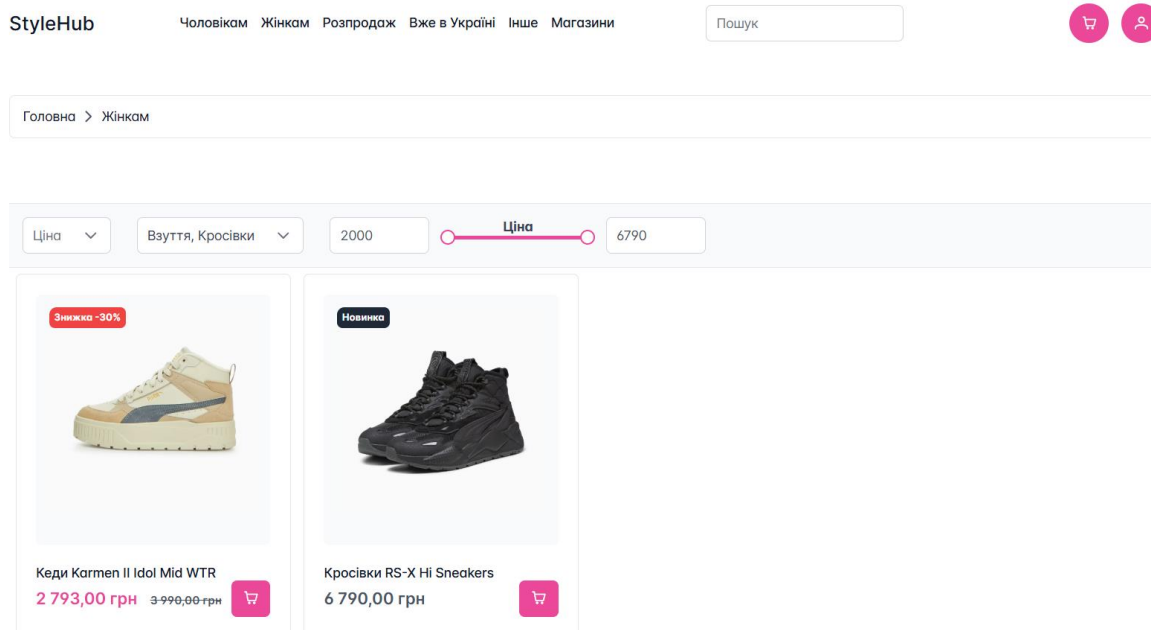


Рисунок 4.5 – Вигляд каталогу товарів із застосуванням фільтрів
Джерело: побудовано автором (знімок з екрану)

Натиснувши на фото або назву товару відкривається окрема картка даного продукту з детальною інформацією (рис. 4.6).

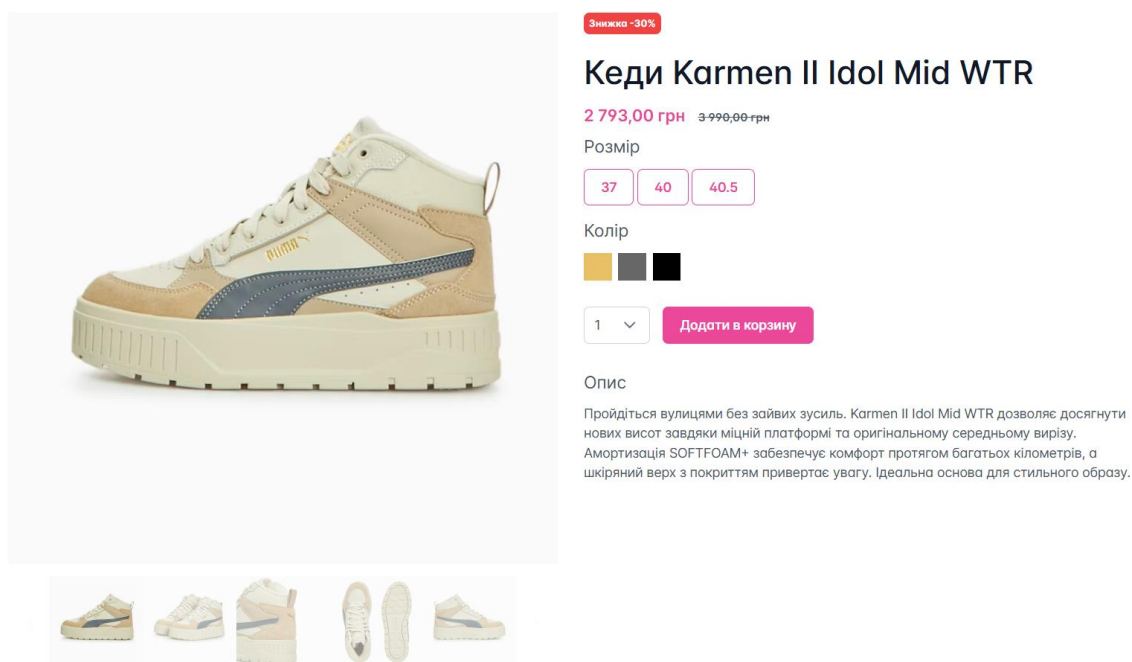


Рисунок 4.6 – Вигляд сторінки окремого товару
Джерело: побудовано автором (знімок з екрану)

Для додавання товару в корзину потрібно натиснути на відповідну кнопку біля самої продукції. Потім він стає доступним у корзині (рис. 4.7). І тепер є можливість оформити замовлення. Для цього треба перейти у корзину. Або можна продовжити шопінг та обрати інший одяг, який сподобався та є бажання придбати.

The screenshot shows the 'КОШИК' (Cart) section of the StyleHub website. At the top, there is a navigation bar with 'StyleHub' and links for 'Чоловікам', 'Жінкам', 'Розпродаж', 'Вже в Україні', 'Інше', and 'Магазини'. A search bar with the text 'Пошук' is also present. On the right, there are icons for a shopping cart and a user profile.

The cart items are listed as follows:

- Кеди Karmen II Idol Mid WTR**
Розмір: 40
Колір: Бежевий
Артикул: 100000-1
Quantity: 1 (dropdown menu)
Price: 2 793,00 грн
Action: [Видалити](#)
- Чоловіча футболка PUMA x LAMELO BALL Phoenix**
Розмір: L
Колір: Жовтий
Артикул: 100001-1
Quantity: 1 (dropdown menu)
Price: 2 000,00 грн
Action: [Видалити](#)

ДЕТАЛІ ЗАМОВЛЕННЯ

Сума: 4 793,00 грн
Всього: 4 793,00 грн

Buttons: [Оформити замовлення](#), [Створити спільне замовлення](#)

Рисунок 4.7 – Вигляд корзини з доданими товарами

Джерело: побудовано автором (знімок з екрану)

Якщо користувач не авторизований, то стає доступним повідомлення про потребу зареєструватись або авторизуватись. На рисунках 4.8-4.11 представлені форми для входу в систему та створення акаунту для нового клієнта.

The screenshot shows a login form with the title "Вхід в акаунт". It contains two input fields: "Email" with the value "user@mail.com" and "Пароль" (Password) with a placeholder "Пароль" and a visibility toggle icon. Below the password field is a link "Забули пароль?". At the bottom, there is a prominent pink button labeled "Увійти" and a smaller pink link "Зареєструватися".

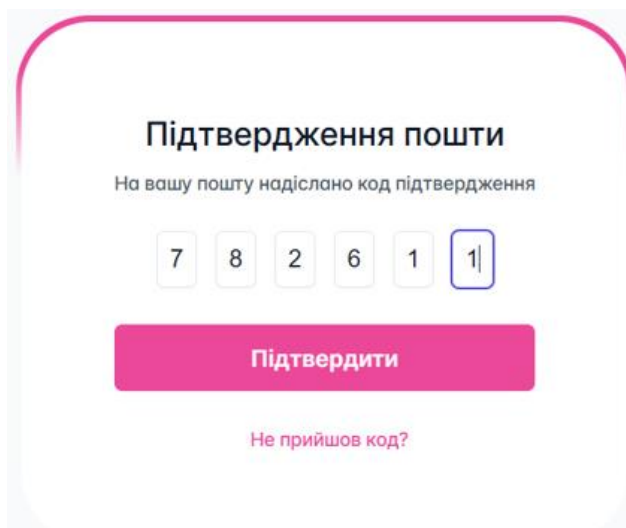
Рисунок 4.8 – Вигляд форми авторизації

Джерело: побудовано автором (знімок з екрану)

The screenshot shows a registration form with the title "Реєстрація акаунту". It features an "Email" input field containing "sydorenkot21@gmail.com". Below the field is a red error message "Email обов'язковий". At the bottom, there is a prominent pink button labeled "Продовжити" and a smaller pink link "Увійти".

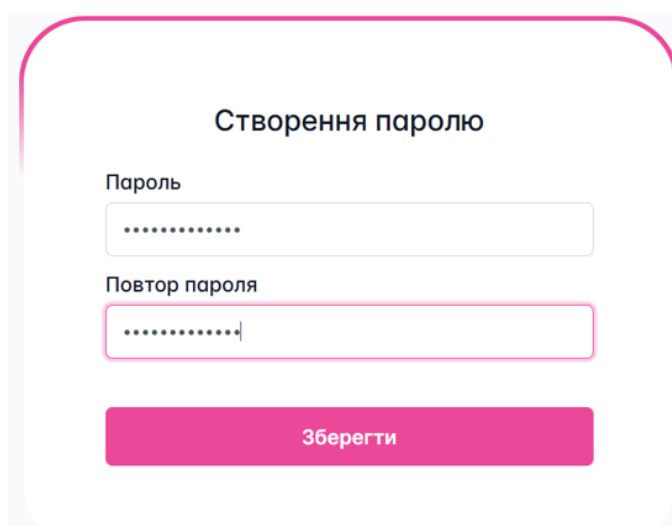
Рисунок 4.9 – Вигляд форми реєстрації через email

Джерело: побудовано автором (знімок з екрану)



The screenshot shows a confirmation code input form. At the top, the title is "Підтвердження пошти" (Post Confirmation). Below it, the text reads "На вашу пошту надіслано код підтвердження" (A confirmation code has been sent to your email). There are six input fields for the code, containing the digits 7, 8, 2, 6, 1, and 1. The last field is active, with a cursor. Below the input fields is a pink button labeled "Підтвердити" (Confirm). At the bottom, there is a link that says "Не прийшов код?" (Code not received?).

Рисунок 4.10 – Вигляд форми вводу коду для підтвердження email
Джерело: побудовано автором (знімок з екрану)



The screenshot shows a password creation form. The title is "Створення паролю" (Create Password). There are two input fields: "Пароль" (Password) and "Повтор пароля" (Repeat Password). Both fields contain masked characters (dots). Below the input fields is a pink button labeled "Зберегти" (Save).

Рисунок 4.11 – Вигляд форми створення паролю
Джерело: побудовано автором (знімок з екрану)

Авторизувавшись користувач може продовжити оформлення замовлення, заповнивши відповідні поля форми (рис. 4.12). Після цього клієнта перенаправляє на модуль для здійснення оплати (рис. 4.13). Для відстеження статусу замовлення користувач натискає на іконку особистого кабінету та обирає пункт «Історія замовлень» (рис. 4.14). Приклад історії статусів замовлення представлено на рисунку 4.15.



Персональні дані **Доставка**

Персональні дані

Ім'я

Валерія

Прізвище

Сидоренко

Номер телефону

+380506985249

Далі

ВАШЕ ЗАМОВЛЕННЯ



Кеди Karmen II Idol Mid WTR

Розмір: 40
Колір: Бежевий
Артикул: 100000-1

1 шт. 2 793,00 грн



Чоловіча футболка PUMA x LAMELO BALL Phoenix

Розмір: L
Колір: Жовтий
Артикул: 100001-1

1 шт. 2 000,00 грн

Сума: 4 793,00 грн

Всього: 4 793,00 грн

Рисунок 4.12 – Вигляд сторінки для оформлення замовлення

Джерело: побудовано автором (знімок з екрану)

The,INC TEST MODE

Замовлення №ORD-0000001

UAH 4,793.00

Оплата покупок на сайті StyleHub

Pay

Or pay with card

Email

Card information

1234 1234 1234 1234

MM / YY CVC

Cardholder name

Full name on card

Country or region

Ukraine

Securely save my information for 1-click checkout
Pay faster on The,INC and everywhere Link is accepted.

Pay

Powered by stripe | Terms Privacy

Рисунок 4.13 – Вигляд сторінки для оплати

Джерело: побудовано автором (знімок з екрану)

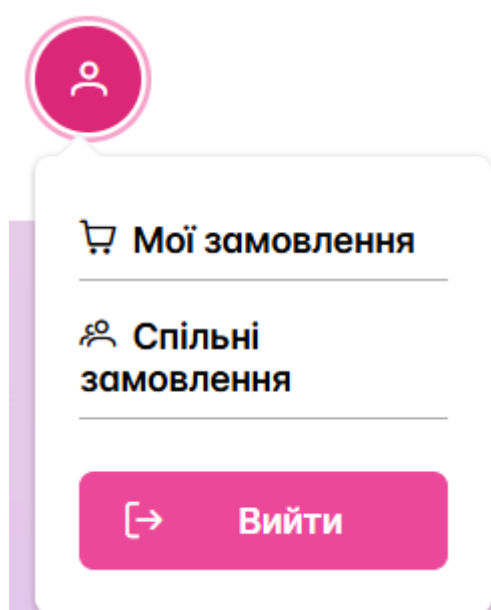


Рисунок 4.14 – Вигляд випадаючого списку з функціями особистого кабінету
Джерело: побудовано автором (знімок з екрану)

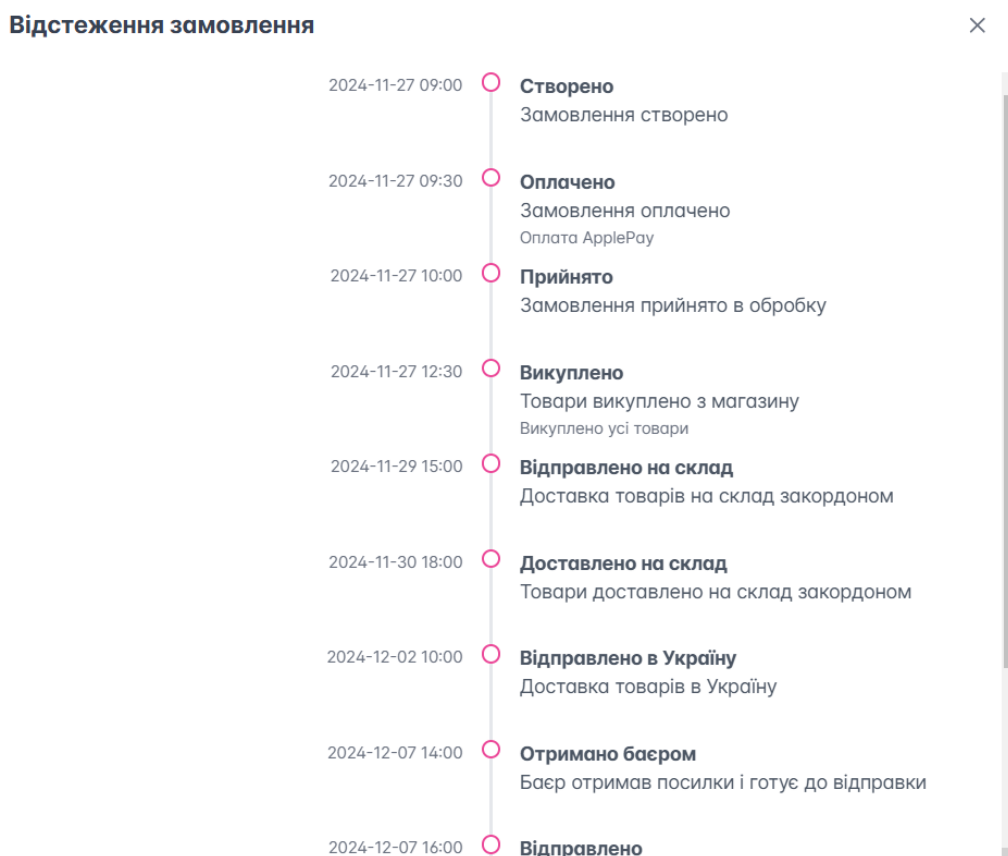


Рисунок 4.15 – Вигляд вікна з історією статусів замовлення
Джерело: побудовано автором (знімок з екрану)

Для зменшення вартості доставки клієнт може запропонувати друзям чи родичам замовити товари разом. Тому було реалізовано функцію спільних покупок. Є таких два варіанти оформлення таких покупок:

- коли корзина порожня: потрібно натиснути кнопку «Створити спільну покупку» (рис 4.16);
- якщо корзина вже сформована: є аналогічна кнопка, що надає посилання, яким можна поділитись (рис. 4.17).

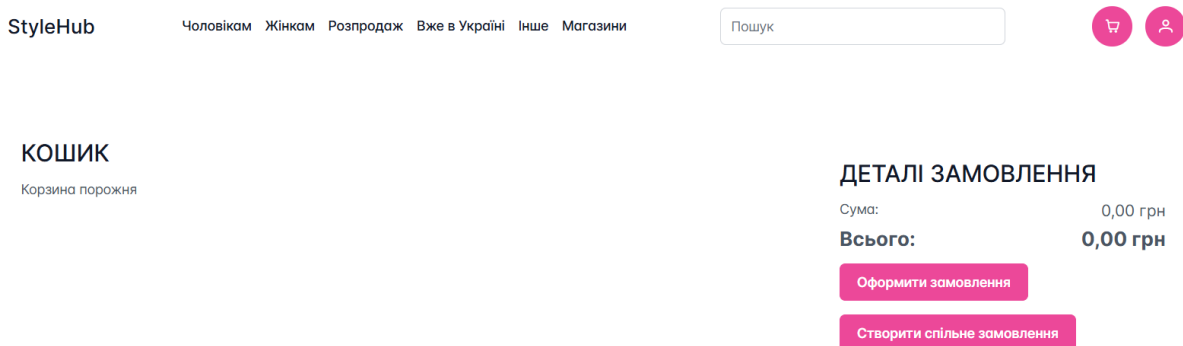


Рисунок 4.16 – Вигляд порожнього кошику

Джерело: побудовано автором (знімок з екрану)



КОШИК



Кеди Karmen II Idol Mid WTR

Розмір: 40
Колір: Бежевий
Артикул: 100000-1

1 2 793,00 грн

Видалити



Чоловіча футболка PUMA x LAMELO BALL Phoenix

Розмір: L
Колір: Жовтий
Артикул: 100001-1

1 2 000,00 грн

Видалити

ДЕТАЛІ ЗАМОВЛЕННЯ

Сума: 4 793,00 грн

Всього: 4 793,00 грн

Оформити замовлення

Створити спільне замовлення

Рисунок 4.17 – Вигляд наповненого кошику товарами

Джерело: побудовано автором (знімок з екрану)

Далі розглянемо функціонал для адміністратора. Останній авторизується за такими даними, як логін: admin, пароль: 11111111. Після успішного входу адміністратора відразу стає доступною сторінка з таблицею замовлень, де може відфільтрувати їх відповідно статусу (рис. 4.18).

№ замовлення	Спосіб доставки	Адреса доставки	Коментар	Сума замовлення	Статус	Деталі
ORD-0000001	Нова Пошта	м. Київ, Відділення №23	Відправте у коробці	4 793,00 грн	Статус	Деталі
ORD-0000002	Meest	м. Одеса, Відділення №10		5 800,00 грн	Статус	Деталі

Рисунок 4.18 – Вигляд сторінки з таблицею замовлень в адміністративній панелі

Джерело: побудовано автором (знімок з екрану)

Для перегляду деталей замовлення потрібно натиснути на стрілочку, що розміщена перед його номером. Приклад замовлення зображено на рисунку 4.19.

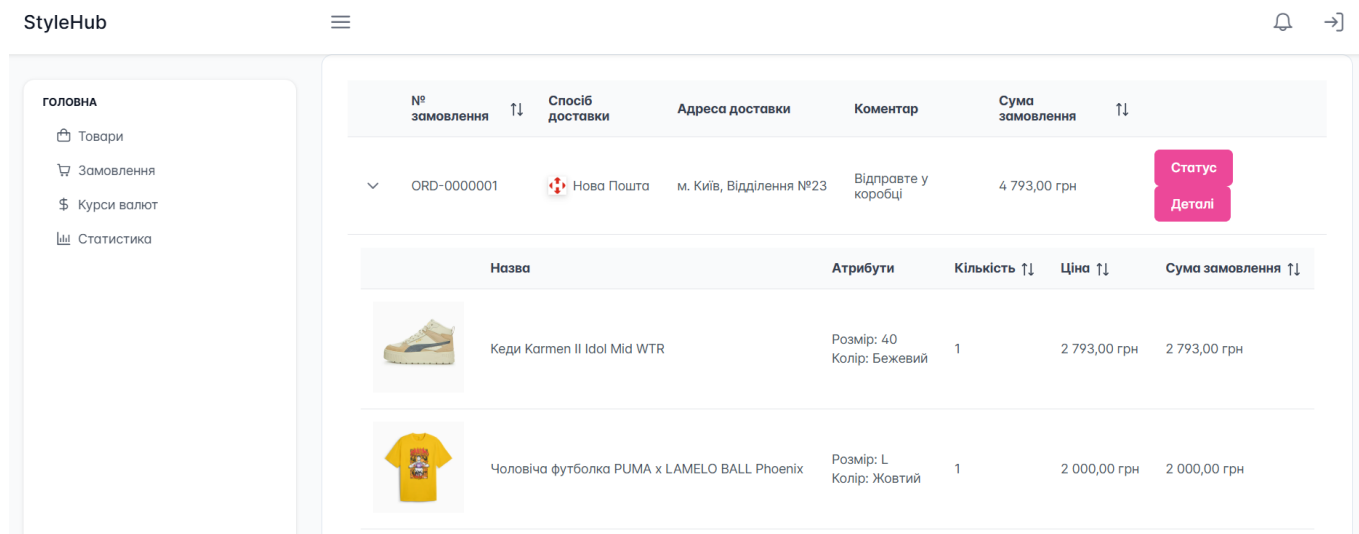


Рисунок 4.19 – Вигляд сторінки з детальною інформацією про замовлення

Джерело: побудовано автором (знімок з екрану)

Головна функція адміністратора – це управління замовленнями, а, отже, зміна їхніх таких статусів, які передбачені системою:

- створено – замовлення створено, але не оплачено;
- оплачено – замовлення створено та оплачено;
- спільне замовлення – доступне додавання товарів до спільного замовлення;
- очікування підтвердження – адміністратор взяв в роботу замовлення та перевіряє наявність товарів;
- прийнято – замовлення прийнято в обробку;
- відхилено – замовлення відхилено адміністратором на етапі перевірки наявності товару;
- відмінено – замовлення відмінено клієнтом;
- викуплено – товари викуплено з магазину байером;
- відправлено на склад – доставка товарів на склад закордоном;

- доставлено на склад – товари доставлено на склад закордоном;
- відправлено в Україну – доставка товарів на склад в Україні;
- доставлено в Україну – замовлення доставлено на склад в Україні;
- отримано байєром – байєр отримав посилки та готує їх до відправки;
- відправлено – відправлено до місця клієнта;
- доставлено – доставлено до місця призначення;
- отримано – отримано покупцем;
- відмова – відмова від отримання;
- очікується – очікуються товари в замовленні (статус для спільних замовлень).

До кожного статусу за потреби можна додати коментар. Наприклад, вписати номер товарно транспортної накладної (ТТН) або причину, із якій було відхилено замовлення. На рисунку 4.20 зображено форму для зміни статусу замовлення.

Змінити статус ×

Статус

Відправлено ▾

Коментар

ТТН 240455151554414

✓ Зберегти

Рисунок 4.20 – Вигляд форми для зміни статусу замовлення

Джерело: побудовано автором (знімок з екрану)

Для наповнення каталогу товарами розроблено зручну адміністративну панель (рис. 4.21). Адміністратор може вставити посилання на товар із закордонного сайту. Після цього система автоматично парсить дані та заповнює відповідні поля, включно з автоматичним перекладом інформації. Байер має можливість редагувати цю інформацію для забезпечення їхньої точності та відповідності. Для зменшення рутинної роботи адміністратора була впроваджена функція автоматичного оновлення даних, яка раз на добу актуалізує ціни та доступність розмірів, гарантуючи, що каталог завжди містить нагальну інформацію для користувачів.

StyleHub ☰ 🔔 →

ГОЛОВНА

- Товари
- Замовлення
- Курси валют
- Статистика

Додавання товару

Посилання на товар

Назва сайту

Назва товару

Переклад назви товару

Категорії

Розмір

Колір

Ціна
 =

[+ Опис товару](#)

— Переклад опису товару

AI **B** *I* U

Опис

Випустіть свого внутрішнього гонщика з цією сезонною курткою. Комір-стійка, витончені значки та надійні прорізи кишені - це все про деталі. Ребристі манжети та поділ забезпечують щільне прилягання, а яскравий логотип PUMA завершує образ. На старт, увага, марш!

Виготовлено щонайменше на 50% з перероблених матеріалів

Облягаючий крій

Тканина твіл

Регулярна довжина

Повна застібка на блискавку

Довгі рукави

Фірмовий логотип PUMA

Стиль: 629914_61

Колір: Glacial Gray

[Читати далі](#)

Рисунок 4.21 – Вигляд форми додавання товару
Джерело: побудовано автором (знімок з екрану)

В адміністративній панелі передбачена функція встановлення курсу валют для автоматичного розрахунку націнки на товари. Дана сторінка продемонстрована на рисунку 4.22.

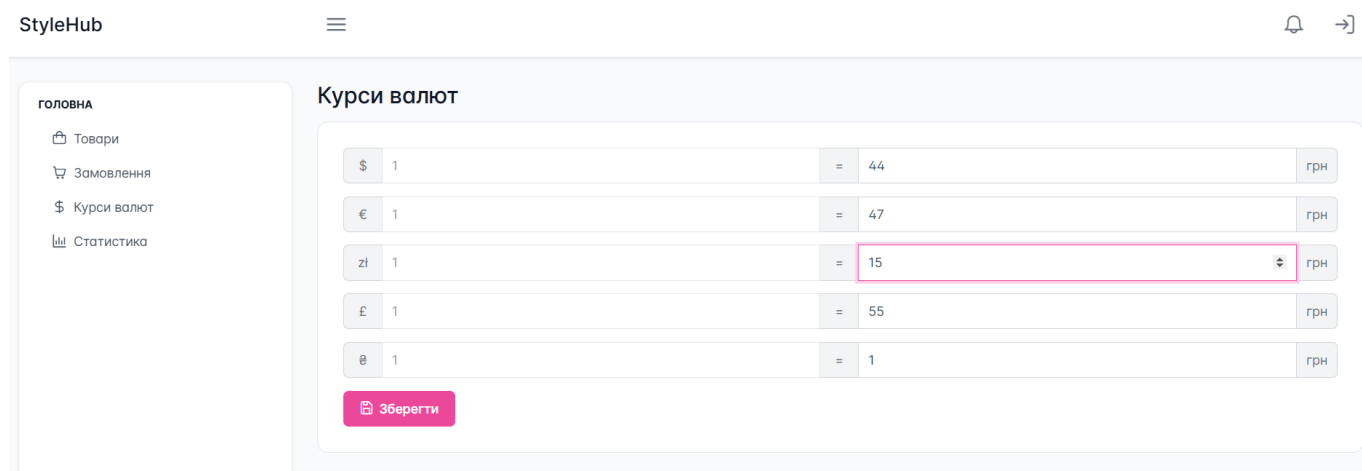


Рисунок 4.22 – Вигляд сторінки для встановлення курсу валют

Джерело: побудовано автором (знімок з екрану)

Розділ «Статистика» створений для зручного відображення ключових показників власної діяльності (рис. 4.23). У цьому розділі адміністратор може переглянути динаміку продажів за місяцями, кількість замовлень зроблених із певного закордонного магазину, а також порівняльну характеристику товарних категорій у вигляді зрозумілих графіків та діаграм.

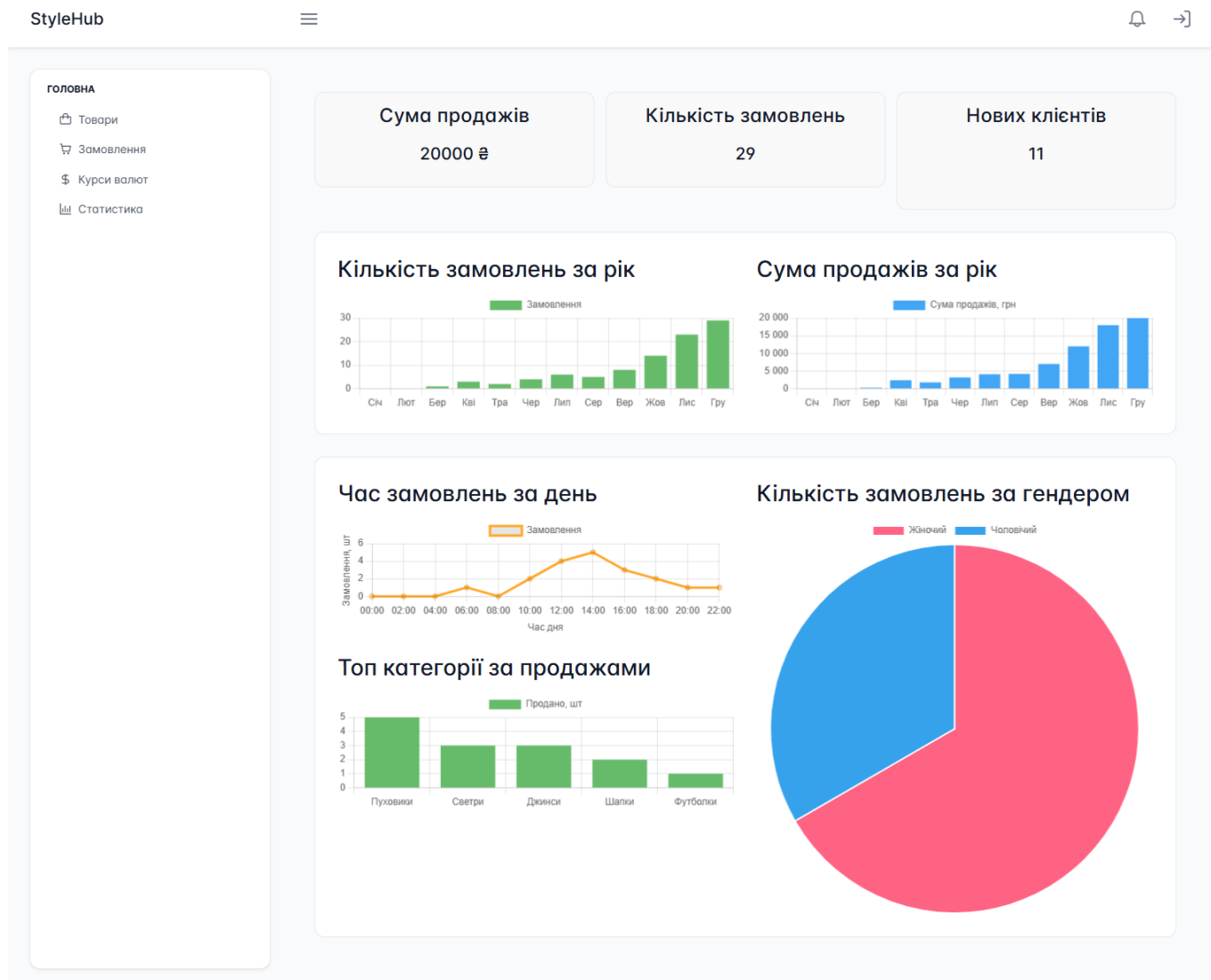


Рисунок 4.23 – Вигляд сторінки зі статистикою
Джерело: побудовано автором (знімок з екрану)

4.5 Тестування веборієнтованої інформаційної системи

Після завершення програмної реалізації було проведено тестування розробленої веборієнтованої інформаційної системи для перевірки її працездатності та відповідності очікуваним результатам. Основна увага була зосереджена на

функціональному тестуванні. Воно охоплювало перевірку коректності роботи посилань, маршрутизації, обробки невалідних даних у формах, а також функціональності ключових компонентів даної системи.

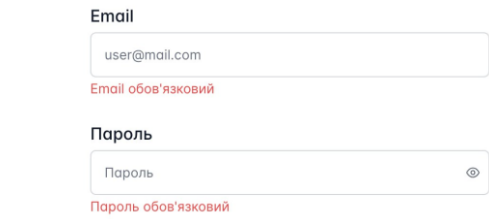
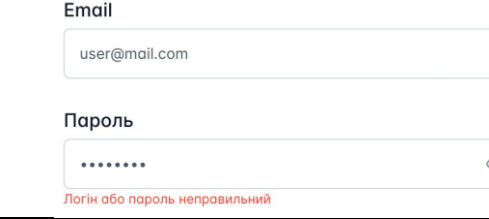
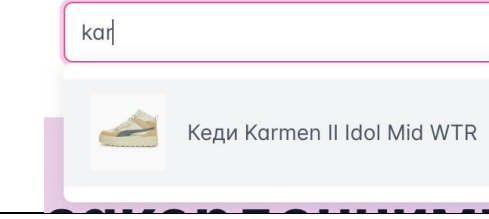
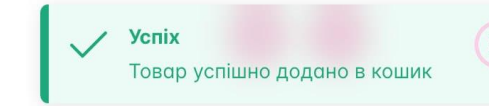
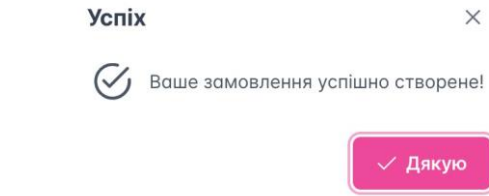
Насамперед протестовано процеси реєстрації та авторизації користувачів, перевірено правильність застосування валідаційних правил у формах. Особлива увага приділялася логіці переспрямування користувачів залежно від їхньої ролі в системі (гостя, зареєстрованого користувача чи адміністратора) та коректності відпрацювання навігаційних елементів.

Для тестування використовувався метод «чорного ящика» [37]. Це дозволило оцінити рівень якості функціонування системи, не аналізуючи її програмний код. Основні результати тестування та висновки щодо роботи запропонованого програмного продукту подано у таблиці 4.1.

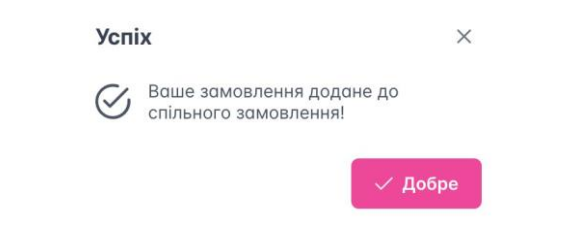
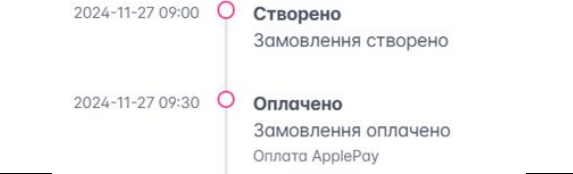
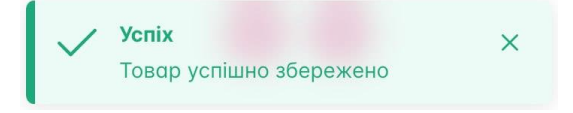



Таблиця 4.1 – Результати тестування веборієнтованої інформаційної системи

№	Назва	Очікуваний результат	Фактичний результат	0/1
1	Перевірка форми реєстрації при спробі використання email, який уже зареєстрований у системі	Виводиться повідомлення про те, що такий email вже існує.	Email <input type="text" value="test@mail.com"/> Такий користувач вже існує	1
2	Перевірка правильності коду для підтвердження email	Виводиться повідомлення про те, що введено неправильний код.	<input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/> Код підтвердження неправильний	1
3	Перевірка форми реєстрації на невідповідність паролів	Виводиться повідомлення про те, що введені паролі не збігаються.	Пароль <input type="password" value="....."/> Повтор пароля <input type="password" value="....."/> Паролі не співпадають	1

Продовження табл. 4.1

№	Назва	Очікуваний результат	Фактичний результат	0/1
4	Перевірка форми авторизації при спробі відправити пусту форму	Виводиться повідомлення про обов'язковість заповнення полів		1
5	Перевірка форми авторизації з неправильними даними	Виводиться повідомлення про невірно введені дані		1
6	Перевірка форми авторизації з правильними даними	Користувача перенаправляє на сторінку, яка відповідає його ролі (адміністратор або користувач)	Відбувається переадресація на відповідну ролі початкову сторінку	1
7	Перевірка функції фільтрації каталогу за обраними параметрами (категорії, ціна, розмір)	У каталозі відображаються товари, які відповідають заданим критеріям	На сторінці лишаються товари, що відповідають критерію фільтрації	1
8	Перевірка пошуку товарів за ключовим словом	Біля рядку пошуку стає доступним список товарів, назва яких містять введені ключове слово		1
9	Перевірка відображення детальної інформації про товар на сторінці товару	Відображається інформація про товар: назва, опис, ціна, доступні розміри та фото	Відображається вся необхідна інформація про товар	1
10	Перевірка функціоналу додавання товару в кошик	Товар додається до кошика, кількість товарів у кошику збільшується, а користувачу показується підтвердження		1
11	Перевірка оформлення індивідуального замовлення через кошик	Виводиться повідомлення про успішне оформлення замовлення		1

Продовження табл. 4.1

№	Назва	Очікуваний результат	Фактичний результат	0/1
12	Перевірка оформлення спільного замовлення через кошик	Користувач отримує підтвердження про додавання свого замовлення до спільного замовлення		1
13	Перевірка відображення статусу замовлення у кабінеті користувача	У кабінеті користувача відображається коректний статус замовлення		1
14	Перевірка функції додавання товару адміністратором через посилання на закордонний сайт	Інформація про товар заповнюється автоматично, з можливістю редагування адміністратором.	Поля для створення нового товару заповнюються автоматично, є можливість їх редагувати	1
15	Перевірка функції редагування товару адміністратором	Після збереження даних оновлена інформація коректно відображається у каталозі товарів. Адміністратору відображається повідомлення		1
16	Перевірка функції видалення товару адміністратором	Товар більше не відображається у каталозі. Адміністратору відображається повідомлення про успішне видалення товару		1
17	Перевірка функції зміни статусу замовлення адміністратором	Статус замовлення коректно оновлюється і відображається у кабінеті користувача та адміністратора. Адміністратору відображається повідомлення про успішне оновлення статусу		1
18	Перевірка функції зміни курсу валют	Усі товари, ціна яких вказана в обраній валюті, автоматично оновлюються відповідно до нового курсу.		1

Продовження табл. 4.1

№	Назва	Очікуваний результат	Фактичний результат	0/1
19	Перевірка відображення статистики для адміністратора	На сторінці статистики коректно відображаються дані у вигляді графіків і таблиць.	Статистика коректно відображається у графічному вигляді	1
20	Перевірка обмеження доступу до функцій адміністратора для звичайних користувачів	Сторінка не відкривається з відсутності відповідних прав	Сторінка не відкрилась	1

Після проведеного тестування було підтверджено стабільну роботу та відповідність її функціональності очікуваним результатам.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено веборієнтовану інформаційну систему підтримки діяльності баєра одягу. Реалізовано наступні функціональні можливості:

- форма реєстрації/авторизації користувачів у особистий кабінет;
- перегляд каталогу товарів та окремих карток товарів;
- пошук товарів за назвою та фільтрації за категоріями, брендами та іншими характеристиками;
- наявність форми для оформлення замовлень;
- можливість спільних покупок з друзями;
- адміністративна панель для швидкого управління замовленнями, контролю закупівельних процесів та додавання асортименту;
- перегляд блоку статистики по продажам для байєра.

Розроблений програмний продукт є актуальним для байєрів одягу, оскільки дозволяє автоматизувати ключові процеси його діяльності, зокрема управління асортиментом, обробку замовлень та аналіз статистики. Для клієнтів система забезпечує зручний доступ до каталогу товарів, можливість швидкого пошуку та фільтрації. А також зрозумілий механізм оформлення замовлень і функціонал для спільних покупок. Усе це робить процес вибору та придбання речей із-за кордону простішим і приємнішим.

У процесі виконання даної роботи було виявлено необхідність створення веборієнтованої інформаційної системи підтримки діяльності байєра одягу. Ця потреба стала результатом аналізу предметної області та вивчення аналогічних систем, які показали як свої сильні сторони, так і недоліки. Отримані висновки враховано при розробці концепції власної системи.

На першому етапі було визначено мету проєкту, сформульовано вимоги до системи, а також детально сплановано етапи її реалізації. Подальші дослідження охоплювали розробку варіантів використання системи, ідентифікацію ключових ролей,

визначення рівнів доступу та формалізацію основних задач за допомогою відповідних моделей і діаграм.

Далі була спроектована архітектура системи, розроблено базу даних, яка забезпечує надійне зберігання інформації, та реалізовано програмний функціонал відповідно до визначених вимог. Здійснено інтеграцію фронтенду та бекенду для забезпечення коректної взаємодії між ними.

Після реалізації системи проведено тестування для перевірки її працездатності та відповідності функціональним вимогам. У результаті запропонований програмний продукт показав високий рівень стабільності, ефективності та відповідність поставленим задачам.

Таким чином, використання даної веборієнтованої інформаційної системи підтримки діяльності байєра одягу дозволить скоротити час, необхідний на виконання рутинних операцій, забезпечить зручний механізм аналізу статистичних даних і спростить процес опрацювання замовлень. Для клієнтів представлений програмний продукт забезпечує простоту взаємодії. А також економію часу на пошук і замовлення товарів і робить взагалі сам процес покупок більш організованим і зручним.

Результати виконаної роботи пройшли апробацію на Всеукраїнській науково-практичній студентській конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку», яка відбулася на базі Харківського національного університету імені В. Н. Каразіна (Додаток Б).

Лістинг основних функцій системи представлений в Додатку В.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Стендер С. В., Лисак О. І., Лук'яненко Н. Е. Розвиток електронної комерції та її вплив на цифрову економіку. URL: <http://dx.doi.org/10.5281/zenodo.8420621> (дата звернення: 27.08.2024).
2. Шевчук О.І., Стежко Н.В. ТЕНДЕНЦІЇ РОЗВИТКУ СВІТОВОЇ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В УМОВАХ ДІДЖИТАЛІЗАЦІЇ БІЗНЕСУ. URL: <https://doi.org/10.32782/dees.5-3> (дата звернення: 30.08.2024).
3. Максютенко І. АНАЛІЗ ТЕНДЕНЦІЙ РОЗВИТКУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ. URL: <https://doi.org/10.32782/2524-0072/2024-64-54> (дата звернення: 30.08.2024).
4. Ярош, . В., Лагута, . В., Смоляков, . М., Тарасяк, . І., Футиляк , С., & Рутар, . Р. (2024). Вплив інноваційного дизайну веб-сайту на конверсію та користувацький досвід в е-торгівлі. Академічні візії, (27). URL: <https://academy-vision.org/index.php/av/article/view/1049> (дата звернення: 03.09.2024).
5. Пуцейло П., СУЧАСНІ ТЕХНОЛОГІЇ БІЗНЕС-АНАЛІТИКИ ЯК ІНСТРУМЕНТ ДЛЯ ПІДВИЩЕННЯ БІЗНЕС-КОМУНІКАЦІЙ КОМПАНІЇ. Інститут бухгалтерського обліку, контроль та аналіз в умовах глобалізації, [S.I.], п. 1-2, р. 29-40, june 2022. ISSN 2410-9274. URL: <http://ibo.wunu.edu.ua/index.php/ibo/article/view/543> (дата звернення: 07.09.2024).
6. Herschel R. Knowledge Management and Business Intelligence. 2021. URL: <http://www.b-eyenetwork.com/view/7621>. (date of access: 07.09.2024).
7. Свидрук, І. І., Сидорчук, О. Г., Коцупей, В. М., & Саврас, І. З. (2022). СОЦІОЛОГІЧНІ ДОСЛІДЖЕННЯ ВНУТРІШНЬОГО РИНКУ ТА ПРОЯВИВ ЛІДЕРСТВА В УПРАВЛІНСЬКІЙ ДІЯЛЬНОСТІ ПІД ЧАС ВОЄННОГО СТАНУ. Підприємництво і торгівля, (34), 56-63. URL: <https://doi.org/10.32782/2522-1256-2022-34-08> (дата звернення: 07.09.2024).

8. Сидоренко В.О., Антипенко В.П. Веборієнтована інформаційна система підтримки діяльності байера одягу. Програмні засоби для вирішення прикладних задач виробництва, освіти, бізнес-аналітики, інтелектуального оброблення даних, прийняття рішень: Збірник тез доповідей Всеукраїнської науково-практичної студентської конференції, 16 жовтня 2024 року, м. Харків, Україна. С. 186–187. URL: http://kbi.karazin.ua/wp-content/uploads/2024/11/proyekt_zbirnik-itmm.pdf (дата звернення: 16.10.2024)
9. Все про Instagram - повний посібник (SEO, факти, статистика) Ranktracker. URL: <https://www.ranktracker.com/uk/blog/all-about-instagram-the-ultimate-guide-seo-facts-stats/> (дата звернення: 20.09.2024).
10. Telegram Business: як нові функції покращують комунікацію з клієнтами. URL: <https://sendpulse.ua/blog/telegram-business> (дата звернення: 20.09.2024).
11. MY Shopping URL: <https://web.telegram.org/k/#-1963242060> (дата звернення: 24.09.2024).
12. Що таке мережа VPN URL: <https://experience.dropbox.com/uk-ua/resources/what-is-vpn> (дата звернення: 24.09.2024).
13. Eduard Store URL: <https://web.telegram.org/k/#@eduardstoree> (дата звернення: 24.09.2024).
14. Романа, ваш баєр vinogradova_shop URL: Store URL: https://www.instagram.com/vinogradova_shop/ (дата звернення: 26.09.2024).
15. Байєр одягу Італія URL: <https://brandshunter.com.ua/> (дата звернення: 26.09.2024).
16. Кунаєва Н. І. Методологія наукових досліджень: навчальний посібник. Київ: Ліра-К, 2018. 234 с. URL: <https://library.lira-k.ua/methodology-research> (дата звернення: 01.10.2024).
17. Попов Є. В., Сидоренко Ю. О. *Основи емпіричних досліджень: посібник*. Харків: Видавництво ХНУ, 2017. 192 с. URL: <https://repository.khnn.edu/emprirical-basics> (дата звернення: 01.10.2024).

18. Мартинюк І. В. *Наукові методи досліджень: теорія і практика*. Львів: Новий Світ, 2020. 210 с. URL: <https://nsvit.lviv.ua/research-methods> (дата звернення: 03.10.2024).
19. *Методологія ідеф0*. Підручники для студентів онлайн. URL: https://stud.com.ua/87184/ekonomika/metodologiya_idef0. (дата звернення: 06.10.2024).
20. Мова UML. Діаграма UML: <http://p4ilka.blogspot.com/2018/12/uml.html> (дата звернення: 08.10.2024).
21. PHP: Documentation. PHP: Hypertext Preprocessor. URL: <https://www.php.net/docs.php> (date of access: 10.10.2024).
22. Introduction to Laravel and MVC Framework - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/introduction-to-laravel-and-mvc-framework/> (date of access: 10.10.2024).
23. Laravel Team. Eloquent ORM. Laravel Documentation. Accessed November 12, 2024. URL: <https://laravel.com/docs/eloquent> (date of access: 10.10.2024).
24. Vue.js. Vue.js - The Progressive JavaScript Framework | Vue.js. URL: <https://vuejs.org/guide/introduction.html> (date of access: 17.10.2024).
25. PrimeTek Informatics. PrimeVue: The Ultimate UI Component Library for Vue 3. 2024. URL <https://www.primefaces.org/primevue/showcase> (date of access: 19.10.2024).
26. Why Learn PHP & MySQL?. Noble Desktop | Design & Coding Classes and Certificate Programs. URL: <https://www.nobledesktop.com/learn/php-mysql/why-learn-php-mysql> (date of access: 19.10.2024).
27. Docs. Redis - The Real-time Data Platform. URL: <https://redis.io/docs/latest/> (date of access: 20.10.2024).
28. MVC Framework - Introduction // Tutorialspoint URL: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (date of access: 20.11.2024).

29. Database Relationship Diagrams Design Tool. URL: <https://dbdiagram.io/home> (date of access: 23.11.2024).
30. PhpStorm. URL: <https://www.jetbrains.com/phpstorm/> (date of access: 25.11.2024).
31. RESTful Design. URL: <https://restfulapi.net/> (date of access: 25.11.2024).
32. Symfony DomCrawler. URL: https://symfony.com/doc/current/components/dom_crawler.html (date of access: 26.11.2024).
33. Puma. URL: <https://www.puma.com/> (date of access: 26.11.2024).
34. The DomCrawler Component. URL: https://symfony.com/doc/current/components/dom_crawler.html (date of access: 26.11.2024).
35. DeepL API (ChrisKonnertz\DeepLy). URL: <https://www.deepl.com/pro-api> (date of access: 28.11.2024).
36. JSON. URL: <https://www.json.org/> (date of access: 30.11.2024).
37. Black Box Testing. URL: <https://www.softwaretestinghelp.com/black-box-testing/> (date of access: 03.12.2024).
38. The SMART method to achieve your goals. Glowbl. URL: <https://www.glowbl.com/blog/en/the-smart-method-to-achieve-your-goals/> (date of access: 01.11.2024).
39. What is a work breakdown structure (WBS). Adobe Experience Cloud. URL: <https://business.adobe.com/blog/basics/what-is-work-breakdown-structure> (date of access: 01.11.2024).
40. Organizational Breakdown Structure Template. Wondershare EdrawMind. URL: <https://www.edrawmind.com/templates/organizational-breakdown-structure-template.html> (date of access: 04.11.2024).
41. Календарне планування проєктів. Бібліотека економіста. URL: <https://library.if.ua/book/96/6614.html> (дата звернення: 05.11.2024).

42. Project Risk: Definition and Advice for Project Managers. Teamwork.com. URL: <https://www.teamwork.com/project-management-guide/project-risk/> (date of access: 06.11.2024).

ДОДАТОК А

Планування робіт

Із початком повномасштабного вторгнення значна кількість закордонних брендів вийшли з ринку України. А ті, хто залишилися, значно підвищили ціни на свій асортимент. Тому, діяльність байєрів почала масово набирати популярність. У даній ситуації створення веборієнтованої інформаційної системи підтримки діяльності такого продавця одягу стає особливо важливим і актуальним сьогодні. Використання запропонованого інструменту представить можливість не тільки спростити виконання процесів пошуку та замовлення модних речей, а й дозволить байєрам ефективно керувати заявками, каталогами товарів і взаємодією з клієнтами в одному місці.

Деталізація мети проєкту методом SMART. Для успішності та конкурентоспроможності представленого проєкту треба на концептуальному етапі правильно визначити мету за допомогою SMART-методу [38]. Формат постановки його SMART-мети наступний: «Розробити веборієнтовану інформаційну систему підтримки діяльності байєра одягу до 10.12.2024 року для автоматизації процесу придбання відповідних товарів модних закордонних магазинів зі сторони клієнта та удосконалення механізму ведення бізнесу для продавця».

Таблиця А.1 – Деталізація мети проєкту методом SMART

Specific (конкретна)	Розробка веборієнтованої інформаційної системи підтримки діяльності байєра одягу з автоматизацією оформлення замовлень і з удосконаленим функціоналом для управління бізнес-процесами, включаючи облік заявок, оновлення каталогу товарів, а також забезпечення зручної взаємодії з клієнтами.
----------------------	--

Продовження табл. А.1

Measurable (вимірювана)	Результатом роботи стане веборієнтована інформаційна система підтримки діяльності байєра одягу, яка міститиме модулі для оформлення замовлень, керування товарними каталогами, відстеження виконання заявок і статистичний аналіз для оцінки клієнтських вподобань.
Achievable (досяжна, узгоджена)	Для розробки проєкту потрібні знання мови програмування PHP з фреймворком Laravel та Vue.js на JavaScript у поєднанні з навичками написання документації. Також є затверджене технічне завдання з сформованими вимогами до системи, враховані та розподілені наявні ресурси. Тому, ціль є досяжною.
Relevant (реалістична)	Застосування розробленої веборієнтованої інформаційної системи допоможе автоматизувати виконання ключових процесів у діяльності байєрів одягу, наприклад, таких, як обробка замовлень, ведення клієнтської бази, оновлення каталогу товарів, а також покращить ефективність ведення бізнесу. Це сприятиме збільшенню прибутків завдяки удосконаленій організації роботи зазначених продавців із клієнтами.
Time-framed (обмежена в часі)	Проєкт заплановано завершити до 10 грудня 2024 року.

Джерело: побудовано автором

Планування змісту робіт. WBS [39] (Work Breakdown Structure – ієрархічна структура робіт) – це графічний вигляд елементів проєкту, які згруповані ієрархією у єдине ціле з його продуктом. Структура декомпозиції робіт орієнтована на досконале їх виконання по частинам. WBS і сама є ключовою складовою проєкту. Вона спрямована на організацію командної роботи. Елементами декомпозиції можуть бути продукти, дані та послуги. Більше того, WBS забезпечує необхідний каркас для подальшої

ретельної оцінки термінів, контролю та графіків роботи при створенні календарного плану.

На найвищому (першому) рівні розміщений продукт проекту. Основні дії та заходи, які забезпечують досягнення його мети, зафіксовані на другому рівні декомпозиції. Остання виконується до тих пір, поки роботи не стануть елементарними задачами. Їх також називають простими.

Елементарні роботи – це дії, які мають однозначний чіткий результат. На кожну з них призначена одна відповідальна конкретна особа, для якої можна обчислити витрати праці й тривалість виконання. На рисунку А.1 представлено WBS проекту розробки веборієнтованої інформаційної системи підтримки діяльності байера одягу.

Планування структури виконавців. Наступним етапом після декомпозиції процесів є розробка організаційної структури виконавців або OBS [40]. Остання є аббревіатурою від Organizational Breakdown Structure. Вона визначається як графічна структура відображення відповідальних осіб. Вони всі є учасниками, які беруть участь у реалізації проекту.

У ролі відповідальних осіб виступають співробітники, які відповідають за організацію та виконання однієї елементарної роботи, яка зазначена у WBS. Кожну з них можна розглядати як окремий проект.

На рисунку А.2 представлено організаційну структуру реалізації даного проекту. Список функціонуючих виконавців описано в таблиці А.2.

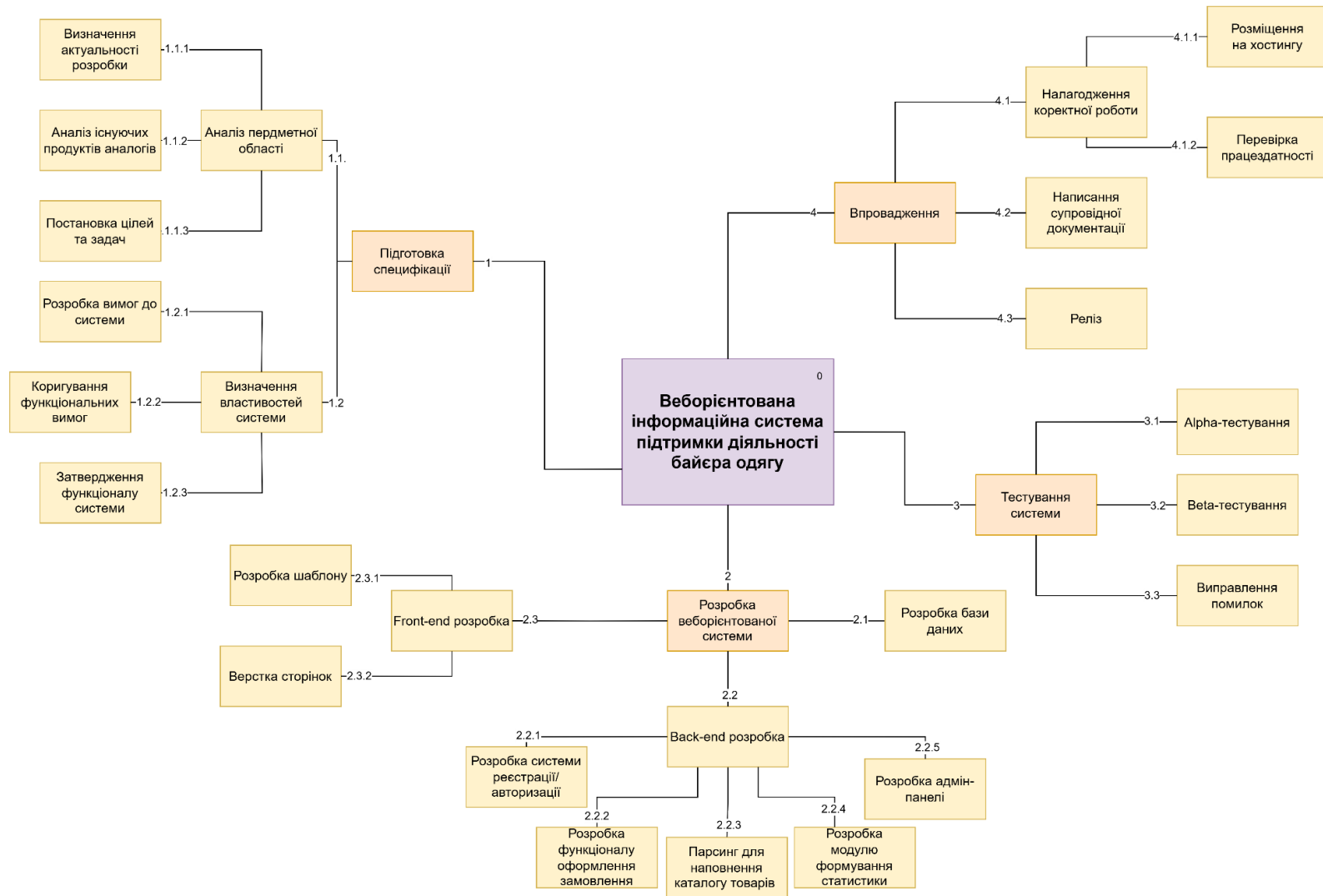


Рисунок А.1 – WBS-структура робіт проекту

Джерело: побудовано автором (знімок з екрану)

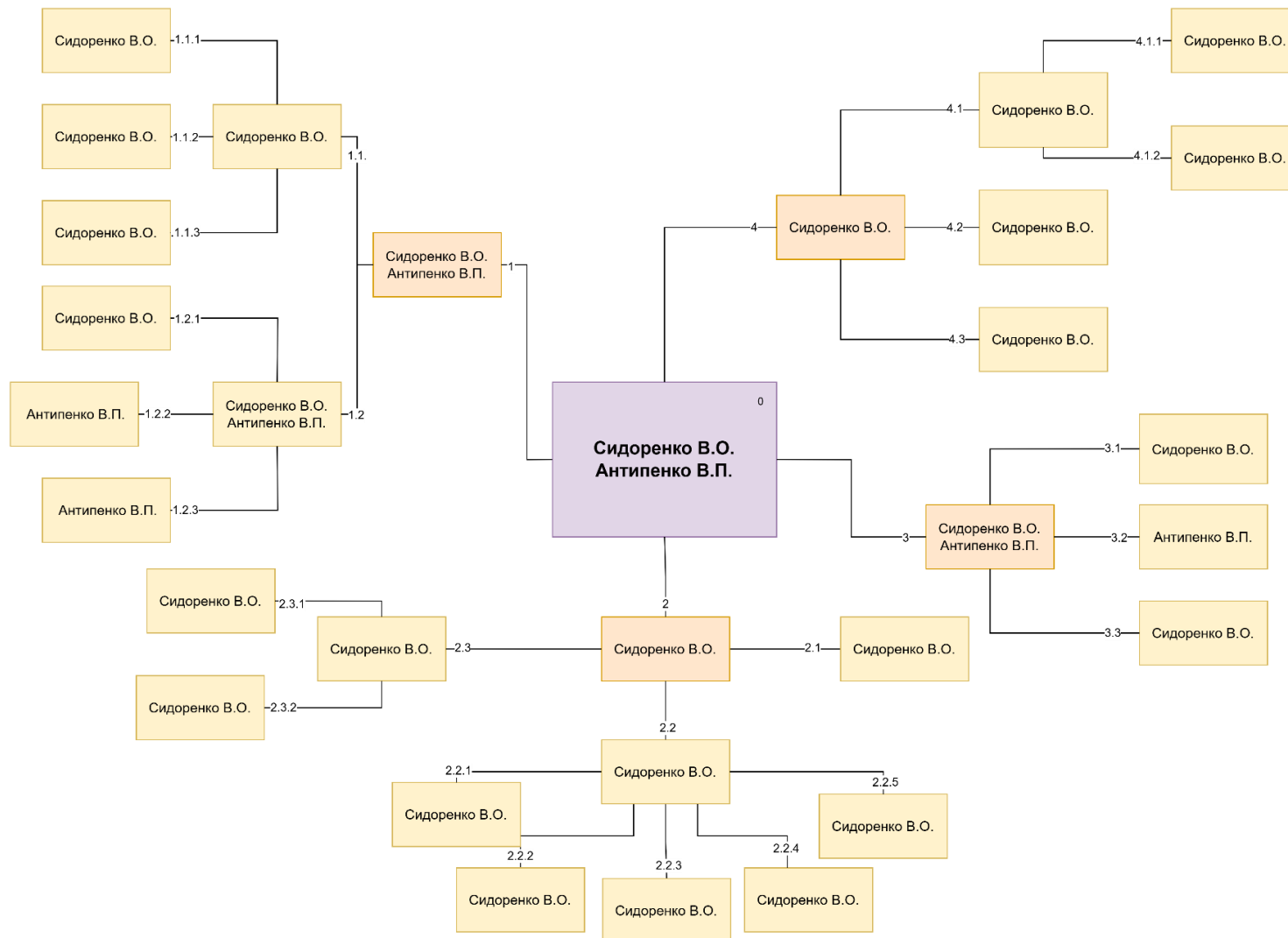


Рисунок А.2 – OBS-структура робіт проекту

Джерело: побудовано автором (знімок з екрану)

Таблиця А.2 – Виконавці проєкту

Роль	Ім'я	Проектна роль
Розробник	Сидоренко В.О.	Виконує front-end та back-end розробку
Проектувальник	Сидоренко В.О.	Відповідає за проєктування бази даних та структуру розроблюваної системи.
Тестувальник методом білого ящика	Сидоренко В.О.	Здійснює перевірку внутрішньої логіки та структури коду, детально аналізуючи його на наявність помилок.
Тестувальник методом чорного ящика	Антипенко В.П.	Тестує функціональності системи, оцінюючи відповідність результатів очікуваним.
Керівник проєкту	Антипенко В.П.	Формує завдання на розробку проєкту та затверджує документацію.
Менеджер проєкту	Сидоренко В.О.	Відповідає за розподіл завдань і ресурсів, а також за дотримання термінів та своєчасне виконання проєкту.

Джерело: побудовано автором

Діаграма Ганта. Побудова календарного графіку (діаграми Ганта) [41] є одним із важливих етапів планування проєкту, що виглядає як розклад виконання робіт з реальним розподілом дат. Завдяки йому можна отримати достовірне уявлення про тривалість процесів із обмеженнями у ресурсах, урахуванням вихідних днів і свят.

Календарний графік даного проєкту представлено на рисунку А.3.

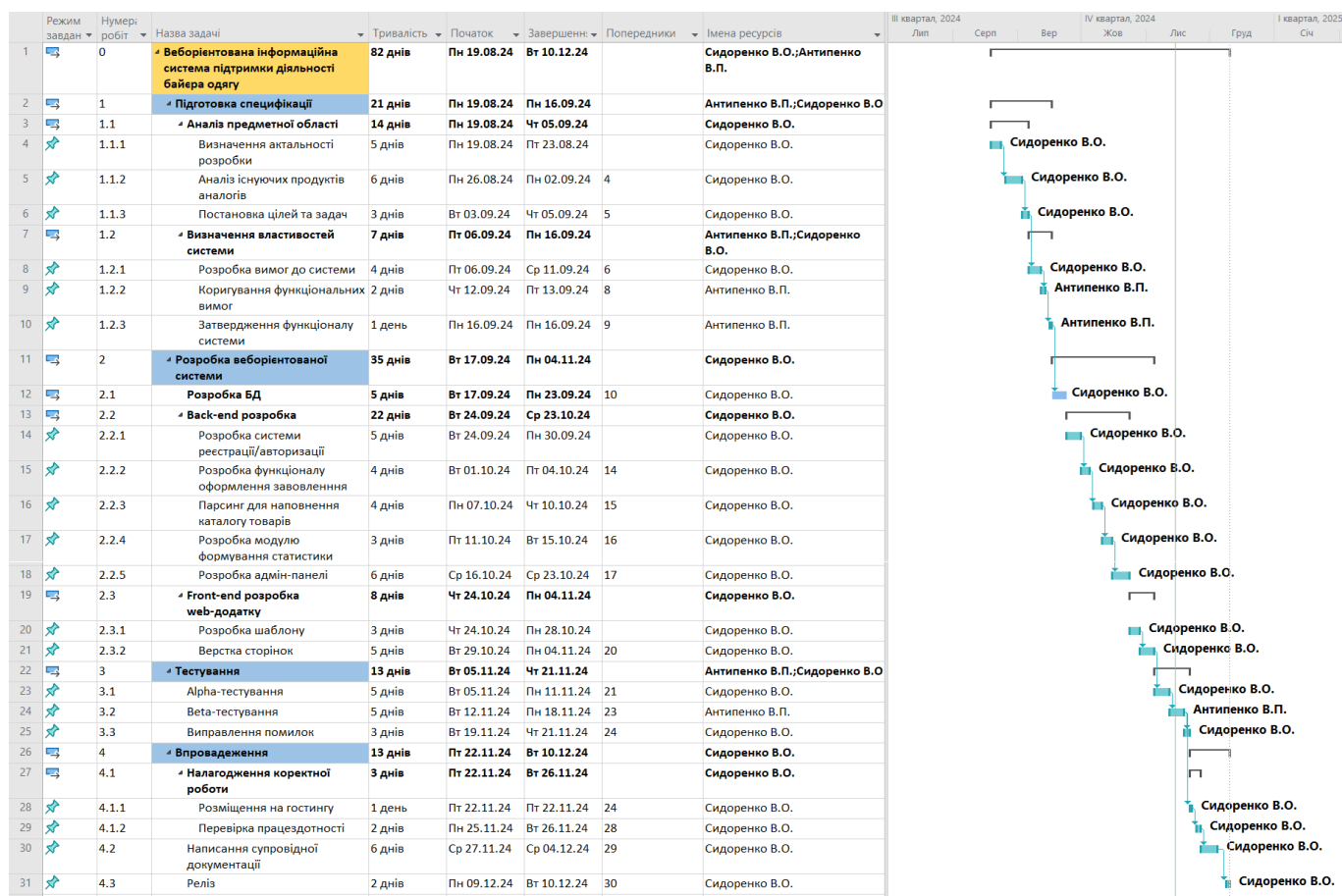


Рисунок А.3 – Діаграма Ганта

Джерело: побудовано автором (знімок з екрану)

Управління ризиками проєкту. При роботі над будь-яким проєктом завжди можуть бути якісь ризики. Проєктний ризик – це невизначена подія, яка може як відбутися так і не відбутися [42]. Сам ризик може впливати, як позитивно так і негативно на хід проєкту. Тому для того, щоб управляти ризиками необхідно заздалегіть ідентифікувати та оцінити їх. Це дозволить розробити стратегії, які допоможуть мінімізувати негативні наслідки та використати можливості для досягнення бажаних результатів. Таблиця А.3 включає в себе потенційні ризики при створенні майбутнього програмного продукту.

Таблиця А.3 – Ідентифікація ризиків проєкту

№	Назва ризику
1	Непорозуміння між розробником та керівником
2	Нечітке завдання на розробку
3	Низька кваліфікація розробників
4	Неоптимальний розподіл часу
5	Помилки проєктування
6	Збої в роботі програмного забезпечення
7	Відсутність резервних копій даних
8	Аварійні відключення світла
9	Загроза ракетних та артилерійських обстрілів
10	Невиконання моніторингу проєкту

Джерело: побудовано автором

Результати оцінки ризиків наведено у таблиці А.4.

Таблиця А.4 – Ймовірність, вплив та ранг ризиків проєкту

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Непорозуміння між розробником та керівником	0,1	0,4	0,04
2	Нечітке завдання на розробку	0,3	0,2	0,06
3	Низька кваліфікація розробників	0,3	0,4	0,12
4	Неоптимальний розподіл часу	0,5	0,2	0,1
5	Помилки проєктування	0,1	0,8	0,08

Продовження таблиці А.4

6	Збої в роботі програмного забезпечення	0,3	0,4	0,12
7	Відсутність резервних копій даних	0,1	0,8	0,08
8	Аварійні відключення світла	0,7	0,2	0,14
9	Загроза ракетних та артилерійських обстрілів	0,5	0,2	0,1
10	Невиконання моніторингу проєкту	0,1	0,2	0,02

Джерело: побудовано автором

Кожен ризик оцінюється по двом параметрам, а саме ймовірність та вплив ризику на проєкт. Таблиця А.5 представляє шкалу оцінювання ризиків за згаданими параметрами.

Таблиця А.5 – Шкала оцінювання ризиків

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Джерело: побудовано автором

У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та впливу ризику, що представлена в таблиці А.6

Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Таблиця А.6 – Матриця ймовірності та впливу

Ймовірність ризику (Й)	Вплив загрози (ризикау)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9	0,045	0,09	0,18	0,36	0,72
0,7	0,035	0,07	0,14 R8	0,28	0,56
0,5	0,025	0,05	0,10 R4, R9	0,20	0,40
0,3	0,015	0,03	0,06 R2	0,12 R3, R6	0,24
0,1	0,005	0,01	0,02 R10	0,04 R1	0,08 R5, R7

Джерело: побудовано автором (знімок з екрану)

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці А.7. У таблиці А.8 описано ризики та стратегії реагування на кожен з них.

Таблиця А.7 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики, які входять(назва)
1	Прийнятні	$0,005 \leq R \leq 0,05$	1, 10
2	Виправдані	$0,05 < R \leq 0,14$	2, 3, 4, 5, 6, 7, 8, 9
3	Недопустимі	$0,14 < R \leq 0,72$	-

Джерело: побудовано автором

Таблиця А.8 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_1	Новий	Непорозуміння між розробником та керівником	0,1	0,4	0,04	Зменшення	Налагодження гарних ділових стосунків між розробником та керівником. Створення комфортних умов для праці.	Оговорення причини непорозуміння та спокійне їх вирішення.
RS_2	Новий	Нечітке завдання на розробку	0,3	0,2	0,06	Зменшення	Ясно і однозначно обговорити із керівником усі види вимог. Періодичний контроль керівником етапів роботи.	Внесення правок у розробку.
RS_3	Новий	Низька кваліфікація розробників	0,3	0,4	0,12	Зменшення	Переглянути онлайн-ресурси для підвищення рівня знань.	Врахувати час на підготовку розробників. Видати літературу, переглянути онлайн-уроки.

Продовження табл. А.8

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_4	Новий	Неоптимальний розподіл часу	0,5	0,2	0,1	Зменшення	Звернути особливу увагу на правильність розподілу часу. Правильно визначити пріоритети виконання робіт. Чітко дотримуватися календарного плану.	Знайти способи оптимізації роботи з вже існуючою розстановкою. Обговорити варіанти внесення поправок до термінів реалізації із керівником.
RS_5	Новий	Помилки проектування	0,1	0,8	0,08	Зменшення	На етапі проектування тісно співпрацювати із керівником та на певних етапах демонструвати поточні результати.	Здійснювати проміжний контроль результатів в ході виконання проєкту.
RS_6	Новий	Збої в роботі програмного забезпечення	0,3	0,4	0,12	Зменшення	Підготувати резерв програмних засобів. Залучити спеціаліста для усунення збоїв.	Замінити програмне забезпечення.

Продовження табл. А.8

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_7	Новий	Відсутність резервних копій даних	0,1	0,8	0,08	Зменшення	Налаштувати автоматичне збереження даних. Зберігати дані на різних носіях інформації.	Робити копію даних після кожного виконаного етапу.
RS_8	Новий	Аварійні відключення світла	0,7	0,2	0,14	Зменшення	Передбачити у календарному плані неможливість роботи через відключення світла.	Оснащення розробників додатковими джерелами живлення.
RS_9	Новий	Загроза ракетних та артилер. обстрілів	0,5	0,2	0,1	Зменшення	Оснастити безпечне місце роботи.	Організувати дистанційний спосіб роботи.
RS_10	Новий	Невиконання моніторингу проекту	0,1	0,2	0,02	Зменшення	Здійснювати проміжний контроль результатів в ході виконання проекту.	Надання проміжних результатів виконання проекту після кожного етапу.

Джерело: побудовано автором

ДОДАТОК Б

Апробація результатів дослідження

На рисунках Б.1-Б.2 представлено тези, опубліковані за результатами даного дослідження.

УДК 004

Сидоренко В.О.
здобувачка вищої освіти,
Сумський державний університет
Науковий керівник
Антипенко В.П.
к.т.н., доцент, доцент кафедри інформаційних технологій,
Сумський державний університет

ВЕБОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ДІЯЛЬНОСТІ БАЙЄРА ОДЯГУ

Веборієнтована інформаційна система для байєрів одягу [1] – це інноваційний інструмент, який забезпечує ефективну організацію закупівлі вбрання й аксесуарів світових брендів та управління асортиментом для відповідних спеціалістів в індустрії моди. Такий програмний продукт насамперед створюється для спрощення здійснення пошуку, аналізу та придбання товарів із-за кордону. Також його використання забезпечить зручним каналом комунікації всіх учасників даного процесу.

Особливо актуальним даний інструмент є в умовах складної економічної ситуації в Україні. Сьогодні багато брендів залишили наш ринок або значно підвищили ціни на свою продукцію. Це змушує байєрів шукати альтернативні варіанти постачань відповідного одягу та аксесуарів із-за кордоном. Оскільки покупки у такий шлях стали вигіднішими. Завдяки інтеграції з іншими онлайн сервісами, наприклад, такими як закордонні інтернет-магазини, запропонована веборієнтована система для байєрів дозволить ефективно управляти ланцюгом придбання товарів поза межами України. Це дозволить відповідним продавцям адаптуватися до нових умов сучасного ринку.

Тому метою даного проекту є створення веборієнтованої платформи, застосування якої дозволить байєрам автоматизувати процес отримання запиту на здійснення замовлення в структурованому електронному вигляді, а також сприятиме удосконаленому управлінню товарними запасами та плануванню закупівель. У свою чергу клієнти отримуватимуть актуальну інформацію щодо цін, знижок та наявності товарів світових брендів в реальному часі. Це дозволить мінімізувати вірогідність виникнення помилок, пов'язані з людським фактором, та налагодити процес шопінгу з інших країн [2].

Представлена веборієнтована система передбачає автоматизацію таких процесів, як розміщення товару, опрацювання замовлень, сортування за постачальником, створення відповідних звітів. Також надається зручний канал комунікації з клієнтами. Її використання забезпечує можливість зберігання та аналізу даних про доступну продукцію, ціни та постачальників. Це допомагає байєрам приймати обґрунтовані рішення щодо подальшого кар'єрного розвитку [3].

186

Рисунок Б.1 – Перша сторінка тез

Джерело: побудовано автором (знімок з екрану)

Таким чином, розробка веборієнтованої інформаційної системи підтримки діяльності байера одягу є важливим кроком у напрямку удосконалення здійснення процесів закупівлі товарів із-за кордону. Її впровадження дозволить скоротити час клієнтів при здійсненні такого виду шопінгу, знизити витрати на доставку придбаної продукції та забезпечити конкурентну перевагу на ринку моди, що є особливо вагомим фактором у сучасних умовах високого економічного суперництва.

Список використаних джерел:

1. Shklar, L., Rosen, R. Архітектура веб-додатків: принципи, протоколи та практики. – 2-е вид. – Чічестер: Wiley, 2009. – 480 с.
2. Laudon, K.C., Traver, C.G. Електронна комерція: бізнес, технології, суспільство. – 15-е вид. – Бостон: Pearson, 2018. – 912 с.
3. 20 Top eCommerce Trends to Watch in 2024 URL: <https://blog.contactpigeon.com/ecommerce-trends-2024/#top10>

Рисунок Б.2 – Друга сторінка тез

Джерело: побудовано автором (знімок з екрану)

ДОДАТОК В

Лістинг програмного коду основних модулів веборієнтованої інформаційної системи

Створення, редагування, видалення, перегляд категорій в адміністративній панелі

CategoryController.php

```
<?php

namespace App\Http\Controllers\Api\Admin;

use App\Helpers\CategoryHelper;
use App\Http\Controllers\Controller;
use App\Http\Requests\StoreCategoryRequest;
use App\Http\Requests\UpdateCategoryRequest;
use App\Http\Resources>Main\CategoryResource;
use App\Models\Category;
use App\Services\ShopParser\Exceptions\UndefinedParserException;
use App\Services\ShopParser\ShopParserFactory;
use Illuminate\Http\Request;

class CategoryController extends Controller
{
    public function list()
    {
        $categories = Category::primary()->with('children')->get();

        return response()->json(CategoryHelper::buildTree($categories));
    }

    public function store(StoreCategoryRequest $request)
    {
        $category = Category::create($request->validated());

        return response()->json($category);
    }

    public function show(Category $category)
    {
        return response()->json(new CategoryResource($category));
    }

    public function update(UpdateCategoryRequest $request, Category $category)
    {
        $category->update($request->validated());

        return response()->json(new CategoryResource($category));
    }
}
```

```

}

public function destroy(Category $category)
{
    $category->delete();

    return response()->json(null, 204);
}
}

```

Створення, редагування, видалення атрибутів, отримання розмірів та кольорів

AttributeController.php

```

<?php

namespace App\Http\Controllers\Api\Admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\StoreAttributeRequest;
use App\Http\Requests\UpdateAttributeRequest;
use App\Http\Resources\AttributeResource;
use App\Models\Attribute;
use Illuminate\Http\JsonResponse;

class AttributeController extends Controller
{
    public function sizes(): JsonResponse
    {
        $sizes = Attribute::sizes()->get();

        return response()->json(AttributeResource::collection($sizes));
    }

    public function colors(): JsonResponse
    {
        $colors = Attribute::colors()->get();

        return response()->json(AttributeResource::collection($colors));
    }

    public function store(StoreAttributeRequest $request)
    {
        $attribute = Attribute::create($request->validated());

        return response()->json(new AttributeResource($attribute), 201);
    }

    public function update(UpdateAttributeRequest $request, Attribute $attribute)
    {
        $attribute->update($request->validated());

        return response()->json(new AttributeResource($attribute));
    }
}

```

```

public function destroy(Attribute $attribute)
{
    $attribute->delete();

    return response()->json(null, 204);
}
}

```

Керування курсами валют

CurrencyController.php

```

<?php

namespace App\Http\Controllers\Api\Admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\UpdateCurrenciesRequest;
use App\Models\Currency;

class CurrencyController extends Controller
{
    public function update(UpdateCurrenciesRequest $request){
        $currencyData = collect($request->validated());
        $ids = $currencyData->pluck('id')->toArray();
        $rates = $currencyData->pluck('rate', 'id')->toArray();

        $currencies = Currency::whereIn('id', $ids)->get();
        $currencies->each(function($currency) use ($rates){
            $currency->update([
                'rate' => $rates[$currency->id]
            ]);
        });

        return response()->json([
            'message' => 'Курс валют оновлено'
        ]);
    }
}

```

Перегляд замовлень та керування статусами

OrderController.php

```

<?php

namespace App\Http\Controllers\Api\Admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\ChangeOrderStatusRequest;
use App\Http\Resources\OrderResource;

```

```

use App\Models\Order;
use Illuminate\Support\Facades\DB;

class OrderController extends Controller
{
    public function index()
    {
        $orders = Order::with('products')->simplePaginate(25);

        return response()->json(OrderResource::collection($orders));
    }

    public function show(Order $order)
    {
        return response()->json(new OrderResource($order));
    }

    public function update(ChangeOrderStatusRequest $request, Order $order)
    {
        try {
            DB::beginTransaction();
            $order->update([
                'updated_at' => now()
            ]);
            $order->timeline()->create([
                'status' => $request->status,
                'comment' => $request->comment
            ]);
            DB::commit();
        } catch (\Throwable) {
            DB::rollBack();
            return response()->json([
                'message' => 'Помилка при оновленні статусу замовлення'
            ], 500);
        }

        return response()->json(new OrderResource($order));
    }

    public function destroy(Order $order)
    {
        $order->delete();

        return response()->json([
            'message' => 'Замовлення видалено'
        ]);
    }
}

```

Створення, редагування, видалення, перегляд товарів

ProductController.php

```

<?php

namespace App\Http\Controllers\Api\Admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\StoreProductRequest;
use App\Http\Requests\UpdateProductRequest;
use App\Http\Resources\ProductResource;
use App\Models\Product;
use App\Services\Helpers\ImageService;
use App\Services\ShopParser\Exceptions\UndefinedParserException;
use App\Services\ShopParser\ShopParserFactory;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Throwable;

class ProductController extends Controller
{
    public function index(Request $request)
    {
        $products = Product::query();

        if ($request->has('query')) {
            $products->where('name', 'like', '%' . $request->get('search') . '%');
        }

        return $products->paginate(25);
    }

    public function store(StoreProductRequest $request, ImageService $image-
Service)
    {
        try {
            DB::beginTransaction();
            $product = Product::create($request->validated());

            $product->categories()->attach($request->get('categories', []));
            $product->attributes()->attach($request->get('attributes', []));

            if ($request->has('images')) {
                $images = $imageService->saveImages($request->get('images'));
                $product->images()->createMany($images);
            }

            DB::commit();
        } catch (Throwable $e) {
            DB::rollBack();
            return response()->json(['message' => $e->getMessage()], 500);
        }

        return response()->json(new ProductResource($product), 201);
    }

    /**
     * Display the specified resource.

```

```

    */
public function show(Product $product)
{
    return response()->json(new ProductResource($product));
}

/**
 * Update the specified resource in storage.
 */
public function update(UpdateProductRequest $request, Product $product, ImageService $imageService)
{
    try {
        DB::beginTransaction();
        $product->update($request->validated());

        $product->categories()->sync($request->get('categories', []));
        $product->attributes()->sync($request->get('attributes', []));

        if ($request->has('images')) {
            $product->images()->delete();
            $images = $imageService->saveImages($request->get('images'));
            $product->images()->createMany($images);
        }

        DB::commit();
    } catch (Throwable $e) {
        DB::rollBack();
        return response()->json(['message' => $e->getMessage()], 500);
    }

    return response()->json(new ProductResource($product));
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(Product $product)
{
    $product->delete();

    return response()->json(['message' => 'Product deleted']);
}
}

```

Парсер товарів

ShopParserController.php

```

<?php

namespace App\Http\Controllers\Api\Admin;

use App\Http\Controllers\Controller;
use App\Http\Requests\ParseProductRequest;

```

```

use App\Http\Resources\ParsedProductResource;
use App\Services\Helpers\CurrencyService;
use App\Services\ShopParser\Exceptions\UndefinedParserException;
use App\Services\ShopParser\ShopParser;
use ChrisKonnertz\DeepLy\DeepLy;
use Illuminate\Http\JsonResponse;

class ShopParserController extends Controller
{
    /**
     * @throws UndefinedParserException
     */
    public function parse(
        ParseProductRequest $request,
        ShopParser $parser,
        DeepLy $deepLy,
        CurrencyService $currencyService
    ): JsonResponse {
        $url = $request->url;
        $product = $parser->parse($url);

        $product->convertedPrice = $currencyService->convertToUAH($product->price, $product->currency);

        $product = $deepLy->translateProduct($product);

        return response()->json(new ParsedProductResource($product));
    }
}

```

Вхід в акаунт

LoginController.php

```

<?php

namespace App\Http\Controllers\Api\Auth;

use App\Helpers\Auth\LoginTypeHelper;
use App\Helpers\Auth\UserHelper;
use App\Http\Controllers\Controller;
use App\Http\Requests\Auth\ConfirmRequest;
use App\Http\Requests\Auth>LoginByCodeRequest;
use App\Http\Requests\Auth>LoginByPasswordRequest;
use App\Models\ConfirmationCode;
use App\Models\User;
use App\Services\ConfirmationService;
use Illuminate\Database\Eloquent\ModelNotFoundException;
use Illuminate\Http\JsonResponse;
use Illuminate\Support\Facades\Log;
use Throwable;

class LoginController extends Controller

```

```

{
  public function login(LoginByPasswordRequest $request): JsonResponse
  {
    $password = $request->password;
    $credentials = array_merge(UserHelper::getLoginData($request->login),
compact('password'));

    $user = User::byLogin($request->login)->first();

    if ($user === null || $user->password === null || !auth()->attempt($cre-
dentials)) {
      return response()->json([
        'errors' => [
          'login' => ['Невірний логін або пароль'],
        ],
      ], 401);
    }

    $user = auth()->user();
    $token = $user->createToken('auth_token')->plainTextToken;

    return response()->json([
      'token' => $token,
      'user' => $user,
    ]);
  }

  public function confirm(ConfirmRequest $request): JsonResponse
  {
    try {
      // Знаходимо код підтвердження
      $confirmationCode = ConfirmationCode::byCodeAndLogin($request->code,
$request->login)
        ->signin()
        ->firstOrFail();
    } catch (ModelNotFoundException $e) {
      return response()->json(['message' => 'Код підтвердження недійсний'],
404);
    }

    try {
      // Пошук користувача за логіном
      $user = User::byLogin($confirmationCode->login)->firstOrFail();
    } catch (ModelNotFoundException $e) {
      return response()->json([
        'errors' => [
          'login' => ['Користувача з таким логіном не знайдено'],
        ],
      ], 404);
    }

    try {
      // Деактивуємо код
      $confirmationCode->update(['confirmed_at' => now()]);

      // Створюємо токен

```



```

        $token = $user->createToken('auth_token')->plainTextToken;
    } catch (\Exception $e) {
        Log::channel('errors')->error($e->getMessage() . $e->getTrace-
AsString());
        return response()->json(['message' => 'Виникла помилка під час обробки
запиту'], 500);
    }

    return response()->json(['token' => $token]);
}
}

```

Реєстрація, підтвердження пошти та створення пароля

RegisterController.php

```

<?php

namespace App\Http\Controllers\Api\Auth;

use App\Helpers\Auth\UserHelper;
use App\Http\Controllers\Controller;
use App\Http\Requests\Auth\ConfirmRequest;
use App\Http\Requests\Auth\RegisterRequest;
use App\Http\Requests\Auth\SetPasswordRequest;
use App\Models\ConfirmationCode;
use App\Models\User;
use App\Services\ConfirmationService;
use Exception;
use Illuminate\Database\Eloquent\ModelNotFoundException;
use Illuminate\Database\QueryException;
use Illuminate\Http\JsonResponse;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;
use Throwable;

class RegisterController extends Controller
{
    public function register(RegisterRequest $request, ConfirmationService
$service): JsonResponse
    {
        try {
            $service->sendConfirmationCode($request->login, Confirma-
tionCode::CODE_TYPE_SIGNUP);
        } catch (Throwable $e) {
            Log::channel('errors')->error($e->getMessage() . $e->getTrace-
AsString());
        }

        return response()->json(['message' => 'Код перевірки успішно
відправлено']);
    }

    public function confirm(ConfirmRequest $request): JsonResponse
    {

```

```

try {
    DB::beginTransaction();

    // Знаходимо код підтвердження
    $confirmationCode = ConfirmationCode::byCodeAndLogin($request->code,
$request->login)
        ->signup()
        ->firstOrFail();

    // Деактивуємо код
    $confirmationCode->update(['confirmed_at' => now()]);

    // Створюємо користувача
    $user = User::create(UserHelper::getUserConfirmationData($confirma-
tionCode));

    // Створюємо токен
    $token = $user->createToken('auth_token')->plainTextToken;

    DB::commit();
} catch (ModelNotFoundException $e) {
    // Якщо код підтвердження не знайдено
    DB::rollBack();
    return response()->json(['message' => 'Код підтвердження недійсний'],
404);
} catch (QueryException $e) {
    // Якщо виникає помилка дублювання запису, відкатуємо транзакцію
    DB::rollBack();
    return response()->json(['message' => 'Користувач з такою поштою або
номером телефону вже існує'], 409);
} catch (Exception $e) {
    // Якщо виникає помилка
    DB::rollBack();
    Log::channel('errors')->error($e->getMessage() . $e->getTrace-
AsString());
    return response()->json(['message' => 'Виникла помилка під час обробки
запиту'], 500);
}

return response()->json([
    'token' => 'test',
    'user' => [],
]);
}

public function password(SetPasswordRequest $request): JsonResponse
{
    $user = $request->user();
    try {
        $user->updatePassword($request->password);
    } catch (Exception) {
        return response()->json(['message' => 'Не вдалося встановити пароль,
спробуйте пізніше'], 500);
    }

    return response()->json(['message' => 'Пароль успішно встановлено']);
}

```

```

    }
}

```

Додавання, видалення та отримання товарів з кошика

CartController.php

```

<?php

namespace App\Http\Controllers\Api>Main;

use App\Http\Controllers\Controller;
use App\Http\Requests\StoreCartProductRequest;
use App\Models\CartProduct;
use Illuminate\Support\Facades\DB;

class CartController extends Controller
{
    public function index()
    {
        $cart = auth()->user()->cartProducts()->get();
        return response()->json($cart);
    }

    public function store(StoreCartProductRequest $request)
    {
        try {
            DB::beginTransaction();

            $cartProduct = auth()->user()->cartProducts()->where('product_id', $re-
quest->product_id)->first();

            if ($cartProduct) {
                $cartProduct->update([
                    'quantity' => $cartProduct->quantity + $request->quantity
                ]);
            } else {
                $cartProduct = auth()->user()->cartProducts()->create([
                    'product_id' => $request->product_id,
                    'quantity' => $request->quantity
                ]);

                $cartProduct->attributes()->attach($request->attributes);
            }
            DB::commit();
        } catch (\Throwable $e) {
            DB::rollBack();
            return response()->json([
                'message' => 'Помилка при додаванні товару до кошика'
            ], 500);
        }

        return response()->json([
            'message' => 'Товар додано до кошика'
        ]);
    }
}

```

```

public function destroy(CartProduct $product)
{
    $product->delete();

    return response()->json([
        'message' => 'Товар видалено з кошика'
    ]);
}
}

```

Отримання даних головного меню та товарів у категоріях

CategoryController.php

```

<?php

namespace App\Http\Controllers\Api>Main;

use App\Helpers\CategoryHelper;
use App\Http\Controllers\Controller;
use App\Http\Resources\CategoryListResource;
use App\Http\Resources>Main\CategoriesMenuPrimaryResource;
use App\Http\Resources>Main\CategoryResource;
use App\Models\Category;
use Illuminate\Http\Request;

class CategoryController extends Controller
{
    // Get all categories with subcategories
    public function index(Request $request)
    {
        $categories = Category::with(['children'])->primary()->active()->get();

        return response()->json(CategoriesMenuPrimaryResource::collection($categories));
    }

    public function list()
    {
        $categories = Category::primary()->get();

        return response()->json(CategoryListResource::collection($categories));
    }

    public function childList(string $slug)
    {
        $category = Category::with(['children'])->where('slug',$slug)->firstOrFail();

        // get only children and grandchildren of the category, but make it flat
        $categories = CategoryHelper::getChildren($category->children);

        return response()->json(CategoryListResource::collection($categories));
    }
}

```

```

    }

    public function show(string $slug)
    {
        $category = Category::with('parent')->where('slug', $slug)-
>findOrFail();
        return response()->json(new CategoryResource($category));
    }

    public function products(Category $category)
    {
        return response()->json($category->products);
    }
}

```

Сервіс для завантаження фото на диск, що спарсались з сайту

ImageService.php

```

<?php

namespace App\Services\Helpers;

use Illuminate\Support\Facades\Storage;

class ImageService
{
    public function saveImages(array $images): array
    {
        $paths = [];

        foreach ($images as $image) {
            $path = $this->saveImage($image);
            $paths[]['url'] = $path;
        }

        return $paths;
    }

    public function saveImage(string $image): string
    {
        $path = 'products/' . uniqid() . '.jpg';
        $content = file_get_contents($image);
        Storage::disk('products')->put($path, $content);

        return $path;
    }

    public function deleteImage(string $path): void
    {
        Storage::disk('products')->delete($path);
    }
}

```

Сервіс для конвертації ціни з інших валют у гривню

CurrencyService.php

```
<?php

namespace App\Services\Helpers;

use App\Models\Currency;
use Illuminate\Database\Eloquent\Collection;

class CurrencyService
{
    private Collection $currencies;

    public function __construct()
    {
        $this->currencies = Currency::all();
    }

    public function getCurrencyBySymbol(string $symbol): ?Currency
    {
        return $this->currencies->first(fn (Currency $currency) => $currency->symbol === $symbol);
    }

    public function convertToUAH(float $price, string $currencyCode): float
    {
        $currency = $this->getCurrencyBySymbol($currencyCode);

        if ($currency === null) {
            return $price;
        }

        return $price * $currency->rate;
    }
}
```

Парсер товару для сайту Puma

PumaShopParser.php

```
<?php

namespace App\Services\ShopParser;

use App\Services\ShopParser\Models\ParsedProduct;
use Symfony\Component\DomCrawler\Crawler;

class PumaShopParser extends ShopParser
{
    public function parse(string $url): ParsedProduct
    {

```

```

$this->crawler->addHtmlContent($this->getContent($url));
$name = $this->crawler->filter('#pdp-product-title')->text();
$subName = $this->crawler->filter('[data-test-id="pdp-product-sub-
header"]');

if ($subName->count() > 0) {
    $subName = $subName->text();
} else {
    $subName = '';
}

$price = $this->crawler->filter('[data-test-id="item-price-pdp"]')->
text();

$color = $this->crawler->filter('.tw-6nh9ql')->text();

$sizes = collect($this->crawler->filter('#size-picker label')->each(func-
tion (Crawler $node) {
    $size = $node->filter('span span[data-content="size-value"]')->text();
    $disabled = $node->attr('data-disabled');
    if ($disabled === 'true') {
        return null;
    }
    return $size;
}))
->filter()
->values()
->toArray();

$images = $this->crawler->filter('[data-test-id="product-image-gallery-
section"] div div img')->each(
    function (Crawler $node) {
        return $node->attr('src');
    }
);

$description = $this->crawler->filter('[data-test-id="pdp-product-de-
scription"]')->html();

return new ParsedProduct(
    $name . ' ' . $subName,
    $description,
    $this->getPrice($price),
    $this->getCurrency($price),
    $color,
    $sizes,
    $images,
    $url,
    $this->getSiteName($url),
);
}

private function getPrice(string $price): float
{
    return (float)preg_replace('/^[^0-9.]/', '', $price);
}

```

```
private function getCurrency(string $price): string
{
    return preg_replace('/[0-9.]/', '', $price);
}
}
```