

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна інженерія»

Група: 2БКС-29

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**здобувача освіти денної форми навчання
БКС.29.15.000.КРБ**

***МАТВЄЄВА МИХАЙЛА
ІГОРОВИЧА***

**м. Одеса
2025 р.**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна інженерія»

Група: 2БКС-29

ПОЯСНЮВАЛЬНА ЗАПИСКА

До кваліфікаційної роботи бакалавра на тему: _____

«Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень.»

Проектний матеріал складається з пояснювальної записки на 62 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах)

Виконавець _____  (Матвеев М.І.)

Керівник проекту _____ (Іванова Л.В.)


Консультанти:

з охорони праці _____  (Чорновол Н.І.)

з дотримання вимог ЄСКД _____  (Петрашова В.І.)

старший консультант _____  (Кривченко Ю.В.)

До захисту допущений

Завідувачка кафедри _____  (Іванова Л.В.)

Завідувач відділення _____ (Краснокутська К.Г.)

Захист «18» _____ 06 _____ 2025 р. Протокол ДКК № 3

Оцінка ЕК 5 (відмінно) / 90

Секретар ДКК _____ 

АНОТАЦІЯ

У роботі представлено результати розробки інтернет магазину з адміністративною панеллю та сторінкою для користувачів.

У цій роботі описаний принцип розробки сайту та телеграм боту на одній базі даних без використання фронтенд фреймворків.

У аналітичній частині приведені технічне завдання до проекту, а також фреймворку Laravel Livewire. В технологічній частині приведені інструменти для розробки сайту. У розділі представлена покласова реалізація сайту з використанням інструментів, таких як PhpStorm, Docker Compose.

Ключові слова: Php, Backend Framework, Laravel, Laravel Livewire, Windows, Telegram-бот, Python.

ANNOTATION

The paper presents the results of developing an online store with an administrative panel and a page for users.

This paper describes the principle of developing a site and telegram bot on one database without using front-end frameworks.

The technical task for the project, as well as the Laravel Livewire framework, is given in the analytical part. The technological part includes tools for website development. The section presents a class-by-class implementation of the site using tools such as PhpStorm, Docker Compose.

Keywords: Php, Backend Framework, Laravel, Laravel Livewire, Windows, Telegram bot, Python, aiogram.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Кафедра комп'ютерної інженерії
Освітньо-професійна програма «Комп'ютерна інженерія»
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Баркань І.В.
« 28 » 08 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

Здобувачеві (здобувачці) освіти Матвеев Михайло Ігорович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень

затверджена наказом по коледжу від « 14 » 11 2025 р. № 246

2. Термін здачі кваліфікаційної роботи 20.06.2025р.

3. Вихідні дані до роботи

1. Вимоги щодо структури сайту та чат-боту
2. Функціональні вимоги щодо сайту та чат-боту
3. Апаратні вимоги до використання сайту та чат-боту
4. Програмні вимоги до сайту та чат-боту


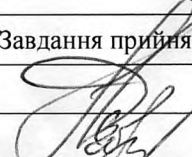
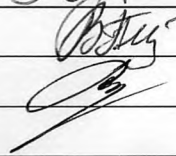
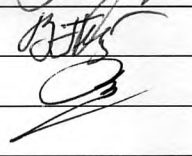
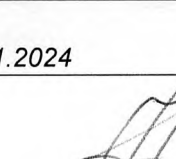
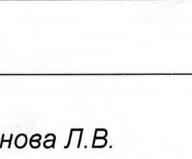
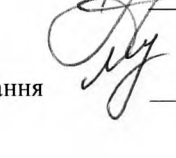
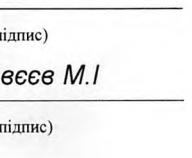
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Ознайомлення з технічним завданням для сайту та телеграмів бота. Установка бібліотек для фреймворку. Реалізація адміністративної панелі. Реалізація сторінки користувача. Створення телеграм бота. Висновок.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

1. Платформа проекту
2. Мова програмування
3. Мова програмування
4. Використання бібліотек
5. Будова сайту

6. Консультанти по кваліфікаційній роботі, із зазначенням розділів роботи, що стосується їх

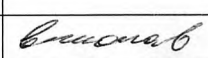
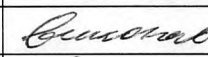
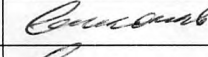
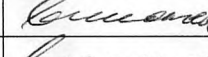
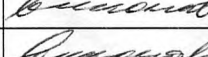
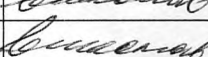
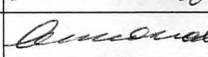
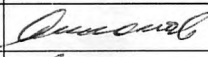
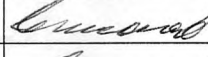
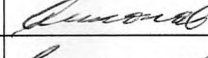
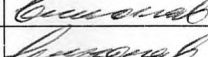
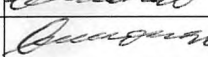
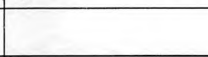

| Розділ | Консультант | ПІДПИС | |
|---------------------|----------------|--|---|
| | | Завдання видав | Завдання прийняв |
| Основний | Іванова Л.В. |  |  |
| Охорона праці | Чорновол В.І. |  |  |
| Нормоконтроль | Петрашова В.І. |  |  |
| Старший консультант | Кривченко Ю.В. |  |  |


7. Дата видачі завдання 30.11.2024

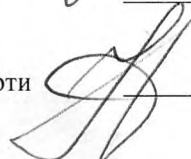
Керівник роботи  Іванова Л.В.
 (підпис)

Завдання прийняв до виконання Матвеев М.І
 (підпис)

КАЛЕНДАРНИЙ ПЛАН

| Пор. № | Назва етапів кваліфікаційної роботи | Термін виконання етапів роботи | Примітка |
|--------|--|--------------------------------|---|
| 1. | Передмова | 4.05.2025 |  |
| 2. | Технологічний розділ | 8.05.2025 |  |
| 3. | Визначення технічного завдання на роботу | 10.05.2025 |  |
| 4. | Розробка сайту на Laravel | 12.05.2025 |  |
| 5. | Створення міграцій | 13.05.2025 |  |
| 6. | Створення адміністративної панелі | 23.05.2025 |  |
| 7. | Створення сторінки користувача | 30.05.2025 |  |
| 8. | Розділ охорони праці | 5.06.2025 |  |
| 9. | Висновок | 14.06.2025 |  |
| 10. | Перелік літератури | 14.06.2025 |  |
| 11. | Оформлення пояснювальної записки | 14.06.2025 |  |
| 12. | Оформлення графічної частини | 14.06.2025 |  |
| 13. | Малий захист кваліфікаційної роботи | 15.06.2025 |  |
| 14. | Захист кваліфікаційної роботи | 20.06.2025 |  |

Виконавець  Матвеев М.І.
 (підпис)

Керівник роботи  Іванова Л.В.
 (підпис)

ЗМІСТ

| | С |
|--|----------|
| | ТОР |
| ПЕРЕДМОВА | 7 |
| РОЗДІЛ 1. Технологічний розділ | 8 |
| 1.1 Технічне завдання для сайту | 8 |
| 1.2 Технічне завдання на створення Telegram-бота з функціями авторизації та замовлення | 1 |
| | 1 |
| 1.3 Аналіз структури сайту на Laravel | 2 |
| | 1 |
| 1.4 Розробка сайту на Laravel | 5 |
| | 1 |
| 1.4.1 Установка Laravel | 5 |
| | 1 |
| 1.4.2. Налаштування проекту в .env файлі | 6 |
| | 1 |
| 1.4.3. Створення міграцій | 6 |
| | 1 |
| 1.4.4. Опис моделей | 8 |
| | 2 |
| 1.4.5. Встановлення додатку Spatie Permission | 0 |
| | 2 |
| 1.4.6. Встановлення доповнення Laravel Fortify | 0 |
| | 2 |
| 1.4.7. Встановлення Laravel UI | 1 |
| | 2 |
| 1.4.8. Сторінка реєстрації та авторизації | 1 |
| | 2 |
| 1.4.9. Створення сторінки адміністратора | 2 |

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 6 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

| | | |
|--|----------|----------|
| | 3 | |
| 1.4.10. Створення планувальника завдань Trello на основі Livewire | 5 | 2 |
| 1.4.11. Створення розділів користувачі, продукти та категорії | 0 | 3 |
| 1.4.12. Створення сторінки користувача та кошика | 6 | 3 |
| 1.5. Розробка телеграм-чат бота для інтернет-магазину | 6 | 4 |
| 2. ОХОРОНА ПРАЦІ | 4 | 5 |
| 2.1. Аналіз та безпека умов праці працівника на робочому місці | 5 | 5 |
| 2.1.2 Вимоги безпеки до мікроклімату, освітлення, шуму, виробничих випромінювань | 5 | 5 |
| 2.2. Пожежна безпека | 7 | 5 |
| ВИСНОВКИ | 9 | 5 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 1 | 6 |

ПЕРЕДМОВА

Використання інтернету є невід'ємною частиною нашого життя. Ми можемо в ньому проводити по годині або по пів дня використовуючи різні ресурси, такі як перегляд відео на youtube, фільмів або серіалів на reza або просто перегляд стрічки в instagram.

Зараз, щоб привернути увагу клієнтів, покупців чи партнерів, необхідно заявити про себе в інтернеті, шляхом створення WEB-сайту. Для цих цілей якраз і служить web-сайт, що містить основну інформацію про організацію,

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | БКС 29.15.000.00 ДП | <i>Арк.</i> |
| | | | | | | 7 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

приватну особу, компанію, товари або послуги, прайс-листи, контактні дані. Сайти дозволяють зберігати, передавати, продавати різні типи інформації, не відходячи від екрану комп'ютера. Люди отримують доступом до цієї інформації у вигляді використання технологій Internet. Для пошуку по інтернету використовують спеціальні програми - Web-браузери, які суттєво полегшують подорож безкрайними просторами інтернету.

Основна мета створення сайту полягає в тому, щоб зробити вашу роботу доступною широкому колу людей. Це може включати вашу наукову спільноту, потенційних роботодавців або зацікавлених користувачів. Сайт надає можливість детального представлення ваших досліджень, методів, результатів і висновків. Він може також включати інтерактивні елементи, такі як графіки, діаграми, відео або анімацію, які допоможуть краще зрозуміти вашу роботу.

У цій роботі було розглянуто створення сайту для комерційної компанії Smile. Завдяки сайту та боту клієнти можуть безпосередньо замовляти та не зв'язуватися з торговим представником. Сайт стає візитівкою компанії і це є кроком у напрямку розширення бізнесу та встановлення присутності в онлайн-середовищі, де потенційні клієнти можуть ознайомитися з її послугами, продуктами, історією, контактною інформацією та іншою важливою інформацією.

РОЗДІЛ 1. Технологічний розділ

1.1 Технічне завдання для сайту

Технічне завдання на створення сайту на PHP з використанням фреймворку Laravel.

- Введення

Технічне завдання описує вимоги до розробки сайту на PHP з використанням фреймворку Laravel. На сайті повинна бути передбачена адміністративна панель для адміністраторів, що включає функціонал створення

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 8 |

продуктів, категорій і сторінок з інформацією про користувачів і можливістю видалення користувачів.

Функціональні вимоги:

- Авторизація та аутентифікація

- Веб-сайт повинен передбачати механізм авторизації та аутентифікації адміністраторів.

- Адміністратори повинні мати можливість увійти в адміністративну панель, ввівши логін і пароль.

- Слід вжити заходів безпеки, таких як хешування паролів та запобігання атакам грубої сили.

- Планувальнику

- Головна сторінка повинна бути реалізована на прикладі Atlassian Trello

- Створення продуктів

- Можливість створення нових продуктів повинна бути реалізована в адміністративній панелі.

- Для кожного товару потрібно вказати наведені нижче атрибути.

- Назва продукту;

- Опис товару;

- Ціна товару;

- Категорія, до якої належить товар;

- Зображення товару (завантаження та зберігання зображень).

- Створення категорій

- Можливість створення нових категорій товарів повинна бути реалізована в адміністративній панелі.

- Для кожної категорії повинні бути передбачені:

- Назва категорії.

- Опис категорії (необов'язково).

Сторінка з усіма користувачами та видалення користувачів

- В адміністративній панелі повинна бути реалізована сторінка, на

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 9 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

якій відображаються всі користувачі системи та інформація:

- ID користувача;
- Логін (ім'я) користувача;
- Електронна пошта Користувача;
- Дата реєстрації користувача;
- Дії, включаючи можливість видалення користувача.

- Технічні вимоги

Фреймворк та мова програмування:

- Для розробки сайту повинен використовуватися фреймворк Laravel версії 8 або вище, сумісний з PHP 8.

- JavaScript або Laravel Livewire повинні використовуватися для розробки планувальника завдань і кошика.

- База даних

- Для зберігання даних має використовуватися реляційна база даних.

- Рекомендується використовувати MySQL, MySQLite або PostgreSQL як базу даних.

- Дизайн та інтерфейс користувача

- Веб-сайт повинен мати привабливий та інтуїтивно зрозумілий користувацький інтерфейс.

- Рекомендується використовувати сучасні методи розробки інтерфейсу користувача, такі як CSS фреймворки (наприклад, Bootstrap), щоб забезпечити адаптивний дизайн.

- Безпека

- Повинні бути впроваджені заходи безпеки, включаючи захист від атак CSRF.

- Паролі повинні зберігатися в зашифрованому вигляді.

Сторінка користувачів

- Переглянути список товарів і додати в кошик

- Користувачі, після успішної авторизації, повинні мати можливість

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| Змін. | Арк. | № докум. | Підпис | Дата | | 10 |

переглянути список доступних товарів.

- Веб-сайт повинен відображати інформацію про кожен товар, включаючи його назву, опис та ціну.

- Повинна бути реалізована функція додавання товарів в кошик користувача.

- Кошик для покупок повинен бути реалізований за допомогою сесій.

- Користувачі повинні мати можливість переглядати вміст свого кошика та видаляти товари з нього.

- На сайті повинна відображатися загальна сума покупки в кошику.

- Сеанси

- Для реалізації функціоналу кошика і авторизації користувача повинна використовуватися механіка сеансу.

- Сеанси повинні бути надійно керовані та повинні забезпечувати захист від підміни сеансів.

- Рекомендується використовувати вбудовані механізми сеансів, передбачені фреймворком Laravel, щоб забезпечити безпечне зберігання даних сеансу та управління сеансом користувача.

1.2 Технічне завдання на створення Telegram-бота з функціями авторизації та замовлення

- Введення

Технічне завдання описує вимоги до розробки Telegram-бота з функціями авторизації та оформлення замовлення. Бот повинен надати користувачам можливість реєстрації та авторизації, після чого користувач має можливість оформити замовлення, написавши вимоги до замовлення.

Функціональні вимоги:

- Реєстрація

- Бот повинен підтримувати механізм реєстрації користувачів за ім'ям, поштою, пароль та номер.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 11 |

- Авторизація

- Бот повинен підтримувати механізм авторизації користувачів за логіном і паролем зареєстрованого користувача.

- Переглянути доступні товари

- Бот повинен надавати можливість перегляду списку доступних товарів.
- Для кожного товару повинна бути надана інформація, включаючи його найменування, кількість і ціну.

- Оформлення замовлення

- Авторизовані користувачі повинні мати можливість оформити замовлення через бота.
- Користувачі повинні мати можливість переглядати товари зі списку доступних товарів або описувати, що їм потрібно.

Технічні вимоги:

- Мова програмування та фреймворк

- Для розробки Telegram-бота рекомендується використовувати мову програмування Python.
- Рекомендується використовувати фреймворк для розробки Telegram-ботів, такий як python-telegram-bot або aiogram.

- База даних

- Вам може знадобитися використовувати базу даних для зберігання даних про користувачів, товари та замовлення.
- Рекомендується вибрати відповідну реляційну базу даних, таку як MySQL або PostgreSQL.

- Безпека

- При розробці Telegram-бота необхідно забезпечити безпеку передачі даних між користувачем і ботом.
- Після введення даних користувача бот видає повідомлення про авторизаційні дані.

1.3 Аналіз структури сайту на Laravel

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 12 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Структура сайту Laravel [1] має певні умовності та рекомендації, які допомагають організувати код і спростити розробку. Про основні елементи структури сайту на Laravel:

1. Каталог ``app``: у цьому каталозі знаходяться основні файли програми, включаючи моделі (``app/Models``), контролери (``app/Http/Controllers``), сервіси (``app/Services``), провайдери (``app/Providers``) та інші класи, необхідні для роботи програми.

2. Каталог ``bootstrap``: тут містяться файли, що відповідають за запуск програми, включаючи завантаження автозавантажувача та встановлення необхідних налаштувань.

3. Каталог ``config``: у цьому каталозі розміщуються файли конфігурації програми, такі як база даних, авторизація, пошта та інші налаштування.

4. Каталог ``database``: тут знаходяться файли міграцій (``database/migrations``), фабрики даних (``database/factories``) та сидери (``database/seeders``), які допомагають створювати та заповнювати базу даних програми.

5. Каталог ``public``: у цьому каталозі знаходяться файли, доступні публічно, включаючи точку входу у вашу програму (``public/index.php``), статичні ресурси (зображення, CSS, JavaScript) та інші публічні файли.

6. Каталог ``resources``: тут містяться ресурси вашої програми, включаючи шаблони уявлень (``resources/views``), мовні файли (``resources/lang``) та файли стилів, скриптів та інших ресурсів (``resources/assets`` або ``resources/js`` та ``resources/css``, якщо використовується Laravel 8+ і Laravel Mix).

7. Каталог ``routes``: у цьому каталозі визначаються маршрути вашої програми. Файл `web.php` містить маршрути для веб-інтерфейсу, а файл `api.php` визначає маршрути для API.

8. Каталог ``storage``: тут розміщуються файли, згенеровані вашим додатком, такі як файли журналів, завантажені користувачем файли та кешовані уявлення. Він також містить підкаталоги для сеансів (``storage/framework/sessions``), кешу (``storage/framework/cache``) та інших

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 13 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

тимчасових файлів.

9. Каталог `tests`: у цьому каталозі можна розміщувати тести для вашої програми. Laravel поставляється із вбудованою підтримкою для тестування, і тут ви можете створювати модульні, функціональні та інтеграційні тести.

10. Каталог `vendor`: цей каталог містить залежності, встановлені за допомогою Composer, включаючи сам фреймворк Laravel.

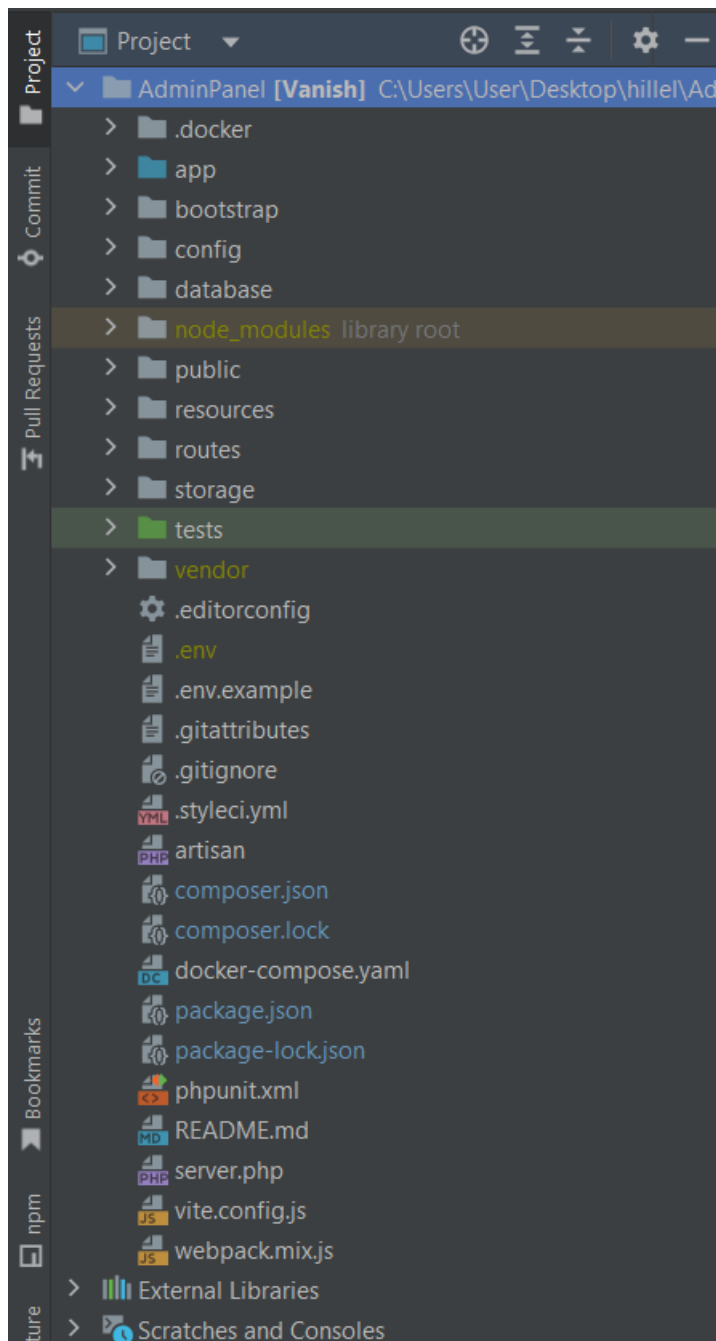


Рисунок 1.1. Вміст каталогу для розробки

Це основна структура каталогів Laravel. Деякі з цих каталогів, такі як `storage` та `vendor`, можуть бути прихованими в репозиторії, і їх вміст може

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 14 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

бути ігнорований системою контролю версій.

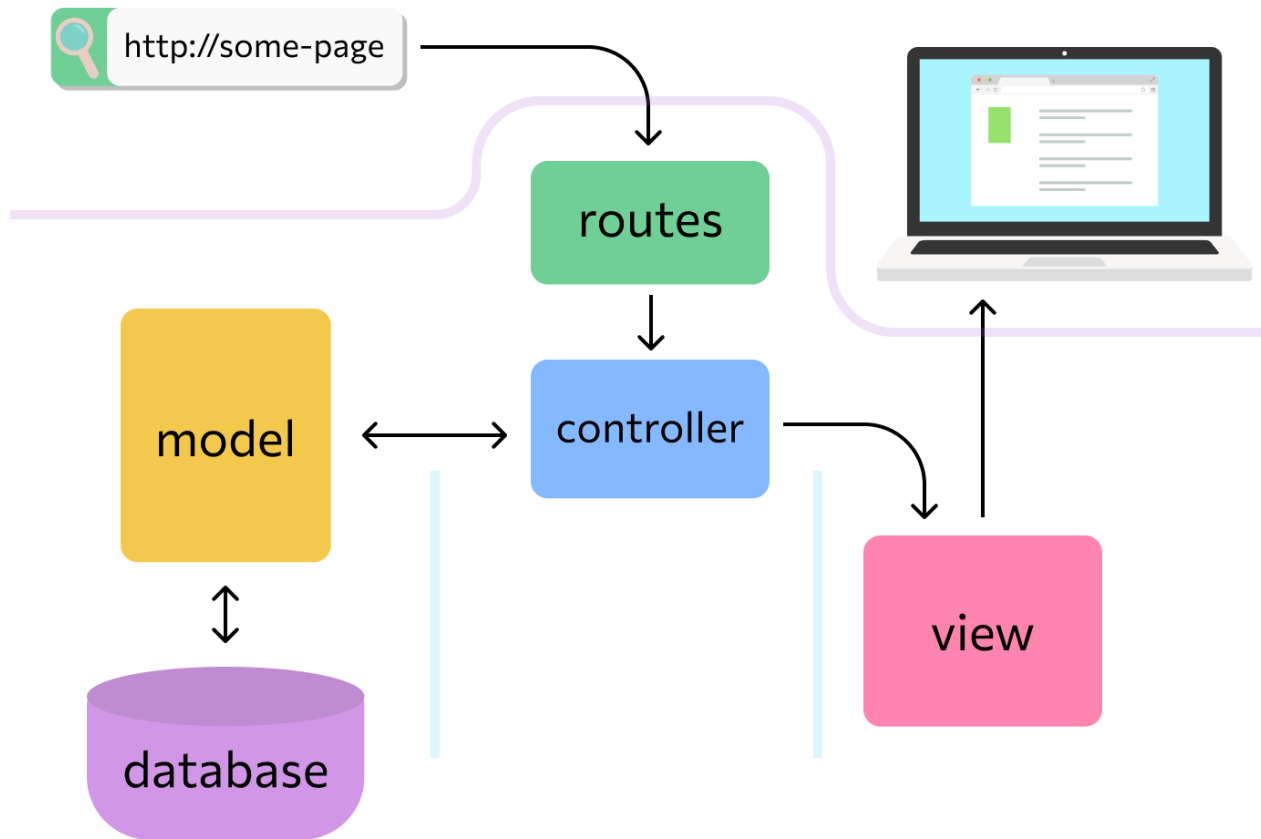


Рисунок 1.2. Схематичне відображення архітектури MVC

1.4 Розробка сайту на Laravel

1.4.1 Установка Laravel

Composer – це менеджер залежностей для PHP, який дозволяє керувати та завантажувати сторонні пакети та бібліотеки, необхідні для проекту.

У терміналі потрібно перейти до каталогу, де створюється новий проект Laravel. Використовуючи Composer потрібно запустити наступну команду: `composer create-project laravel/laravel AdminPanel` створюється папка з проектом[2].

Після створення проекту з'являється можливість запустити локальний сервер командою: `php artisan serve`

Після запуску сервера проект буде доступний за адресою `http://localhost:8000`

| | | | | | | |
|-------|------|----------|--------|------|---------------------|------|
| | | | | | БКС 29.15.000.00 ДП | Арк. |
| | | | | | | 15 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

1.4.2. Налаштування проекту в .env файлі.

APP_NAME="DiplomProject"

APP_ENV=local

APP_KEY=base64:xJti7423bX7cbW5F2O4ZQkXNW2AHad9snJHygPrwt28=

APP_DEBUG=true

APP_URL=http://127.0.0.1:8000

LOG_CHANNEL=stack

LOG_LEVEL=debug

DB_CONNECTION=sqlite

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=C:/Users/User/Desktop/hillel/AdminPanel/database/Diplom

DB.sqlite

DB_USERNAME=root

DB_PASSWORD=secret

1.4.3. Створення міграцій

У терміналі прописуємо команду[3]:

```
php artisan make:migration create_users_table--create=users
```

Прописується команда для продуктів, категорій та користувачів.

Заповнення таблиці users:

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->integer('phone')->unique();
    });
}
```

```

$table->string('role_id')->default('1');
$table->rememberToken();
$table->timestamps();
});
}

```

У цій міграції будуть описані ID користувача, ім'я, пошта, коли була верифікована пошта, пароль, номер телефону, номер ролі, зберігання токена сесії для підтримки функціональності «запам'ятати мене», час реєстрації.

Для пошти та номера телефону виставлено додатковий параметр unique(), щоб повторень у базі даних не було. Для email_verified_at виставлено nullable(), щоб стовпець не був заповнений до моменту підтвердження пошти. Стовпець role_id має початкове значення 1 щоб визначити кожного нового зареєстрованого користувача роль user.

Заповнення таблиці posts:

```

public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->string('img');
        $table->text('text');
        $table->integer('quantity');
        $table->float('price', 8, 2);
        $table->bigInteger('cat_id')->unsigned();
        $table->timestamps();
    });
}

```

У цій таблиці кожному товару буде надано id, назву, картинка, текст для опису товару, ціна, id категорії, час створення.

Для ціни ми додаємо параметри, щоб ціна не була більше 8 знаків і не

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 17 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

більше 2 знаків після коми.

Заповнення таблиці categories:

```
public function up()
{
    Schema::create('categories', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->timestamps();
    });
}
```

У цій таблиці описується id категорії, назва та час створення.

Командою `php artisan migrate` запускаються міграції, після чого до бази даних додаються нові таблиці, які були описані.

1.4.4. Опис моделей

Для моделі posts:

```
class Post extends Model
{
    use HasFactory;

    public function category()
    {
        return $this->belongsTo('App\Models\Category', 'cat_id');
    }

    protected $guarded = [];

    public function getUnitPriceAttribute($value)
    {
        return number_format($value, 2);
    }
}
```

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 18 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

```

}
    Для моделі User:
    class User extends Authenticatable
    {
    use HasApiTokens, HasFactory, Notifiable, HasRoles;

    * @var array<int, string>
    */
    protected $fillable = [
        'name',
        'email',
        'password',
        'phone',
    ];

    * @var array<int, string>
    */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    * @var array<string, string>
    */
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
    }

```

1.4.5. Встановлення додатку Spatie Permission

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 19 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Spatie для Laravel - це набір пакетів (бібліотек) від команди розробників Spatie, які надають додаткові функціональні можливості та інструменти для розробки програм на основі фреймворку Laravel[4].

Laravel Permission: пакет, що надає інструменти для керування дозволами та ролями користувачів у Laravel. Він дозволяє легко визначати дозволи, прив'язувати їх до користувачів та ролей, а також перевіряти доступ до певних дій або ресурсів у додатку.

Командою `composer require spatie/laravel-permission` відбувається встановлення доповнення до проекту[5].

У файл `config/app.php` додається провайдер:

```
'providers' => [  
    // ...  
    Spatie\Permission\PermissionServiceProvider::class,  
];
```

1.4.6. Встановлення доповнення Laravel Fortify

Laravel Fortify полегшить процес аутентифікації, реєстрації, скидання пароля та підтвердження електронної пошти у веб-додатках на основі фреймворку.

Команда установки: `composer require laravel/fortify`[6]

Додатково в терміналі прописується команда: `php artisan vendor:publish --provider="Laravel\Fortify\FortifyServiceProvider"`

Після всього заново запускається міграція даних командою `php artisan migrate`

1.4.7. Встановлення Laravel UI

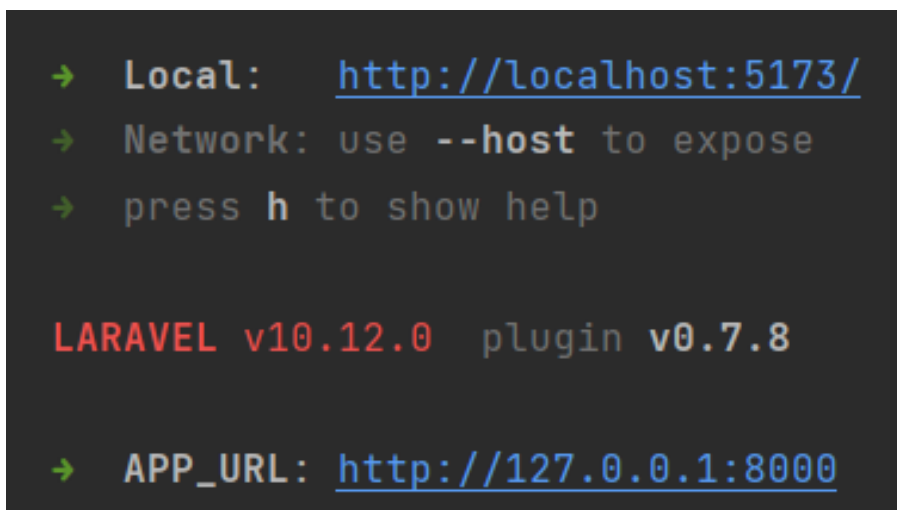
Для встановлення Laravel UI попередньо завантажується `node modules` командою: `npm install`.

Laravel UI дозволяє швидко створювати ефективні інтерфейси користувача в Laravel, використовуючи попередньо створені компоненти та інтеграцію з іншими інструментами Laravel.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 20 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Командою `php artisan ui bootstrap -auth` запускається установка UI з використанням `bootstrap`[7].

Командою `npm run dev` запускається сервер.



```
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h to show help

LARAVEL v10.12.0  plugin v0.7.8

→ APP_URL: http://127.0.0.1:8000
```

Рисунок 1.3. Команда `npm run dev` у консолі

1.4.8. Сторінка реєстрації та авторизації

До файлу `register.blade.php` додається розділ номер телефону та в моделі описується обов'язковим для заповнення[8].

У контролері `Auth/RegisterController` заповнюється функція:

```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
        'phone'=>['required','integer','min:8', 'max:13', 'unique:users']
    ]);
}
```

The image shows a registration form titled "Зареєструватись" (Register). It contains five input fields: "Ім'я" (Name), "Email Address", "Номер" (Number), "Пароль" (Password), and "Підтвердження пароля" (Password Confirmation). Below the fields is a blue button labeled "Зареєструватись" (Register).

Рисунок 1.4. Сторінка реєстрації

На сторінці реєстрації обов'язково мають бути заповнені всі поля. Пошта повинна мати знак @, обов'язково латиницею. Номер телефону обов'язково цифри, не менше 8 символів і не більше 13. Пошта та номер телефону обов'язково мають бути унікальними, тобто не співпадати з даними, які були введені раніше в базу. Пароль спочатку прихований, повинен мати не менше 8 символів, рекомендується використовувати символи та літери для більшої складності пароля. Після заповнення всіх полів використовується кнопка «Зареєструватися» для введення даних у базу.

Команда `php artisan permission:create-role user` створює роль користувача, так само додається роль адміністратора[9].

Кожен новий користувач, який зареєструвався на сайті, отримує роль «user».

```
protected          function          create(array          $data)
{
    $user=User::create([
        'name'          =>          $data['name'],
        'email'         =>          $data['email'],
        'phone'=>          $data['phone'],
        'password'      =>          Hash::make($data['password']),
    ]);
    $user->assignRole('user');
```

```

return $user;
}

```

Щоб змінити користувача на адміна, у базі даних таблиці users і Model_has_roles змінюється стовпець role_id на 2.

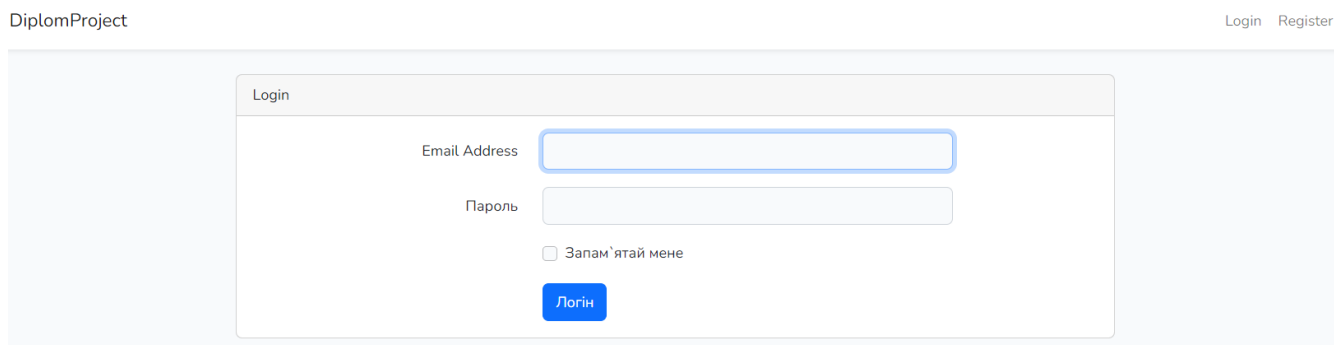


Рисунок 1.5. Сторінка авторизації

Сторінка авторизації має поля "Email Address" та пароль.

Функціонал "Запам'ятай мене" надає можливість користувачеві залишатися авторизованим навіть після закриття браузера та перезапуску сесії. Цей функціонал забезпечується за допомогою токенів "remember_token", що зберігаються у базі даних.

Коли користувач успішно авторизується з опцією "Запам'ятай мене" обраною, Laravel створює випадковий та унікальний токен "remember_token" для даного користувача. Потім цей токен зберігається у базі даних у полі "remember_token" для цього користувача.

При наступній спробі авторизації Laravel перевіряє наявність токена "remember_token" у базі даних. Якщо токен присутній і збігається з токеном, який було збережено раніше, Laravel автоматично авторизує користувача.

Таким чином, функціонал "Запам'ятай мене" забезпечує автоматичну авторизацію користувача при наступних відвідуваннях сайту, навіть після закінчення терміну дії сесії.

Після авторизації користувача переносить на сторінку відповідний role_id.

1.4.9. Створення сторінки адміністратора

На головну сторінку виводиться три блоки інформації для більш чіткого розуміння статистики для адміністратора. Перший блок складається із кількості

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 23 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Рисунок 1.6. Головна сторінка панелі адміністратора

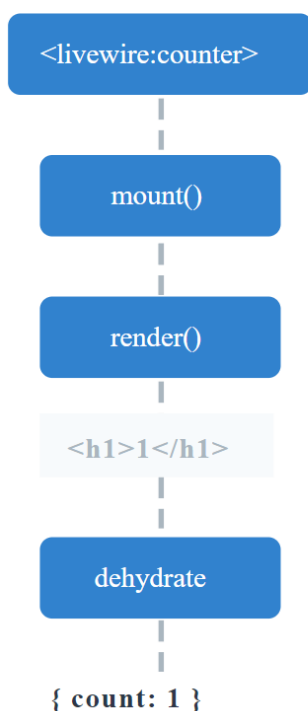
1.4.10. Створення планувальника завдань Trello на основі Livewire

Приклад Trello береться з сайту Atlassian Trello, який використовується великою кількістю програмістів.

Завдяки доповненню Livewire, який встановлюється командою `composer require, livewire/livewire` можна замінити JavaScript мовою PHP.

Командою `php artisan make: livewire Trello` додається контролер, view та модель[10].

Початковий запит



Подальший запит

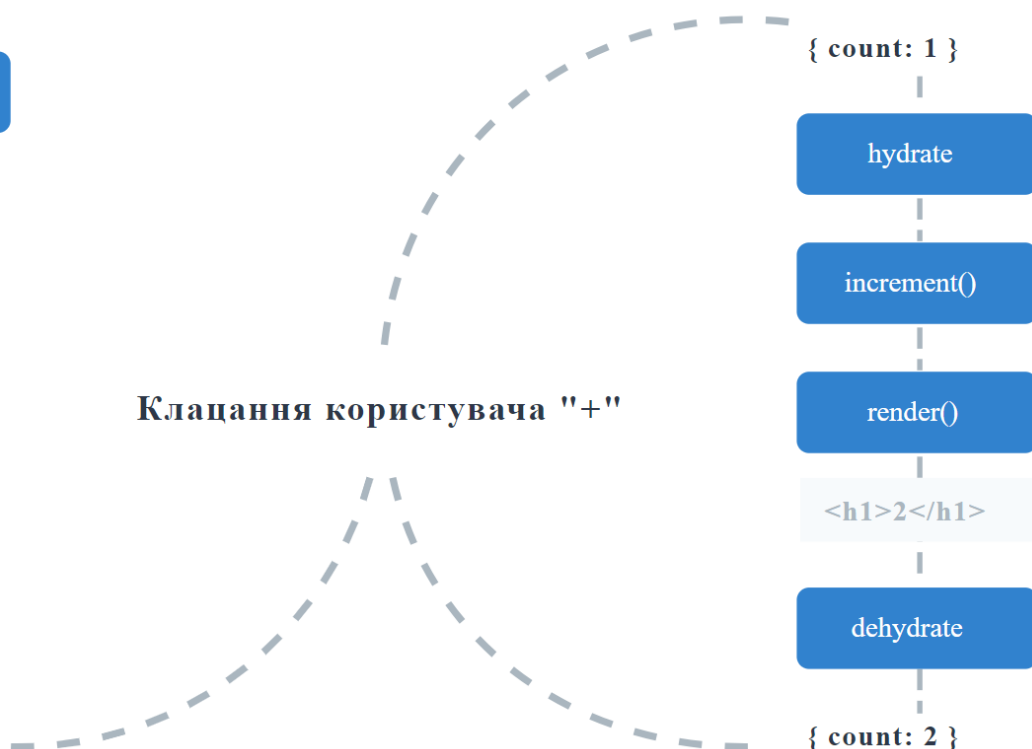


Рисунок 1.7. Схема обробки запиту користувача за допомогою Livewire

Для створення планувальника завдань додається дві додаткові міграції Cards і Groups.

Для створення таблиці Groups ід групи, текстове значення title, числове значення sort з стандартним параметром 999 для того, щоб кожна нова група була в самому кінці після її створення.

public

function

up()

| | | | | | | |
|-------|------|----------|--------|------|---------------------|------|
| | | | | | БКС 29.15.000.00 ДП | Арк. |
| | | | | | | 25 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

```

{
  Schema::create('groups', function (Blueprint $table) {
    $table->id();
    $table->string('title');
    $table->integer('sort')->default(999);
    $table->timestamps();
  });
}

```

Для створення таблиці Cards використано id картки, зовнішній ключ від group_id, текстове значення title, числове сортування зі значенням за замовчуванням 999 та часові позначки створення.

При перенесенні картки груп оновлюється group_id. При видаленні групи картка видаляється разом із нею.

```

public function up()
{
  Schema::create('cards', function (Blueprint $table) {
    $table->id();
    $table->foreignId('group_id')->constrained()
      ->onUpdate('cascade')
      ->onDelete('cascade');

    $table->string('title');
    $table->integer('sort')->default(999);
    $table->timestamps();
  });
}

public $title;
public $addGroup = false;
public $addCard = ";
protected $rules = [

```

'title'

=>'required'

];

При описі бізнес-логіки контролера Trello додаються функції:

Контролер Trello

Таблиця 1.1

| | |
|------------------|--|
| Створення групи | public function addGroup() { \$this->addGroup = true; } |
| Створення картки | public function addCard(\$group_id) { \$this->addCard = \$group_id; } |
| Видалення групи | public function deleteGroup(\$id) { Group::destroy(\$id); } |
| Видалення картки | public function deleteCard(\$id) { Card::destroy(\$id); } |

Продовження таблиці 1.1

| | |
|------------------|--|
| Збереження даних | public function save() { \$data = \$this->validate(); } |
|------------------|--|

```

if($this->addGroup){
    Group::create($data);
}
else {
    $data['group_id']=$this->addCard;

    Card::create($data);
}
$this->reset();
}

```

| | |
|--|--|
| <p>Сортуван ня груп і карток</p> | <pre> public function sorting(\$order) { foreach (\$order as \$group){ Group::where(['id'=>\$group['value']]- >update(['sort'=>\$group['order']]); if(isset(\$group['items'])){ foreach (\$group['items'] as \$item) { Card::where(["id"=>\$item["value"]]- >update(["sort"=>\$item["order"],"group_id"=>\$group["value"]]); } } } } </pre> |
|--|--|

Продовження таблиці 1.1

| | | | |
|----------|--------|----------|----------|
| Відображ | public | function | render() |
|----------|--------|----------|----------|

| | |
|---|---|
| ення груп і карток на сторінці LiveWire | { \$groups=Group::orderBy('sort')->get(); return view('livewire.trello', ['groups'=>\$groups,]); } |
|---|---|

Після опису дій кожної функції треба додати дії у views/livewire/trello опису до кнопок:

```

@if($addGroup)
<form wire:submit.prevent="save">
  <label>
    <input wire:model.defer="title" type="text">
  </label>
</form>
@else
<button wire:click="addGroup">
  Додати групи
</button>
@endif

```

При заповненні blade файлу використовується bootstrap для стилізації компонентів.

Визначення дій у blade файлі

Таблиця 1.2

| | |
|-------------------|--|
| Сортування груп | wire:sortable-group.item-group="{{ \$group->id }}" |
| Сортування карток | wire:key="card-{{ \$card->id }}" wire:sortable-group.item="{{ \$card->id }}" |

Продовження таблиці 1.2

| | |
|-------------------|---|
| Видалення карток | wire:click="deleteCard({ {\$card->id} })" |
| Збереження | wire:submit.prevent="save" |
| Визначення моделі | wire:model.defer="title" type="text" |
| Створення картки | wire:click="addCard({ {\$group->id} })" |

У папці `views/layouts` знаходиться `admin_layout.blade.php`, в якому для коректної роботи Livewire прописується в тезі `<head>` `@livewireStyles` і після всіх скриптів JavaScript в тезі `<body>` `@livewireScripts`. Після цього достатньо послатися на цей файл і доповнення Livewire буде працювати правильно.

1.4.11. Створення розділів користувачі, продукти та категорії

Для кожного розділу створюється свій контролер та view файл.

Для розділу "Користувачі" створено view файл у папці `views/admin/user`.

На сторінці відображається ID, ім'я, Email, дата додавання та номер телефону кожного користувача.

У контролері описується відображення та видалення користувача з бази даних.

```
$users=User::latest()->get();
```

```
return view('admin.user.index', [
    'users' => $users
]);

public function destroy(User $user)
{
    $user->delete();

    return redirect()->back()->withSuccess('Користувач був вилучений'); // Після
    видалення користувача з'явиться значок про те, що користувач видалений.
}
```

Для ролі користувача прописується логіка видалення на сторінці. Якщо `role_id=1`, то з'являється кнопка навпроти користувача видалити.

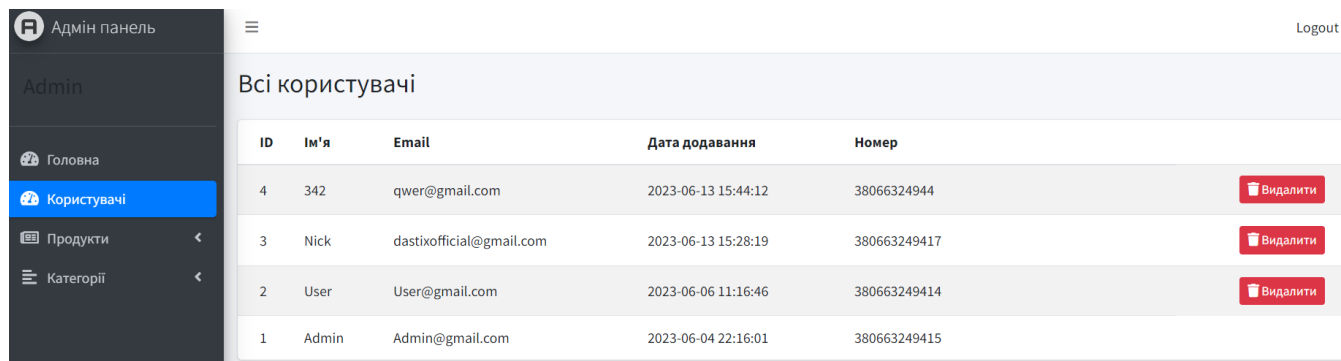


Рисунок 1.8. Сторінка з усіма користувачами

Для розділу "Продукти" створюється два view файли, в яких буде описано додавання та перегляд усіх товарів.

При створенні товару мають бути такі параметри для введення:

- заголовок;
- вибір категорії;
- ціна;
- кількість;
- зображення;
- опис.

Для продукту описуються функції у контролері PostController:

Визначення функцій у контролері PostController

Таблиця 1.3

| | |
|----------------------------|--|
| Відображен ня продуктів | <pre> public function index() { \$posts = Post::orderBy('created_at', 'DESC')->get(); return view('admin.post.index', ['posts' => \$posts]); } </pre> |
|----------------------------|--|

Продовження таблиці 1.3

| | |
|---|---|
| Відображен ня категорій у товарах | <pre> public function create() { \$categories = Category::orderBy('created_at', 'DESC')- >get(); return view('admin.post.create', ['categories' => \$categories]); } </pre> |
| Створення товару | <pre> public function store(Request \$request) { \$post = new Post(); \$post->title = \$request->title; \$post->img = \$request->img; \$post->quantity = \$request->quantity; \$post->price = \$request->price; \$post->text = \$request->text; \$post->cat_id = \$request->cat_id; \$post->save(); return redirect()->back()->withSuccess('Товар було успішно додано!'); } </pre> |
| Редагування товару | <pre> public function edit(Post \$post) { \$categories = Category::orderBy('created_at', 'DESC')- >get(); return view('admin.post.edit', ['categories' => \$categories, 'post' => \$post,]); } </pre> |

Продовження таблиці 1.3

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 32 |

| | |
|-----------------------------|---|
| <p>Оновлення товару</p> | <pre> public function update(Request \$request, Post \$post) { \$post->title = \$request->title; \$post->img = \$request->img; \$post->quantity = \$request->quantity; \$post->price = \$request->price; \$post->text = \$request->text; \$post->cat_id = \$request->cat_id; \$post->save(); return redirect()->back()->withSuccess('Товар було успішно оновлено!'); } </pre> |
| <p>Видалення товару</p> | <pre> public function destroy(Post \$post) { \$post->delete(); return redirect()->back()->withSuccess('Товар було успішно видалено!'); } </pre> |

При перегляді всіх товарів адміністратором показується таблиця, в якій кожен товар має наступні параметри:

- ID;
- Зображення;
- Назва;
- Категорії;
- Ціну;
- Кількість;
- Дату створення.

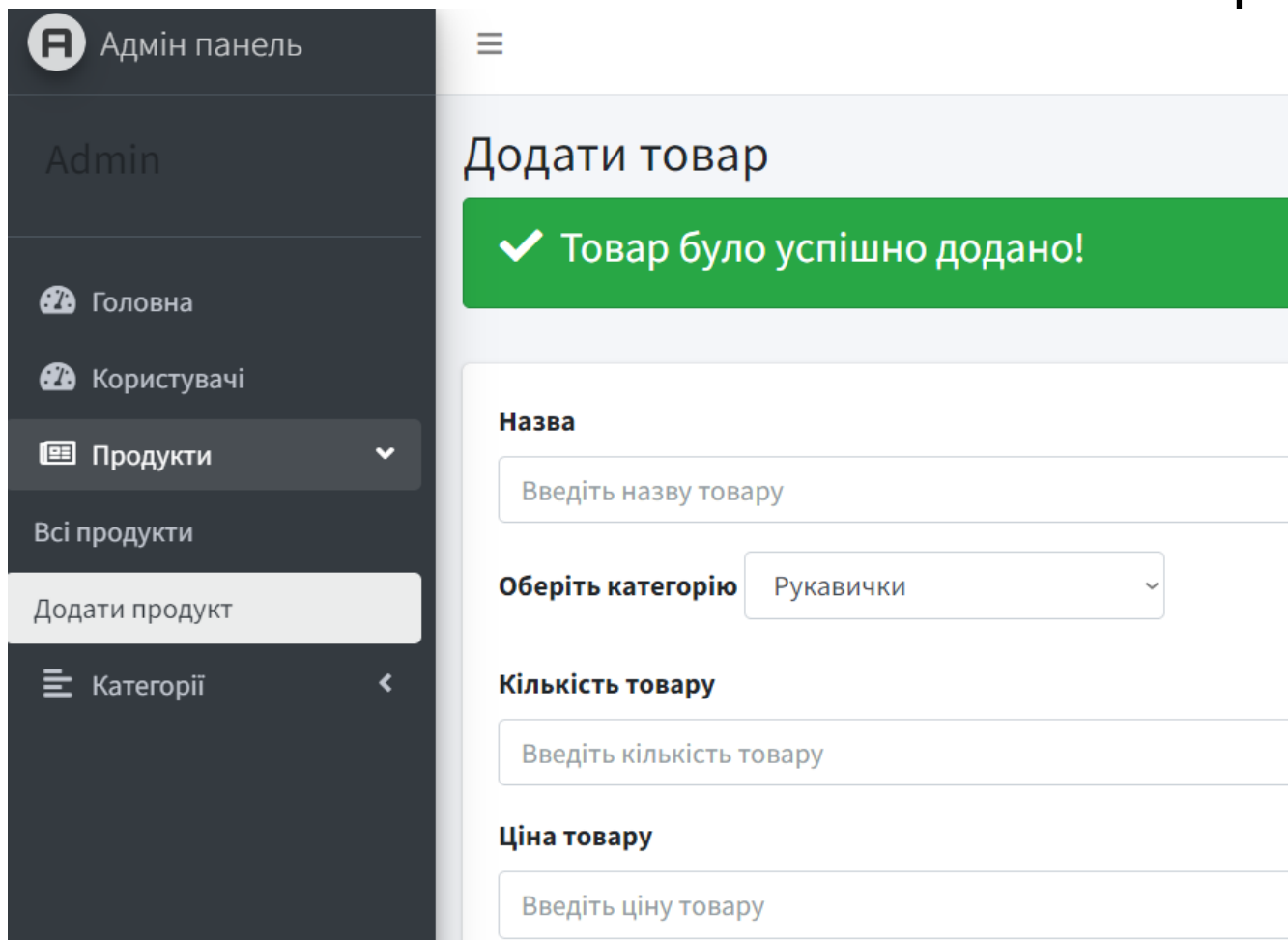


Рисунок 1.9. Підтвердження додавання товару

Кожен продукт має кнопки «редагувати» та «видалити».

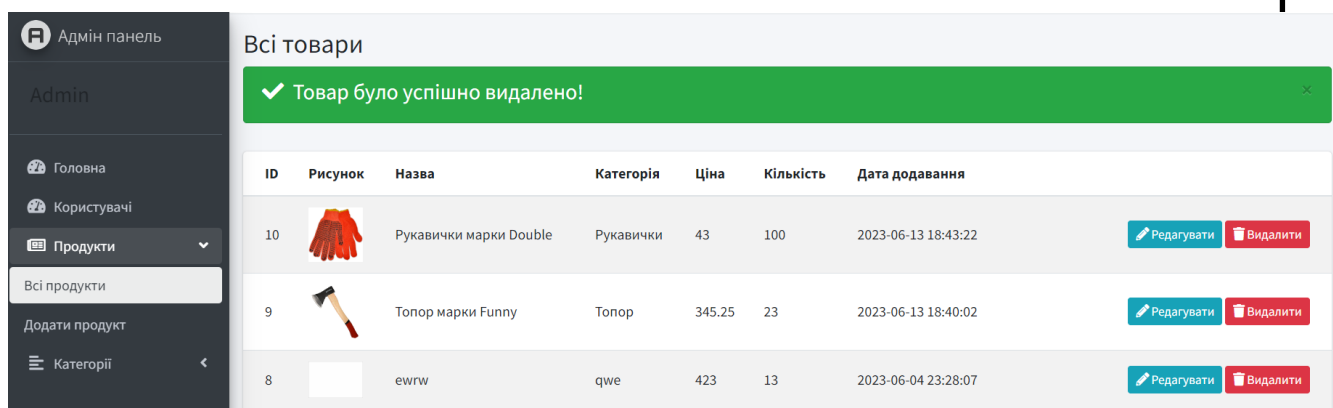


Рисунок 1.10. Відображення всіх продуктів

Для розділу «Категорії» створюється view файл із додаванням та відображенням усіх категорій.

Під час створення категорії повинні бути такі параметри для введення:

– Заголовок.

Для категорій описується функції в CategoryController:

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | БКС 29.15.000.00 ДП | Арк. |
| | | | | | | 34 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

Визначення функцій у контролері CategoryController

Таблиця 1.4

| | |
|-------------------------------|--|
| <p>Відображення категорій</p> | <pre> public function index() { \$categories = Category::orderBy('created_at', 'desc')->get(); return view('admin.category.index', ['categories' => \$categories]); } </pre> |
| <p>Заповнення даними</p> | <pre> public function store(Request \$request) { \$new_category = new Category(); \$new_category->title = \$request->title; \$new_category->save(); return redirect()->back()->withSuccess('Категорія була успішно додана!'); } </pre> |

Продовження таблиці 1.4

| | |
|------------------------------|--|
| <p>Редагування категорій</p> | <pre> public function update(Request \$request, Category \$category) { \$category->title = \$request->title; \$category->save(); } </pre> |
|------------------------------|--|

| | |
|---------------------|--|
| | <pre> return redirect()->back()->withSuccess('Категорія була успішно оновлена!'); } </pre> |
| Видалення категорій | <pre> public function destroy(Category \$category) { \$category->delete(); return redirect()->back()->withSuccess('Категорія була успішно вилучена!'); } </pre> |

1.4.12. Створення сторінки користувача та кошика

Для створення кошика використовуються сесії. З Livewire створюються компоненти CartComponent, ProductComponent[11].

У CartComponent описуються початкові дані:

```

protected $total; // Визначення загальної суми кошика
protected $content; //Вміст кошика
protected $listeners = [
'productAddedToCart' => 'updateCart',
]; // слухачі, коли додається до кошика продукт – кошик оновлюється

```

Опис функцій CartComponent

Таблиця 1.5

| | |
|--------------------------|--|
| Оновлення ціни | <pre> public function mount(): void { \$this->updateCart(); } </pre> |
| Повернення View з даними | <pre> public function render(): View { return view('livewire.cart-component', ['total' => \$this->total, </pre> |

| | | |
|---------------------|---|---|
| | | 'content' => \$this->content,); } |
| Видалення кошика | з | public function removeFromCart(string \$id): void { Cart::remove(\$id); \$this->updateCart(); } |
| Очищення кошика | | public function clearCart(): void { Cart::clear(); \$this->updateCart(); } |

Продовження таблиці 1.5

| | | |
|---|-----|--|
| Оновлення кошика додаванні товару | при | public function updateCartItem(string \$id, string \$action): void { Cart::update(\$id, \$action); \$this->updateCart(); } |
| Оновлення вартості кошика | | public function updateCart() { \$this->total = Cart::total(); \$this->content = Cart::content(); } |

У ProductComponent описують початкові дані:

public \$post;

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 37 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

public \$quantity;

Опис функцій ProductComponent

Таблиця 1.6

| | |
|--------------------------|--|
| Початкова кількість | <pre>public function mount(): void { \$this->quantity = 1; }</pre> |
| Повернення View з даними | <pre>public function render(): View { return view('livewire.product-component'); }</pre> |

Продовження таблиці 1.6

| | |
|---------------------|--|
| Додавання до кошику | <pre>public function addToCart(): void { Cart::add(\$this->post->id, \$this->post->title, \$this->post->getRawOriginal('price'), \$this-> >quantity); \$this->emit('productAddedToCart'); }</pre> |
|---------------------|--|

У папці Http для більшої практичності створюється папка Services та додається файл CartService.

Початкові дані класу мають бути визначені як мінімальне значення, захищений модифікатор protected \$session та \$instance.

```
const MINIMUM_QUANTITY = 1;
const DEFAULT_INSTANCE = 'shopping-cart';

protected $session;
protected $instance;
```

Важливий параметр @param \Illuminate\Session\SessionManager \$session, який описується перед функцією для створення сесій.

```

public function __construct(SessionManager $session)
{
    $this->session = $session;
}

```

Магічним методом `__construct` та менеджером сесій зберігається інформація користувачів з унікальним ідентифікатором. При кожному оновленні кошика в сесії сторінка не оновлюватиметься.

Для визначення наступної функції прописуються додаткові параметри:

```

- ID;
- Заголовок;
- Ціна;
- Кількість;
- Параметри.
* @param string $id
* @param string $title
* @param string $price
* @param string $quantity
* @param array $options
* @return void
*/

```

```

public function add($id, $title, $price, $quantity, $options = []): void
{
    $cartItem = $this->createCartItem($title, $price, $quantity, $options);

    $content = $this->getContent();

    if ($content->has($id)) {
        $cartItem->put('quantity', $content->get($id)->get('quantity') + $quantity);
    }
}

```

```

$content->put($id, $cartItem);

$this->session->put(self::DEFAULT_INSTANCE, $content);
}

```

Оновлення кількості в кошику описується функцією з додатковими параметрами:

```

* @param string $id
* @param string $action
* @return void

public function update(string $id, string $action): void
{
    $content = $this->getContent();

    if ($content->has($id)) {
        $cartItem = $content->get($id);

        switch ($action) {
            case 'plus':
                $cartItem->put('quantity', $content->get($id)->get('quantity') + 1);
                break;
            case 'minus':
                $updatedQuantity = $content->get($id)->get('quantity') - 1;

                if ($updatedQuantity < self::MINIMUM_QUANTITY) {
                    $updatedQuantity = self::MINIMUM_QUANTITY;
                }

                $cartItem->put('quantity', $updatedQuantity);
                break;
        }
    }
}

```

```

    }

    $content->put($id, $cartItem);

    $this->session->put(self::DEFAULT_INSTANCE, $content);
}
}

```

У разі if використовується конструкція switch і case. У switch вводиться змінна \$action для перевірки оновлення випадків натискання + або користувачем при зміні кількості товару, який додається до кошика. Мінімальне значення описувалося на самому початку CartService - MINIMUM_QUANTITY який спочатку 1. Змінна \$cartItem бере дані кількості та оновлює кількість, що вводиться в кошик. Через змінну \$content завантажуються дані: \$id та \$cartItem. Посилаючись на сесію, додається DEFAULT_INSTANCE і \$content.

Опис видалення товару з кошика початкові параметри: \$id

```

*                @param                string                $id
*                @return                void
*/
public function remove(string $id): void
{
    $content = $this->getContent();

    if ($content->has($id)) {
        $this->session->put(self::DEFAULT_INSTANCE, $content->except($id));
    }
}

```

Використовуючи команду ехсерт, у сесії видаляється ID товару, який був доданий до кошика.

Очищення кошика відбувається за допомогою команди, вбудованої в Laravel – forget[12].

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 41 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

```

public          function          clear():          void
{
    $this->session->forget(self::DEFAULT_INSTANCE);
}

```

Повернення вмісту кошика за допомогою Collection.

```

public          function          content():          Collection
{
    return is_null($this->session->get(self::DEFAULT_INSTANCE)) ? collect([]) :
    $this->session->get(self::DEFAULT_INSTANCE);
}

```

Повернення загальної суми товарів у кошику. Змінна \$total обчислюється ціною товару помноженою на кількість товарів у кошику. Повертає числовий формат із 2 цифрами після коми.

```

public          function          total():          string
{
    $content          =          $this->getContent();

    $total          =          $content->reduce(function          ($total,          $item)          {
        return          $total          +=          $item->get('price')          *          $item->get('quantity');
    });

    return          number_format($total,          2);
}

```

Створення нових елементів кошика із заданих вхідних даних.

Початкові змінні для захищеної функції:

- назва;
- ціна;
- кількість;
- параметри.

```

*          @param          string          $title

```

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 42 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

```

*           @param           string           $price
*           @param           string           $quantity
*           @param           array            $options
*           @return           \Illuminate\Support\Collection
*/

protected function createCartItem(string $title, string $price, string $quantity, array
$options): Collection
{
    $price           =           floatval($price);
    $quantity        =           intval($quantity);

    if ($quantity < self::MINIMUM_QUANTITY) {
        $quantity    =           self::MINIMUM_QUANTITY;
    }

    return collect([
        'title'       =>           $title,
        'price'       =>           $price,
        'quantity'    =>           $quantity,
        'options'     =>           $options,
    ]);
}

```

У кошику кількість одного товару не може бути меншою за 1. Після повертається колекція із заголовка, ціни, кількості та параметрів.

В view файле cart-component.blade.php создаём условие проверки содержимого в корзине.

```

@if ($content->count() > 0)
@foreach ($content as $id => $item)
    wire:click="updateCartItem({{ $id }}, 'minus'
    {{ $item->get('title') }} x {{ ($item->get('quantity')) }}

```

```

wire:click="updateCartItem({{ $id }}, 'plus'
wire:click="removeFromCart({{ $id }})"
@endforeach

```

Всього: {{ \$total }} UAH

```

wire:click="clearCart">Clear Cart

```

```

@else

```

```

<p>

```

Кошик

порожній!</p>

```

@endif

```

Для `product-component.blade.php` описується максимум кількість, яку можна додати за раз і кнопка додавання товару в кошик.

Рисунок 1.11. Кошик на сторінці користувача

1.5 Розробка телеграм-чат бота для інтернет-магазину

Інтернет-магазини стали надзвичайно популярними в останні роки, і не дивно, що все більше бізнесів переходять в онлайн-простір. Але просто мати електронний магазин більше не є достатньою умовою для успіху. Зараз більшість клієнтів очікують наявності зручних інструментів для комунікації, які дозволять їм отримати необхідну інформацію швидко та без зайвих зусиль.

Один з таких інструментів - це телеграм-чат бот, який може значно полегшити спілкування між клієнтами та інтернет-магазином. Розробка власного телеграм-чат бота для магазину може бути корисною не тільки для

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | БКС 29.15.000.00 ДП | Арк. |
| | | | | | | 44 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

споживачів, але й для самого бізнесу. Він може автоматизувати процеси замовлення, підтримки та продажу, що зменшить навантаження на персонал та дозволить ефективніше використовувати ресурси.

У цьому тексті ми розглянемо основні переваги створення телеграм-чат бота для інтернет-магазину та процес його розробки.

Телеграм-чат бот може автоматизувати процеси замовлення, оплати та доставки товарів, що зменшить навантаження на персонал та забезпечить ефективніше використання ресурсів.

Покращення клієнтського досвіду, забезпечує швидку та якісну підтримку клієнтів, що позитивно позначиться на їхньому досвіді користування сервісом та зробить їх задоволеними.

Для створення ботів, популярна мова програмування Python і бібліотека aiogram. Aiogram є потужним інструментом для розробки ботів та пов'язаних з ними додатків в Telegram.

Ця бібліотека дозволяє з легкістю отримувати та оброблювати повідомлення, команди, події, інлайн-запити, клавіатури, а також зберігати дані користувачів. Для створення ботів за допомогою aiogram не потрібно знати складний синтаксис Telegram API, оскільки ця бібліотека приховує всі складнощі та надає простий та зрозумілий інтерфейс.

Крім того, Python має велику спільноту розробників, яка постійно розширює функціональність мови та робить налагодження проектів швидким і безболісним. Тому використання Python та aiogram є дуже популярним в середовищі розробки ботів для Telegram.

Робота з БД буде безпосередньо, минаючи сервер сайту. Це дасть змогу гнучкіше керувати інформацією користувачів. Для хешування паролів користувачів використовується bcrypt. Це дасть змогу захистити призначені для користувача паролі від зломів, оскільки звичайні люди використовують один пароль на всі свої сервіси, що досить небезпечно.

Для взаємодією з БД Sqlite використовуватимемо ORM бібліотеку

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 45 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

SQLAlchemy, яка дасть змогу надалі відв'язатися від sqlite і перейти на продуктивніші ядра, в яких можна підключатися віддалено. Це бібліотека мови Python, що надає інструменти для роботи з базами даних. SQLAlchemy підтримує роботу з багатьма типами баз даних, включаючи SQLite, PostgreSQL, MySQL та багатьма іншими.

Одна з ключових особливостей SQLAlchemy - це ORM (Object-Relational Mapping), що дозволяє працювати з базами даних у термінах об'єктно-орієнтованих програм. Це дає можливість програмістам працювати з базами даних на рівні абстракцій, що вище, ніж класичний SQL.

Окрім ORM, SQLAlchemy також надає інструменти для створення SQL-запитів на рівні рядка коду, операцій з транзакціями та з'єднаннями з базами даних.

Отримання токена для Telegram-бота - це необхідний крок, який потрібно зробити, щоб розпочати розробку свого бота в мережі Telegram. Цей токен дає доступ до функціоналу бота і повинен бути захищеним, тому що його можна використовувати для з'єднання з вашим ботом.

Для отримання токена необхідно перш ніж все завести акаунт користувача в Telegram, потім знайти бота "BotFather" в полі пошуку і запустити його. Далі слід створити нового бота, надіславши повідомлення "/newbot", вказати назву свого бота і юзернейм (у форматі "@username_bot"). Після успішного створення бота BotFather надішле токен, який можна використовувати для з'єднання з ботом.

Щоб захистити токен, рекомендується зберігати його в окремому файлі конфігурації, який не публікується в мережі і не потрапляє в репозиторій проекту. Для захисту токена від крадіжок, використовуватимемо змінні оточення, який рідко використовують, тому менший шанс того, що він буде поцуплений. Також необхідно уникати передачі токена користувачам і не публікувати його в коді.

Приступаємо до коду. Для початку потрібно зробити взаємодію з базою даних, створюємо моделі таблиць, до яких будемо звертатися.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 46 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

```
class MainTable(base):
    __tablename__ = "users"

    id = Column(Integer, primary_key=True)
    telegram_id = Column(Integer)
    name = Column(String)
    email = Column(String)
    email_verified_at = Column(DateTime)
    password = Column(String)
    phone = Column(Integer)
    role_id = Column(String)
    remember_token = Column(String)
    created_at = Column(DateTime(timezone=True), default=datetime.datetime.utcnow)
    updated_at = Column(DateTime(timezone=True), default=datetime.datetime.utcnow)
    two_factor_secret = Column(String)
    two_factor_recovery_codes = Column(String)

class postsTable(base):
    __tablename__ = "posts"

    id = Column(Integer, primary_key=True)
    title = Column(String)
    img = Column(String)
    text = Column(String)
    quantity = Column(Integer)
    price = Column(Integer)
    cat_id = Column(Integer())
    created_at = Column(DateTime(timezone=True), default=datetime.datetime.utcnow)
    updated_at = Column(DateTime(timezone=True), default=datetime.datetime.utcnow)
```

Рисунок 1.14. Моделі для звернення до таблиць БД

Завдяки тому, що ми використовуємо ORM бібліотеку, до полів баз даних можна звертатися через зрозумілі параметри класу, що покращує читабельність коду.

Далі, налаштуємо файл конфігурації, який буде брати з оточення потрібні параметри. Для цього можна використовувати бібліотеку ruidantic, яка заодно буде перевіряти цілісність даних і в разі невірних даних, виведе помилку про це.



```
class Settings(BaseSettings):  
    TOKEN: str  
    DB_ECHO: bool  
  
    class Config:  
        env_file = '.env'  
        env_file_encoding = 'utf-8'  
        case_sensitive = True  
  
settings = Settings()  
  
storage = MemoryStorage()  
bot = Bot(settings.TOKEN)  
dp = Dispatcher(bot, storage=storage)
```

Рисунок 1.15. Реалізація налаштувань через PATH

Далі потрібно реалізувати функції авторизації та реєстрації користувача, а також перевірку на те, що користувач авторизований, і можливість пошуку товарів та їх купівля

Перша дія в будь-якому телеграм-боті - надсилання боту команду /start, почнемо з неї.

| Змін. | Арк. | № докум. | Підпис | Дата |
|-------|------|----------|--------|------|
| | | | | |

```

StartButton = InlineKeyboardMarkup()
StartCallbackData = CallbackData('act', 'action')
StartButton.add(InlineKeyboardButton('Авторизація',
callback_data=StartCallbackData.new(action='auth')))
StartButton.add(InlineKeyboardButton('Зареєструватися',
callback_data=StartCallbackData.new(action='reg')))

@dp.message_handler(commands=["start"])
@dp.throttled(rate=1)
async def start_command(message: Message):
    await message.answer("Hello")

    async with engine.begin() as conn:
        result = (await conn.execute(select(MainTable).where(MainTable.telegram_id ==
message.from_user.id))).first()
        if result != None:
            await message.answer(f"Здрастуйте, {result[2]}. Введіть /product 'назва продукту', щоб ви
змогли отримати список доступних продуктів")
        else:
            await message.answer(
                f"Вітаємо у нашому магазині!\n"
                f"Якщо ви шукаєте якісні та стильні товари, виготовлені з найкращих матеріалів, то ви
потрапили за адресою.\n"
                f"Наші продукти допоможуть вам відчувати себе комфортно та стильно в будь-яких
обставинах.\n"
                f"Звертайтеся до нас із запитаннями та пропозиціями, ми завжди раді допомогти вам
зробити правильний вибір!\n"
                f"Щоб почати покупки в боті, слід спочатку авторизуватися/зареєструватися",
                reply_markup=StartButton)

```

Рисунок 1.16. Реалізація команди /start

Створюємо клавіатуру з кнопками Авторизації/Реєстрації та за допомогою SQL-запиту дивимося, чи прив'язаний користувач свій обліковий запис до акаунту на сайті, якщо прив'язаний, пишемо йому про те, що він може почати пошук товарів.

Інакше, просимо його авторизуватися. Для авторизації просимо ввести пошту і пароль до свого акаунту. Для того, щоб користувач зміг помилитися і не заходити в /start, щоб почати процес від початку, треба використати так звану машину станів, яка задає та зберігає поточний стан користувача, наприклад те, що він вводить логін із паролем.

```

class StartAuth(StatesGroup):
    auth = State()
    reg = State()

@dp.callback_query_handler(StartCallbackData.filter(action=['auth']))
async def auth(callback_query: CallbackQuery, callback_data: CallbackData, state: FSMContext):
    await bot.edit_message_text(
        message_id=callback_query.message.message_id,
        chat_id=callback_query.message.chat.id,
        text="Введіть свою пошту/пароль у форматі 'email password'"
    )
    await state.set_state(StartAuth.auth)

```

Рисунок 1.17. Реалізація обробки натискання на кнопку Авторизуватися

Якщо користувач правильно ввів свої дані, зберігаємо ID його телеграм користувача до його акаунту. Інакше, просимо ввести ще раз.

Для реєстрації потрібно більше даних, а точніше потрібно ввести пошту, пароль і номер телефону.

```

@dp.message_handler(state=StartAuth.reg)
async def reg_state(message: Message, state: FSMContext):
    print(message.text)
    login_password = message.text.split()
    if len(login_password) != 4:
        answer = await message.answer(
            text="Невірно ввели, спробуйте змінити текст",
        )
        await sleep(5)
        await message.delete()
        return await answer.delete()

    async with engine.begin() as conn:
        await conn.execute(insert(MainTable).values(
            name=login_password[0],
            email=login_password[1],
            password=bcrypt.hashpw(login_password[2].encode('utf-8'), bcrypt.gensalt()),
            phone=login_password[3],
            telegram_id=message.from_user.id
        ))
    await message.answer(
        text="Ви увійшли у свій акаунт. /start",
    )
    await state.finish()

```

Рисунок 1.18. Обробник повідомлень у стані реєстрації

Для генерації хеш через алгоритм bcrypt можна використовувати готову бібліотеку, яка захистить пароль, навіть якщо БД буде віддана третім особам.

| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 50 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |

Як видно з першого рядка, ми використовуємо машину станів, поки користувач у стані "reg", тобто в процесі реєстрації, постійно просимо його ввести вірні дані.

Після цього, користувач зможе знайти необхідні йому товар і купити, отримавши тим самим чек.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 51 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

2 ОХОРОНА ПРАЦІ

Охорона праці - це сукупність заходів, які спрямовані на запобігання травматизму та захист здоров'я працівників на робочому місці. Кожен працівник має право на безпечні та здорові умови праці, але це не завжди можливо без виконання певних норм та правил. З метою забезпечення безпеки та охорони здоров'я працівників багато країн світу створюють законодавчу базу, яка регулює питання охорони праці на різних видовищах.

Для поліпшення умов та забезпечення безпеки праці необхідно ефективне співробітництво між усіма рівнями державної влади та громадськістю, а також впровадження програм на державному та місцевому рівнях. Реалізація цих програм може сприяти створенню системи нагляду, навчання та контролю в галузі охорони праці, гармонізації законодавства з європейськими стандартами, забезпеченню інформаційного та науково-методичного супроводу, а також створенню безпечних умов праці на підприємствах та в організаціях різних форм власності. Все це сприятиме захисту життя та здоров'я працюючих, а також забезпечить комплексне вирішення проблем охорони праці.

Крім того, необхідно встановити ефективні механізми контролю за дотриманням норм охорони праці та належного виконання законодавства. Інспекційні органи та органи влади повинні мати необхідні ресурси та повноваження для проведення регулярних перевірок та покарання порушників стандартів безпеки праці.

Також важливо забезпечити систему постійного навчання та підвищення кваліфікації працівників у галузі охорони праці, щоб вони були свідомі своїх прав та обов'язків, а також знали сучасні методи та практики безпечної праці.

У світі існують різні види професій та виробництв, де вимагається дотримання особливих правил та норм, щоб забезпечити безпеку працівників. Охорона праці стала невід'ємною складовою в будь-якій сфері діяльності, де є люди. Вона дає можливість не тільки забезпечити безпеку працівників, але й ефективно вирішувати проблеми на робочому місці та знижувати ризик

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 52 |

виникнення небезпечних ситуацій.

2.1. Аналіз та безпека умов праці працівника на робочому місці

Шкідливі фактори, пов'язані з використанням комп'ютерної техніки, можуть мати негативний вплив на здоров'я користувачів. Ці фактори включають неправильну поставу, напругу очей, шум, електромагнітне випромінювання, біологічні фактори та дезорганізацію режиму роботи та відпочинку.

Неправильна постава може призвести до болей у спині, шиї, руках та зап'ястях. Розглядання екрану комп'ютера може призвести до напруження очей, що може спричинити головні болі, сухість очей та інші проблеми зі здоров'ям очей. Комп'ютери можуть видавати шум, що може викликати стрес та зниження працездатності.

Електромагнітне випромінювання може мати шкідливий вплив на здоров'я людини. Клавіатури та миші комп'ютерів можуть бути джерелом мікробів та інших патогенних організмів. Працюючи за комп'ютером, людина може забути про перерви для відпочинку, що може призвести до зниження працездатності та збільшення ризику розвитку хронічних захворювань.

2.1.1 Організація робочого місця

Для забезпечення комфортної та безпечної праці з ВДТ, необхідно застосовувати відповідну організацію робочого місця та обладнання. Всі елементи робочого місця та їх взаємне розташування повинні відповідати ергономічним вимогам, які враховують характер і особливості трудової діяльності (ДСанПіН 3.3.2.-007-98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

Робочі місця з ВДТ краще розташовувати так, щоб природне світло падало збоку, переважно зліва. Для робочих столів з ВДТ слід дотримуватися

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 53 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

відстаней: між бічними поверхнями ВДТ - 1,2 м; від тильної поверхні одного ВДТ до екрану іншого - 2,5 м. Екран ВДТ має бути розміщений на оптимальній відстані від очей користувача, що становить 600-700 мм, але не ближче, ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів.

Клавіатуру слід розташовувати на поверхні столу на відстані 100-300 мм від краю, зверненого до працюючого. Конструкція клавіатури повинна передбачати опорний пристрій, який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5-150.

При обладнанні робочого місця лазерним принтером необхідно враховувати вимоги СанПіН № 5804-91 щодо параметрів лазерного випромінювання.

2.1.2 Вимоги безпеки до мікроклімату, освітлення, шуму, виробничих випромінювань

Основними документами, які визначають параметри мікроклімату виробничих приміщень, є ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. Ці параметри регулюються для робочої зони - визначеного простору, де знаходяться робочі місця. Згідно з нормативним документом ДСН 3.3.6.042-99 параметри мікроклімату мають відповідати значенням, зазначеним у таблиці 1.

Параметри мікроклімату приміщення. Таблиця 1

| Період року | Категорія робіт | Температура, С | | Відносна вологість, % |
|-------------|-----------------|----------------|-----------|-----------------------|
| | | оптимальна | допустима | |
| Холодний | Легка – Іа | 22-24 | 21-25 | 40-60 |
| Теплий | Легка – Іа | 23-25 | 22-26 | 40-60 |

Приміщення, в яких встановлені персональні комп'ютери, повинні мати природне та штучне освітлення. Природне освітлення здійснюється через світові прорізи (вікна), орієнтовані переважно на північ чи північний схід. Штучне освітлення в приміщенні здійснюється системою загального рівномірного освітлення. На поверхні столу в зоні розміщення документів штучне освітлення має становити 300-500лк.

Також важливо забезпечити рівномірність та постійність рівня освітленості у всіх виробничих приміщеннях. Це допоможе уникнути частого переадаптування органів зору та зменшить ризик появи засліплювальної дії. Додатково, на робочій поверхні не повинно бути ніяких різних тіней, а також має бути достатньо контрасту між освітлюваними поверхнями. Також важливо уникати небезпечних та шкідливих виробничих чинників, таких як шум, теплове випромінювання, електрична небезпека, пожежо- та вибухонебезпека світильників.

Нарешті, виробниче освітлення має бути надійним, простим у експлуатації, економічним та естетичним.

Так як шум має 35Дб, сприйняття шуму людським вухом межується від 20Дб до 120 дб, це означає, що при роботі за ЕОМ шум не заважає, працівнику працювати.

Для запобігання виникнення інших шумів у відповідності з ГОСТ 12.1.029-80 зниження шуму й вібрації в приміщенні дипломним проектом передбаченні звукоізоляція вікон та дверей.

2.2. Пожежна безпека

Під визначенням терміну пожежна безпека розуміють систему заходів, які спрямовані на захист людей та майна від вогню. В приміщеннях з електричними мережами, дотримання пожежної безпеки регулюється НАПК А.01.001-2014 Правила пожежної безпеки в Україні. Крім того, для роботи оператора ЕОМ необхідно мати приміщення, яке відповідає категорії Д пожежної безпеки, тобто містить негорючі речовини та матеріали в холодному

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 55 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

стані.

До засобів пожежогасіння належать внутрішні пожежні водопроводи (крани - ПК), вогнегасники (вуглекислотні та порошкові), сухий пісок та інші. У будівлях пожежні крани зазвичай розміщують у коридорах та на майданчиках сходових кліток. Кожен пожежний кран обладнаний пожежним рукавом, який знаходиться в спеціальному ящику на висоті 1,35 м від полу.

Для загасання пожеж на початкових етапах широко використовують вогнегасники. У виробничих приміщеннях основною формою вогнегасників є вуглекислотні, які мають високу ефективність у гасінні пожеж, а також дозволяють зберегти електричне обладнання. Вогнегасники розміщують на видному місці на висоті не більше 1,5 м від полу.

Для вирішення евакуаційних задач у виробничих приміщеннях зазвичай є запасні виходи. Двері, які ведуть до запасних виходів, повинні мати освітлений напис "Запасний вихід". План евакуації вивішують на видному місці біля основного виходу. Приміщення, в яких встановлені персональні комп'ютери, повинні мати природне та штучне освітлення. Природне освітлення здійснюється через світові прорізи (вікна), орієнтовані переважно на північ чи північний схід. Штучне освітлення в приміщенні здійснюється системою загального рівномірного освітлення. На поверхні столу в зоні розміщення документів штучне освітлення має становити 300-500лк.

У разі виникнення пожежі потрібно відключити електропостачання, негайно повідомити за номером 101 професійну пожежну команду, дотримуючись плану евакуації, евакуювати людей з будівлі і розпочати процедуру гасіння пожежі.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 56 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

ВИСНОВКИ

Для сайту було створено та реалізовано поділ ролей на адміністратор та користувачів, що дозволяє розділити видимі сторінки між користувачами. Для адміністративної панелі були створені контролери та view файли для більш інтуїтивної навігації. На сторінках було створено @csrf захист. При реєстрації у всіх користувачів паролі зберігаються у зашифрованому вигляді

Під час створення адміністративної панелі було досягнуто такі результати:

- Реалізовано реєстрацію та авторизацію.
- У базі даних зберігаються зашифровані паролі.
- Кожному новому користувачеві надається роль користувача.
- На головній сторінці відображається загальна кількість різновидів товару, користувачів та адміністраторів.
- Створено бічна панель де реалізовані розділи: головна, користувачі, продукти та категорії.
- У розділі користувачі можуть переглядати всіх користувачів і їх віддаляти.
- У розділі продукти можливість створювати, видаляти, редагувати та переглядати всі товари.
- У розділі категорії реалізовано створення та перегляд усіх існуючих категорій.

- Сторінки захищені від злому.

Після створення сторінки для користувачів було створено:

- Реалізація сесії користувача.
- Кошик з можливістю додавання товарів до нього.
- Кошик без перезавантаження сторінки автоматично оновлює її вартість.

В результаті створення сайту адміністратори та користувачі отримали зручні та інтуїтивно зрозумілі сторінки для майбутнього інтернет магазину.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| | | | | | | 57 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Для подальшої розробки рекомендується розширити функціонал кошика з можливістю додавання замовлень до бази даних, використовуючи сесії.

Розробка телеграм-чат бота для інтернет-магазину є важливим кроком у поліпшенні якості обслуговування клієнтів. Завдяки такому чат-боту користувачам стає зручніше звертатися за допомогою, оскільки вони можуть отримати відповіді на свої запитання у будь-який час без необхідності чекати на відповідь оператора.

Крім цього, розробка телеграм-чат бота дозволяє знизити витрати на підтримку клієнтів, оскільки програма може виконувати багато функцій автоматично без участі живої людини.

Зокрема, телеграм-чат бот може надавати користувачам інформацію про товари, допомагати з оформленням замовлень, повідомляти про акції та знижки, відповідати на запитання про оплату та доставку тощо.

Отже, розробка телеграм-чат бота може стати важливим інструментом для покращення обслуговування клієнтів та зниження витрат на підтримку.

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| | | | | | | 58 |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ

1. <https://laravel.su/docs/8.x/structure>
2. <https://laravel.com/docs/10.x>
3. <https://laravel.com/docs/10.x/migrations>
4. <https://spatie.be/docs/laravel-permission/v5/introduction>
5. <https://spatie.be/docs/laravel-permission/v5/installation-laravel>
6. <https://laravel.com/docs/10.x/fortify>
7. <https://github.com/laravel/ui>
8. <https://laravel.com/docs/7.x/authentication>
9. <https://spatie.be/docs/laravel-permission/v5/basic-usage/artisan>
10. <https://laravel-livewire.com/docs/2.x/quickstart>
11. <https://adevait.com/laravel/create-shopping-cart-laravel-livewire>
12. <https://laravel.com/docs/10.x/collections>

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 59 |

ВСП «Одеський технічний фаховий коледж ОНТУ»

Тема дипломного проєкту:

Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень

Виконав: Матвеев М.І,
група 2БКС-29
Керівник: Іванова Л.В.

ВСП «Одеський технічний фаховий коледж ОНТУ»

Вступ

Актуальність

Ріст популярності інтернет продажів, чат-ботів, зручність та доступність, покращення комунікації та залучення

Мета

Розробити інтернет магазин та чат-бот для комерційного підприємства

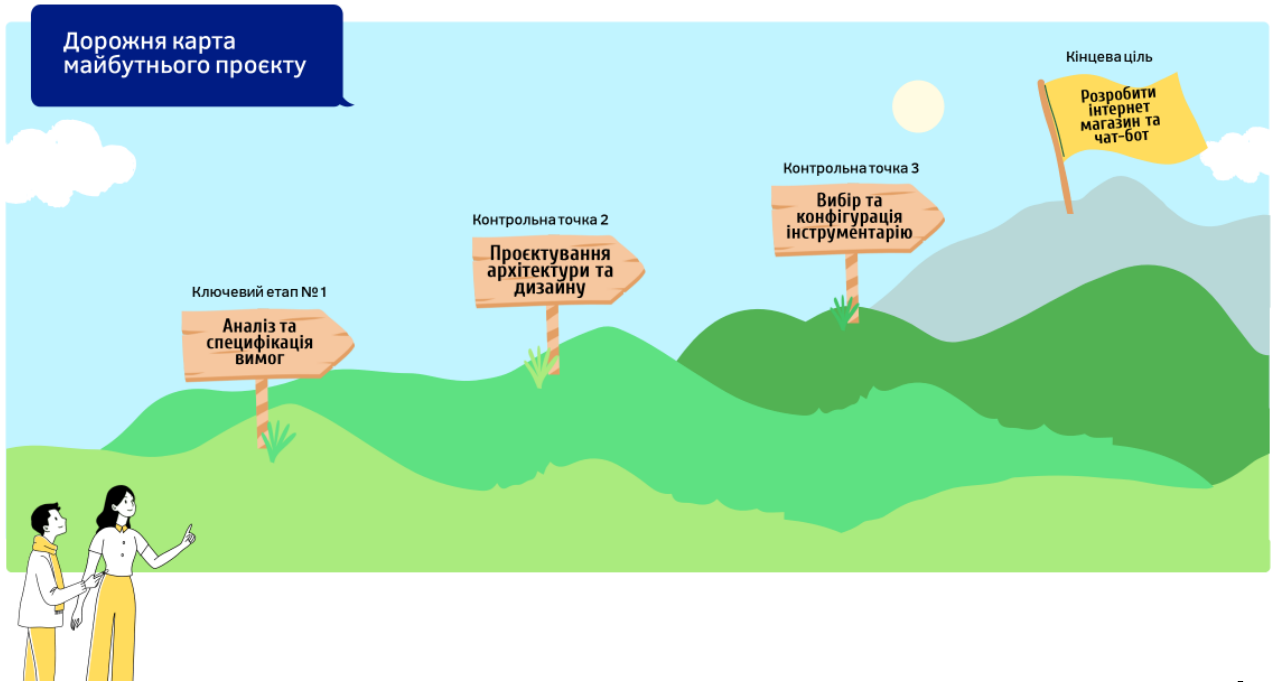
Об'єкт

Розробка та використання інтернет ресурсу та чат-бота для ознайомлення з асортиментом

Предмет

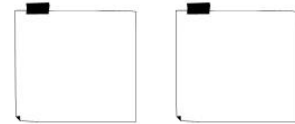
Функціональність та ефективність розробленого чат-бота для оформлення замовлень

| | | | | | | |
|-------|------|----------|--------|------|---------------------|------|
| | | | | | БКС 29.15.000.00 ДП | Арк. |
| | | | | | | 60 |
| Змін. | Арк. | № докум. | Підпис | Дата | | |



Етап №1

Під час аналізу вимог було проведено детальний огляд потреб користувачів та вимог.



| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | БКС 29.15.000.00 ДП | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 61 |

Етап №2

Під час проєктування визначалися основні компоненти, структура та взаємодія між ними, а також розроблявся зовнішній вигляд та інтерфейс користувача.



Етап №3

Основні фактори, які враховувалися при виборі засобів реалізації.



| | | | | | | |
|-------|------|----------|--------|------|----------------------------|------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | Арк. |
| Змін. | Арк. | № докум. | Підпис | Дата | | 62 |

Кінцева ціль

Процес реалізації чат-боту з використанням SmartSender можна розділити на кілька етапів.

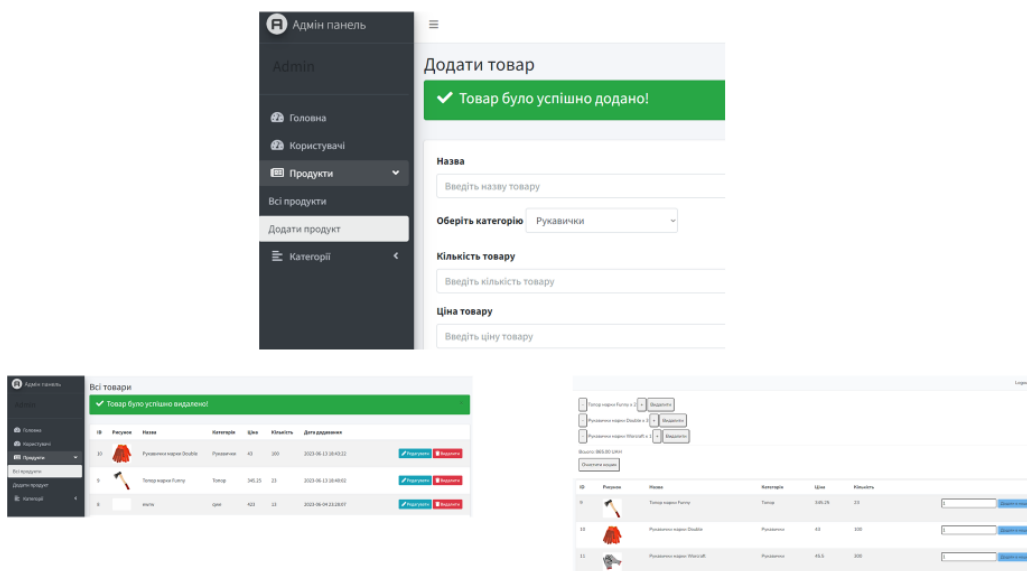


Результати тестування та оцінка функціональності та ефективності чат-боту

Під час тестування інтернет магазину та чат-боту, були отримані наступні результати:

- 1 Функціональність
- 2 Ефективність
- 3 Взаємодія з користувачами
- 4 Інтеграція з іншими системами

Тестування



Висновки та практичні рекомендації для подальшого вдосконалення

Отже, розроблений інтернет магазин та чат-бот виявився успішним та ефективним рішенням для ознайомлення клієнтів з товаром та оформлення замовлення. Його функціональність, швидкість та зручність використання сприяють поліпшенню комунікації та забезпечують задоволення потреб абітурієнтів. Запропоновані рекомендації дають змогу подальшого вдосконалення чат-боту та його впровадження на практиці з метою забезпечення ще кращих результатів і задоволення вимог світового технічного рівня.

- 1 Розширення функціоналу
- 2 Покращення навчання чат-боту
- 3 Розширення інтеграцій
- 4 Вдосконалення інтерфейсу

| | | | | |
|-------|------|----------|--------|------|
| | | | | |
| Змін. | Арк. | № докум. | Підпис | Дата |

БКС 29.15.000.00 ДП

Арк.

64

Дякую за увагу!

Гарного дня!

| | | | | | | |
|--------------|-------------|-----------------|---------------|-------------|----------------------------|-------------|
| | | | | | <i>БКС 29.15.000.00 ДП</i> | <i>Арк.</i> |
| <i>Змін.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 65 |

РЕЦЕНЗІЯ

на кваліфікаційну роботу здобувача (здобувачки) освіти

відділення комп'ютерних систем

Матвєєва Михайла Ігоровича

(прізвище, ім'я та по батькові)

Спеціальність 123 Комп'ютерна інженерія

ОПП Комп'ютерна інженерія

Керівник кваліфікаційної роботи _____

Іванова Лілія Вікторівна

(прізвище, ім'я та по батькові)

Тема кваліфікаційної роботи _____

«Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень»

Обсяг розрахунково-пояснювальної записки 61 сторінок

Обсяг графічної (презентаційної) частини 10 аркушів (слайдів)

ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

а) заключення про ступінь відповідності виконаного кваліфікаційної роботи завданню
кваліфікаційна робота у повному обсязі відповідає темі та завданню

б) характеристика виконання кожного розділу кваліфікаційної роботи _____

Кваліфікаційна робота складається з розділів: Передмова. Технічне завдання для сайту.

Технічне завдання на створення Telegram-бота з функціями авторизації та замовлення.

Аналіз структури сайту на Laravel. Охорона праці. Висновки. Список використаних джерел

в) оцінка якості виконання пояснювальної записки та графічної частини кваліфікаційної роботи

Пояснювальна записка виконана якісно, у достатньому обсязі, відповідно до

індивідуального завдання та теми дипломного проекту, розділи пояснювальної записки

відповідають етапам рішення завдання, поставленого у кваліфікаційній роботі.

Презентація виконана якісно, у достатньому обсязі. Презентація наочно демонструє

результати роботи.

г) перелік позитивних якостей кваліфікаційної роботи _____
Актуальна тематика

Сучасні технології реалізації програмного продукту

Якісне подання результатів роботи

д) основні недоліки кваліфікаційної роботи _____
Не визначені механізми забезпечення захисту інформації

Відсутність можливості здійснення оплати за товар

Типовий дизайн

Оцінка розрахункової частини _____ *Відмінно*
Оцінка графічної частини _____ *Добре*
Загальна оцінка _____ *Відмінно*

Прізвище, ім'я, по батькові рецензента _____ *к.т.н. Шibaєва Наталя Олегівна*

Місце роботи і посада рецензента _____ *Національний університет «Одеська політехніка»,
доцент кафедри інформаційних технологій*

Підпис: _____
« _____ 2025 р.



Відокремлений структурний підрозділ
Одеський технічний фаховий коледж ОНТУ

ВІДГУК

Керівника про кваліфікаційну роботу бакалавра

Матвеева Михайла Ігоровича

(прізвище, ім'я та по батькові)

Освітньо-професійна програма *«Комп'ютерна інженерія»*

Спеціальність *123 «Комп'ютерна інженерія»*

Тема кваліфікаційної роботи

«Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) Обсяг і якість виконання роботи (розрахунково-пояснювальної записки)

Пояснювальна записка виконана якісно, у достатньому обсязі, відповідно до індивідуального завдання та теми дипломного проекту, розділи пояснювальної записки відповідають етапам рішення завдання, поставленого у дипломному проекті

Презентація виконана якісно, у достатньому обсязі. Презентація наочно демонструє результати роботи.

б) Самостійність роботи над кваліфікаційною роботою

Студент самостійно обрала напрям та тематику кваліфікаційної роботи. Провів аналіз існуючих рішень і зробив необхідні висновки для реалізації проекту. Виявив навички самостійно опрацьовувати новий матеріал та виконувати пошук необхідної літератури та інших джерел інформації

в) Теоретична підготовка бакалавра _____

відповідає вимогам, що надаються до бакалавра зі спеціальності

«Комп'ютерна інженерія»

г) Вміння розв'язувати виробничі і конструкторські питання на базі останніх досліджень науки і техніки, передових методів виробництва _____

У дипломному проекті розглянута та реалізована сучасна тема створення мобільних інформаційних чат-ботів для соціальних мереж описані принципи розробки чат-боту у сучасному середовищі програмування та процес підбору інструментів та технологій для розробки графічного дизайну та ергономіки. Обґрунтовано вибір програмних засобів для реалізації програмного забезпечення, проведено його тестування

Загальна оцінка _____ 5(відмінно) _____

Прізвище, ім'я, по батькові _____ Іванова Лілія Вікторівна _____

Місце роботи і посада керівника проекту ВСП «Одеський технічний фаховий коледж ОНТУ» к.т.н., зав. кафедрою Комп'ютерної інженерії _____

Підпис _____

« 20 »

96

2025р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Матвеев Михайло Ігорович,
здобувач освіти гр. 2БКС-29, та

Іванова Лілія Вікторівна,
керівник випускної кваліфікаційної роботи,

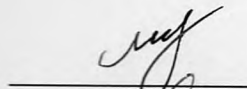
не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи бакалавра на тему:

«Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень» (автор роботи – Матвеев М.І., керівник роботи – Іванова Л.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

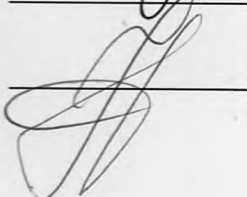
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Матвеев М.І. /

Керівник



/ Іванова Л.В. /

«20» червня 2025 р.

Д О В І Д К А

кафедри комп'ютерної інженерії
про допуск до захисту кваліфікаційної роботи
здобувача (здобувачки) освіти II курсу
відділення комп'ютерних систем групи 2БКС-29

Матвєєва Михайла Ігоровича

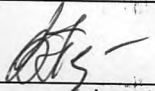
на тему Розробка інтернет-магазину для комерційної організації та чат-боту
в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень

Висновок відповідальної особи за проведення нормоконтролю:

пояснювальна записка до кваліфікаційної роботи виконана з деякими

порушеннями ДСТУ та оформлена відповідно до вимог Положення про

дипломне проєктування


(підпис)

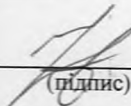
25.06.2025

(дата)

Петрашова В.І.

(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагиату згідно звіту про перевірку від 11.06.2025 р. значення коефіцієнту
подібності в роботі становить 21,10%, коефіцієнт цитування – 3,89%.


(підпис)

25.06.2025

(дата)

Краснокутська К.Г.

(П.І.Б.)

Попередня експертиза (малий захист) кваліфікаційної роботи

здобувача (здобувачки) освіти

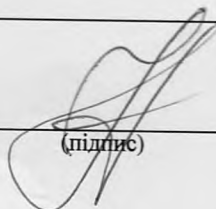
Матвєєва М.І.

(П.І.Б.)

проведена « 25 » червня 2025 р.

Висновки Пояснювальна записка до кваліфікаційної роботи виконана у
повному обсязі. Випускна кваліфікаційна робота відповідає вимогам
Положення про дипломне проєктування та рекомендована до захисту.

Зав. кафедри КІ


(підпис)

Іванова Л.В.

(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка інтернет-магазину для комерційної організації та чат-боту в соціальній мережі Telegram для реєстрації користувачів і оформлення замовлень

Автор

Науковий керівник / Експерт

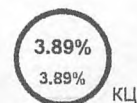
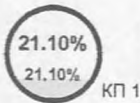
Матвеев Михайло Ігорович/Еванова Лілія Вікторівна

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

8203

Кількість слів

70097

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

| | | |
|------------------------|--|----|
| Заміна букв | | 0 |
| Інтервали | | 0 |
| Мікропробіли | | 0 |
| Білі знаки | | 0 |
| Парафрази (SmartMarks) | | 78 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

| ПОРЯДКОВИЙ НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ) | Колір тексту |
|---------------------|---|---|
| | | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
| 1 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content | 85 1.04 % |
| 2 | https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download | 80 0.98 % |
| 3 | https://adevait.com/laravel/create-shopping-cart-laravel-livewire | 64 0.78 % |
| 4 | https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download | 62 0.76 % |

| | | |
|----|---|-----------|
| 5 | https://card-file.ontu.edu.ua/items/c1578562-5733-4ac6-994c-f14e0d3bda3d/full | 59 0.72 % |
| 6 | https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download | 58 0.71 % |
| 7 | https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download | 55 0.67 % |
| 8 | https://card-file.ontu.edu.ua/bitstreams/7b1e10b9-0ac2-4b07-afc4-8cdf7db780/download | 53 0.65 % |
| 9 | https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download | 50 0.61 % |
| 10 | https://card-file.ontu.edu.ua/bitstreams/7b1e10b9-0ac2-4b07-afc4-8cdf7db780/download | 50 0.61 % |

з домашньої бази даних (0.00 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|-----------|--|
|------------------|-----------|--|

з програми обміну базами даних (0.48 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|--|--|
| 1 | ФКПІ_2024_121_Жипка_О.В 7/11/2024 Ukrainian national aviation university (Ukrainian national aviation university) | 11 (2) 0.13 % |
| 2 | Вебресурс «Освітня платформа» з особистим кабінетом студента за допомогою фреймворку Laravel 3/15/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute) | 10 (2) 0.12 % |
| 3 | Automation of search for funding startups and IT projects 6/3/2019 International IT University (Информационные системы) | 10 (2) 0.12 % |
| 4 | Kolomoisev_mag_rob.docx 12/22/2022 Sumy State University (Кафедра комп'ютерних наук) | 8 (1) 0.10 % |

з Інтернету (20.63 %)

| ПОРЯДКОВИЙ НОМЕР | ДЖЕРЕЛО URL | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|---|--|
| 1 | https://adevait.com/laravel/create-shopping-cart-laravel-livewire | 315 (16) 3.84 % |
| 2 | https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download | 292 (11) 3.56 % |
| 3 | https://card-file.ontu.edu.ua/bitstreams/7b1e10b9-0ac2-4b07-afc4-8cdf7db780/download | 141 (6) 1.72 % |
| 4 | https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download | 104 (7) 1.27 % |
| 5 | https://larainfo.com/blogs/laravel-9-insert-category-in-posts-crud-example | 99 (9) 1.21 % |
| 6 | https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download | 98 (3) 1.19 % |
| 7 | https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download | 86 (10) 1.05 % |
| 8 | https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content | 85 (1) 1.04 % |
| 9 | https://card-file.ontu.edu.ua/items/c1578562-5733-4ac6-994c-f14e0d3bda3d/full | 80 (3) 0.98 % |
| 10 | https://masters.donntu.org/2020/fknt/yrieva/diss/indexu.htm | 74 (4) 0.90 % |
| 11 | https://laravel.sillo.org/creer-un-blog-les-donnees/ | 62 (5) 0.76 % |

