

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Гордіюка Дениса Володимировича
(ПІБ)

академічної групи 121-19-1
(шифр)

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

освітньої програми Інженерія програмного забезпечення
(назва освітньої програми)

на тему: Розробка веб-орієнтованого програмного забезпечення
та проектування бази даних за допомогою фреймворка laravel на
мові програмування PHP

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи				
розділів:				
спеціальний	<i>проф. Бердник М.Г.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент	<i>доц. Шедловський І.А.</i>			
Нормоконтролер	<i>ст. викл. Мартиненко А.А.</i>			

Дніпро
2023

РЕФЕРАТ

Пояснювальна записка: 74 с., 21 рис., 3 дод., 22 джерела.

Об'єкт розробки: веб-додаток для продажі безнікотинових заправок для кальяну з повноцінною адміністративною панеллю.

Мета кваліфікаційної роботи: розробка веб-орієнтованого програмного забезпечення за допомогою фреймворка Laravel на мові програмування PHP.

У вступній частині кваліфікаційної роботи проведено аналіз та розглянуто сучасний стан проблеми, визначено конкретну мету дослідження та галузь її застосування, наведено обґрунтування актуальності обраної теми та деталізовано постановку задачі.

У першому розділі кваліфікаційної роботи було проведено детальний аналіз предметної галузі, розглянуто актуальність поставленої задачі та цілей розробки, сформульовано конкретну постановку завдання, а також визначено вимоги до програмної реалізації, використовуваних технологій та програмних засобів.

У другому розділі кваліфікаційної роботи було розглянуто функціональне призначення програми. Було описано архітектуру та шаблони програмування, технології та мови, що були обрані для створення продукту. Крім того, в розділі була надана детальна інформація стосовно повного функціонування програми від технічних засобів до інтерфейсів користувача та адміністратора.

У економічному розділі кваліфікаційної роботи була визначена трудомісткість розробки програмного забезпечення для цієї кваліфікаційної роботи. Також, було проведено розрахунок усіх витрат на створення даного веб-застосування.

Практичне значення полягає у розробці веб-додатку, що дозволяє покупати та продавати товар з дотриманням усіх сучасних норм від безпеки до зручності користування.

Актуальність даного продукту визначається попитом як зі сторони споживачів, так і продавців, що значно розширює ринок збуту товарів.

Список ключових слів: ІНТЕРНЕТ, САЙТ, ВЕБ-ДОДАТОК, PHP, LARAVEL, JAVASCRIPT, HTML, CSS.

ABSTRACT

Explanatory note: 74 pages, 21 figures, 3 appendices, 22 sources.

Development object: a web application for selling nicotine-free shisha refills with a fully functional administrative panel.

Objective of the qualification work: development of web-oriented software using the Laravel framework and PHP programming language.

The introductory part of the qualification work includes an analysis and examination of the current state of the problem, determination of the specific research goal and its application area, justification of the relevance of the chosen topic, and detailed task formulation.

The first chapter of the qualification work conducted a detailed analysis of the subject area, discussed the relevance of the stated problem and development goals, formulated specific task requirements, and defined the software implementation, technologies, and tools used.

The second chapter of the qualification work discussed the functional purpose of the program. It described the architecture and programming patterns, technologies, and languages chosen for creating the product. Additionally, the chapter provided detailed information on the complete functioning of the program, from technical components to user and administrator interfaces.

The economic chapter of the qualification work determined the development effort required for the software. It also conducted calculations of all expenses related to creating this web application.

The practical significance lies in the development of a web application that allows buying and selling goods while adhering to modern standards, ranging from safety to user convenience.

The relevance of this product is determined by the demand from both consumers and sellers, significantly expanding the market for goods distribution.

Keywords: INTERNET, WEBSITE, WEB APPLICATION, PHP, LARAVEL, JAVASCRIPT, HTML, CSS.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ЗМІСТ	5
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	8
1.1. Загальні відомості з предметної області	8
1.2. Призначення розробки та область застосування	9
1.3. Підстава для розробки	9
1.4. Постановка завдання.....	9
1.4.1. Функціональні особливості сайту	10
1.4.2. Опис інтерфейсу користувача.....	10
1.5. Вимоги до програми або програмного виробу.....	13
1.5.1. Вимоги до функціональних характеристик	13
1.5.2. Вимоги до інформаційної безпеки	13
1.5.3. Вимоги до складу та параметрів технічних засобів	13
1.5.4. Вимоги до інформаційної та програмної сумісності.....	13
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	15
2.1. Функціональне призначення програми.....	15
2.2. Опис застосованих математичних методів.....	15
2.3. Опис використаної архітектури та шаблонів проектування.....	16
2.4. Опис використаних технологій та мов програмування.....	18
2.5. Опис структури програми та алгоритмів її функціонування.....	20
2.6. Обґрунтування та організація вхідних та вихідних даних програми	23
2.7. Опис розробленого програмного продукту	24
2.7.1. Використані технічні засоби	24
2.7.2. Використані програмні засоби.....	25
2.7.3. Виклик та завантаження програми.....	25
2.7.4. Опис інтерфейсу користувача. Основна частина.....	26

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	40
3.1. Визначення трудомісткості розробки програмного забезпечення.....	40
3.2. Розрахунок витрат на створення програми	44
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ДОДАТОК А. КОД ПРОГРАМИ.....	50
ДОДАТОК Б. ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	73
ДОДАТОК В. ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ.....	74

ВСТУП

Завдання даної кваліфікаційної роботи та об'єкт його діяльності безпосередньо пов'язані з напрямом підготовки «Комп'ютерні науки» та відповідає узагальненій тематиці кваліфікаційних робіт і переліку зазначених виробничих функцій, типових задач діяльності, умінню та компетенціям, якими повинні володіти бакалаври напряму 121 «Програмна інженерія» галузі знань 0501 «Інформатика та обчислювальна техніка» згідно Освітньо-кваліфікаційній характеристиці. Виконання кваліфікаційної роботи надає можливість отримання автору кваліфікації «фахівець з розробки та тестування програмного забезпечення».

Тематика даної кваліфікаційної роботи присвячена розробці веб-орієнтованого програмного забезпечення та проектування бази даних засобами фреймворка Laravel на мові програмування php.

Метою даної кваліфікаційної роботи є створення повноцінного та багатофункціонального веб-сайту для онлайн-продажу смакових безтабачних заправок для кальяна на території сучасного ЄС.

У сучасному світі неможливо уявити продаж іноваційної продукції, особливо з розрахунком на міжнародний ринок, без використання веб-додатків, що дозволяють замовити товар з будь-якої точки світу.

Одним з найбільш популярних інструментів для розробки веб-додатків є фреймворк Laravel на мові програмування PHP. Він надає розробникам зручні інструменти для швидкої розробки веб-застосунків та забезпечує чітку архітектуру, що дозволяє розробникам ефективно працювати із складними проектами.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної області

Використання веб-сайтів для продажу товарів є надзвичайно важливим для сучасних підприємств. За останні кілька років інтернет-торгівля стала однією з найбільш швидко зростаючих галузей бізнесу, завдяки чому більшість компаній, що працюють в галузі роздрібної торгівлі, мають веб-сайти для продажу своїх товарів.

Однією з переваг використання веб-сайтів для продажу товарів є можливість залучення більшої кількості клієнтів. Веб-сайти дають змогу підприємствам привернути клієнтів з будь-якої точки світу, що розширює межі ринку і збільшує продажі. Крім того, веб-сайти дозволяють підприємствам створювати персоналізовані пропозиції та рекламні кампанії, які допомагають збільшити свідомість про бренд та привернути більше клієнтів.

Ще однією важливою перевагою використання веб-сайтів для продажу товарів є можливість зниження витрат на зберігання та розповсюдження товарів. Завдяки електронному способу продажу, підприємства можуть зменшити витрати на складські приміщення та транспортування товарів. Крім того, веб-сайти дозволяють підприємствам швидко та легко оновлювати свої каталоги та пропозиції, що зменшує час та витрати на зміну та друк нових брошур та каталогів.

Однак, використання веб-сайтів для продажу товарів має й свої недоліки. Найбільш відчутним з них є необхідність в інтернет-маркетингу та просуванні веб-сайту, що вимагає додаткових витрат та зусиль. Крім того, веб-сайти не забезпечують можливості фізичного огляду товарів, що може бути недоліком для деяких клієнтів.

У підсумку, використання веб-сайтів для продажу товарів є важливим та необхідним елементом бізнесу в епоху цифрових технологій. Він дозволяє

підприємствам залучати більше клієнтів, знижувати витрати та збільшувати прибуток. Однак, варто мати на увазі недоліки та вимоги до інтернет-маркетингу для успішного використання веб-сайту як засобу продажу товарів.

1.2. Призначення розробки та область застосування

Сайт який розробляється, призначен для використання співробітниками компанії, що займається виготовленням та продажем натуральної безтабачної заправки для кальяну. Областю застосувань проекту є застосування інтернету для продажу продукції.

1.3. Підстава для розробки

Підставами для розробки та виконання кваліфікаційної роботи є:

- освітня програма 121 «Інженерія програмного забезпечення»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 350-с від 16.05.2023 р;
- завдання на кваліфікаційну роботу на тему «Розробка веб-орієнтованого програмного забезпечення та проектування бази даних за допомогою фреймворка Laravel на PHP».

1.4. Постановка завдання

Розробити веб-додаток для продажу продукції та можливістю повного адміністрування цього сайту. Сайт має відповідати сучасним потребам споживачів та виробників продукції.

Веб-додаток має включати веб-клієнт, що реалізовано за допомогою фреймворку Laravel на мові програмування PHP. Веб-додаток повинен працювати у всіх сучасних браузерях.

1.4.1. Функціональні особливості сайту

Функціональні особливості:

- Можливість змінювати мову застосунку;
- Можливість перегляду списку товару;
- Можливість перегляду характеристик і ціни товару;
- Можливість перегляду рейтингу товару;
- Можливість написання власних відгуків на товар;
- Можливість відправлення повідомлень до адміністратора;

Актуальність сайту:

Використання веб-сайтів для продажу товарів є надзвичайно важливим для сучасних підприємств. За останні кілька років інтернет-торгівля стала однією з найбільш швидко зростаючих галузей бізнесу, завдяки чому більшість компаній, що працюють в галузі роздрібної торгівлі, мають веб-сайти для продажу своїх товарів.

1.4.2. Опис інтерфейсу користувача

1. Головна сторінка – звичайна інформаційна сторінка з короткою відомістю про продукцію.
2. Сторінка списку продукції – сторінка з каталогом товару. У кожного товару відображається головна картинка, назва, ціна та середній рейтинг товару. Реалізован посторінковий вивід каталогу. На кожній сторінці виводиться по 8 екземплярів товару.
3. Сторінка продукта – сторінка з повною інформацією про товар: назва, ціна, рейтинг, опис, склад, кнопка замовлення та повний список усіх ревію з постраничним виводом. Реалізован функціонал оцінки продукції: оцінка від 1 до 5 та сам текст з можливістю зворотнього зв'язку.
4. Сторінка списку новин – сторінка зі списком усіх новин у вигляді постраничного виводу. Спочатку виводяться новини, що були додані пізніше. У кожній новини виводиться головна картинка, назва, короткий

опис та дата додавання новини. На кожній сторінці виводиться по 6 екземплярів новин.

5. Сторінка новини – сторінка з конкретною новиною.
6. Сторінка деталей – сторінка з додатковою інформацією про продукцію.
7. Сторінка для зворотного зв'язку – сторінка з формою для зворотного зв'язку, повідомлення з якою приходить на спеціальну пошту, що належить фірмі.
8. Можливість зміни мови на сайті. Сайт підтримує дві мови: англійську та французьку. По замовчуванню встановлюється мова на основі основною мови, вибраної у браузері. Якщо у браузері вибрана французька – по замовчуванню на сайті буде французька, якщо будь-яка інша мова – англійська.

Адміністративна частина:

1. Сторінка для авторизації у вигляді адміністратора.
2. Новини:
 - сторінка зі списком новин – можливість короткого огляду списку усіх новин (номер новини, назва, короткий опис, картинка) та кнопка переходу до редагування конкретної новини;
 - сторінка редагування новини – можливість зміни будь-якого з полів новини (назва, опис, текст) на двох мовах та картинки;
 - сторінка додавання новини – сторінка для створення нового екземпляра новини – назва, опис, текст на двох мовах та картинка.
3. Відгуки:
 - сторінка для модерування останніх відгуків на продукції (можливість підтвердження або видалення не опублікованих відгуків)
 - сторінка всіх підтверджених відгуків на усі екземпляри продукції з можливістю сортування на основі продукції або оцінки. Та можливість видалення відгука;

4. Продукція

- сторінка зі списком продукції – можливість огляду списку усіх продуктів (номер, назва, ціна, склад), сортування по усім полям та наявна кнопка для переходу до редагування продукта;
- сторінка редагування продукції – можливість зміни будь-якого з полів продукції та додавання нових картинок;
- сторінка додавання продукції – сторінка для створення нового екземпляра продукції.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Сайт повинен відкриватись на будь-яких комп'ютерах, які підключені до інтернету и мають браузер.

1.5.2. Вимоги до інформаційної безпеки

Так як сайт буде розміщений на сервері загального користування, необхідно буде задати вхід адміністратора використовуючи логін і пароль на самому сервері, так як сайт не має даних о користувачі. Сайт має гарний захист від злому, дотримані усі основні засоби захисту.

1.5.3. Вимоги до складу та параметрів технічних засобів

Технічні засіби користувача повинні відповідати таким вимогам:

- Процесор 2 ядра по 2ГГц;
- ОЗУ 512мб або більше;
- Будь який браузер.

Рекомендовані системні вимоги:

- Процесор 2 ядра 3ГГц;
- ОЗУ 1024мб;
- Будь який браузер.

1.5.4. Вимоги до інформаційної та програмної сумісності

У якості основної мови для розробки дипломного проекту повинна бути використана мова php, html, CSS, JavaScript. Середовище розробки обрано PhpStorm — сучасне багатофункціональне середовище розробки з підтримкою усіх вищеназваних мов.

HTML і CSS - це основа будь-якого сайту. Від їх коректності залежить якість відображення сайту в різних браузерах і на різних пристроях. Валідність семантичного HTML коду також визначає якість сайту з точки зору пошукових систем.

Html - це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

JavaScript - мультіпарадігменна мова програмування. Підтримує об'єктно-орієнтовану, імперативний і функціональний стилі. Є реалізацією стандарту ECMAScript. JavaScript зазвичай використовується як вбудований мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок.

PHP - це скриптова мова програмування, яка використовується для створення веб-сайтів та додатків. PHP код може бути вбудований в HTML-документи, що дозволяє динамічно генерувати вміст сторінок залежно від різних умов та параметрів.

Наприклад, допустимо, що ми створюємо веб-сайт для онлайн магазину, і потрібно відображати список продуктів з бази даних. Замість того, щоб вручну створювати HTML-код для кожного продукту, ми можемо використовувати PHP для отримання списку продуктів з бази даних та автоматично генерувати HTML-код для кожного елемента.

Також PHP підтримує роботу з різними базами даних, зокрема MySQL та PostgreSQL, що дозволяє зберігати та отримувати дані з веб-сайту.

Крім того, PHP має широку підтримку серед веб-хостинг провайдерів, що дозволяє запускати PHP скрипти на більшості серверів. Це робить PHP однією з найбільш популярних мов програмування для веб-розробки.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

Розроблений веб-додаток повинен бути сумісний з будь-яким клієнтом та працювати на будь-якій платформі, для автоматизації системи продажу товарів (у цьому випадку заправки для кальянів), інтуїтивно зрозумілий та зручний у використанні.

Експлуатаційне призначення веб-додатка – автоматизація та спрощення системи продажу та купівлі безтабачних заправок для кальяна. Розроблена система значно зменшує витрати часу та людських ресурсів для пошуку, аналізу та продажу існуючих товарів.

Функціональне призначення веб-додатка – надати можливість користувачеві у ролі покупця зручно переглядати існуючі товари, замовляти їх, залишати відгуки на будь-яку продукцію та надати можливість продавцю швидко та зручно додавати нову продукцію або оновлювати інформацію про вже додані товари.

2.2. Опис застосованих математичних методів

Оскільки особливості предметної області розв'язуваної задачі не передбачають застосування математичних методів, при розробці веборієнтованого додатку з продажу військової амуніції математичні методи не використовувалися.

2.3. Опис використаної архітектури та шаблонів проектування

Вибір використання фреймворка Laravel для створення веб-орієнтованого програмного забезпечення на мові програмування PHP [1] є одним із найкращих варіантів для подібних застосунків.

Laravel [2] - це потужний фреймворк, який надає зручні інструменти для швидкої розробки веб-додатків із забезпеченням чистоти коду та структурованості проекту.

Оптимальна архітектура для Laravel-проектів - це Model-View-Controller (MVC)[3]. В MVC модель даних (Model) представляє бізнес-логіку, яка включає роботу з базою даних, валідацію та маніпуляцію даними. Представлення (View) відповідає за відображення даних користувачу і зазвичай містить шаблони, які використовуються для генерації HTML-сторінок. Контролер (Controller) взаємодіє з моделлю та представленням, обробляючи запити користувачів, виконуючи необхідну логіку і повертаючи відповіді.

Шаблони проектування [4] також грають важливу роль у розробці програмного забезпечення. Декілька шаблонів, які часто використовуються в Laravel-проектах, включають:

- шаблон роутингу (Route pattern): Laravel надає потужний маршрутизатор, який дозволяє визначати шаблони URL-адрес, які відповідають певним діям у контролерах. Використання шаблонів роутингу спрощує налаштування маршрутів і забезпечує зручний спосіб керування маршрутизацією;
- шаблон моделей (Model pattern): Laravel використовує шаблон активного запису (Active Record) для взаємодії з базою даних. Кожна модель представляє окрему таблицю бази даних та надає методи для виконання операцій з даними, таких як створення, зчитування, оновлення та видалення записів. Шаблон моделей дозволяє зручно працювати з даними та забезпечує легкість їх валідації та маніпуляції;

- шаблон фасаду (Facade pattern): Фасади в Laravel надають простий інтерфейс для взаємодії з різними складними підсистемами фреймворку. Вони дозволяють зменшити складність коду та забезпечують зручний доступ до функціональності Laravel, такої як робота з сесіями, кешуванням, електронною поштою та іншими;
- шаблон патернів (Template pattern): Laravel використовує шаблон шаблонів Blade для розробки візуальної частини веб-додатків. Blade надає потужні можливості, такі як умовні оператори, цикли, наслідування шаблонів та компоненти, що дозволяють створювати розширювані та повторно використовувані шаблони.

Ці шаблони проектування та архітектура MVC [5] спільно допомагають забезпечити чистоту, структурованість та розширюваність вашого у проекту на Laravel.

Також, Laravel [6] надає декілька вбудованих функцій та механізмів для забезпечення безпеки вашого веб-застосунку. Ось кілька важливих аспектів безпеки, які Laravel дозволяє реалізувати:

- захист від атак на міжсайтове підкорення (Cross-Site Scripting, XSS)[7]: Laravel автоматично застосовує безпечне екранування до вхідних даних, що дозволяє запобігти XSS-атакам. Це означає, що ви можете відображати вхідні дані безпечно, не ризикуючи виконання зловісного JavaScript-коду;
- захист від атак на впровадження коду (Code Injection)[7]: Laravel забезпечує захист від атак на впровадження коду шляхом використання параметризованих запитів та підготовлених виразів для роботи з базою даних. Це дозволяє уникнути проблем, пов'язаних зі впровадженням шкідливого коду у запитах до бази даних;
- захист від фальшивих маршрутів (CSRF)[7]: Laravel автоматично генерує та перевіряє маркери CSRF (Cross-Site Request Forgery), які захищають ваш додаток від атак CSRF. Це дозволяє переконатись, що

лише валідні та схвалені користувачем запити можуть виконувати дії зміни стану на вашому веб-застосунку;

- автентифікація та авторизація: Laravel має вбудовану систему автентифікації, яка дозволяє зручно реалізувати процес входу в систему та ідентифікацію користувачів. Ви також можете використовувати вбудовану систему авторизації для контролю доступу користувачів до різних частин вашого додатку;
- хешування паролів: Laravel надає можливість хешувати паролі користувачів, що забезпечує їх безпеку у випадку витоку даних. Laravel використовує сучасні алгоритми хешування, такі як bcrypt, для забезпечення безпеки паролів;
- захист файлової системи: Laravel забезпечує безпечну збереження завантажених файлів, контролюючи доступ до них через механізми авторизації. Ви можете встановити обмеження на типи файлів та розміри завантажуваних файлів, щоб забезпечити безпеку вашого додатку;
- захист від SQL-ін'єкцій: Laravel використовує підготовлені запити та механізми екранування, що дозволяють уникнути проблем з SQL-ін'єкціями. Це допомагає забезпечити безпеку ваших запитів до бази даних.

Ці інструменти та механізми, надані Laravel, допомагають забезпечити безпеку вашого веб-застосунку та захистити його від різних видів атак[8]. Однак, варто пам'ятати, що безпека - це постійний процес, і вам слід також враховувати й інші аспекти безпеки, такі як перевірка валідності даних, захист від витоку інформації та інші аспекти, які можуть бути специфічні для вашого веб-застосунку.

2.4. Опис використаних технологій та мов програмування

В рамках кваліфікаційної роботи про створення веб-орієнтованого програмного забезпечення з використанням фреймворка Laravel на мові програмування PHP, основні технології та мови програмування, які використовувались, включають наступні:

- PHP (Hypertext Preprocessor) [9] - це мова програмування загального призначення, яка спеціально створена для розробки веб-додатків. Вона має широку популярність у розробці веб-сайтів і забезпечує потужні можливості для роботи з базами даних, обробки форм, роботи з файлами та іншими веб-задачами. Laravel побудований на мові PHP і використовує її як основу для розробки веб-додатків;
- Laravel: Laravel [10] - це високопродуктивний веб-фреймворк на основі мови PHP, який забезпечує зручні інструменти для швидкої розробки веб-додатків. Він пропонує широкий набір готових компонентів, які спрощують роботу з маршрутизацією, сесіями, аутентифікацією, базами даних, шаблонами та іншими аспектами веб-розробки. Laravel також підтримує модель розробки MVC, що сприяє структурованості та розширюваності проекту;
- HTML (Hypertext Markup Language) [11] та CSS (Cascading Style Sheets) [12] - це основні мови для створення та оформлення веб-сторінок. HTML використовується для створення структури веб-сторінки, визначення заголовків, текстів, таблиць, форм і т. д. CSS використовується для задання стилів та вигляду елементів HTML. Laravel співпрацює з HTML/CSS, дозволяючи генерувати динамічний HTML з використанням шаблонів та компонентів;
- JavaScript [13] - це мова програмування, яка використовується для реалізації інтерактивності на веб-сторінках. Вона надає можливості для маніпуляції елементами сторінки, обробки подій, взаємодії з сервером та іншими динамічними функціями. Laravel дозволяє включати JavaScript-скрипти в ваші веб-додатки та взаємодіяти з ними для покращення користувацького досвіду;

- Бази даних [14]. Laravel підтримує різні системи управління базами даних (СУБД), включаючи MySQL [15], PostgreSQL, SQLite та інші. Ви можете використовувати мову запитів Eloquent ORM для зручної роботи з базами даних в Laravel;
- Git [16] - це розподілена система керування версіями, яка дозволяє вам ефективно керувати та відстежувати зміни у вашому програмному коді. Використання Git допомагає вам зберігати версії вашого проекту, співпрацювати з іншими розробниками та забезпечувати стабільність вашого коду.

Ці технології та мови програмування використовуються в комбінації для розробки веб-орієнтованого програмного забезпечення з використанням фреймворка Laravel на мові програмування PHP. Вони надають потужні можливості для створення функціональних та безпечних веб-додатків.

2.5. Опис структури програми та алгоритмів її функціонування

Структура програми, розробленої з використанням фреймворка Laravel, базується на моделі розробки MVC (Model-View-Controller) (рис. 2.1).

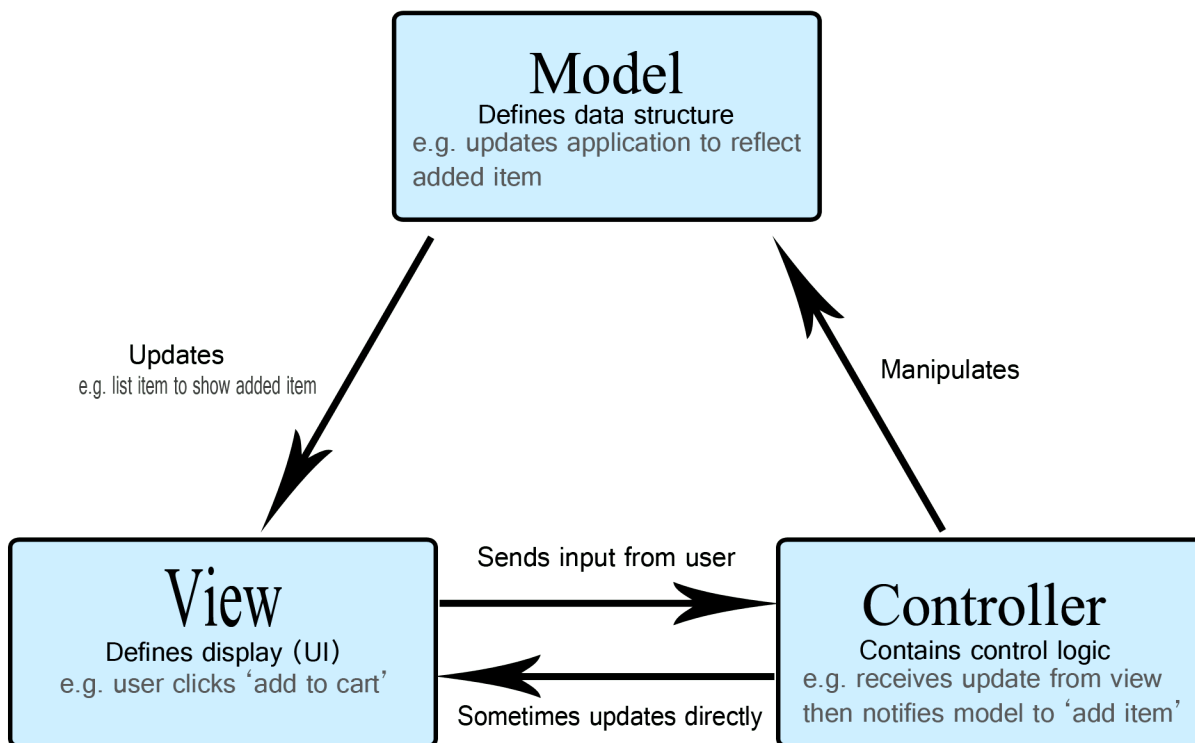


Рис. 2.1. Схема моделі розробки MVC

Ця модель дозволяє розділити логіку програми, представлення даних та обробку запитів на окремі компоненти. Основні компоненти структури програми включають моделі, контролери, маршрути та представлення. Давайте розглянемо кожен з цих компонентів більш детально:

- моделі (Models) в Laravel представляють бізнес-логіку та взаємодію з базою даних. Вони відповідають за зберігання та отримання даних, виконання запитів до бази даних і валідацію даних. Ви можете визначити моделі для різних сутностей вашої програми, таких як користувачі, продукти, замовлення тощо. Кожна модель зазвичай відповідає таблиці в базі даних і надає методи для виконання операцій над цими даними;
- контролери (Controllers) відповідають за обробку запитів користувача та взаємодію з моделями та представленнями. Вони отримують запити

від користувача, виконують необхідні дії та повертають відповіді. Контролери визначають методи, які відповідають різним діям, наприклад, отримання списку користувачів, створення нового продукту тощо;

- маршрути (Routes) визначають, які URL-шляхи будуть співпадати з конкретними методами контролерів. Вони визначають, які дії виконуватимуться при обробці запитів на певні URL-шляхи. Маршрути можуть бути визначені в файлі `routes/web.php` для веб-маршрутів або в `routes/api.php` для API-маршрутів. Вони також можуть мати параметри, які передаються до методів контролерів;
- представлення (Views) відповідають за відображення даних користувачу. Вони використовуються для відображення HTML-сторінок або шаблонів, які містять дані з моделей. Laravel використовує шаблонізатор `Blade` для створення динамічних та перевикористовуваних представлень;

Алгоритм функціонування програми зазвичай включає такі етапи:

- 1) Користувач робить запит до веб-додатку шляхом відкриття веб-сторінки або відправки форми.
- 2) Запит перенаправляється до відповідного маршруту, визначеного у файлі маршрутів.
- 3) Маршрут викликає відповідний метод контролера, який обробляє запит та взаємодіє з моделями для отримання необхідних даних.
- 4) Контролер використовує представлення для відображення даних і генерації HTML-сторінки або відповіді API [18].
- 5) Згенерована відповідь передається назад до клієнта, який бачить результат на веб-сторінці або отримує відповідь API у вигляді JSON або іншого формату даних.

Цей алгоритм повторюється при кожному новому запиті до програми, що дозволяє веб-додатку реагувати на дії користувача та надавати йому актуальну інформацію.

2.6. Обґрунтування та організація вхідних та вихідних даних програми

Обґрунтування та організація вхідних та вихідних даних є важливою частиною розробки програмного забезпечення, включаючи веб-орієнтовані додатки на базі фреймворка Laravel.

Вхідні дані - це дані, які ваша програма отримує від користувача або з інших джерел, тоді як вихідні дані - це дані, які ваша програма генерує або надсилає користувачу чи іншим системам.

Обґрунтування вхідних даних:

- Веб-застосунок може містити HTML-форми, які дозволяють користувачам вводити дані. Можливо використовувати елементи форм, такі як текстові поля, прапорці, списки, кнопки тощо, для отримання різних типів даних від користувача;
- Вхідні дані також можуть бути передані через URL-параметри або параметри запиту. Можливо визначати маршрути в додатку, які приймають певні параметри, наприклад, ідентифікатори ресурсів чи додаткові параметри для фільтрації даних;
- Якщо веб-застосунок має систему автентифікації, то вхідні дані можуть включати дані, які ідентифікують користувача, такі як ім'я користувача та пароль. Функціонал авторизації Laravel використовується для перевірки та обробки цих даних.

Обґрунтування вихідних даних:

- представлення (View). Вихідні дані програми можуть бути відображені у вигляді HTML-сторінок, які користувач може переглядати у своєму веб-браузері. У Laravel використовується шаблонізатор Blade для генерації HTML-коду, який включає дані з програми;
- JSON та інші формати даних. Частина даних генерується у форматі JSON [18] або інших форматах, які можуть бути використані для обміну даними з іншими системами чи API. Laravel надає зручні інструменти для серіалізації даних у різних форматах;
- системні повідомлення використовуються, щоб повідомляти користувачів про певні події або статуси виконання операцій. Laravel має вбудовану підтримку для таких повідомлень через механізм сесій.

Важливо забезпечити безпеку та валідацію вхідних даних для запобігання атакам та некоректному виконанню програми. Laravel надає багато інструментів для перевірки та фільтрації вхідних даних, включаючи вбудовані правила валідації та захист від хрестоматричних сайтів (Cross-Site Scripting, XSS) та інших загроз безпеки.

2.7. Опис розробленого програмного продукту

2.7.1. Використані технічні засоби

Результатом виконання кваліфікаційної роботи є веб-сайт, тому для його завантаження потрібен лише браузер, що дозволяє користуватися їм за допомогою комп'ютера, планшета, телефона та будь-якого іншого пристрою, що має доступ до інтернету та браузер.

Характеристики комп'ютера, на якому розроблявся додаток:

- монітор 21.8'' (1920x1080) Full HD;
- процесор Intel Core i3-9100F 3.60 Ghz;
- RAM: 16 GB;
- накопичувачі: HDD 1TB, SSD 512GB;

- Відеокарта NVIDIA GeForce GTX 1050TI.

Розміщення сайту організовано на зовнішніх серверах веб-хостингу Ukraine.com.ua із наступними характеристиками:

- PHP 8;
- RAM 1536 MB;
- Місце на диску: 30 GB;
- База даних: MySQL;
- Корпоративна пошта: наявна.

2.7.2. Використані програмні засоби

Для розробки даного кваліфікаційного проекту були використані наступні програмні засоби:

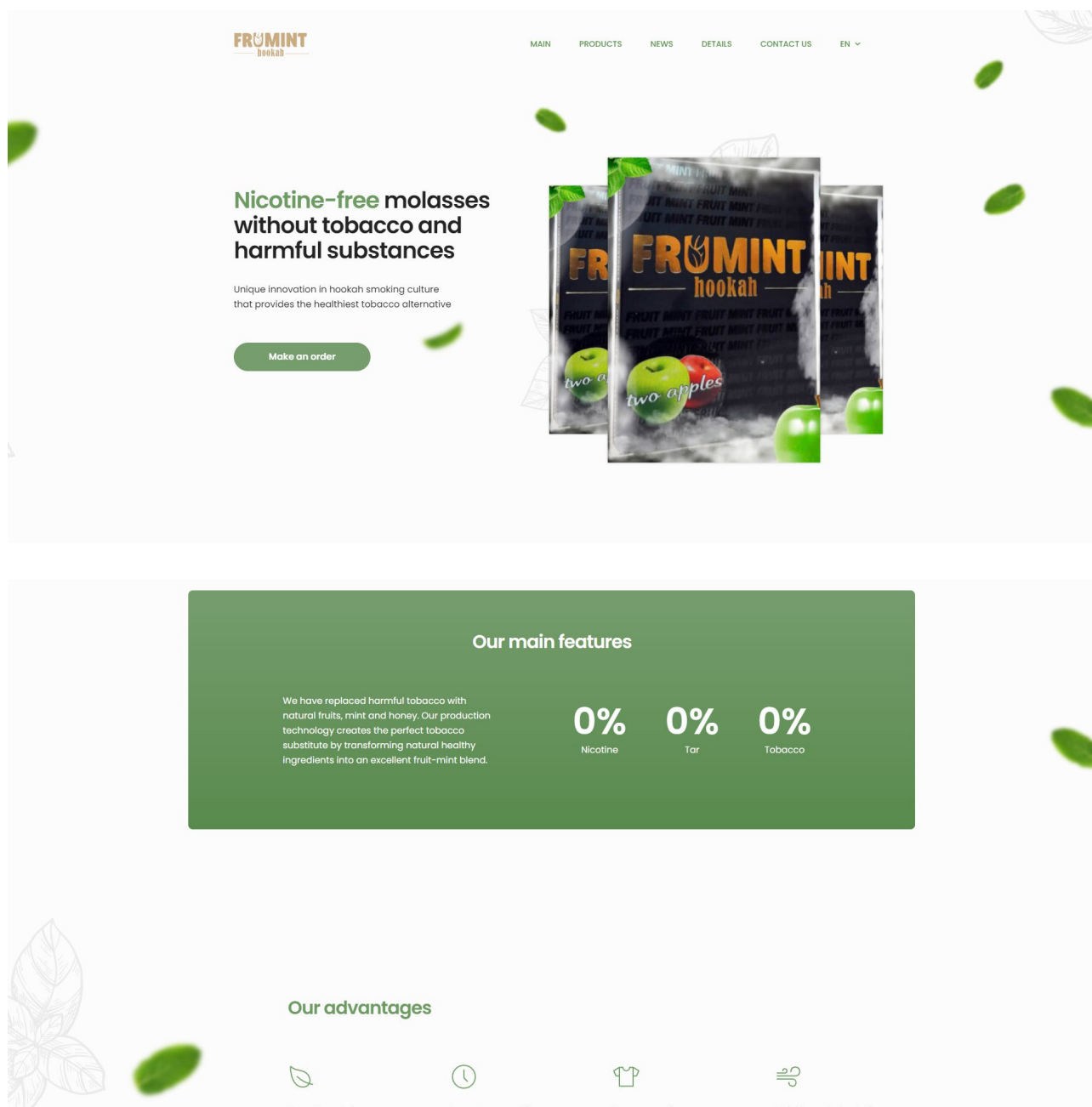
- Open Server – локальний веб сервер для Windows;
- PHP – мова програмування;
- NPM [19] – менеджер пакетів;
- Composer [20] – пакетний менеджер програмного рівня для мови програмування PHP;
- JetBrains PhpStorm IDE – професійна середа розробки з підтримкою PHP, JavaScript, HTML, CSS та великою кількістю додаткових плагінів;
- Git – система контролю версій.

2.7.3. Виклик та завантаження програми

Для роботи з розробленим веб-додатком потрібно відкрити браузер та перейти за посиланням <https://www.frumint.com>.

2.7.4 Опис інтерфейсу користувача. Основна частина

Головна сторінка – звичайна інформаційна сторінка з короткою відомістю про продукцію (рис. 2.2).



Our advantages



Doesn't contain harmful substances



Long-term smoking without loss of taste



Leaves no odor on clothes, skin and hair



Quickly vented out of the room



Does not cause addiction



Enormous, sweet and fragrant smoke



Convenient packaging



Less charcoal is used for smoking



How to use?

- 1 Fluff the hookah mix with your hands



How to use?

- 1 Fluff the hookah mix with your hands
- 2 Freely put in any bowl
- 3 Level the top layer of the mixture
- 4 Maintain a small margin from the edges

Compound

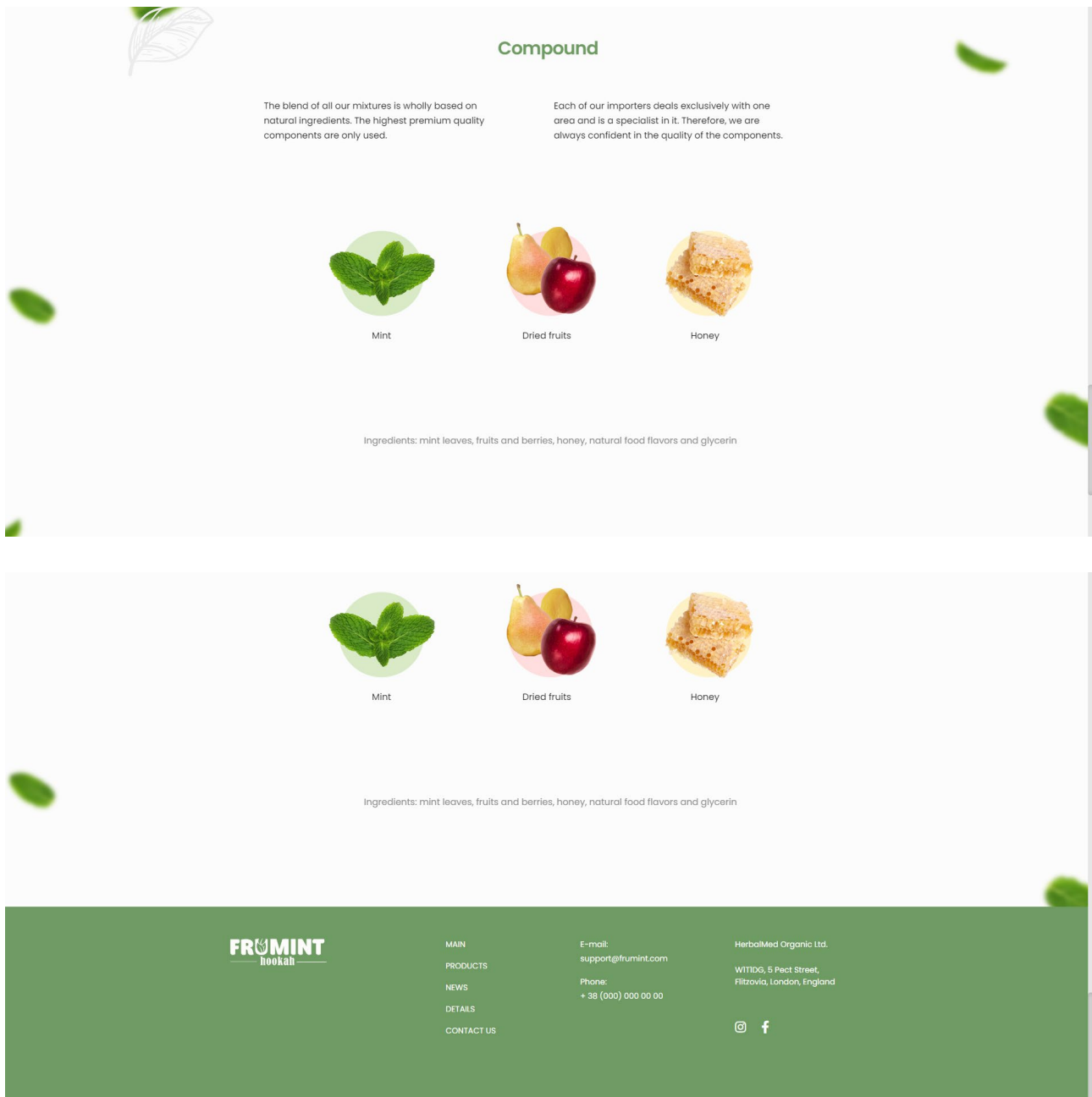


Рис. 2.2. Демонстрація головної сторінки

Сторінка списку продукції – сторінка з каталогом товару. У кожного товару відображається головна картинка, назва, ціна та середній рейтинг товару. Реалізований посторінковий вивід каталогу. На кожній сторінці виводиться по 8 екземплярів товару (рис. 2.3, рис. 2.4).

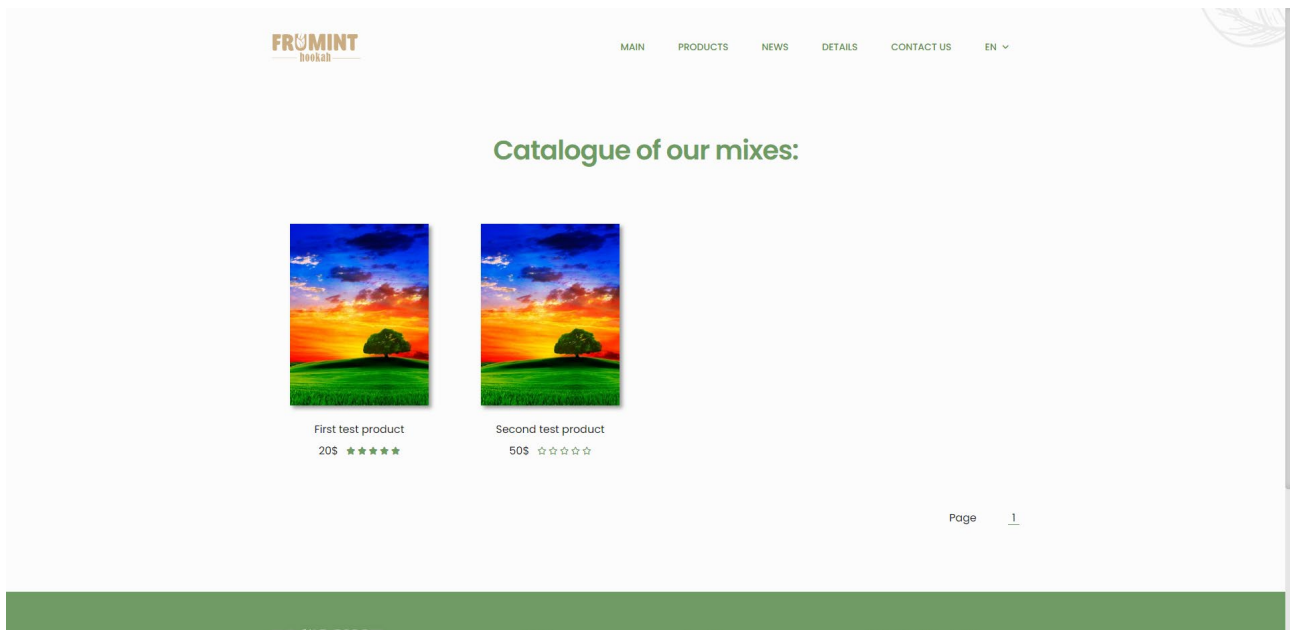


Рис. 2.3. Демонстрація сторінки зі списком продукції (без ховеру)

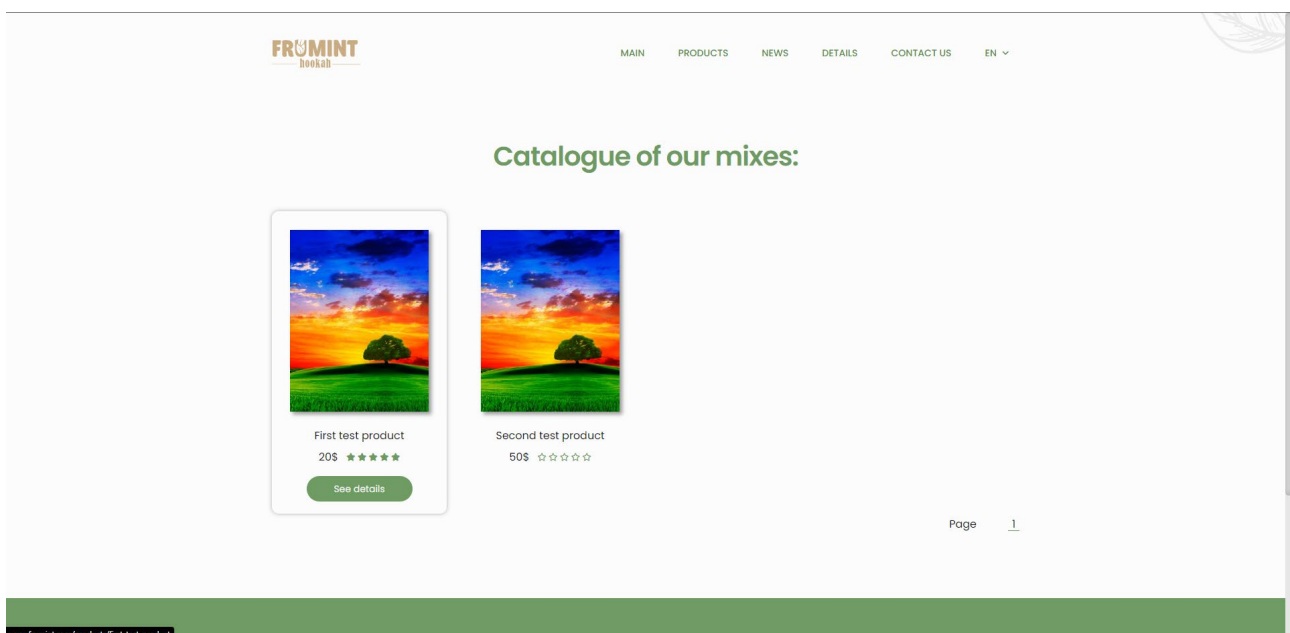


Рис. 2.4. Демонстрація сторінки зі списком продукції (з наведенням з один із товарів)

Сторінка продукта – сторінка з повною інформацією про товар: назва, ціна, рейтинг, опис, склад, кнопка замовлення (рис. 2.5) та повний список усіх ревію з постраничним виводом (рис. 2.7). Реалізован функціонал оцінки продукції: оцінка від 1 до 5 та сам текст з можливістю зворотнього зв'язку (рис. 2.6).

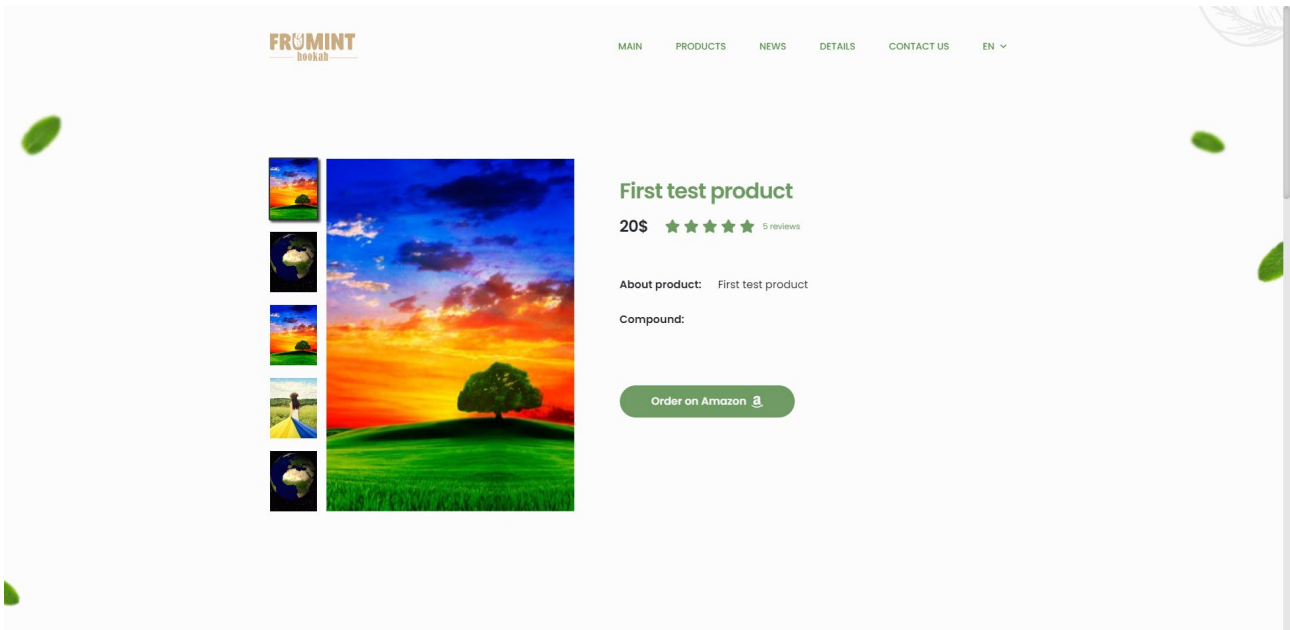


Рис. 2.5. Демонстрація сторінки продукту (галерея картинок та інформація про товар)

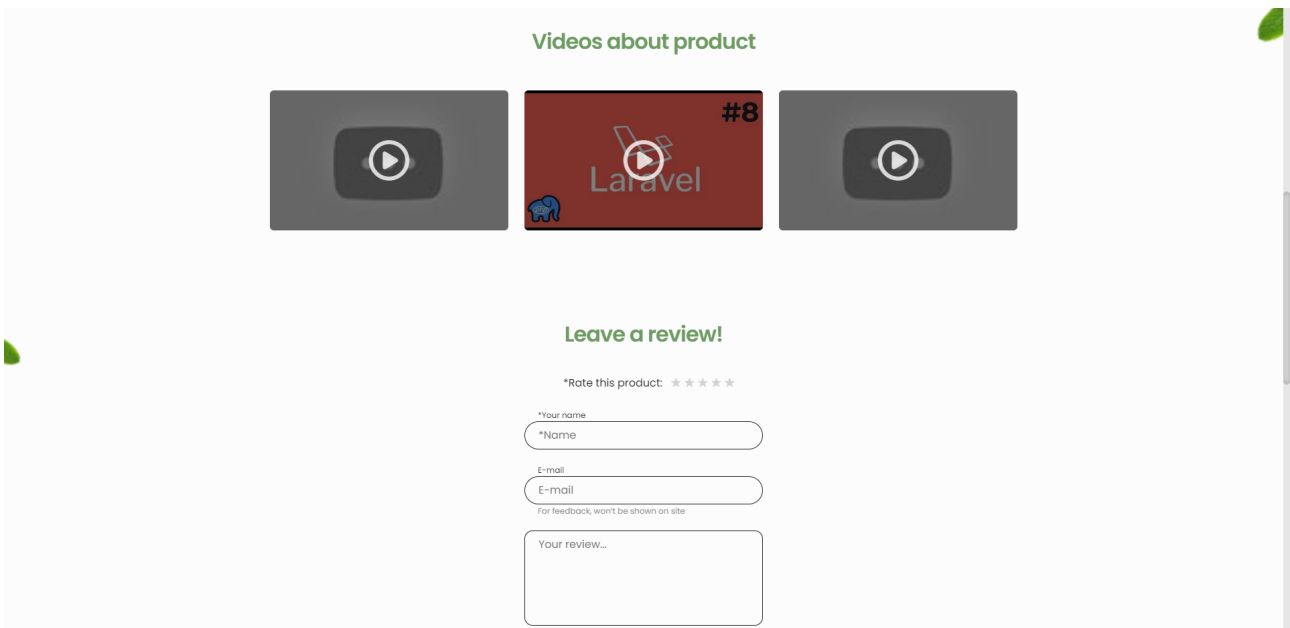


Рис. 2.6. Демонстрація сторінки продукту (відео про товар та форма для відгуків)

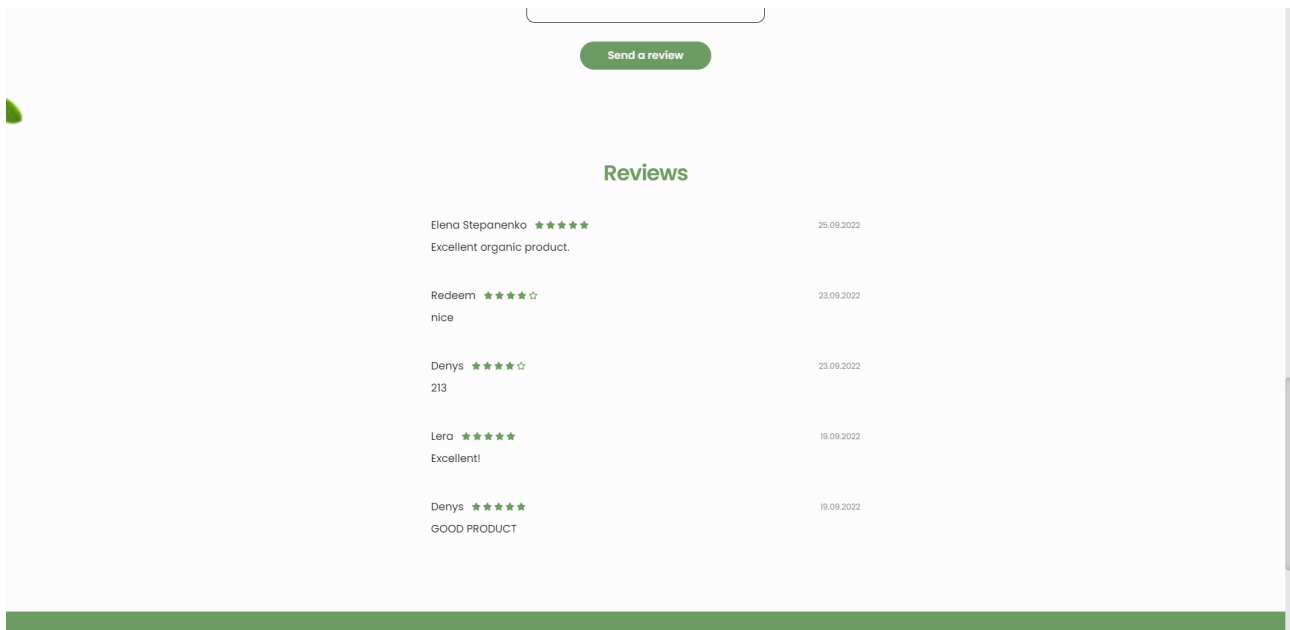


Рис. 2.7. Демонстрація сторінки продукта (список відгуків)

Сторінка списку новин – сторінка зі списком усіх новин у вигляді постраничного виводу. Спочатку виводяться новини, що були додані пізніше. У кожній новини виводиться головна картинка, назва, короткий опис та дата додавання новини. На кожній сторінці виводиться по 6 екземплярів новин (рис. 2.8 та рис. 2.9).

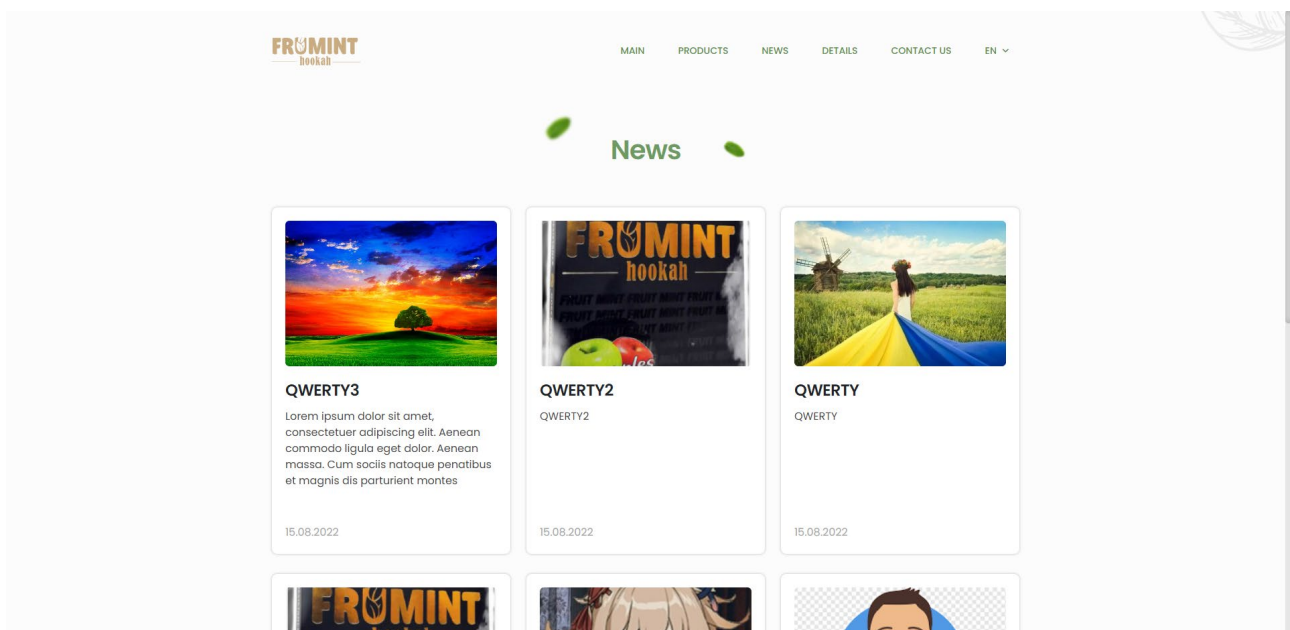


Рис. 2.8. Демонстрація сторінки зі списком новин (1 частина)

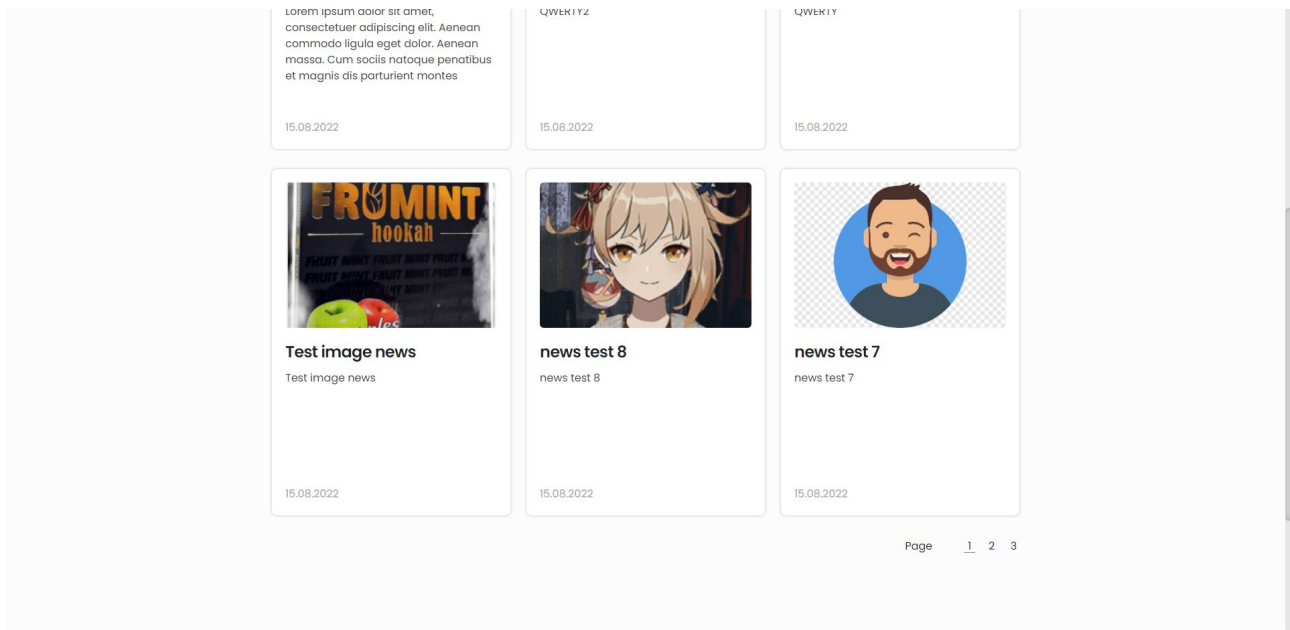


Рис. 2.9. Демонстрація сторінки зі списком новин (2 частина)

Сторінка новини – сторінка з конкретною новиною (рис. 2.10).

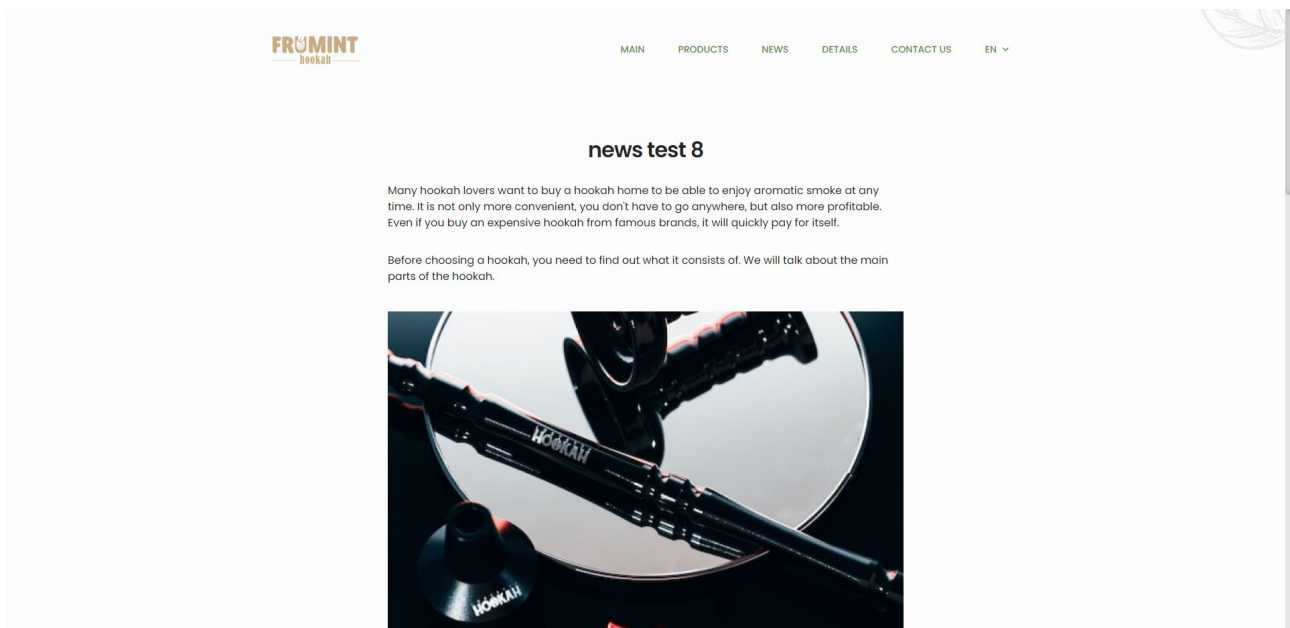


Рис. 2.10. Демонстрація сторінки новини

Сторінка деталей – сторінка з додатковою інформацією про продукцію (рис 2.11).

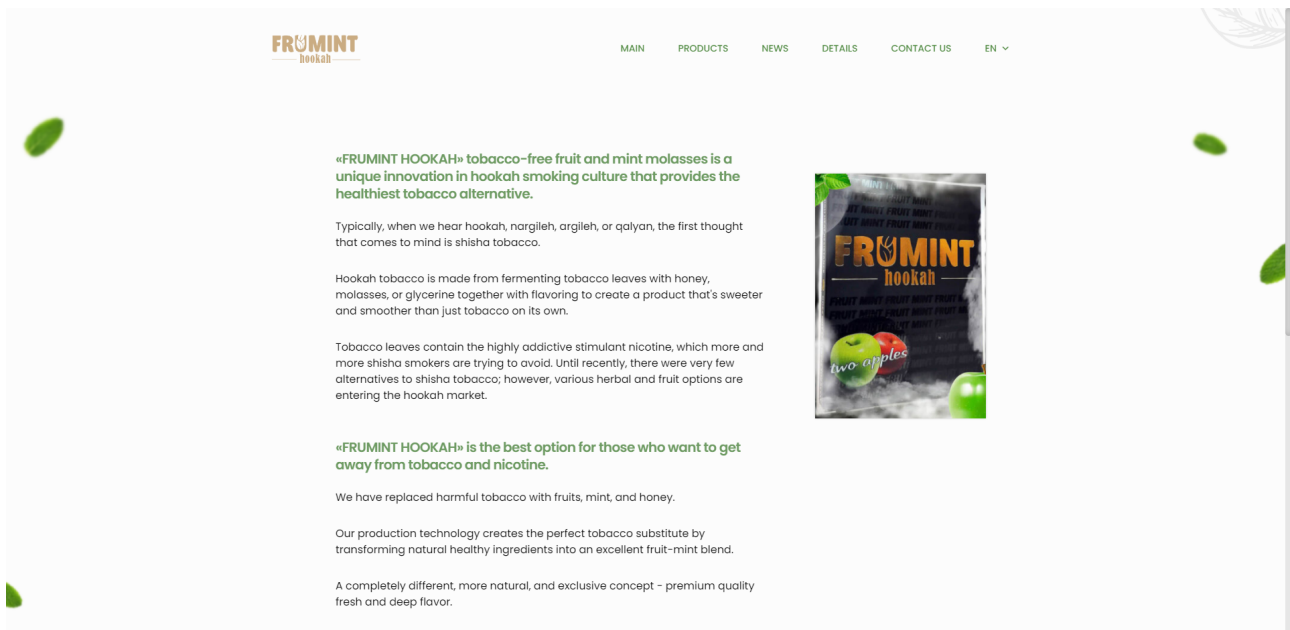


Рис. 2.11. Демонстрація сторінки деталей

Сторінка для зворотного зв'язку – сторінка з формою для зворотного зв'язку, повідомлення з якою приходить на спеціальну пошту, що належить фірмі (рис 2.12).

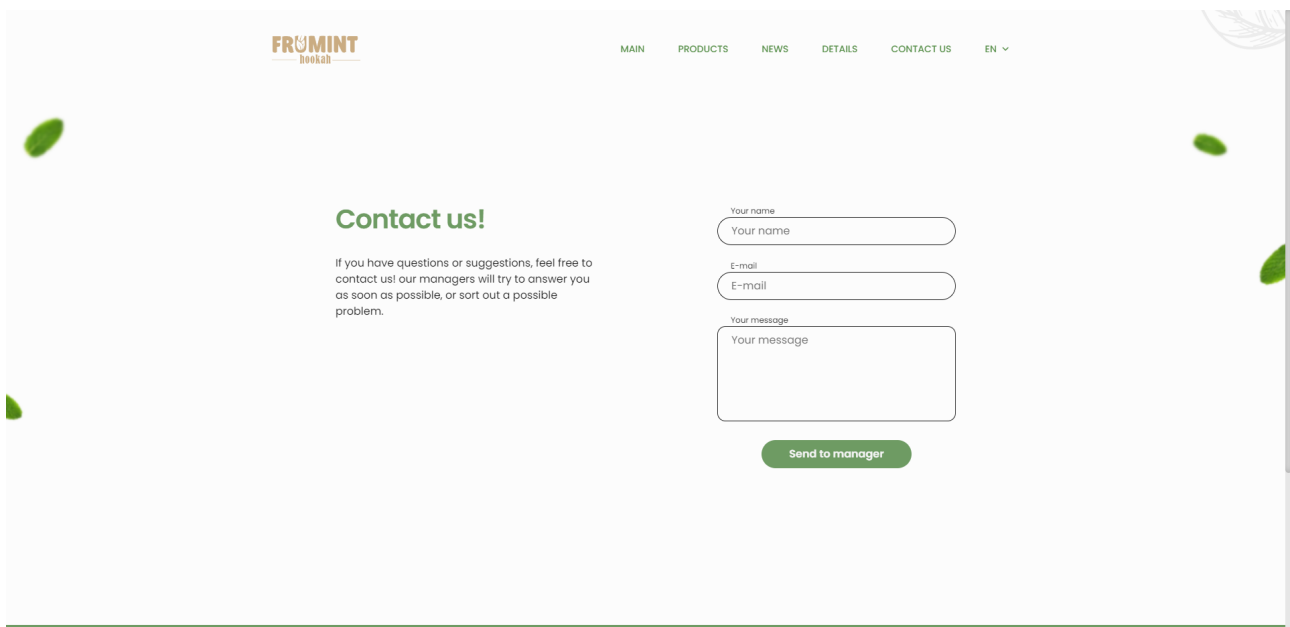


Рис. 2.12. Демонстрація сторінки для зворотного зв'язку

Можливість зміни мови на сайті. Сайт підтримує дві мови: англійську та французьку. По замовчуванню встановлюється мова на основі основною мови, вибраної у браузері. Якщо у браузері вибрана французька – по замовчуванню на сайті буде французька, якщо будь-яка інша мова – англійська.

Адміністративна частина:

Сторінка для авторизації у вигляді адміністратора (рис. 2.13).

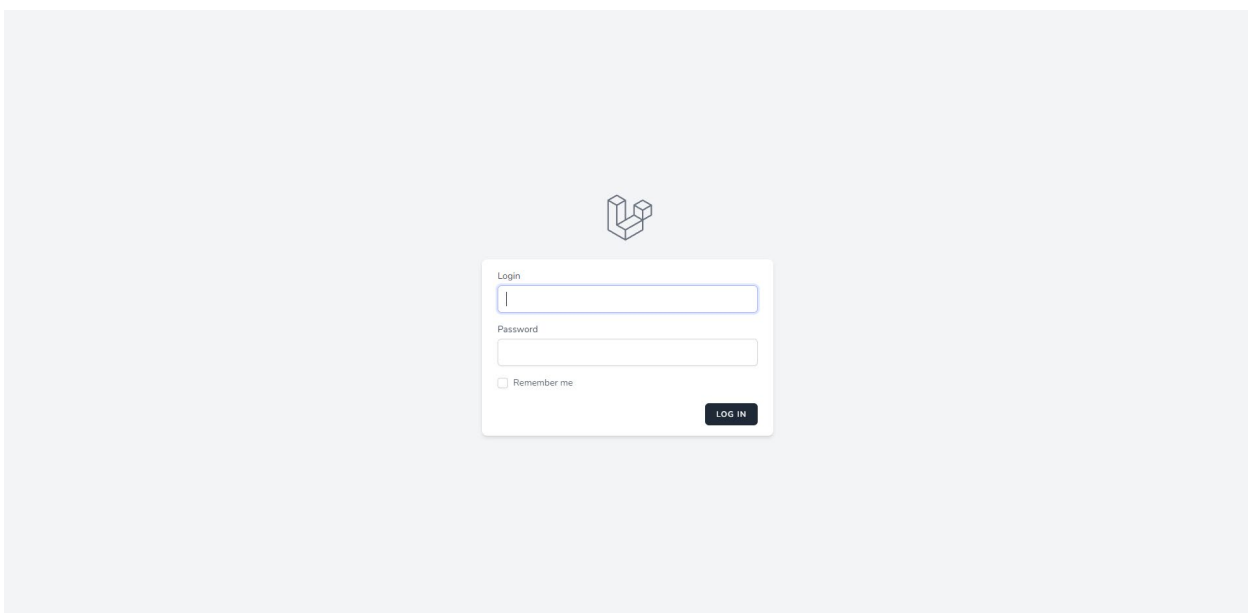


Рис. 2.13. Демонстрація сторінки авторизації

Сторінка зі списком новин (рис. 2.14) – можливість короткого огляду списку усіх новин (номер новини, назва, короткий опис, картинка) та кнопка переходу до редагування конкретної новини.

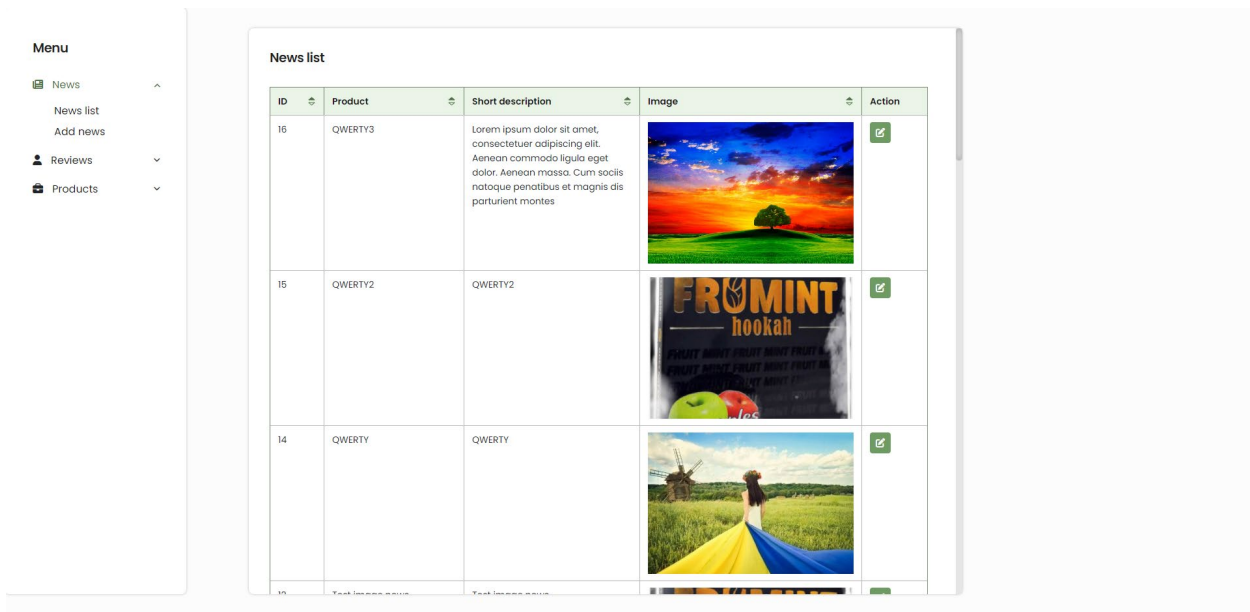


Рис. 2.14. Панель Адміністратора. Демонстрація сторінки зі списком новин

Сторінка редагування новини (рис. 2.15) – можливість зміни будь-якого з полів новини (назва, опис, текст) на двох мовах та картинки.

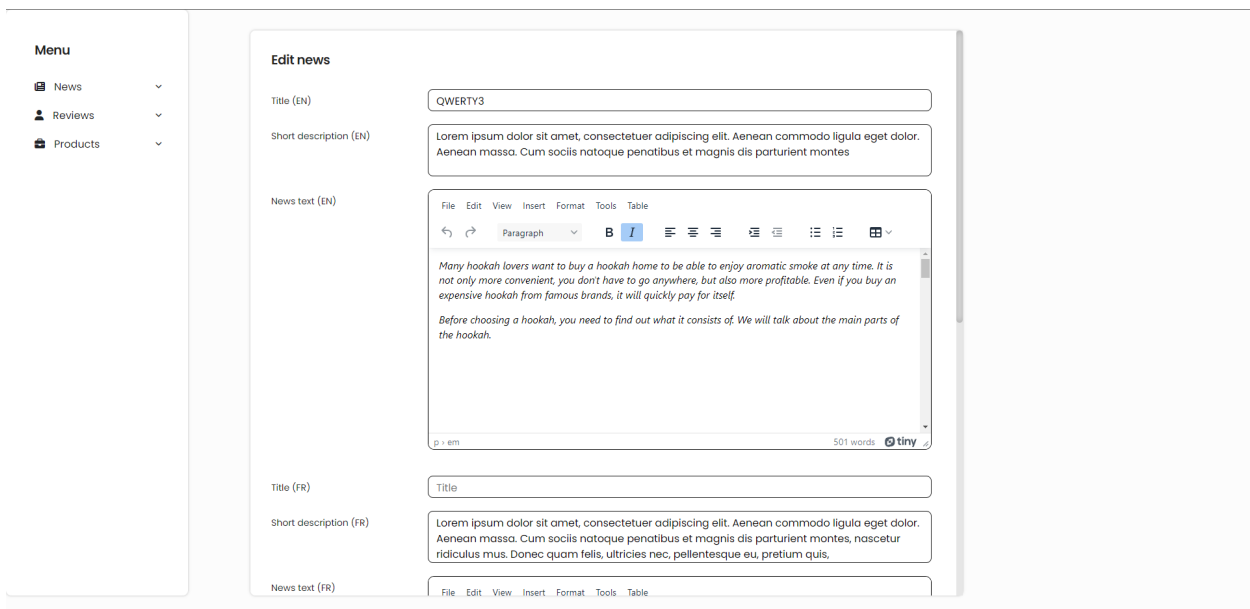


Рис. 2.15. Панель Адміністратора. Демонстрація сторінки редагування новини

Сторінка додавання новини (рис. 2.16) – сторінка для створення нового екземпляра новини – назва, опис, текст на двох мовах та картинка.

Add news

Title (EN)

Short description (EN)

News text (EN)

File Edit View Insert Format Tools Table

← → Paragraph **B** *I* [Text color] [Background color] [Bulleted list] [Numbered list] [Link] [Unlink] [Image]

p 0 words tiny

Title (FR)

Short description (FR)

News text (FR)

File Edit View Insert Format Tools Table

Рис. 2.16. Панель Адміністратора. Демонстрація сторінки із додаванням новини

Сторінка для модерування останніх відгуків на продукції (можливість підтвердження або видалення не опублікованих відгуків) (рис. 2.17).

Reviews moderation

Product	Rate	Name	E-mail	Review text	Date	Action
---------	------	------	--------	-------------	------	--------

Рис. 2.17. Панель Адміністратора. Демонстрація сторінки модерування відгуків

Сторінка всіх підтверджених відгуків на усі екземпляри продукції з можливістю сортування на основі продукції або оцінки. Та можливість видалення відгука (рис. 2.18).

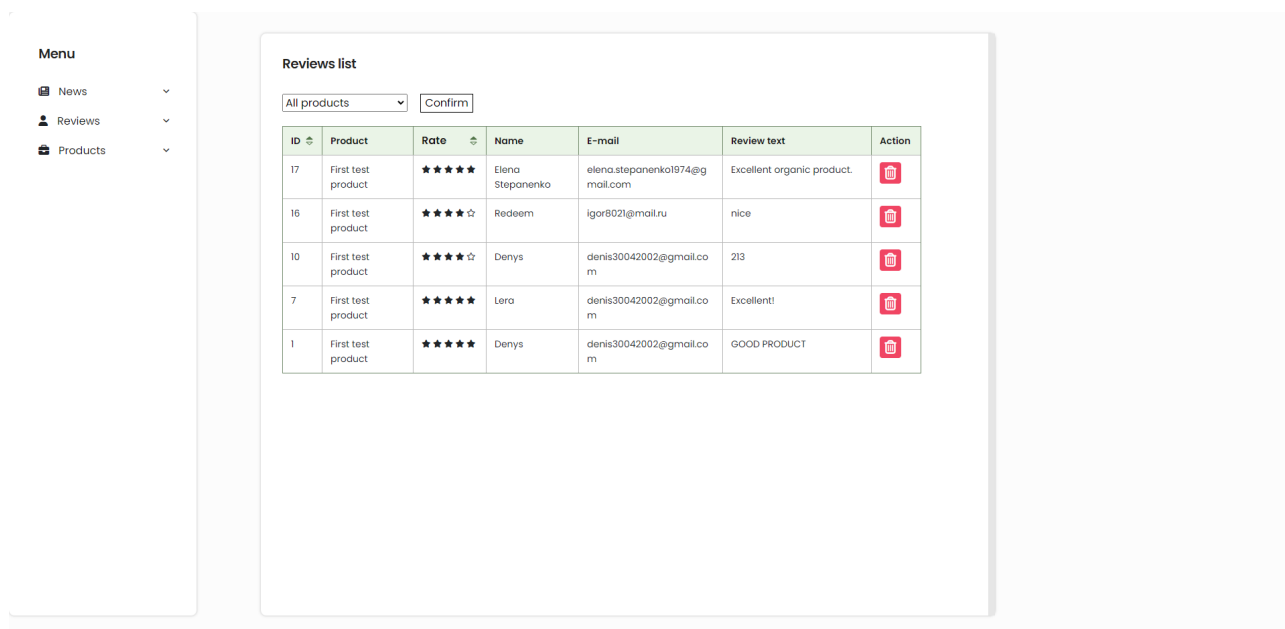


Рис. 2.18. Панель Адміністратора. Демонстрація сторінки відгуків

Сторінка зі списком продукції (рис. 2.19) – можливість огляду списку усіх продуктів (номер, назва, ціна, склад), сортування по усім полям та наявна кнопка для переходу до редагування продукту.

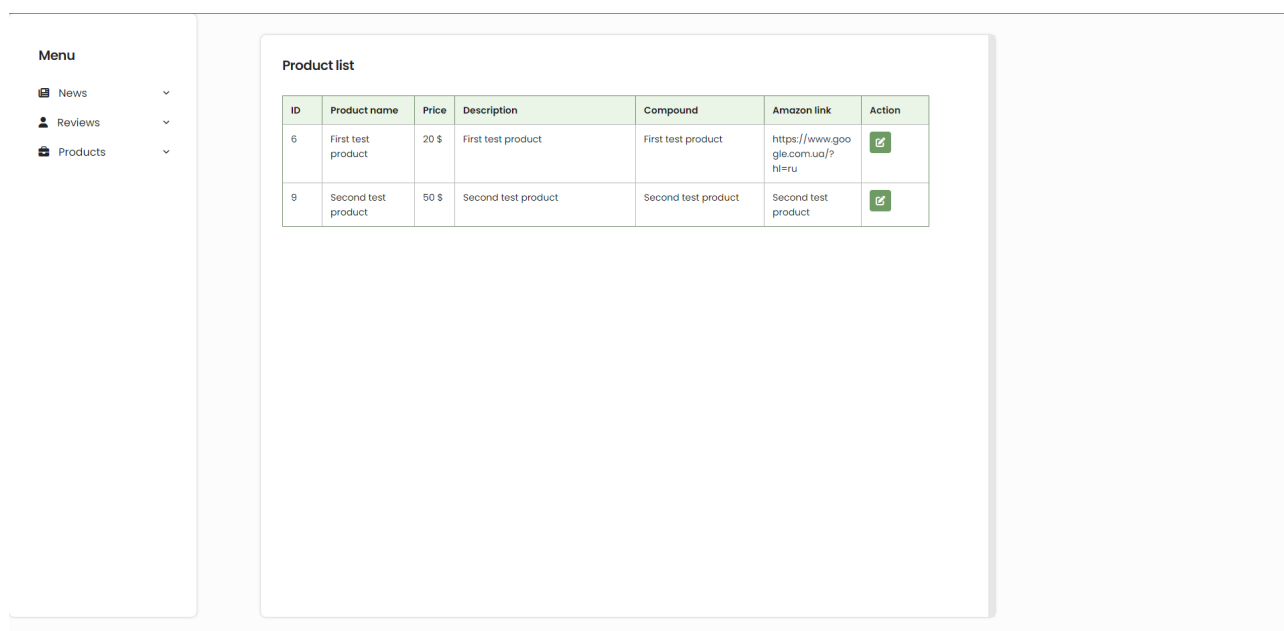


Рис. 2.19. Панель Адміністратора. Демонстрація зі списком продукції

Сторінка редагування продукції (рис. 2.20) – можливість зміни будь-якого з полів продукції та додавання нових картинок.

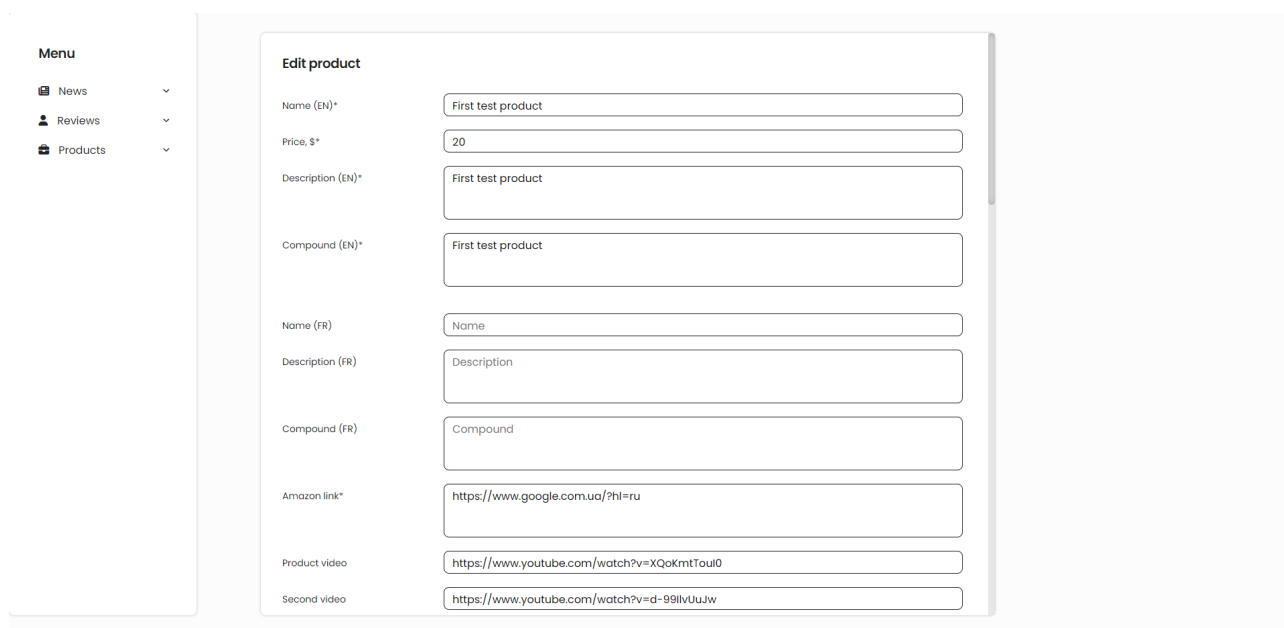


Рис. 2.20. Панель Адміністратора. Демонстрація сторінки редагування продукції

Сторінка додавання продукції – сторінка для створення нового екземпляра продукції (рис. 2.21).

The image shows a screenshot of an administrator panel with a sidebar menu and a main content area. The sidebar menu is titled "Menu" and contains three items: "News", "Reviews", and "Products", each with a downward arrow. The main content area is titled "Add product" and contains a form with the following fields:

- Name (EN): Input field with placeholder "Name"
- Price, \$: Input field with placeholder "Price"
- Description (EN): Input field with placeholder "Description"
- Compound (EN): Input field with placeholder "Compound"
- Name (FR): Input field with placeholder "Name"
- Description (FR): Input field with placeholder "Description"
- Compound (FR): Input field with placeholder "Compound"
- Amazon link: Input field with placeholder "Amazon link"
- Product video: Input field with placeholder "https://www.youtube.com/watch?v=..."
- Second video: Input field with placeholder "https://www.youtube.com/watch?v=..."

Рис. 2.21. Панель Адміністратора. Демонстрація сторінки додавання продукції

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Визначення трудомісткості розробки програмного забезпечення

Початкові дані:

1. передбачуване число операторів програми – 2207;
2. коефіцієнт складності програми – 1,8;
3. коефіцієнт корекції програми в ході її розробки – 0,3;
4. годинна заробітна плата програміста – 176 грн/год;

За даними найбільшої ІТ-спільноти України DOU [21] початок 2023 року, в середньому зарплата Junior Backend PHP Developer в Україні становить 800\$ за місяць. У гривнях на сьогоднішній день це еквівалентно 29600 грн (курс 1 USD = 37 UAH). Якщо враховувати, що більшість програмістів працює по 8 годин в день та має приблизно 21 робочий день (у середньому). Можна підрахувати, що погодинна ставка становить приблизно $29600 / (8 * 21) = 176$ грн/год.

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,3;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,3;
7. вартість машино-години ЕОМ – 1,18 грн/год.

Завдяки «Калькулятору живлення» [22] встановлюємо, що ПК споживає приблизно 180 Вт за годину. Також, за даними виробника, монітор споживає 20 Вт за годину.

За даними постачальнику електроенергії Yasnо [23], тариф для населення на момент виконання кваліфікаційної роботи становив 1,44 грн/кВт*год. Отже, вартість електроенергії для ПК та монітору становить $(180 + 20) * 1,44 / 1000 = 0,29$ грн/год. Домашнє інтернет-підключення від «Triolan» коштує 150 грн/міс. Таким чином, загальна вартість машино-години ЕОМ виходить $0,29 * 8 * 21 +$

150 = 198,72 грн/місяць або 1,18 грн/год за умови, що інтернет використовувався лише для роботи над кваліфікаційною роботою.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,} \quad (3.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

t_{oml} - витрати праці на налагодження програми на ЕОМ;

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де q - передбачуване число операторів (2207);

C - коефіцієнт складності програми (1,8);

p - коефіцієнт корекції програми в ході її розробки (0,3).

Отже, умовне число операторів в програмі має значення:

$$Q = 2207 \cdot 1,8 \cdot (1 + 0,3) = 5164,38$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин} \quad (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи близько 3 років він складає 1,3.

Будемо вважати збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1,2$). З урахуванням коефіцієнта кваліфікації $k = 1,3$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (5164,38 \cdot 1,2) / (85 \cdot 1,3) = 56,08 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин} \quad (3.4)$$

де Q – умовне число операторів програми; (2207)

k – коефіцієнт кваліфікації програміста. (1,3)

Підставивши відповідні значення в формулу (3.4), отримуємо:

$$t_a = 5164,38 / (24 \cdot 1,3) = 165,51 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин} \quad (3.5)$$

Підставимо значення у формулу (3.5) та отримаємо витрати на складання програми по готовій блок-схемі:

$$t_n = 5164,38 / (25 \cdot 1,3) = 158,89 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин} \quad (3.6)$$

$$t_{oml} = 5164,38 / (5 \cdot 1,3) = 794,52 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин} \quad (3.7)$$

$$t_{oml}^k = 1,5 \cdot 794,52 = 1191,48 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{dp} + t_{do}, \text{ людино-годин} \quad (3.8)$$

де t_{dp} -трудомісткість підготовки матеріалів і рукопису:

$$t_{dp} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин} \quad (3.9)$$

t_{do} - трудомісткість редагування, печатки й оформлення документації:

$$t_{do} = 0,75 \cdot t_{dp}, \text{ людино-годин.} \quad (3.9)$$

Підставляючи відповідні значення, отримаємо:

$$t_{dp} = 5164,38 / (20 \cdot 1,3) = 198,63 \text{ людино-годин.}$$

$$t_{do} = 0,75 \cdot 198,63 = 148,97 \text{ людино-годин.}$$

$$t_d = 198,63 + 148,97 = 347,6 \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 56,08 + 165,51 + 158,89 + 794,52 + 347,6 = 1572,6 \text{ людино-години.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{ПО}$ включають витрати на заробітну плату виконавця програми $Z_{ПЛ}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн} \quad (3.10)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн} \quad (3.11)$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 176 грн / год, отримуємо:

$$Z_{ЗП} = 1572,6 \cdot 176 = 276777,6 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн}, \quad (3.12)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (1,18 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{МВ} = 794,52 \cdot 1,18 = 937,53 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{\text{ПО}} = 276777,6 + 937,53 = 277715,13 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.} \quad (3.13)$$

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Очікуваний період створення ПЗ:

$$T = 1572,6 / 1 \cdot 176 \approx 8,94 \text{ міс.}$$

Висновок: Розроблений продукт є повністю функціонуючим веб-застосунком, який дозволяє покупцям зручно та швидко вибирати та купувати продукт, а продавцеві швидко за ручно додавати нові види продукції або редагувати вже існуючі. Вартість даного програмного забезпечення, згідно розрахункам, становить 277715,13 грн. Також, згідно розрахункам, термін створення цього продукту становить майже 9 місяців. Цей термін обумовлений складністю поставлених задач та великою кількістю правок від замовника, щоб отримати фінальний результат.

ВИСНОВКИ

У рамках даної кваліфікаційної роботи був розроблений веб-додаток для продажу безнікотинової заправки для кальяну. Була створена багатофункціональна адміністративна панель, що дозволяє повністю змінювати будь-яку інформацію про товар, додавати нові або видаляти вже існуючі.

Цей додаток має практичне застосування, оскільки надає користувачам можливість швидко переглядати, замовляти продукцію. Також, дає можливість залишати відгуки на будь-який товар, переглядати відгуки інших користувачів, а також надсилати безпосередньо відгук на корпоративну пошту компанії.

Архітектура розробленого програмного забезпечення базується на шаблоні MVC, що передбачає поділ системи на три частини – модель даних, інтерфейс користувача та модуль керування.

Розробка застосунку відбувалася за допомогою фреймворка Laravel на мові програмування PHP. Це дозволило дотримати усі сучасні норм безпеки та дотримуватись чіткій архітектурі. Основною базою даних є MySQL, що дозволяє зберігати дані на сервері без доступу до них користувачів, які не мають відношення до супроводження сайту. Крім того, для повноцінного функціонування використовувались такі мови, як HTML, CSS та мова програмування PHP, що дозволило зробити не тільки якісний функціонал з гарними дизайном та швидкодією для користувачів, що у свою чергу сприяє притоку нових клієнтів.

У процесі виконання кваліфікаційної роботи було визначено трудомісткість розробленої системи, що становить 1572, людино-годин. Також, була розрахована вартість роботи, що склала 277715 гривень. Крім того, було розраховано очікуваний період створення додатку, що, згідно розрахункам, становить майже 9 місяців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ZANDSTRA M. PHP Objects, Patterns, and Practice. Apress, 2016. 576 с.
2. Matula T. Laravel Application Development Cookbook. Packt Publishing, 2013. 272 с.
3. Ali J. Mastering PHP Design Patterns. Packt Publishing - ebooks Account, 2016. 270 с.
4. Fowler M. Patterns of enterprise application architecture. Boston : Addison-Wesley, 2003. 533 с.
5. Rees D. Code Happy, 2012. 126 с.
6. Pecoraro C. J. Mastering Laravel. Packt Publishing, 2015. 232 с.
7. Stuttard D. The web application hacker's handbook: Finding and exploiting security flaws. 2nd ed. Indianapolis : Wiley, 2011. 912 с.
8. Bean M. Laravel 5 essentials: Explore the fundamentals of Laravel, one of the most expressive and robust PHP frameworks available. Birmingham, UK : Packt Publishing, 2015. 144с
9. Lockhart J. Modern PHP: New Features and Good Practices. O'Reilly Media, 2015. 270 с.
10. Laravel - The PHP Framework For Web Artisans. Laravel - The PHP Framework For Web Artisans. URL: <https://laravel.com/> (дата звернення: 01.05.2023).
11. HTML5: Up and Running. O'Reilly Media, 2010. 222с
12. Robbins J. N. Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. 4th ed. Beijing : O'Reilly, 2012. 603 с.
13. Flanagan D. JavaScript: The Definitive Guide. 7th ed. O'Reilly Media, 2020. 706 с.
14. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017. 624 с.
15. Taylor A. G. SQL For Dummies. For Dummies, 2018. 512 с.
16. Loeliger J. Version control with Git. 2nd ed. Beijing : O'Reilly, 2012. 434 с.

17. Jin B., Sahni S., Shevat A. Designing Web APIs: Building APIs That Developers Love. O'Reilly Media, 2018. 232 с.
18. Marrs T. JSON at Work: Practical Data Integration for the Web. O'Reilly Media, 2017. 376 с.
19. npm. npm. URL: <https://www.npmjs.com/> (дата звернення: 01.05.2023).
20. Composer. Composer. URL: <https://getcomposer.org/> (дата звернення: 01.05.2023).
21. Dou – найбільша IT-спільнота України. Dou. URL: <https://dou.ua/> (дата звернення: 01.05.2023)
22. Power Supply Calculator - PSU Calculator | OuterVision. eXtreme Outer Vision - Recommended Power Supply Calculator. URL: <https://outervision.com/power-supply-calculator> (дата звернення: 01.05.2023).
23. Офіційний сайт YASNO. YASNO. URL: <https://dp.yasno.com.ua> (дата звернення: 01.05.2023).

КОД ПРОГРАМИ

MainController.php:

```
<?php
namespace App\Http\Controllers;

use App\Http\Requests\ContactUsRequest;
use App\Mail\ContactUsMail;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Facades\Session;

class MainController
{
    public function index()
    {
        return view('user.main');
    }

    public function detailsPage()
    {
        return view('user.details.details');
    }

    public function contactPage()
    {
        return view('user.contact_us.contact_us');
    }

    public function contactUs(ContactUsRequest $request)
    {
        $mailData = [
            'username' => $request->username,
            'email' => $request->email,
            'message' => $request->message
        ];

        Mail::to('test@gmail.com')->send(new ContactUsMail($mailData));

        return redirect()->route('contact.page');
    }

    public function en()
    {
        Session::put('locale', 'en');

        return redirect()->back();
    }

    public function fr()
    {
        Session::put('locale', 'fr');

        return redirect()->back();
    }
}
```

NewsController.php:

```
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\Http\Requests\AddNewsRequest;
use App\Http\Requests>EditNewsRequest;
use App\Models\News;
use Illuminate\Http\Request;

use Illuminate\Support\Str;
use Intervention\Image\Facades\Image;

class NewsController extends Controller
{
    public function newsPage()
    {
        $newsList = News::orderBy('created_at', 'DESC')->paginate(6);

        return view('user.news.news_list', compact('newsList'));
    }

    public function currentNewsPage($slug)
    {
        $news = News::where('slug', $slug)->firstOrFail();

        return view('user.news.news', compact('news'));
    }

    public function newsListPage()
    {
        $newsList = News::orderBy('id','DESC')->get();

        return view('admin.news.news_list', compact('newsList'));
    }

    public function addNewsPage()
    {
        return view('admin.news.news_add');
    }

    public function editNewsPage($id)
    {
        $news = News::findOrFail($id);
        return view('admin.news.news_edit', compact('news'));
    }

    public function addNews(AddNewsRequest $request)
    {
        $request->file('news_image')?
            $imagePath = $this->saveNewsImage($request->file('news_image')):
            $imagePath = null;

        $slug = Str::slug($request['news_title'], '-');

        News::create([
            'news_image' => $imagePath,
            'slug' => $slug,
            'news_title' => $request['news_title'],
            'news_short_description' => $request['news_short_description'],
        ]);
    }
}
```

```

        'news_description' => $request['news_description'],
        'news_title_fr' => $request['news_title_fr'],
        'news_short_description_fr' => $request['news_short_description_fr'],
        'news_description_fr' => $request['news_description_fr']
    ]);
    return redirect()->route('admin.news.list');
}

public function editNews(EditNewsRequest $request)
{
    $news = News::findOrFail($request['id']);

    if($request->file('news_image')) {
        $imagePath = $this->saveNewsImage($request->file('news_image'));
        if($news->news_image) unlink($news->news_image);
    } else {
        $imagePath = $news->news_image;
    }

    $slug = Str::slug($request['news_title'], '-');

    $news->update([
        'slug' => $slug,
        'news_image' => $imagePath,
        'news_title' => $request['news_title'],
        'news_short_description' => $request['news_short_description'],
        'news_description' => $request['news_description'],
        'news_title_fr' => $request['news_title_fr'],
        'news_short_description_fr' => $request['news_short_description_fr'],
        'news_description_fr' => $request['news_description_fr']
    ]);

    return redirect()->route('admin.news.list');
}

public function deleteNews($id)
{
    $news = News::findOrFail($id);

    if($news->news_image) unlink($news->news_image);

    $news->delete();

    return redirect()->route('admin.news.list');
}

private function saveNewsImage($image)
{
    $generateName = hexdec(uniqid()) . '.' . $image->getClientOriginalExtension();

    $imagePath = 'upload/news_images/' . $generateName;

    Image::make($image)->save($imagePath);

    return $imagePath;
}
}

```

ProductsController.php:

```
<?php
```

```

namespace App\Http\Controllers;

use App\Http\Requests\AddProductRequest;
use App\Http\Requests>EditProductRequest;
use App\Models\Product;
use App\Models\ProductReview;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Str;
use Intervention\Image\Facades\Image;

class ProductsController extends Controller
{
    public function productsPage()
    {
        $productsList = Product::query()
            ->leftJoin('product_reviews', function($join) {
                $join->on('product_reviews.product_id', '=', 'products.id')->where('product_reviews.accepted', '!=', 0);
            })
            ->select([
                'products.id',
                'products.slug',
                'products.product_name',
                'products.product_price',
                'products.product_image',
                DB::raw('avg(product_reviews.product_evaluation) as rating')
            ])
            ->groupBy('products.id')
            ->orderBy('rating', 'desc')
            ->paginate(8);

        return view('user.products.products_list', compact('productsList'));
    }

    public function currentProductPage($slug)
    {
        $product = Product::where('slug', $slug)->firstOrFail();
        $reviewsList = ProductReview::where('product_id', $product->id)->where('accepted', '!=', true)-
>orderBy('created_at', 'desc')->get();
        $averageRate = round($reviewsList->avg('product_evaluation'));

        return view('user.products.product', compact('product', 'reviewsList', 'averageRate'));
    }

    // admin part
    public function productsListPage()
    {
        $productsList = Product::get();

        return view('admin.products.products_list', compact('productsList'));
    }

    public function addProductPage()
    {
        return view('admin.products.products_add');
    }

    public function addProduct(AddProductRequest $request)
    {
        $request->file('product_image')?
            $productImage = $this->saveProductImage($request->file('product_image')):
            $productImage = null;
    }
}

```

```

$request->file('product_image2') && !$request->delete_image2?
$productImage2 = $this->saveProductImage($request->file('product_image2'));
$productImage2 = null;

$request->file('product_image3') && !$request->delete_image3?
$productImage3 = $this->saveProductImage($request->file('product_image3'));
$productImage3 = null;

$request->file('product_image4') && !$request->delete_image4?
$productImage4 = $this->saveProductImage($request->file('product_image4'));
$productImage4 = null;

$request->file('product_image5') && !$request->delete_image5?
$productImage5 = $this->saveProductImage($request->file('product_image5'));
$productImage5 = null;

$slug = Str::slug($request['product_name'], '-');

$productVideo = $this->getVideoId($request->product_video);
$productVideo2 = $this->getVideoId($request->product_video2);
$productVideo3 = $this->getVideoId($request->product_video3);

Product::create([
    'product_name' => $request['product_name'],
    'product_name_fr' => $request['product_name_fr'],
    'slug' => $slug,
    'product_image' => $productImage,
    'product_image2' => $productImage2,
    'product_image3' => $productImage3,
    'product_image4' => $productImage4,
    'product_image5' => $productImage5,
    'product_video' => $productVideo,
    'product_video2' => $productVideo2,
    'product_video3' => $productVideo3,
    'product_description' => $request['product_description'],
    'product_description_fr' => $request['product_description_fr'],
    'product_price' => $request['product_price'],
    'product_compound' => $request['product_compound'],
    'product_compound_fr' => $request['product_compound_fr'],
    'product_amazon_link' => $request['product_amazon_link']
]);

return redirect()->route('admin.products.list');
}

public function editProductsPage($id)
{
    $product = Product::findOrFail($id);

    return view('admin.products.products_edit', compact('product'));
}

public function editProduct(EditProductRequest $request)
{
    $product = Product::findOrFail($request->id);

    if($request->delete_image) {
        if($product->product_image) unlink($product->product_image);
        $productImage = null;
    } else if($request->file('product_image')) {
        $productImage = $this->saveProductImage($request->file('product_image'));
        if($product->product_image) unlink($product->product_image);
    }
}

```

```

} else {
    $productImage = $product->product_image;
}

if($request->delete_image2) {
    if($product->product_image2) unlink($product->product_image2);
    $productImage2 = null;
} else if($request->file('product_image2')) {
    $productImage2 = $this->saveProductImage($request->file('product_image2'));
    if($product->product_image2) unlink($product->product_image2);
} else {
    $productImage2 = $product->product_image2;
}

if($request->delete_image3) {
    if($product->product_image3) unlink($product->product_image3);
    $productImage3 = null;
} else if($request->file('product_image3')) {
    $productImage3 = $this->saveProductImage($request->file('product_image3'));
    if($product->product_image3) unlink($product->product_image3);
} else {
    $productImage3 = $product->product_image3;
}

if($request->delete_image4) {
    if($product->product_image4) unlink($product->product_image4);
    $productImage4 = null;
} else if($request->file('product_image4')) {
    $productImage4 = $this->saveProductImage($request->file('product_image4'));
    if($product->product_image4) unlink($product->product_image4);
} else {
    $productImage4 = $product->product_image4;
}

if($request->delete_image5) {
    if($product->product_image5) unlink($product->product_image5);
    $productImage5 = null;
} else if($request->file('product_image5')) {
    $productImage5 = $this->saveProductImage($request->file('product_image5'));
    if($product->product_image5) unlink($product->product_image5);
} else {
    $productImage5 = $product->product_image5;
}

$slug = Str::slug($request['product_name'], '-');

$productVideo = $this->getVideoId($request->product_video);
$productVideo2 = $this->getVideoId($request->product_video2);
$productVideo3 = $this->getVideoId($request->product_video3);

$product->update([
    'product_name' => $request['product_name'],
    'product_name_fr' => $request['product_name_fr'],
    'slug' => $slug,
    'product_image' => $productImage,
    'product_image2' => $productImage2,
    'product_image3' => $productImage3,
    'product_image4' => $productImage4,
    'product_image5' => $productImage5,
    'product_video' => $productVideo,
    'product_video2' => $productVideo2,
    'product_video3' => $productVideo3,
    'product_price' => $request['product_price'],

```

```

        'product_description' => $request['product_description'],
        'product_description_fr' => $request['product_description_fr'],
        'product_compound' => $request['product_compound'],
        'product_compound_fr' => $request['product_compound_fr'],
        'product_amazon_link' => $request['product_amazon_link']
    ]);

    return redirect()->route('admin.products.list');
}

public function deleteProduct($id)
{
    $product = Product::findOrFail($id);

    if($product->product_image) unlink($product->product_image);

    $product->delete();

    return redirect()->route('admin.products.list');
}

private function saveProductImage($image)
{
    $generateName = hexdec(uniqid()) . '.' . $image->getClientOriginalExtension();

    $imagePath = 'upload/products_images/' . $generateName;

    $width = Image::make($image)->width();
    $height = Image::make($image)->height();

    $new_width = $height * 0.7;
    $new_height = $height;

    Image::make($image)->resizeCanvas((int) $new_width, (int) $new_height)->resize(370, 526)->save($imagePath);

    return $imagePath;
}

private function getVideoId($url = null)
{
    {
        if(!$url) {
            return null;
        }

        parse_str(parse_url($url, PHP_URL_QUERY), $id);
        if(isset($id['v'])) return $id['v'];
        else return null;
    }
}
}

```


ProductReviewsController.php:

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\ReviewRequest;
use App\Models\Product;
use App\Models\ProductReview;
use Illuminate\Http\Request;

class ProductReviewsController extends Controller
{
    public function reviewsListPage()
    {
        $reviewsList = ProductReview::join('products', 'product_id', '=', 'products.id')
            ->where('accepted', true)
            ->select(
                'product_reviews.id',
                'products.product_name',
                'product_evaluation',
                'user_name',
                'review_text',
                'user_email'
            )->orderBy('id', 'desc')->get();

        $productsList = Product::select('id','product_name')->get();

        return view('admin.reviews.reviews_list', compact('productsList', 'reviewsList'));
    }

    public function reviewsListPageFiltered(Request $request)
    {
        if($request->product_id == 'all products') return redirect()->route('admin.reviews.list');

        $reviewsList = ProductReview::join('products', 'product_id', '=', 'products.id')->where('products.id', $request-
        >product_id)
            ->where('accepted', true)
            ->select(
                'product_reviews.id',
                'products.product_name',
                'product_evaluation',
                'user_name',
                'review_text',
                'user_email'
            )->orderBy('id', 'desc')->get();

        $productsList = Product::select('id','product_name')->get();

        return view('admin.reviews.reviews_list', compact('productsList', 'reviewsList'));
    }

    public function deleteReview($id)
    {
        $review = ProductReview::findOrFail($id);

        $review->delete();

        return redirect()->back();
    }
}
```

```

public function addReview(ReviewRequest $request) {

    ProductReview::create([
        'product_id' => $request->product_id,
        'user_email' => $request->user_email,
        'user_name' => $request->user_name,
        'product_evaluation' => $request->product_evaluation,
        'review_text' => $request->review_text
    ]);

    return redirect()->back();
}

public function moderateReviewsPage()
{
    $reviewsList = ProductReview::join('products', 'product_id', '=', 'products.id')
        ->where('accepted', false)
        ->select(
            'product_reviews.id',
            'products.product_name',
            'product_evaluation',
            'user_name',
            'review_text',
            'user_email',
            'product_reviews.created_at'
        )->orderBy('id', 'desc')->get();

    return view('admin.reviews.reviews_moderate', compact('reviewsList'));
}

public function acceptReview($id)
{
    $review = ProductReview::findOrFail($id);

    $review->update([
        'accepted' => true
    ]);

    return redirect()->route('admin.reviews.moderate');
}
}

```

GetPreferLocalization.php:

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\App;
use Illuminate\Support\Facades\Config;
use Illuminate\Support\Facades\Session;

class GetPreferLocalization
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request

```

```

    *@param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)
    $next
    *@return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
    */
    public function handle(Request $request, Closure $next)
    {
        if( !Session::has('locale' )){
            $browserLanguage = substr($_SERVER['HTTP_ACCEPT_LANGUAGE'], 0, 2); //read browser language

            if ( in_array($browserLanguage, Config::get('app.localizations')) ) {
                Session::put('locale', $browserLanguage);
            } else {
                Session::put('locale', 'en');
            }
        }

        //Simply set language from session
        App::setlocale(Session::get('locale'));

        return $next($request);
    }
}

```

admin_main.js:

```

$(function () {
    $(document).ready(function() {
        $("#table-sort").tablesorter();
    });

    $(document).ready(function(){
        $('#inputImage').change(function(e){
            var reader = new FileReader();
            reader.onload = function(e){
                $('#showImage').attr('src',e.target.result);
            }
            reader.readAsDataURL(e.target.files[0]);
        });
    });

    $(document).ready(function(){
        $('#inputImage2').change(function(e){
            var reader = new FileReader();
            reader.onload = function(e){
                $('#showImage2').attr('src',e.target.result);
            }
            reader.readAsDataURL(e.target.files[0]);
        });
    });
});

```

```

$(document).ready(function(){
  $('#inputImage3').change(function(e){
    var reader = new FileReader();
    reader.onload = function(e){
      $('#showImage3').attr('src',e.target.result);
    }
    reader.readAsDataURL(e.target.files['0']);
  });
});

```

```

$(document).ready(function(){
  $('#inputImage4').change(function(e){
    var reader = new FileReader();
    reader.onload = function(e){
      $('#showImage4').attr('src',e.target.result);
    }
    reader.readAsDataURL(e.target.files['0']);
  });
});

```

```

$(document).ready(function(){
  $('#inputImage5').change(function(e){
    var reader = new FileReader();
    reader.onload = function(e){
      $('#showImage5').attr('src',e.target.result);
    }
    reader.readAsDataURL(e.target.files['0']);
  });
});

```

```

});

```

main.js:

```

$(function () {
  $('.burger').click (function () {
    $('.burger__btn').toggleClass('cross');
    $('.menu-collapse').toggleClass('drive').css('order', '1 ');
    $('.menu').toggleClass('menu-opened');
    $('body').toggleClass('lock');
  });
});

```

```
    $('html').toggleClass('lock');
    $('.overlay').toggleClass('fone');
});
```

```
$(document).ready(function () {
    $('#list > li').click(function () {
        $(this).toggleClass('active');
        $(this).children("ul").slideToggle();
    });
    $('ul').click(function (event) {
        event.stopPropagation();
    });
});
```

```
$(document).ready(function () {
    $('.product-list__wrapper').hover(function () {
        $(this).children(".hint-button").toggleClass('pull');
    });
});
```

```
$('.slider__item').slick({
    slidesToShow: 1,
    slidesToScroll: 1,
    arrows: false,
    fade: true,
    waitForAnimate: false,
    asNavFor: '.slider__dots'
});
```

```
$('.slider__dots').slick({
    slidesToShow: 5,
    asNavFor: '.slider__item',
    dots: false,
    waitForAnimate: false,
    focusOnSelect: true,
    vertical: true
});
```

```
$(document).ready(function() {
    var list = $(".review-list__item");
    var numToShow = 6;
```

```

var button = $(".review-list__more");
var numInList = list.length;
list.hide();
if (numInList > numToShow) {
    button.show();
} else {
    button.hide();
}
list.slice(0, numToShow).show();
button.click(function() {
    var showing = list.filter(':visible').length;
    list.slice(showing - 1, showing + numToShow).fadeIn(600);
    var nowShowing = list.filter(':visible').length;
    if (nowShowing >= numInList) {
        button.hide();
    }
});
});

```

```

$(document).ready(function() {
    $('.popup-youtube').magnificPopup({
        type: 'iframe',
        mainClass: 'mfp-fade',
        removalDelay: 160,
        preloader: false,
        fixedContentPos: true
    });
});

```

```
});
```

main.blade.php:

```

@extends('user.user_master')
@section('content')
<section id="content" class="content">
    <div class="container">
        <div class="main">
            <div class="row">
                <div class="col-sm-8 mx-auto col-md-6 col-xl-5">
                    <div class="description">
                        <h1 class="description__title">
                            <span class="green-font">@lang('user/main.first_block_header_green')</span>

```

```

        @lang('user/main.first_block_header_black')
    </h1>
    <p class="description__text">
        @lang('user/main.first_block_description')
    </p>
    <a href="{{ route('products.page') }}" class="description__btn">
        @lang('user/main.make_an_order_button')
    </a>
</div>
</div>
<div class="col-md-6 col-lg-6 offset-xl-1 col-xl-6">
    <div class="product">
        
        
        

        <div class="leaf-2 d-none d-lg-block"></div>
        <div class="leaf-3 d-none d-lg-block"></div>
    </div>
</div>
</div>
<div class="parallax-bg1" data-type="background" data-speed="2.5">
    <div class="petal-1" ></div>
    <div class="petal-2 d-none d-md-block"></div>
    <div class="petal-3"></div>
    <div class="petal-4 d-none d-lg-block"></div>
    <div class="petal-5"></div>
    <div class="petal-6 d-none d-lg-block"></div>
</div>
<div class="leaf-4 d-none d-lg-block">
    
</div>
</div>
</div>
</section>
<section id="features" class="features">
    <div class="dignity">
        <div class="container">
            <div class="row">
                <div class="col g-0">

```

```

    <h2 class="dignity__title">
      @lang('user/main.second_block_header')
    </h2>
  </div>
</div>
<div class="wrapper">
  <div class="row">
    <div class="col-md-5 m-lg-auto">
      <div class="dignity__descr">
        <p class="dignity__text">
          @lang('user/main.second_block_description')
        </p>
      </div>
    </div>
    <div class="col-12 col-sm-4 offset-md-1 col-md-2 my-md-auto">
      <div class="dignity__item">
        <span>
          0%
        </span>
        <p>
          @lang('user/main.second_block_nicotine')
        </p>
      </div>
    </div>
    <div class="col-12 col-sm-4 col-md-2 my-md-auto">
      <div class="dignity__item">
        <span>
          0%
        </span>
        <p>
          @lang('user/main.second_block_tar')
        </p>
      </div>
    </div>
    <div class="col-12 col-sm-4 col-md-2 my-md-auto">
      <div class="dignity__item">
        <span>
          0%
        </span>
        <p>
          @lang('user/main.second_block_tobacco')
        </p>
      </div>
    </div>
  </div>

```



```

<div class="col-12 col-sm-6 col-md-4 col-lg-3 offset-xl-1 col-xl-2">
  <div class="advantage__item">
    
    <p class="advantage__text">
      @lang('user/main.third_block_subheader4')
    </p>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-4 col-lg-3 offset-xl-1 col-xl-2">
  <div class="advantage__item">
    
    <p class="advantage__text">
      @lang('user/main.third_block_subheader5')
    </p>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-4 col-lg-3 offset-xl-1 col-xl-2">
  <div class="advantage__item">
    
    <p class="advantage__text">
      @lang('user/main.third_block_subheader6')
    </p>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-4 col-lg-3 offset-xl-1 col-xl-2">
  <div class="advantage__item">
    
    <p class="advantage__text">
      @lang('user/main.third_block_subheader7')
    </p>
  </div>
</div>
<div class="col-12 col-sm-6 col-md-4 col-lg-3 offset-xl-1 col-xl-2">
  <div class="advantage__item">
    
    <p class="advantage__text">
      @lang('user/main.third_block_subheader8')
    </p>
  </div>
</div>
</div>
<div class="parallax-bg1" data-type="background" data-speed="2.5">

```

```

    <div class="petal-7"></div>
    <div class="petal-8"></div>
</div>
<div class="leaf-5">
    
</div>
</div>
</div>
</section>
<section id="How-to-use" class="How-to-use">
    <div class="container">
        <div class="manual">
            <div class="row">
                <div class="col-12 col-md-4 offset-lg-1">
                    <div class="hookah">
                        
                    </div>
                </div>
                <div class="col-9 mx-auto offset-md-1 col-md-7 col-lg-6 offset-xl-1">
                    <div class="usage">
                        <h2 class="usage__title">
                            @lang('user/main.fourth_block_header')
                        </h2>
                        <div class="rule">
                            <span class="big-green">
                                1
                            </span>
                            <p class="rule__text">
                                @lang('user/main.fourth_block_subheader')
                            </p>
                        </div>
                        <div class="rule">
                            <span class="big-green">
                                2
                            </span>
                            <p class="rule__text">
                                @lang('user/main.fourth_block_subheader2')
                            </p>
                        </div>
                        <div class="rule">
                            <span class="big-green">
                                3

```

```

        </span>
        <p class="rule__text">
            @lang('user/main.fourth_block_subheader3')
        </p>
    </div>
    <div class="rule">
        <span class="big-green">
            4
        </span>
        <p class="rule__text">
            @lang('user/main.fourth_block_subheader4')
        </p>
    </div>
</div>
</div>
<div class="parallax-bg1" data-type="background" data-speed="2.5">
    <div class="petal-9 d-none d-lg-block"></div>
    <div class="petal-10 d-none d-lg-block"></div>
    <div class="petal-11 d-none d-lg-block"></div>
</div>
<div class="leaf-6 d-none d-md-block">
    
</div>
</div>
</div>
</section>
<section id="compound" class="compound">
    <div class="container">
        <div class="mix">
            <div class="row">
                <div class="col">
                    <h2>
                        @lang('user/main.fifth_block_header')
                    </h2>
                </div>
                <div class="row">
                    <div class="col-11 col-md-6 mx-auto offset-xl-1 col-xl-5">
                        <div class="technology">
                            <p>
                                @lang('user/main.fifth_block_description')
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

    </div>
</div>
<div class="col-11 col-md-6 mx-auto">
  <div class="technology">
    <p>
      @lang('user/main.fifth_block_description2')
    </p>
  </div>
</div>
</div>
<div class="row">
  <div class="col-sm-6 col-md-4 offset-xl-2 col-xl-2">
    <div class="mint">
      <div class="circle-1 circle">
        
      </div>
      <p class="mint__text text">
        @lang('user/main.fifth_block_subheader')
      </p>
    </div>
  </div>
  <div class="col-sm-6 col-md-4 offset-xl-1 col-xl-2">
    <div class="dried-fruit">
      <div class="circle-2 circle">
        
        
        
      </div>
      <p class="dried-fruit__text text">
        @lang('user/main.fifth_block_subheader2')
      </p>
    </div>
  </div>
  <div class="col-sm-12 col-md-4 offset-xl-1 col-xl-2">
    <div class="honey">
      <div class="circle-3 circle">
        
      </div>
      <p class="honey__text text">
        @lang('user/main.fifth_block_subheader3')

```

```

        </p>
    </div>
</div>
</div>
<div class="row">
    <div class="col-11 mx-auto col-lg-12 offset-xl-2 col-xl-8">
        <p class="ingridients">
            @lang('user/main.fifth_block_description3')
        </p>
    </div>
</div>
</div>
<div class="parallax-bg1" data-type="background" data-speed="2.5">
    <div class="petal-12"></div>
    <div class="petal-13"></div>
</div>
<div class="leaf-7 d-none d-lg-block">
    
</div>
</div>
</div>
</section>
@endsection

```

ContactUsMail.php:

```

<?php
namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;

class ContactUsMail extends Mailable
{
    use Queueable, SerializesModels;

    public $mailData;

    /**
     * Create a new message instance.
     *
     * @return void
     */
    public function __construct($mailData)
    {
        $this->mailData = $mailData;
    }
}

```

```

/**
 * Build the message.
 *
 * @return $this
 */
public function build()
{
    return $this->view('emails.contact_us_mail');
}
}

```

web.php:

```
<?php
```

```

use App\Http\Controllers\Auth\RegisteredUserController;
use App\Http\Controllers\Auth\AuthenticatedSessionController;

```

```

use App\Http\Controllers>ContactUsController;
use App\Http\Controllers/MainController;
use App\Http\Controllers\NewsController;
use App\Http\Controllers\ProductReviewsController;
use App\Http\Controllers\ProductsController;

```

```
use Illuminate\Support\Facades\Route;
```

```
// Admin Login routes
```

```

Route::controller(AuthenticatedSessionController::class)->prefix('admin')->group(function () {
    Route::get('/login', 'create')->middleware('guest')->name('login');
    Route::post('/login', 'store')->middleware('guest');
    Route::post('/logout', 'destroy')->middleware('auth')->name('logout');
});

```

```
// Admin routes
```

```
Route::prefix('admin')->middleware('auth')->group(function () {
```

```

    Route::controller(NewsController::class)->group(function () {
        Route::get('/news/list', 'newsListPage')->name('admin.news.list');

```

```

        Route::get('/news/add', 'addNewsPage')->name('admin.news.add.page');
        Route::post('/news/add', 'addNews')->name('admin.news.add');

```

```

        Route::get('/news/edit/{id}', 'editNewsPage')->name('admin.news.edit.page');
        Route::post('/news/edit', 'editNews')->name('admin.news.edit');

```

```

        Route::get('/news/delete/{id}', 'deleteNews')->name('admin.news.delete');
    });

```

```

Route::controller(ProductsController::class)->group(function() {
    Route::get('/products/list', 'productsListPage')->name('admin.products.list');

```

```

        Route::get('/products/add', 'addProductPage')->name('admin.products.add.page');
        Route::post('/products/add', 'addProduct')->name('admin.products.add');

```

```

        Route::get('/products/edit/{id}', 'editProductsPage')->name('admin.products.edit.page');
        Route::post('/products/edit', 'editProduct')->name('admin.products.edit');

```

```

        Route::get('/products/delete/{id}', 'deleteProduct')->name('admin.products.delete');
    });

```

```

});

Route::controller(ProductReviewsController::class)->group(function() {
    Route::get('/reviews', 'reviewsListPage')->name('admin.reviews.list');
    Route::post('/reviews', 'reviewsListPageFiltered')->name('admin.reviews.list');

    Route::get('/reviews/moderate', 'moderateReviewsPage')->name('admin.reviews.moderate');
    Route::get('/reviews/accept/{id}', 'acceptReview')->name('admin.reviews.accept');

    Route::get('/reviews/delete/{id}', 'deleteReview')->name('admin.reviews.delete');
});

Route::controller(ContactUsController::class)->group(function() {
    Route::get('/messages/list', 'messagesListPage')->name('admin.messages.page');

    Route::post('/messages/delete/{id}', 'deleteMessage')->name('admin.messages.delete');
});
});

// User routes
Route::middleware('localization')->group(function() {

    Route::controller(MainController::class)->group(function() {
        Route::get('/', 'index')->name('main.page');
        Route::get('/details', 'detailsPage')->name('details.page');
        Route::get('/contact', 'contactPage')->name('contact.page');
        Route::post('/contact_us', 'contactUs')->name('contact.send');
        Route::get('/en', 'en')->name('language.en');
        Route::get('/fr', 'fr')->name('language.fr');
    });

    Route::controller(NewsController::class)->group(function() {
        Route::get('/news', 'newsPage')->name('news.page');
        Route::get('/news/{slug}', 'currentNewsPage')->name('news.page.current');
    });

    Route::controller(ProductsController::class)->group(function() {
        Route::get('/products', 'productsPage')->name('products.page');
        Route::get('/products/{slug}', 'currentProductPage')->name('products.page.current');
    });

    Route::controller(ProductReviewsController::class)->group(function() {
        Route::post('/products/leave_a_review', 'addReview')->name('reviews.leave');
    });

});
});

```


ДОДАТОК Б

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
ПЗ_Гордіюк.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
ПЗ_Гордіюк.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Frumint.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація Гордіюк.ppt	Презентація кваліфікаційної роботи