

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

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: Web-додаток підтримки діяльності магазину садового інвентарю

Здобувача групи ІТ-92 Дудки Максима Олександровича
(шифр групи) (прізвище, ім'я, по батькові)

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

(підпис)

Дудка Максим Олександрович
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник к.т.н. доц. Неня Віктор Григорович _____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

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

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ С.М. Ващенко
«__» _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Дудки Максима Олександровича

1 Тема роботи Web-додаток підтримки діяльності магазину
садового інвентарю

керівник роботи Неня Віктор Григорович, к.т.н., доцент,

затверджені наказом по університету від «29» травня 2023 р. №0588-VI

2 Строк подання студентом роботи «11» червня 2023 р.

3 Вхідні дані до роботи Фото садового інвентарю, текстовий опис

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ, аналіз предметної області, постановка задачі та
методи дослідження, практична реалізація web-додатка, висновки,
список використаних джерел, технічне завдання, планування робіт

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
вступ, актуальність, постановка задачі, аналіз web-додатків для торгівлі
садового інвентарю, порівняння сайтів-аналогів, моделювання процесу
підтримки діяльності магазину садового інвентарю з використанням
web-додатку, контекстна діаграма, діаграма декомпозиції, діаграма варіантів
використання, реалізація проекту, база даних, архітектура web-додатку,
демонстрація web-додатку (користувач), демонстрація web-додатку
(адміністратор), висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд предметної області	До 16.02.2023	
2	Аналіз конкурентів	До 19.02.2023	
3	Планування	До 24.02.2023	
4	Реалізація	До 12.04.2023	
5	Розробка прототипу	До 17.04.2023	
6	Верстка	До 07.05.2023	
7	Робота над програмною частиною	До 22.05.2023	
8	Заповнення контентом	До 23.05.2023	
9	Тестування	До 28.05.2023	
10	Перенесення web-додатку на сервер	До 01.06.2023	

Студент

(підпис)

Дудка М.О.

Керівник роботи

(підпис)

к.т.н., доц. Неня В.Г.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки діяльності магазину садового інвентарю».

Кваліфікаційна робота містить 92 сторінок, 12 таблиць, 56 рисунків, список літератури 24 найменувань, 3 додатки.

Головна мета проекту – створення web-додатку для магазину садового інвентарю з вбудованим конструктором. Цей додаток буде надавати швидку та зручну можливість оформлення замовлень, а також різних типів замовлень.

У першому розділі було проведено детальний аналіз вибраної сфери дослідження. Було розглянуто і проаналізовано актуальні дослідження та публікації, оглянуто наявні web-додатки, що відповідають даній області. Поставлено чіткі цілі та завдання дослідження і розробки, а також описано інструменти та технології, що використовуються в реалізації проекту.

У другому розділі надано детальний опис проектування web-додатку. Було виконано структурно-функціональне моделювання за допомогою нотації IDEF0, де була показана взаємодія між компонентами системи. Також була побудована діаграма варіантів використання, яка демонструє різні можливі шляхи взаємодії користувача з додатком. Не менш важливою була розробка структури бази даних, яка забезпечує зберігання необхідної інформації.

У третьому розділі була здійснена програмна реалізація проекту та детальний опис використання web-додатку зі сторони замовника та адміністратора. Були розроблені необхідні функціональні модулі і реалізована взаємодія з користувачем, як замовником, так і адміністратором.

Результатом проведеної роботи є розроблений web-додаток підтримки діяльності магазину садового інвентарю.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, WEB-ДОДАТОК, СЕРВІС, ПЛАТІЖНА СИСТЕМА, ТОВАР, КОРИСТУВАЧ, ЗАМОВНИК, ПРОПОЗИЦІЯ, ВІДГУК, ЗВІТ, ПРОЕКТ.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Огляд останніх досліджень і публікацій.....	6
1.2 Аналіз програмних продуктів	8
1.3. Мета та задачі дослідження web-додатку.....	16
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	19
2.1. Вибір засобів реалізації	19
2.2. Проектування web-додатку	20
2.3. Проектування бази даних	22
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ.....	27
3.1 Архітектура web-додатку	27
3.2 Програмна реалізація	28
3.3 Використання web-додатку зі сторони користувача	33
3.4 Адміністрування сайту	42
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ	51
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ.....	63
ДОДАТОК В. КОД РЕАЛІЗАЦІЇ WEB-ДОДАТКУ	76

ВСТУП

Аграрна промисловість постійно розвивається та розширює свої горизонти. Враховуючи постійний розвиток не лише комерційного потенціалу цієї галузі, а й глибоке впровадження з інформаційними технологіями, то можна вважати даний напрямок актуальним. Перед початком розробки потрібно дослідити проблеми конкурентів, визначитися з баченням майбутнього проекту та вирішити головну ціль інформаційної системи – розробити web-додаток підтримки діяльності магазину садового інвентарю.

Об'єкт дослідження – інформаційна система для підтримки діяльності магазину садового інвентарю.

Предмет дослідження – web-додаток для покупки або продажу садового знаряддя та інвентарю різних брендів.

Мета кваліфікаційної роботи бакалавра – створення web-додатку підтримки діяльності магазину садового інвентарю. Магазин буде надавати послуги продажі товарів різних виробників.

Реалізовано такі задачі кваліфікаційної роботи:

- аналіз предметної області інформаційної системи та огляд web-додатків конкурентів, проектування вимог та календарного плану;
- розробка функціональних вимог web-додатку;
- створення інформаційної частини, додавання контенту;
- реалізація web-додатку в інформаційному просторі та тестування.

Створений проект буде реалізовано у вигляді web-додатку, з визначеним напрямком підтримки аграрного сектору для зручного використання потенційними покупцями.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Сама аграрна промисловість являє собою галузь економіки та пов'язана з виробництвом сільськогосподарської продукції та переробленням сільськогосподарських сировинних матеріалів. Вона охоплює різні сектори, такі як рослинництво, тваринництво, рибне господарство, лісове господарство та інші діяльності, пов'язані за збиранням, вирощуванням, виробництвом та переробленням сільськогосподарських продуктів [1]. Аграрна промисловість має велике значення для економіки багатьох країн, особливо тих, де сільське господарство відіграє ключову роль. Вона забезпечує харчову безпеку, зайнятість та експортні можливості. Аграрна промисловість також охоплює дослідження та розвиток нових технологій і методів, спрямованих на покращення продуктивності, ефективності та стійкості сільськогосподарського виробництва [2,3]. Останні роки характеризуються розвитком сучасних технологій в аграрній промисловості, таких як сільськогосподарська механізація, використання дронів і автоматизація процесів. Ці технології сприяють підвищенню виробничої потужності, якості продукції та зменшенню впливу на навколишнє середовище. Однак, аграрна промисловість також стикається з викликами, такими як зміна клімату, забруднення довкілля, скорочення оброблюваних земель, втрата біорізноманіття та соціально-економічні проблеми для сільськогосподарських виробників [4]. Тому важливо розвивати стійке, екологічно відповідальне та соціально зорієнтоване сільське господарство, що сприятиме сталому розвитку аграрної промисловості.

Магазин садового інвентарю - це спеціалізований роздрібний заклад, який надає широкий асортимент товарів для саду, городу та прилеглої території. Він пропонує різноманітні інструменти, пристрої, аксесуари і матеріали, які допомагають садівникам та городникам в догляді за рослинами, створенні та облаштуванні простору для вирощування рослин [5].

Діяльність магазину садового інвентарю може включати наступні аспекти:

- продаж інструментів;
- продаж рослини та насіння;
- пропонувати добрива та засоби захисту рослин;
- надавати вибір в садових аксесуарів;
- надання консультації та підтримка.

Деякі магазини садового інвентарю можуть також проводити семінари або тренінги для клієнтів, на яких пояснюють основні аспекти догляду за рослинами та надають практичні навички [6].

Метою магазину садового інвентарю є задоволення потреб клієнтів у всьому, що стосується садівництва та догляду за природним середовищем. Він надає необхідні засоби та інформацію, щоб клієнти могли насолоджуватися своїм садом чи городом та досягати успіху у вирощуванні рослин.

Інтернет-магазини садового інвентарю є платформами, де клієнти можуть замовляти товари для саду та городу через Інтернет. Вони пропонують широкий вибір продуктів, які допомагають садівникам та городникам в догляді за рослинами та створенні комфортного садового простору [7].

Особливості роботи інтернет-магазинів садового інвентарю:

1. Онлайн каталог: Інтернет-магазини мають веб-сайти, де клієнти можуть переглядати широкий асортимент товарів за допомогою онлайн каталогу. Каталог містить детальні описи продуктів, зображення, ціни та іншу необхідну інформацію.

2. Зручне замовлення: Клієнти можуть легко здійснити замовлення, вибравши бажані товари через веб-сайт. Вони можуть додавати товари до кошика, вказувати кількість, обирати опції, які можуть бути доступні для певних товарів.

3. Безпека платежів: Інтернет-магазини зазвичай пропонують безпечні способи оплати, такі як кредитні картки, електронні платіжні системи або банківські перекази. Вони забезпечують захист особистих даних та фінансової інформації клієнтів [8].

4. Доставка: Інтернет-магазини забезпечують доставку замовлених товарів до вказаної адреси клієнта. Вони співпрацюють з кур'єрськими службами або поштовими операторами для забезпечення швидкої та надійної доставки.

5. Клієнтська підтримка: Багато інтернет-магазинів надають клієнтську підтримку через електронну пошту, онлайн-чат або телефонний зв'язок. Вони готові відповідати на запитання клієнтів, надавати консультації щодо вибору товарів та допомагати розв'язувати проблеми, які можуть виникнути під час процесу замовлення або доставки.

6. Відгуки та рейтинги: Багато інтернет-магазинів мають систему відгуків та рейтингів, де клієнти можуть залишати свої відгуки про куплені товари. Це допомагає іншим клієнтам прийняти рішення щодо покупки та додає додатковий рівень довіри до магазину.

Інтернет-магазини садового інвентарю надають зручну та ефективну платформу для покупок, де клієнти можуть швидко знайти необхідні товари і замовити їх з доставкою безпосередньо до свого дому [9].

Web-додаток може включати різноманітні функції, такі як система управління замовленнями, інвентаризація товарів, облік фінансів, аналітичні інструменти, звітність та багато іншого. Він розроблений з урахуванням потреб та вимог конкретного бізнесу з метою спрощення та автоматизації рутинних операцій, що дозволяє підприємству ефективніше функціонувати. Web-додаток може забезпечувати централізовану систему керування, що дозволяє власникам та керівникам підприємства моніторити та контролювати ключові процеси, отримувати доступ до актуальних даних та звітності в режимі реального часу. Користувачі можуть здійснювати різноманітні операції та взаємодіяти з системою через зручний інтерфейс, доступний з будь-якого пристрою з підключенням до Інтернету.

1.2 Аналіз програмних продуктів

Для розробки конкурентоспроможного web-додатка потрібно дослідити інформаційні системи аналогів.

Досліджено три приклади інтернет-магазинів садового інвентарю:

- GardenTools.com;
- GreenThumbGardenSupplies.com;

– TheGardenDepot.com.

Перший приклад інтернет-магазину садового інвентарю представлено на рис.1.1.

Цей магазин пропонує широкий асортимент садових інструментів, аксесуарів та приладдя. Вони мають великий вибір ручних та електричних інструментів для роботи у саду та городі, таких як граблі, лопати, газонокосарки, системи поливу, багаторічники та багато іншого [10]. GardenTools.com також пропонує швидку доставку та забезпечує професійну консультацію щодо вибору інструментів (рис.1.2).

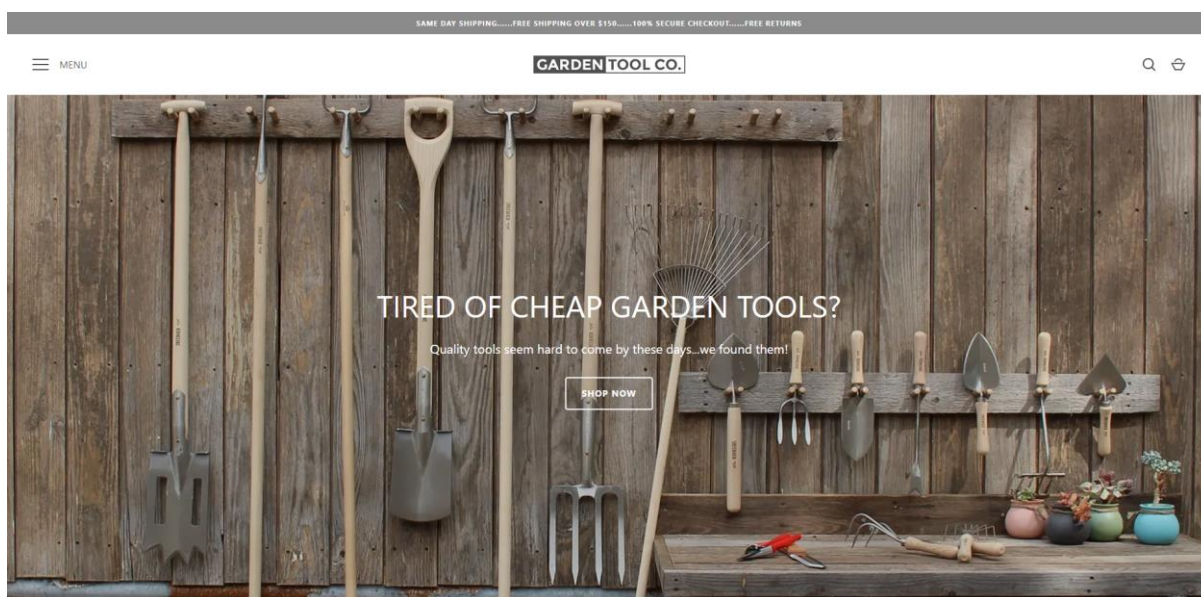


Рисунок 1.1 – Головна сторінка «GardenTools.com»

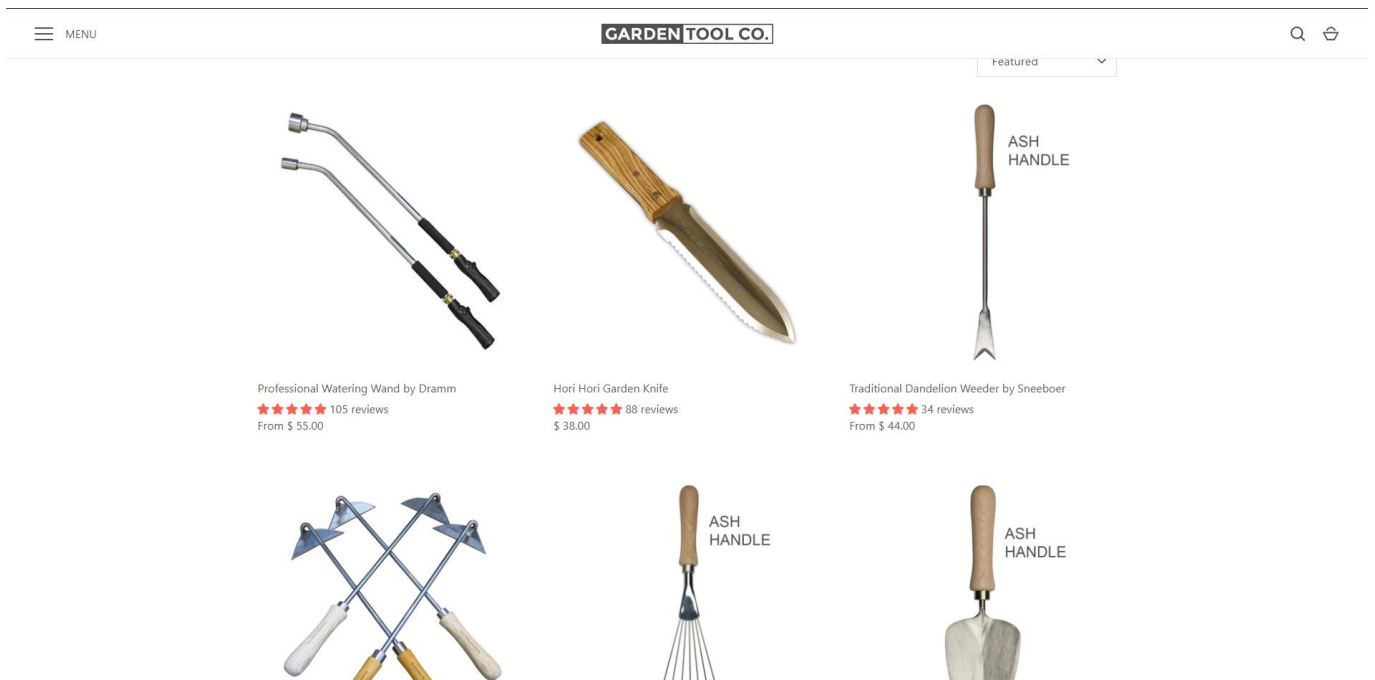


Рисунок 1.2 – Набір товарів на «GardenTools.com»

Сайт підкупляє своїм мінімалізмом та простотою, чим привертає увагу (рис.1.3).

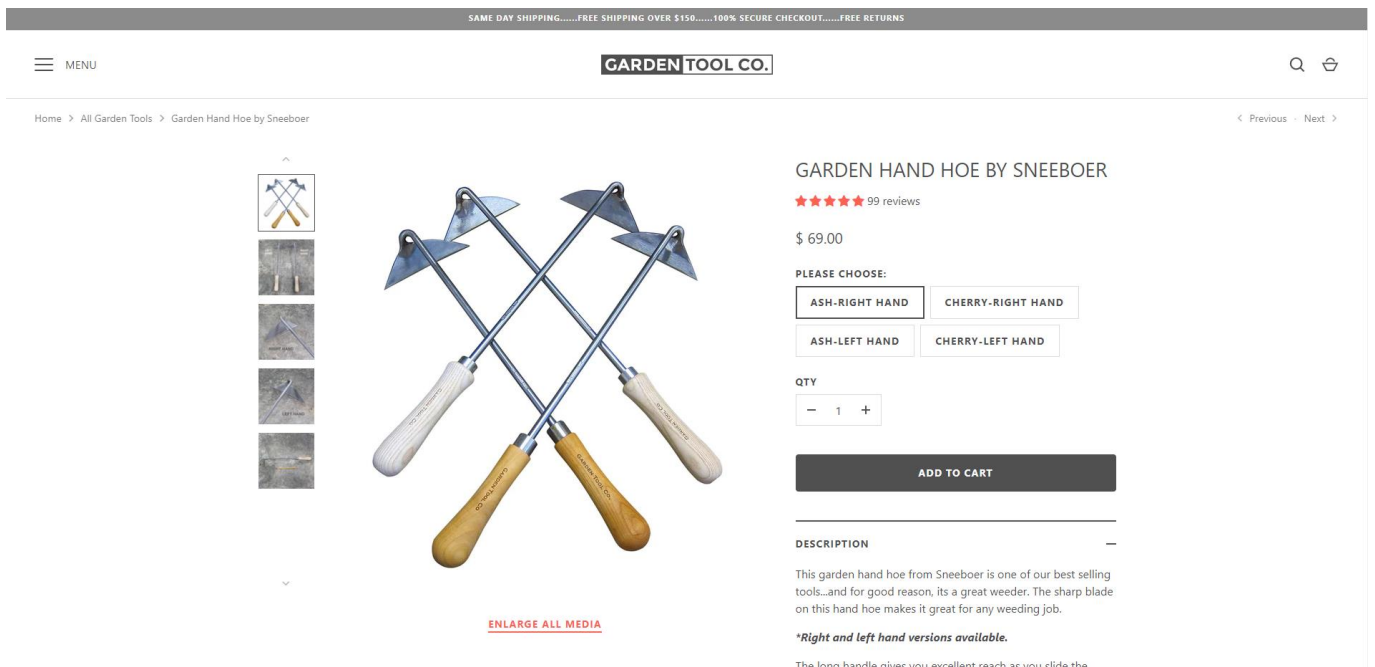


Рисунок 1.3 – Сторінка одного з товарів «GardenTools.com»

GreenThumbGardenSupplies.com інтернет-магазин, який спеціалізується на садових та городніх засобах догляду. Вони пропонують широкий вибір добрив, засобів захисту рослин, насіння, розсади та інших необхідних матеріалів. GreenThumbGardenSupplies.com забезпечує зручне замовлення через Інтернет [11], а також надає детальні описи продуктів та поради щодо їх використання зображена на рис.1.4-1.5.

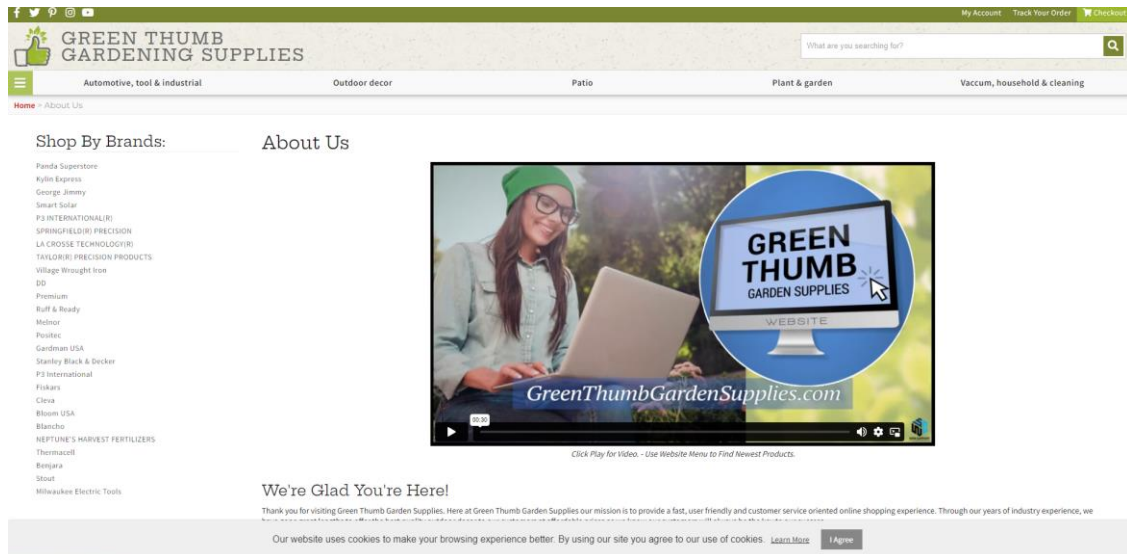


Рисунок 1.4 – Головна сторінка сайту «GreenThumbGardenSupplies.com»



Рисунок 1.5 – Набір товарів «GreenThumbGardenSupplies.com»

"TheGardenDepot.com" – Цей магазин пропонує повний спектр товарів для саду та городу. Він має великий вибір рослин, інструментів, меблів, декоративних елементів та інших аксесуарів для створення та прикрасення садового простору. TheGardenDepot.com також надає консультації щодо ландшафтного дизайну та догляду за рослинами, що дозволяє клієнтам отримати комплексну підтримку у вирощуванні та декоруванні свого саду показано на рис.1.6.

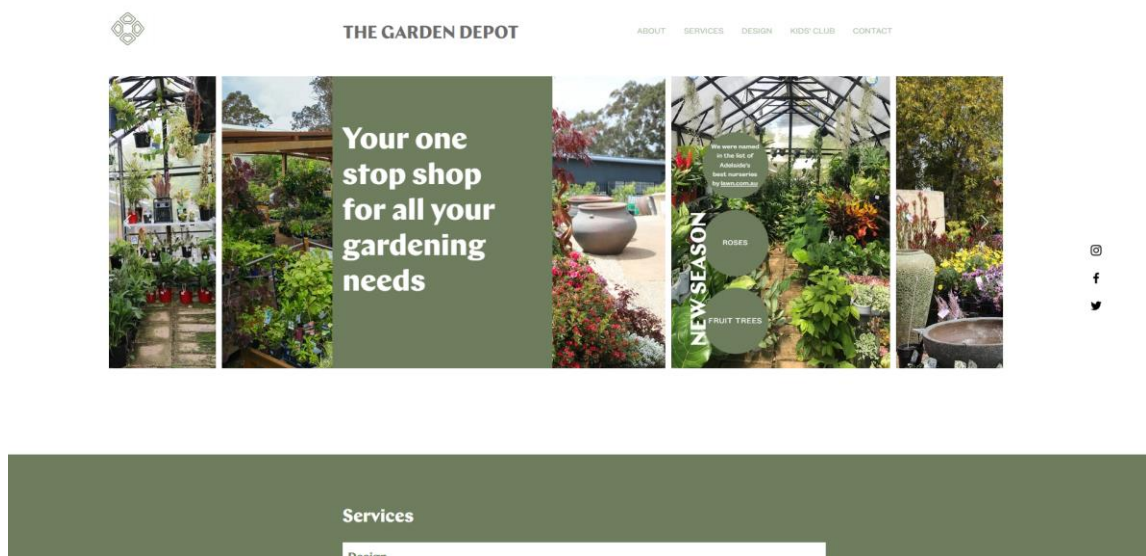


Рисунок 1.6 – Сторінка сайту «TheGardenDepot.com»

При описі інших словами можна сказати, що дизайн сайту видається незавершеним і не повністю задовольняє потреби користувачів. Використання білого фону, розташування елементів на сторінці та колірна гамма потребують додаткової роботи та удосконалення [13].

Також можна зауважити, що сайт не має кошика для замовлень і є не зовсім зручним у процесі формування замовлення. Заповнення інформації може бути незручним для користувача. Крім того, при виборі способу доставки замовлення користувач може обрати лише з обмеженого переліку доступних пунктів видачі та відсутність перегляду товарів без реєстрації (рис.1.7-1.8).

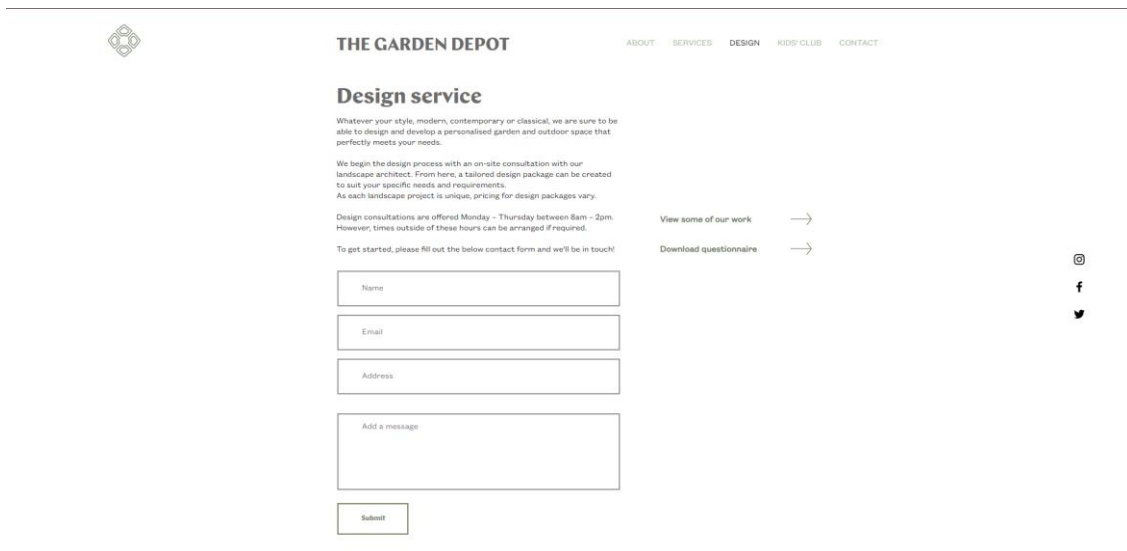


Рисунок 1.7 – Сторінка реєстрації «TheGardenDepot.com»

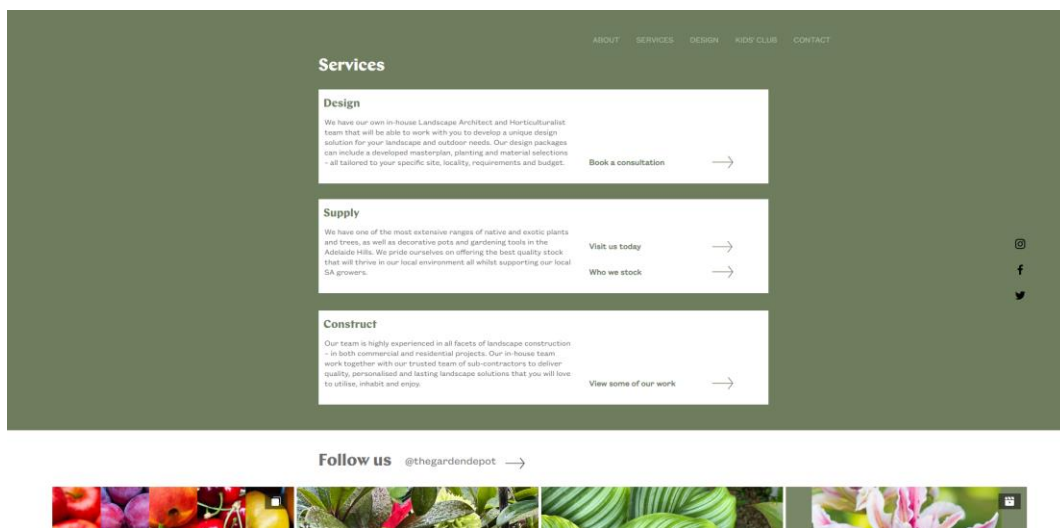


Рисунок 1.8 – Приклад сервісу на «TheGardenDepot.com»

Після аналізування комерційних web-сторінок, які продають садовий інвентар і представлені в пошукових системах, був проведений порівняльний аналіз (див. табл. 1.1), щоб виявити позитивні та негативні аспекти, які слід враховувати при створенні web-додатку для підтримки продажу садового інвентарю.

Серед недоліків можна виділити наступне:

- відсутність валідації даних, що може призвести до проблем з некоректними введеними даними користувачів;
- не чітке розподілення інформації на сайті, що може збентежити відвідувачів і ускладнити їхню навігацію;

- дизайн інтернет-магазину продажу садового інвентарю, який може бути недостатньо привабливим або не дотримуватися стандартів користувальницької зручності;

- не завжди доступна можливість детального перегляду товару, що може ускладнити прийняття рішення покупки для клієнтів;

Серед переваг аналогічних сайтів можна виділити:

- розподіл садового інвентарю за категоріями, що полегшує пошук і спрощує навігацію по сайту;

- можливість зворотного зв'язку, що дозволяє користувачам ставити питання або надсилати відгуки, покращуючи комунікацію з клієнтами;

- зручність навігації, що сприяє легкому переміщенню по сайту та зручному доступу до необхідної інформації.

Ці фактори слід враховувати при розробці web-додатку для продажу садового інвентарю з метою забезпечення позитивного користувацького досвіду та підвищення ефективності продажу.

Таблиця 1.1 – Аналіз комерційних сайтів продажу садового інвентарю

Критерії	GardenTools	GreenThumb GardenSupplies	TheGarden Depot
Валідація даних	+	+	–
Розподіл інформації	+	–	–
Зворотній зв'язок	–	+	–
Структурність інформації	–	+	+
Комфортна навігація	+	+	+
Привабливий дизайн інтернет-магазину	+	+	–
Наявність детальної інформації про товар	+	+	–
Оптимізація мобільної версії	+	+	+

Отже, web-додаток повинен мати інтуїтивно зрозумілу та зручну навігацію, щоб користувачі могли легко переміщатися між різними сторінками та розділами. Дизайн web-додатку повинен бути простим, лаконічним і приємним для очей, з увагою до читабельності тексту та розташування елементів.

Особлива увага повинна бути приділена функціональності, пов'язаній з оформленням замовлення. Користувачі повинні мати можливість зручно вибрати продукти, додавати їх у кошик, змінювати кількість товарів і виконувати інші операції, пов'язані з процесом замовлення. Крім того, додаток повинен надавати можливість користувачам вказати доставку, спосіб оплати та надати необхідну інформацію для успішного оформлення замовлення [14].

Загалом, мета web-додатку полягає в тому, щоб забезпечити зручне, ефективне та безпроблемне оформлення замовлення для користувачів, враховуючи їх потреби та забезпечуючи їх задоволення від процесу покупки.

1.3. Мета та задачі дослідження web-додатку

Головна мета проекту полягає у створенні web-додатку для магазину садового інвентарю. Цей додаток буде надавати швидку та зручну можливість оформлення замовлень, а також різних типів замовлень.

Web-додаток, який розробляється, спрямований на розвиток та поліпшення сервісів, пов'язаних з продажем садового інвентарю. Web-додаток буде розроблений з урахуванням зручного використання його споживачами. У ньому будуть реалізовані різні сервіси, такі як онлайн-замовлення, перегляд інформації, оформлення доставки та інші.

Крім того, ефективне просування web-сайту в соціальних мережах за допомогою таргетованої реклами дозволить досягти хороших результатів і привернути більше уваги до магазину садового інвентарю.

Web-додаток повинен бути організований у взаємозалежні розділи з чітко визначеними функціями. Основні розділи, які повинні бути присутні в додатку, включають:

- авторизація: цей розділ дозволяє користувачам увійти до свого облікового запису за допомогою ідентифікаційних даних, таких як логін і пароль;
- додавання відгуків: користувачі можуть залишати свої відгуки та оцінки для товарів, що допомагає іншим покупцям прийняти рішення;
- перегляд товарів: цей розділ дозволяє користувачам переглядати детальну інформацію про товари, включаючи фотографії, опис та ціни;
- пошук товарів: функція пошуку дозволяє користувачам швидко знайти певний товар за допомогою ключових слів або фільтрів;
- додавання товарів в кошик: користувачі можуть додавати обрані товари до свого кошика покупок перед оформленням замовлення;
- редагування даних: цей розділ дозволяє користувачам змінювати свої особисті дані, такі як адреса доставки або контактна інформація;
- перегляд замовлень: користувачі можуть переглядати історію своїх замовлень та статуси доставки;

- оформлення замовлень: ця функція дозволяє користувачам оформити замовлення, вибрати спосіб доставки та оплати;
- підтвердження замовлення: після оформлення замовлення, користувачі отримують підтвердження про прийняття замовлення з деталями та номером замовлення.

Ці розділи та функції допоможуть забезпечити зручну та комплексну функціональність web-додатка для продажу садового інвентарю.

Розробка web-додатка передбачає кілька етапів. Перший етап - розробка макета, який буде створений за допомогою інструменту Figma. Наступним етапом буде робота над візуальною частиною, що включає створення сторінок сайту з використанням HTML та CSS [15].

Для розробки візуальної частини також буде використовуватися Vue.js. Цей фреймворк працює на клієнтській стороні та не вимагає використання серверного програмного забезпечення або back-end. Vue.js добре поєднується з іншими проектами та бібліотеками, і має широкий функціонал для розробки потужних додатків.

При розробці back-end частини зазвичай використовуються різні бібліотеки, API, web-сервіси та фреймворки [16]. Вони відповідають за роботу з базами даних, забезпечують комунікацію між різними web-сервісами та реалізацію серверної функціональності.

Кожен фреймворк має свої переваги та недоліки, і вибір конкретного фреймворку залежить від вимог проекту та вподобань розробників.

СУБД MySQL є надійним, швидким та ефективним рішенням для зберігання великих обсягів даних. Вона підтримує багато функцій, таких як індексування, транзакції, реплікація та безпека даних. MySQL проста у використанні, має дружній інтерфейс і широкий спектр інструментів для роботи з базами даних [17].

MySQL має активну спільноту розробників і регулярно оновлюється, що забезпечує підтримку та вдосконалення цієї СУБД. Загалом, MySQL є потужним інструментом для роботи з базами даних, який дозволяє зберігати і керувати даними ефективно та надійно.

Можливості технологій для back-end розробки в табл.1.2.

Таблиця 1.2 – Порівняльна таблиця технологій для back-end розробки

№	Назва	Переваги	Недоліки
1	Laravel	Комфортний спосіб визначення шляхів до сторінок.	Проблеми та не точності під час роботи.
		Фреймворк надає інструмент командного рядка.	Потребує значних зусиль.
		Наявність бібліотеки шаблонів.	
		Присутня документація.	
2	Node.js	Технологічна структура стека.	Нестабільна апаратна частина.
		Велика швидкість обробки.	Недостатня кількість фахівців.
		Присутня підтримка спільноти.	Важкість освоєння програмного забезпечення.
		Наявна підтримка JSON.	
3	Spring	Моделює окремі Spring.	Не підходить для великих проєктів.
		Має максимальне програмне охоплення.	Потрібен базовий рівень Java.
		Підтримує хмарні технології в масштабі.	Spring Boot знімає з розробника більшу частину функцій.
		Чудова документація.	

Після уважного аналізу було вирішено обрати Laravel для розробки проєкту на PHP. Цей фреймворк є відмінним вибором, оскільки він містить всі необхідні функції для створення додатків.

Для роботи з базою даних буде використовуватися MySQL, яка є безкоштовною системою управління реляційними базами даних з відкритим кодом [18]. Ця система широко використовується у всьому світі та замінює стандартний SQL. Вона часто використовується разом з мовою сценаріїв.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1. Вибір засобів реалізації

Структурно-функціональна модель процесу роботи web-додатку була створена за допомогою нотації IDEF0. Ця модель демонструє всі дані та вхідну інформацію, які використовуються в web-додатку для замовлення кондитерських виробів. Діаграма нульового рівня, яка відображає цей процес, зображена на рисунку 2.1.

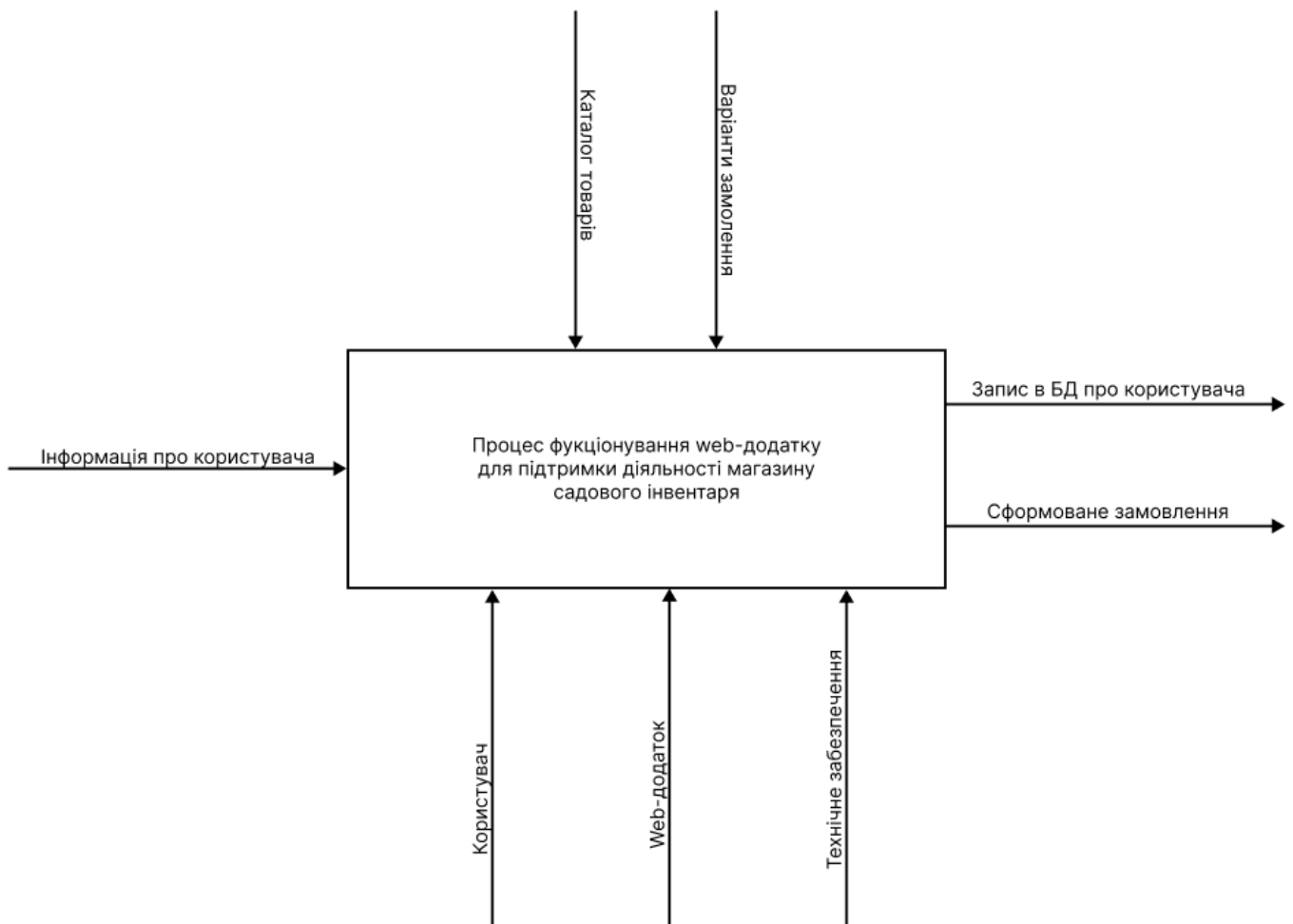


Рисунок 2.1 – Контекстна діаграма IDEF0

На рисунку 2.2 показано варіант діаграми декомпозиції в нотації IDEF0, яка включає детальне розкриття кожного етапу проекту. Ця діаграма відображає послідовність кроків та процесів, які виконуються в проекті [19]. Вона допомагає

зрозуміти структуру та взаємозв'язки між різними елементами проекту і детально представляє кожен етап його виконання.

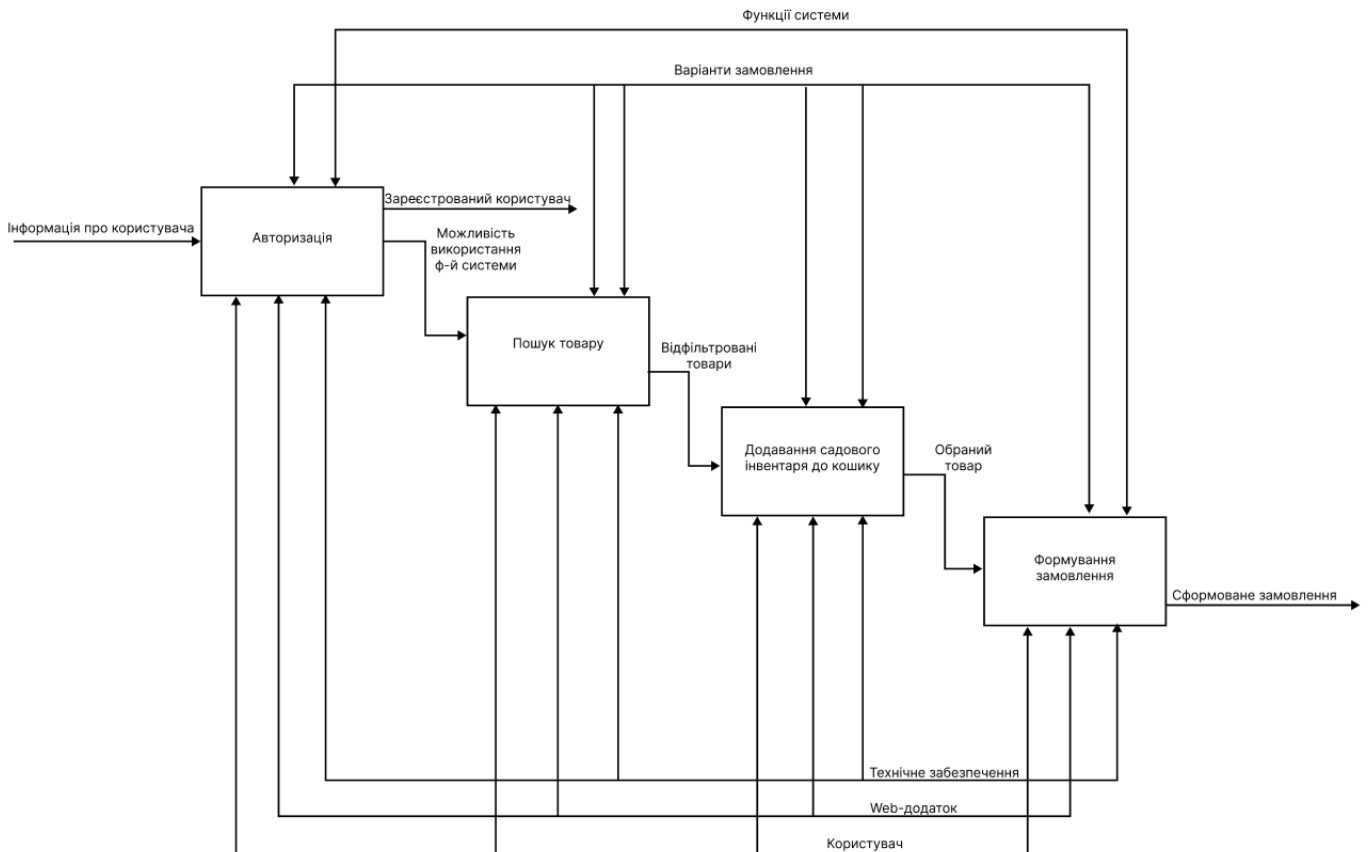


Рисунок 2.2 – Діаграма декомпозиції IDEF0

2.2. Проектування web-додатку

Розгляд системи web-додатку (табл.2.1) та варіантів використання (табл.2.2).

Табл. 2.1 – Актори-користувачі

Назва	Опис
Користувач	Звичайний користувач web-додатку.
Адміністратор	Користувач у ролі «адміністратора».

Табл. 2.2 – Опис варіантів використання

Назва	Опис
Авторизація	Можливість входу за допомогою облікових даних.
Відгуки	Можливість розповсюдити досвід використання садового інвентарю.
Огляд товару	Можливість для користувача переглядати різні товари для покупки.
Фільтр товару	Можливість для користувача здійснювати пошук товарів за назвою, ціною та категорією.
Придбання товару	Можливість користувача додати певний товар до свого кошика.
Редагування даних	Можливість адміністратора змінювати дані.
Доповнювання асортименту	Можливість адміністратора доповнювати асортимент товарів.
Огляд замовлень	Можливість адміністратора переглядати покупки користувачів.
Створення замовлень	Можливість користувача створити заявку на замовлення певного товару.
Підтвердження отриманих замовлень	Можливість адміністратора підтверджувати отримані замовлення.

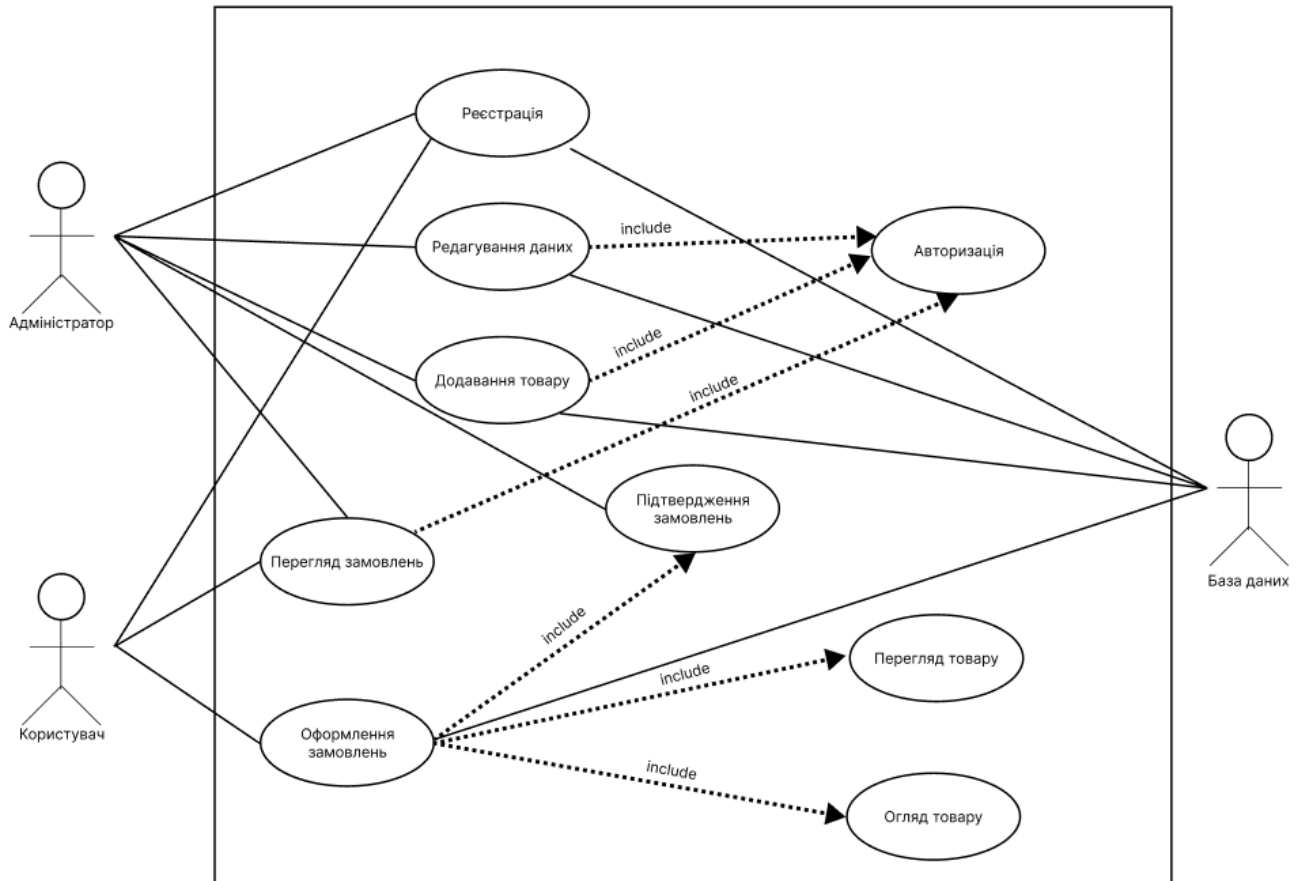


Рисунок 2.3 – Варіанти використання web-додатку підтримки діяльності садових інвентаря

2.3. Проектування бази даних

Головний користувач web-додатку – покупець садового інвентарю. Користувач, озброївшись web-додатком, може вибрати товар та створити замовлення.

Побудовано ER-діаграму бази даних для web-додатку (рис.2.4). Після аналізу сутностей, що використовуються в інформаційній системі, переходимо до реалізації структури бази даних. Детальна інформація про кожну таблицю наведена в таблиці 2.3.

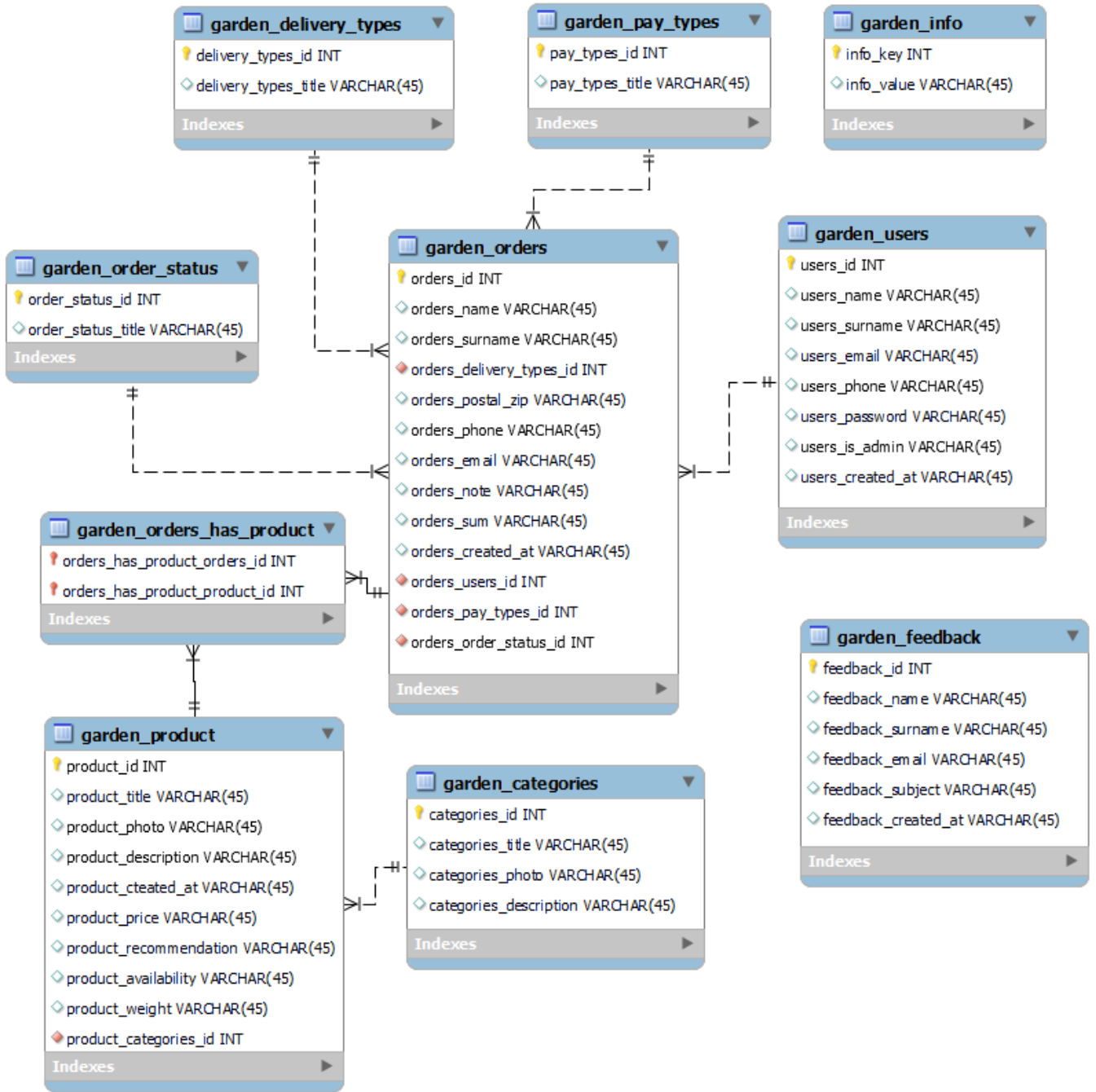


Рисунок 2.4 – ER-діаграма бази даних web-додатку підтримки діяльності магазину садового інвентаря на фізичному рівні

Таблиця 2.3 – Таблицями ER-діаграми web-додатку підтримки магазину садового інвентаря

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
1	garden_users	users_id	Ідентифікатор аккаунта користувача	INTEGER	PK	Не пустий
		users_name	Ім'я користувача технології	VARCHAR(100)		Не пустий
		users_surname	Прізвище користувача	VARCHAR(100)		Не пустий
		users_email	Електронна пошта користувача	VARCHAR(50)		Не пустий
		users_phone	Телефон користувача	VARCHAR(50)		Не пустий
		users_password	Пароль користувача	VARCHAR(50)		Не пустий
		users_is_admin	Помітка що користувач адміністратор	TINYINT		Не пустий
		users_created_at	Дата реєстрація	DATETIME		Не пустий
2	garden_orders	orders_id	Ідентифікатор замовлення	INTEGER	PK	Не пустий
		orders_name	Ім'я замовника	VARCHAR(100)		Не пустий
		orders_surname	Прізвище замовника	VARCHAR(100)		Не пустий
		orders_delivery_types_id	Ідентифікатор типу доставки	INTEGER	FK	Не пустий
		orders_postal_zip	Номер відділення	INTEGER		
		orders_phone	Телефон замовника	VARCHAR(100)		Не пустий
		orders_email	Email замовника	VARCHAR(100)		Не пустий
		orders_note	Додатковий коментар	TEXT		
		orders_sum	Сума замовлення	DECIMAL		Не пустий

Продовження таблиці 2.3

		orders_created_at	Дата створення замовлення	DATETIME		Не пустий
		orders_users_id	Ідентифікатор замовника (якщо користувач зареєстрований)	INTEGER	FK	
		orders_pay_types_id	Ідентифікатор типу оплати	INTEGER	FK	Не пустий
		orders_order_status_id	Ідентифікатор статусу замовлення	INTEGER	FK	Не пустий
3	garden_product	product_id	Ідентифікатор товару	INTEGER	PK	Не пустий
		product_title	Назва товару	VARCHAR(255)		Не пустий
		product_photo	Фото товару	VARCHAR(255)		
		product_description	Опис товару	TEXT		Не пустий
		product_created_at	Дата публікації товару	DATETIME		Не пустий
		product_price	Ціна товару	DECIMAL		Не пустий
		product_recommendation	Рекомендований товар	TINYINT		Не пустий
		product_availability	Наявність товару	TINYINT		Не пустий
		product_weight	Вага товару	DECIMAL		Не пустий
		product_categories_id	Ідентифікатор категорії товару	INTEGER	FK	Не пустий
4	garden_orders_has_product	orders_has_product_orders_id	Ідентифікатор замовлення	INTEGER	FK	Не пустий
		orders_has_product_product_id	Ідентифікатор товару	INTEGER	FK	Не пустий
5	garden_categories	categories_id	Ідентифікатор категорії	INTEGER	PK	Не пустий
		categories_title	Назва категорії	VARCHAR(255)		Не пустий

Продовження таблиці 2.3

		categories_photo	Фото категорії	VARCHAR(255)		Не пустий
		categories_description	Опис категорії	TEXT		
6	garden_order_statuses	order_status_id	Ідентифікатор статусу	INTEGER	PK	Не пустий
		order_status_title	Назва статусу	VARCHAR(100)		Не пустий
7	garden_delivery_types	delivery_types_id	Ідентифікатор типу доставки	INTEGER	PK	Не пустий
		delivery_types_title	Назва типу доставки	VARCHAR(100)		Не пустий
8	garden_pay_types	pay_types_id	Ідентифікатор типу оплати	INTEGER	PK	Не пустий
		pay_types_title	Назва типу оплати	VARCHAR(100)		Не пустий
9	garden_feedback	feedback_id	Ідентифікатор відгуку	INTEGER	PK	Не пустий
		feedback_name	Ім'я користувача	VARCHAR(100)		Не пустий
		feedback_surname	Прізвище користувача	VARCHAR(100)		Не пустий
		feedback_email	Email користувача	VARCHAR(100)		Не пустий
		feedback_subject	Коментар	TEXT		Не пустий
		feedback_created_at	Дата створення	DATETIME		Не пустий
10	garden_info	info_key	Ключ інформації	VARCHAR(100)		Не пустий
		info_value	Значення	TEXT		Не пустий

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ

3.1 Архітектура web-додатку

Для створення web-додатку використовувався фреймворк PHP під назвою Laravel. Цей фреймворк дозволяє розробляти web-додатки, використовуючи архітектурний підхід MVC (Model-View-Controller) [20]. Архітектура web-додатку, яку було використано, зображена на рисунку 3.1.

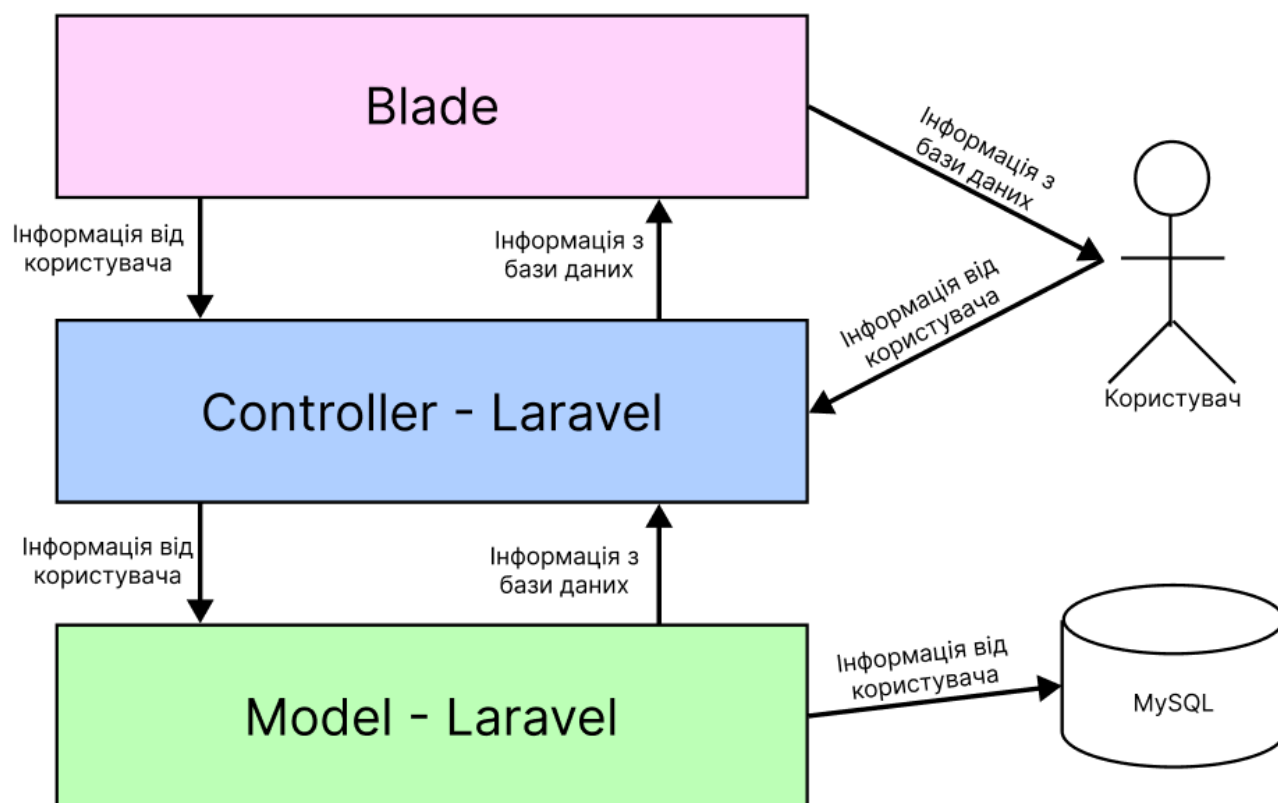


Рисунок 3.1 – Архітектура web-додатку

Модель (Model) - це компонент, який містить дані та логіку, пов'язану з цими даними. Він відповідає за зберігання та обробку даних, взаємодію з базою даних, валідацію та виконання бізнес-логіки. Модель визначає, які дані будуть доступні та як їх можна маніпулювати.

Візуальне представлення (Blade) - це компонент, який відповідає за відображення даних користувачеві та обробку даних, які надходять від користувача. Він містить шаблони HTML-сторінок, які можуть містити вбудований PHP-код для виведення даних, циклів, умовних операторів та інших динамічних елементів [21].

Контролер (Controller) - це компонент, який діє як посередник між компонентами Model та Blade. Він приймає запити від користувача, взаємодіє з моделлю для отримання необхідних даних, і передає ці дані візуальному представленню для подальшого відображення. Контролер також відповідає за обробку подій та взаємодію з іншими компонентами системи.

Таким чином, web-додаток, побудований на основі архітектури MVC, використовує модель для роботи з даними, візуальне представлення для відображення цих даних та контролер для забезпечення взаємодії між ними.

3.2 Програмна реалізація

У процесі розробки web-додатку було використано текстовий редактор Visual Studio Code. Visual Studio Code - це редактор сценаріїв, який був створений компанією Microsoft і працює на різних платформах. Він надає широкий набір функцій для редагування сценаріїв, і будь-хто може використовувати його для написання сценаріїв PowerShell [22].

Крім того, була розроблена база даних, для чого було використано додаток phpMyAdmin. phpMyAdmin - це програмний інструмент, який дозволяє керувати базами даних MySQL через веб-інтерфейс. З його допомогою можна створювати таблиці, виконувати запити, налаштовувати права доступу та багато іншого. Розглянуті створені таблиці можна знайти на рисунках 3.2-3.11.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 categories_title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	3 categories_description	varchar(255)	utf8mb4_unicode_ci		Да	NULL			
<input type="checkbox"/>	4 categories_photo	varchar(255)	utf8mb4_unicode_ci		Нет	/img/no-photo-available.png			
<input type="checkbox"/>	5 created_at	timestamp			Да	NULL			
<input type="checkbox"/>	6 updated_at	timestamp			Да	NULL			

Рисунок 3.2 – Таблица «categories»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 delivery_types_title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	3 created_at	timestamp			Да	NULL			
<input type="checkbox"/>	4 updated_at	timestamp			Да	NULL			

Рисунок 3.3 – Таблица «delivery»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 feedback_name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	3 feedback_surname	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	4 feedback_email	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	5 feedback_subject	text	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	6 created_at	timestamp			Да	NULL			
<input type="checkbox"/>	7 updated_at	timestamp			Да	NULL			

Рисунок 3.4 – Таблица «feedback»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 info_key	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	3 info_value	longtext	utf8mb4_unicode_ci		Нет	Нет			

Рисунок 3.5 – Таблица «info»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 order_products_orders_id	bigint(20)		UNSIGNED	Нет	Нет			
<input type="checkbox"/>	3 order_products_product_id	bigint(20)		UNSIGNED	Нет	Нет			
<input type="checkbox"/>	4 order_products_quantity	int(11)			Нет	Нет			

Рисунок 3.6 – Таблица «order_products»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 order_status_title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			

Рисунок 3.7 – Таблица «order_status»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 orders_name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	3 orders_surname	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	4 orders_delivery_types_id	bigint(20)		UNSIGNED	Нет	Нет			
<input type="checkbox"/>	5 orders_postal_zip	varchar(255)	utf8mb4_unicode_ci		Да	NULL			
<input type="checkbox"/>	6 orders_email	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	7 orders_phone	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	8 orders_note	text	utf8mb4_unicode_ci		Да	NULL			
<input type="checkbox"/>	9 orders_users_id	bigint(20)		UNSIGNED	Да	NULL			
<input type="checkbox"/>	10 orders_sum	text	utf8mb4_unicode_ci		Нет	Нет			
<input type="checkbox"/>	11 orders_pay_type_id	int(11)			Нет	Нет			
<input type="checkbox"/>	12 orders_order_status_id	bigint(20)		UNSIGNED	Нет	1			
<input type="checkbox"/>	13 created_at	timestamp			Да	NULL			

Рисунок 3.8 – Таблица «orders»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 pay_types_title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			

Рисунок 3.9 – Таблица «pay_types»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2 product_title	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	3 product_description	longtext	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	4 product_price	double(8,2)			Нет	Нем			
<input type="checkbox"/>	5 product_photo	varchar(255)	utf8mb4_unicode_ci		Нет	/img/no-photo-available.png			
<input type="checkbox"/>	6 product_recommendation	tinyint(1)			Нет	0			
<input type="checkbox"/>	7 product_availability	tinyint(1)			Нет	1			
<input type="checkbox"/>	8 product_weight	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	9 product_category_id	bigint(20)		UNSIGNED	Нет	Нем			
<input type="checkbox"/>	10 created_at	timestamp			Да	NULL			

Рисунок 3.10 – Таблица «product»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2 users_name	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	3 users_email	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	4 users_password	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	5 users_is_admin	tinyint(1)			Нет	0			
<input type="checkbox"/>	6 users_phone	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	7 users_surname	varchar(255)	utf8mb4_unicode_ci		Нет	Нем			
<input type="checkbox"/>	8 remember_token	varchar(100)	utf8mb4_unicode_ci		Да	NULL			
<input type="checkbox"/>	9 created_at	timestamp			Да	NULL			

Рисунок 3.11 – Таблица «users»

Програмна реалізація web-додатку реалізована з використанням фреймворку Laravel та Vue.js, а саме використовуванні пакети представлені у табл.3.1.

Таблиця 3.1 – Таблица рівнів доступу web-додатку

№	Пакет	Короткий опис
1	app	Файл ядра Laravel web-додатку містить основний код, який визначає функціональність та роботу додатка.
2	bootstrap	Файли, які завантажують фреймворк Bootstrap і налаштовують автозавантаження, відповідають за інтеграцію і налаштування Bootstrap web-компонентів у web-додатку.

Продовження таблиці 3.1

3	config	Конфігураційні файли web-додатку включають налаштування, які визначають поведінку та параметри роботи додатка.
4	database	Міграції та класи, що містяться в цій частині web-додатку, відповідають за створення та оновлення бази даних. Міграції визначають структуру бази даних, включаючи таблиці, стовпці, зв'язки та інші елементи.
5	public	Папка, що містить початковий код та не компільовані ресурси, містить основний набір файлів, які потрібні для роботи web-додатку.
6	resources	Визначення маршрутів web-додатку знаходяться в цій папці.
7	routes	Скомпільовані шаблони, кеш-файли та інші файли, створені фреймворком Laravel, знаходяться в цій папці.
8	storage	В цілому, ця папка містить важливі компоненти та файли, які допомагають фреймворку Laravel працювати ефективно та надавати необхідну функціональність для web-додатку.

3.3 Використання web-додатку в режимі користувача

При переході за посиланням, користувач відкриває головну сторінку web-додатку (показано на рисунках 3.12-3.13). На цій сторінці розміщена загальна інформація, яка включає рекламний банер, вказівки щодо переваг використання цього web-додатка та інформацію про лідерів продажу.

Рекламний банер може бути графічним елементом або текстовою рекламою, які привертають увагу користувача. Його мета - заохотити користувачів до подальшого використання додатку і надати інформацію про його основні переваги та можливості [23,24].

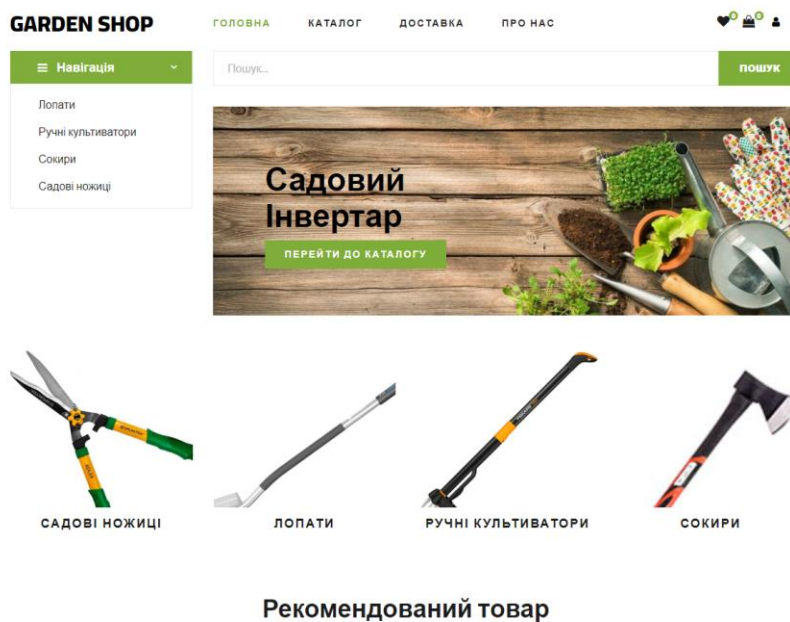


Рисунок 3.12 – Головна сторінка web-додатку (1 частина)

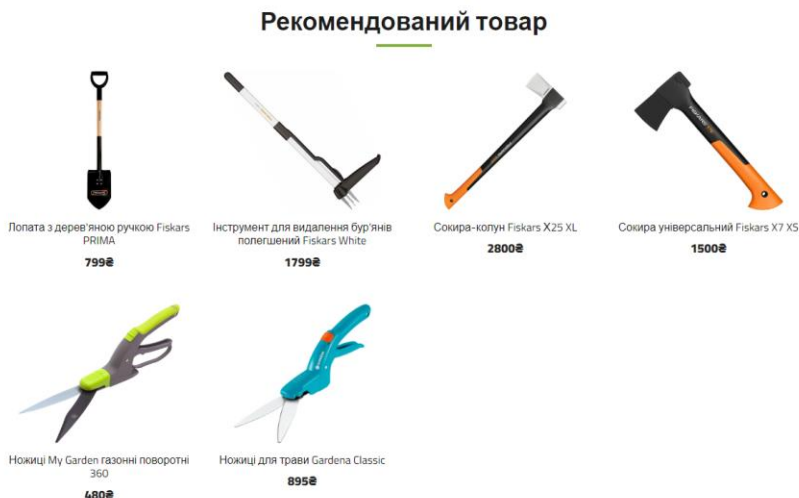


Рисунок 3.13 – Головна сторінка додатку (2 частина)

Реалізовано випадаючий список з категоріями товарів: «Лопати», «Ручні культиватори», «Сокири» та «Садові ножиці» (рис.3.14). Випадаючий список має мінімалістичний дизайн та плавну анімацію появи.

На прикладі показано сторінку «Лопати» (рис.3.15).

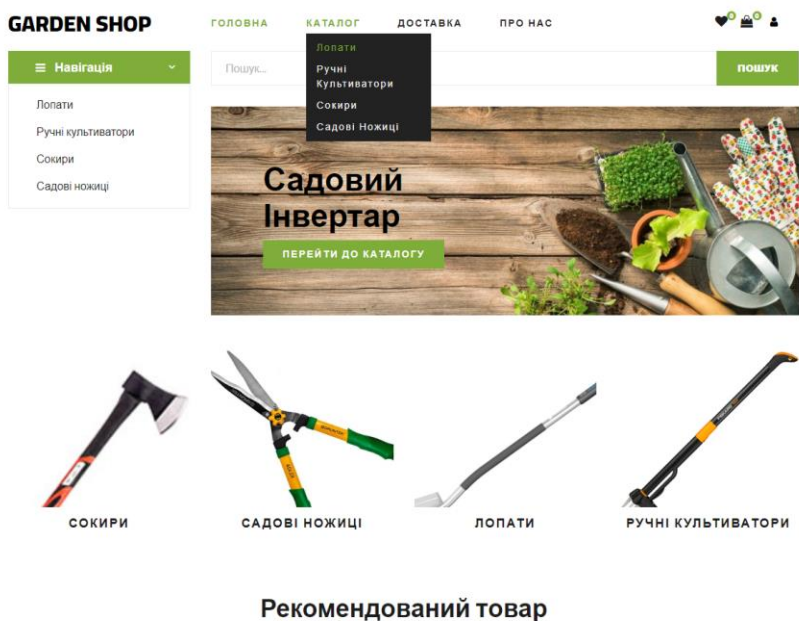


Рисунок 3.14 – Випадаючий список каталогу

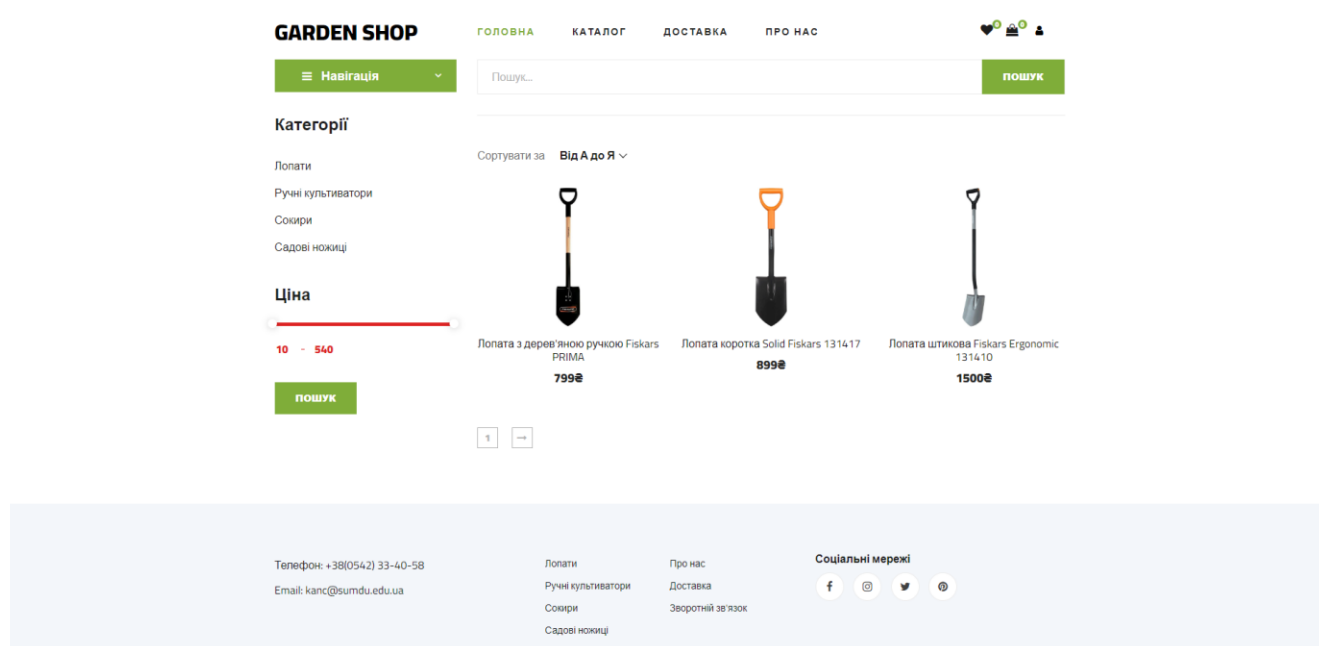


Рисунок 3.15 – Сторінка «Лопати»

На рис.3.16 та рис.3.17 представлено приклад фільтрації та сортування. Сторінка з набором товарів має базовий набір налаштувань сортування та фільтрації, що повністю задовольняє потреби користувача.

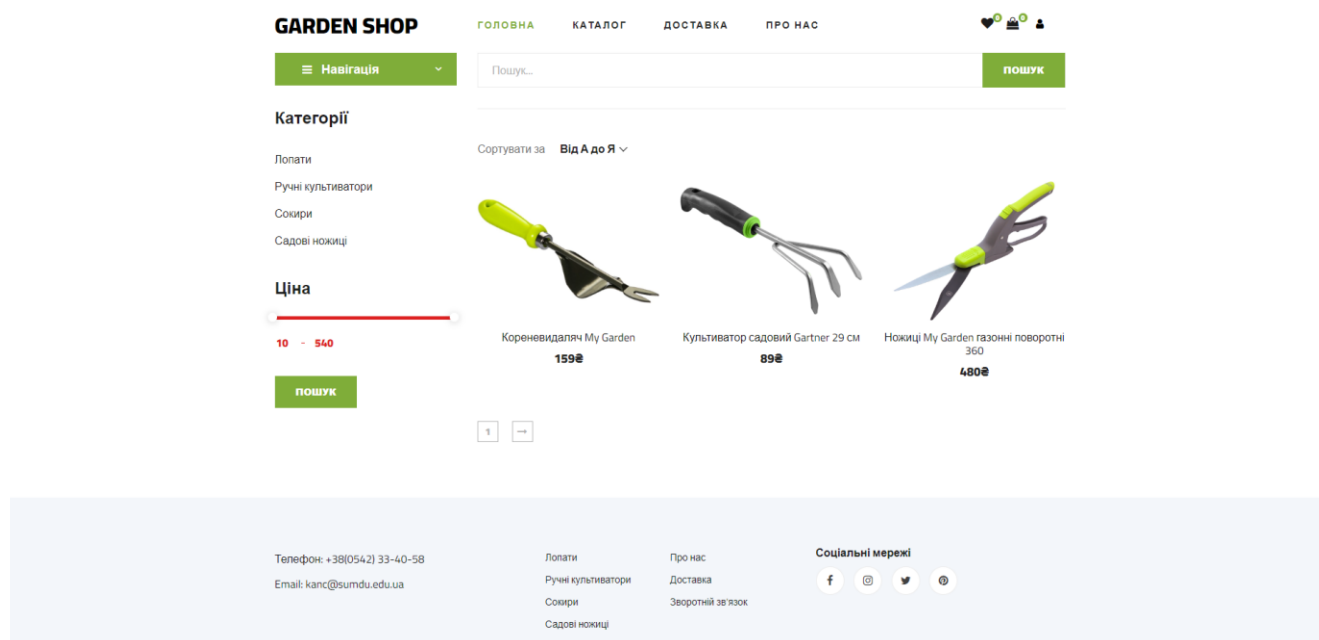


Рисунок 3.16 – Фільтрація за ціною

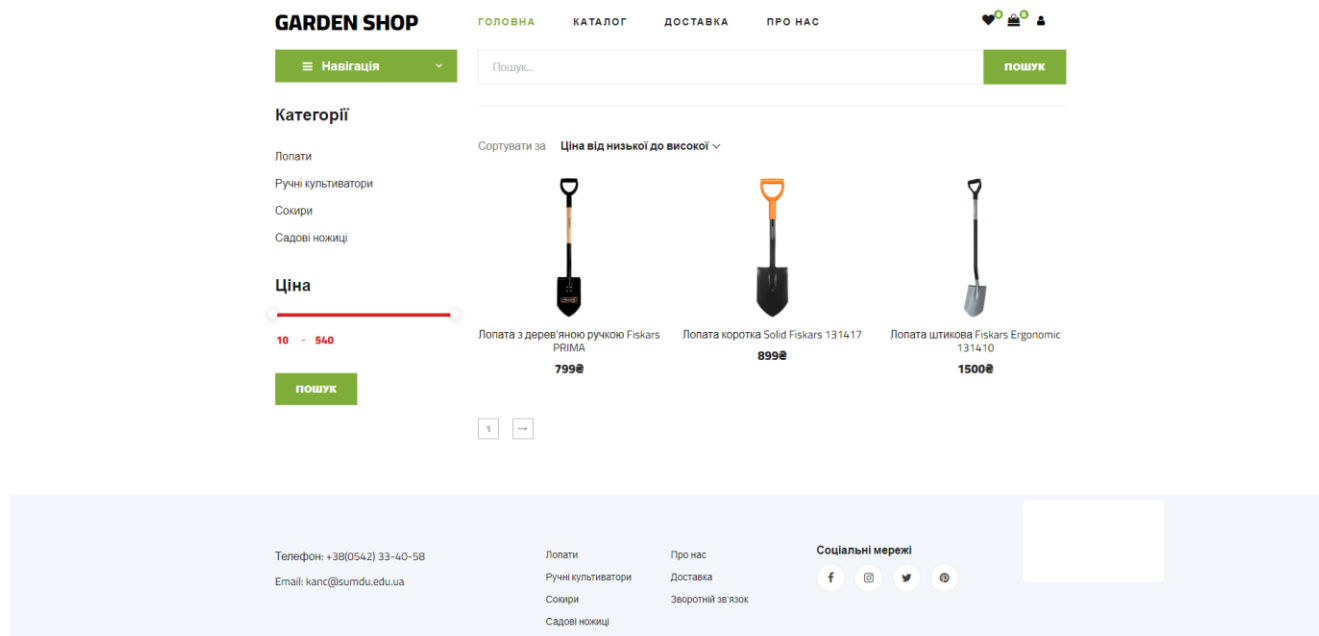


Рисунок 3.17 – Сортування за ціною

Сторінка товару має фото товару, рекомендовані товари та короткий опис (рис.3.17-3.19). Рекомендації показані з урахуванням інтересів користувачів та з поглядом на останні замовлення та переглянуті товари.

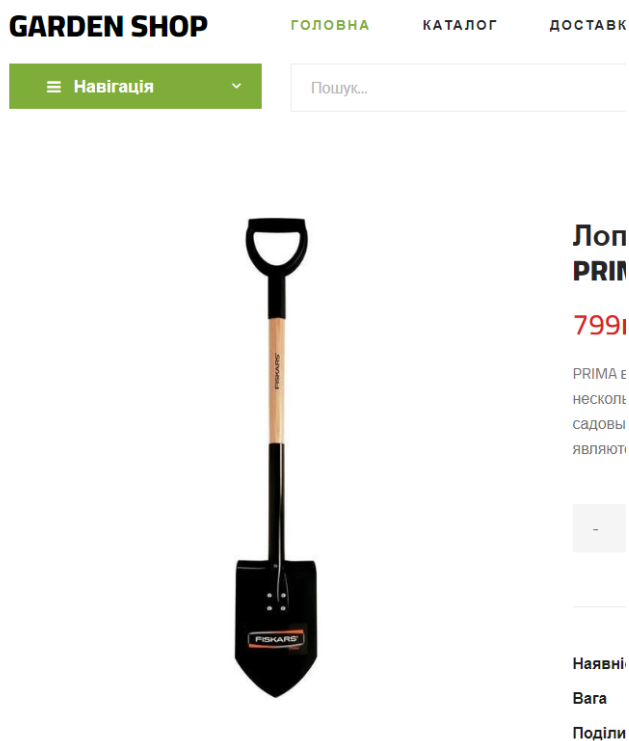


Рисунок 3.17 – Фото товару



Рисунок 3.18 – Рекомендовані товари

Лопата з дерев'яною ручкою Fiskars PRIMA

799грн

PRIMA в перекладі з латинського означає «первокласний». Вот уже несколько лет так называют некоторые из самых популярных в Швеции садовых инструментов — серию качественных продуктов, которые являются безопасным выбором в первую очередь.



Наявність на складі
Вага 12 кг
Поділитись

Рисунок 3.19 – Короткий опис

Користувач може зареєструватися в будь-який момент (рис.3.20). Оскільки зробити замовлення може лише зареєстрований користувач, то виникла потреба надати можливість зареєструватися в будь-який момент.

GARDEN SHOP ГОЛОВНА КАТАЛОГ ДОСТАВКА ПРО НАС

Навігація

Пошук...

Богдан

Дряпко

katamaran.g@gmail.com


Телефон

.....

.....

ЗАРЕЄСТРУВАТИСЬ


Авторизуватись



**Лопата з дерев'янок
PRIMA**





799грн

PRIMA в перекладі з латинської означає кілька років так називають деякі і садові інструменти — серію якісних які є безпечним вибором в першу

- 1 + **ДОДАТИ ДО КОШИКУ** 

Наявність на складі

Вага 12 кг

Поділитись    

Рекомендовані товари

Рисунок 3.20 – Реєстрація користувача

Оформлення замовлення доступне з особистого кошика на сторінці (рис.3.21-3.22). Також на особистій сторінці є можливість збільшити кількість замовлених лотів.

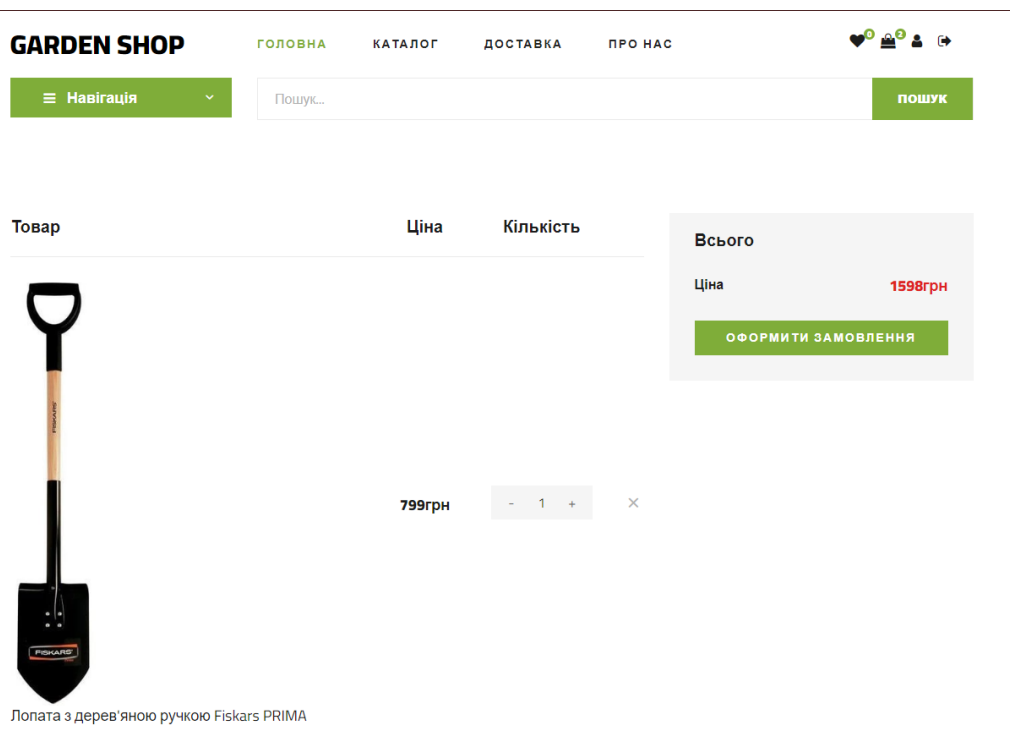


Рисунок 3.21 – Перегляд кошика

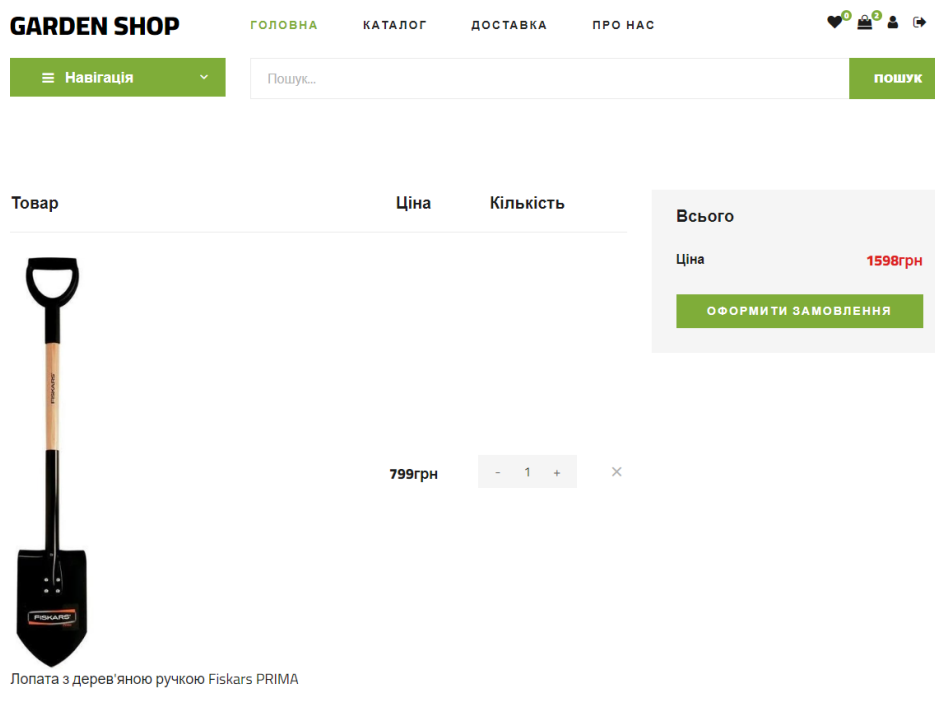


Рисунок 3.22 – Оформлення замовлення

Дані про доставку розміщені на спеціальній сторінці (рис.3.23).

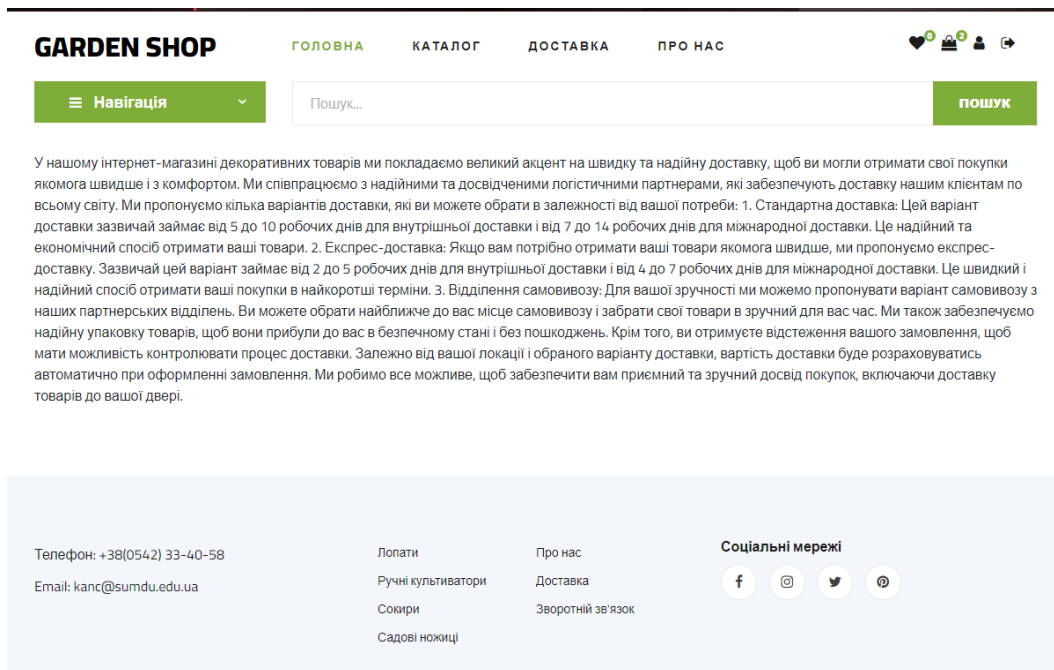


Рисунок 3.23 – Сторінка «Доставка»

Інформацію «Про нас» можна переглянути на відведеній окремій сторінці (рис.3.24).

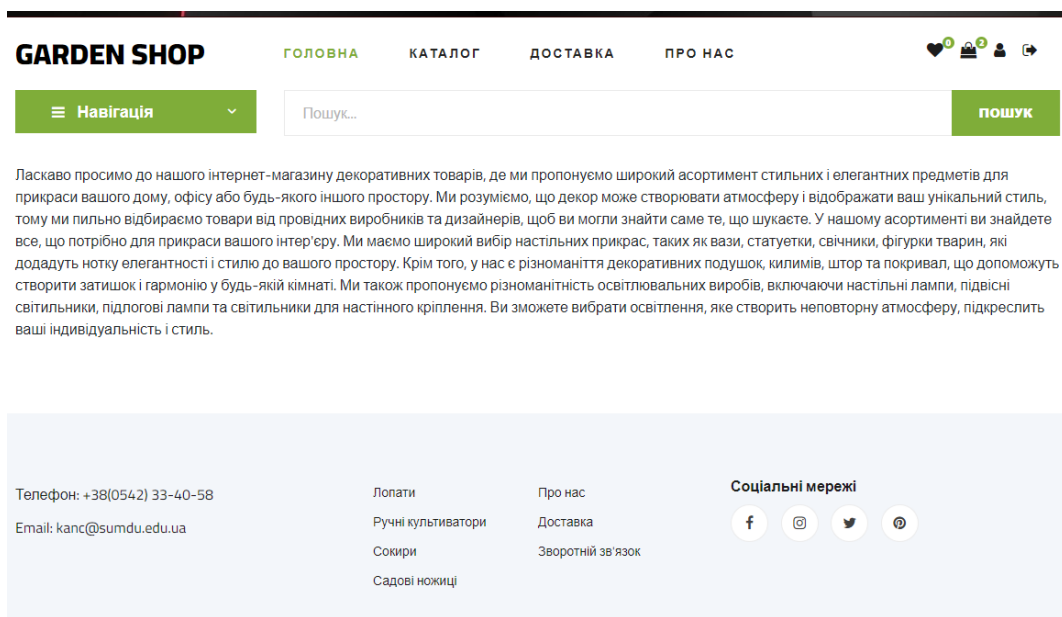


Рисунок 3.24 – Сторінка «Про нас»

Оформити замовлення можна лише після заповнення всіх потрібних полів (рис.3.25). Також реалізована підтримка вибору способу оплати під потреби користувачів.

Контактні дані

Ім'я*	Прізвище*
<input type="text" value="Богдан"/>	<input type="text" value="Дрялко"/>
Спосіб доставки*	Відділення*
<input type="text" value="Нова пошта"/>	<input type="text" value="№3"/>
Телефон*	Email*
<input type="text" value="0626986522"/>	<input type="text" value="katamaran.g@gmail.com"/>
Коментар*	
<input type="text"/>	

Ваше замовлення

Товар

Лопата з дерев'яною ручкою Fiskars

PRIMA	799 грн
Всього	1598

Спосіб оплати

Карта банку

Рахунок

Готівка

ЗАМОВИТИ

Рисунок 3.25 – Заповнення всіх полів

Після успішного оформлення замовлення, користувач отримує подяку у візуальному стилі для створення ефекту діалогу між користувачем та web-додатком (рис.3.26).



Дякуємо за замовлення!

Найблизчим часом менеджер підтвердить ваше замовлення.

ПРОДОВЖИТИ ПОКУПКИ

Рисунок 3.26 – Повідомлення про успішне оформлення замовлення

3.4 Адміністрування сайту

Для успішного адміністрування web-додатку потрібно перейти на спеціальну сторінку та авторизуватися (рис.3.27).

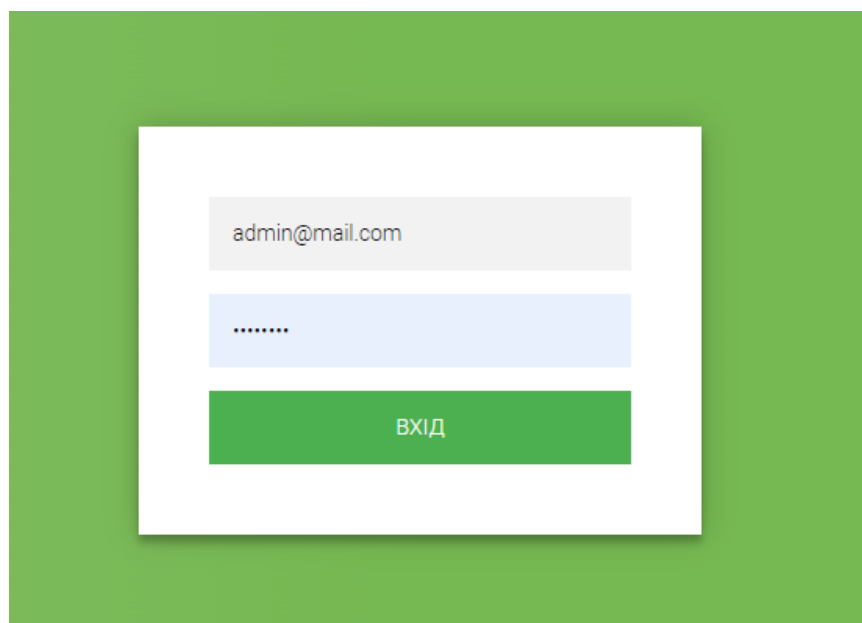


Рисунок 3.27 – Авторизація адміністратора

Першим, що зустрічає адміністратор є сторінка з замовленнями, на якій можна переглянути замовлення користувачів, та детальний опис на рис.3.28-3.29.

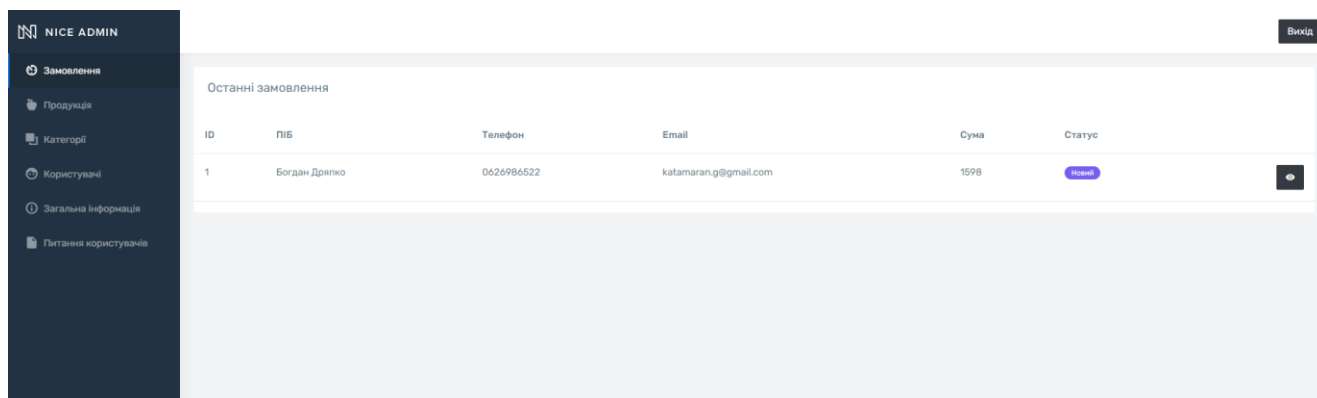


Рисунок 3.28 – Сторінка «Замовлення»

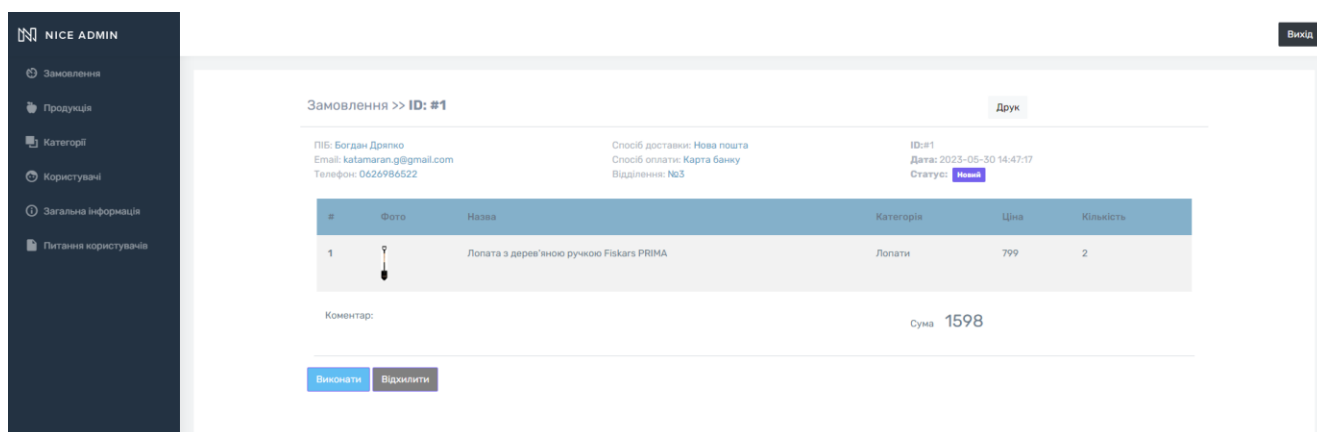


Рисунок 3.29 – Сторінка «Замовлення» з детальним описом

«Продукція» - це слідуюча сторінка для маніпуляцій. На цій сторінці показано товари з якими можна взаємодіяти: додавати, редагувати та видаляти (рис.3.30-3.32).

Зображення	Назва	Категорія	Ціна	Дата додавання	Додати
	Лопата з дерев'яною рукою Fiskars PRIMA	Лопати	799	2023-05-28 18:52:53	
	Лопата коротка Solid Fiskars 131417	Лопати	899	2023-05-28 18:53:45	
	Лопата штикова Fiskars Ergonomic 131410	Лопати	1500	2023-05-28 18:54:28	
	Інструмент для видалення бур'янів полегшений Fiskars White	Ручні культиватори	1799	2023-05-28 18:58:12	
	Кореневидалач My Garden	Ручні культиватори	159	2023-05-28 18:58:49	

Рисунок 3.30 – Сторінка «Продукція»

Назва
Лопата з дерев'яною рукою Fiskars PRIMA

Категорія
Садові ножиці

Ціна
799

Вага Kg
12

Опис
PRIMA в перекладі з латинського означає «перекласний». Вже кілька років так називають деякі з найкращих серій садових інструментів – серію якісних продуктів, які є безпечним вибором в першу чергу.

Рекомендований товар

Наявність на складі

Выберите файл | Файл не выбран

Зберегти

Рисунок 3.31 – Редагування товару

Назва

Категорія
Лопати

Ціна

Вага Kg

Опис

Рекомендований товар

Наявність на складі

Выберите файл | Файл не выбран

Зберегти

Рисунок 3.32 – Додавання товару

«Категорії» - це слідуюча сторінка для маніпуляцій. На цій сторінці показано набір категорій з якими можна взаємодіяти: додавати, редагувати та видаляти (рис.3.33-3.35).

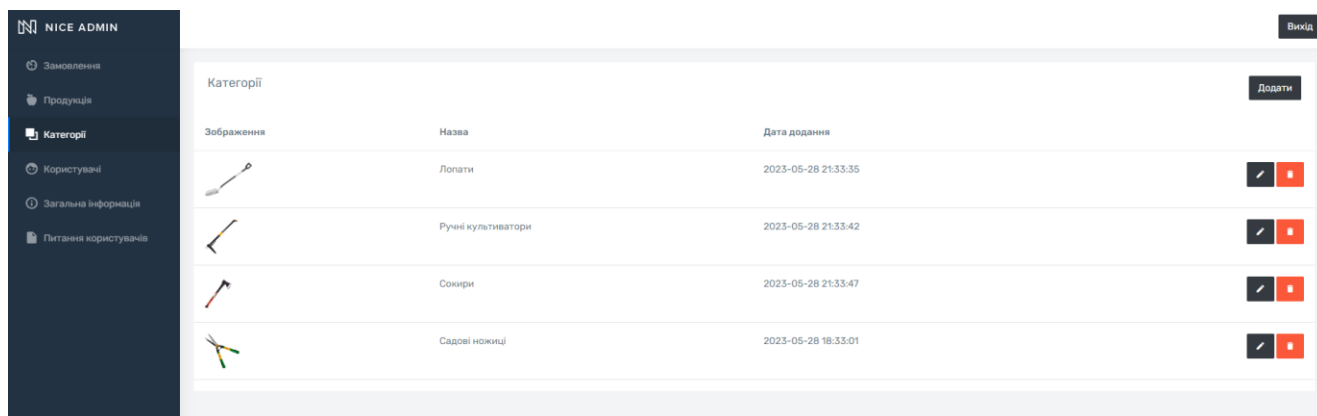


Рисунок 3.33 – Сторінка «Категорії»

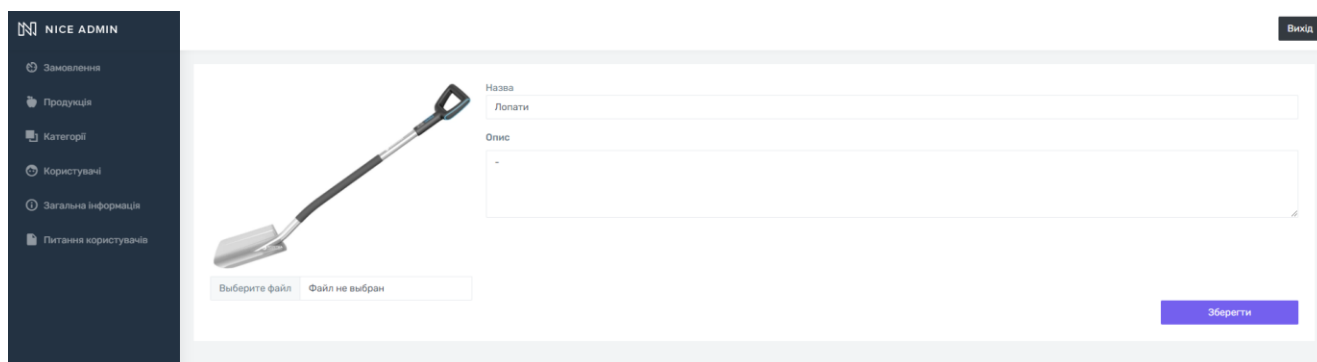


Рисунок 3.34 – Редагування категорії

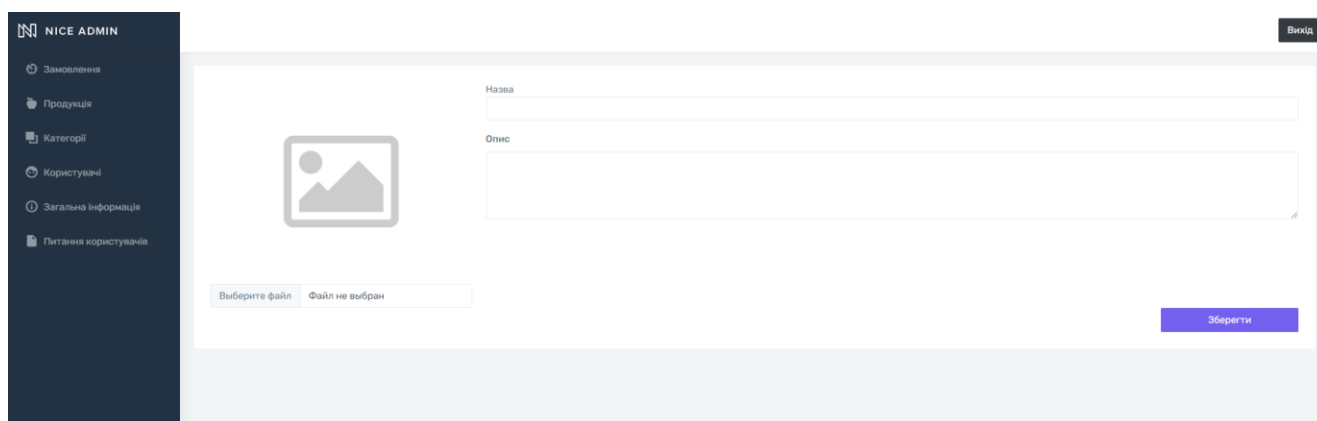


Рисунок 3.35 – Додавання нової категорії

Користувачів та детально інформацію про них можна побачити на сторінці «Користувачі» (рис.3.36).

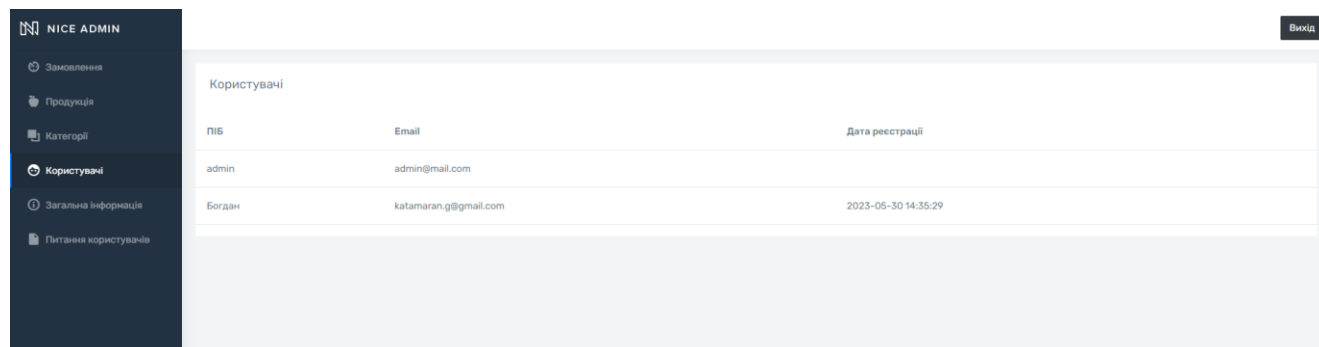


Рисунок 3.36 – Сторінка «Користувачі»

На сторінці «Загальна інформація» присутній інструментарій для редагування контактної інформації та опису доставки (рис.3.37).

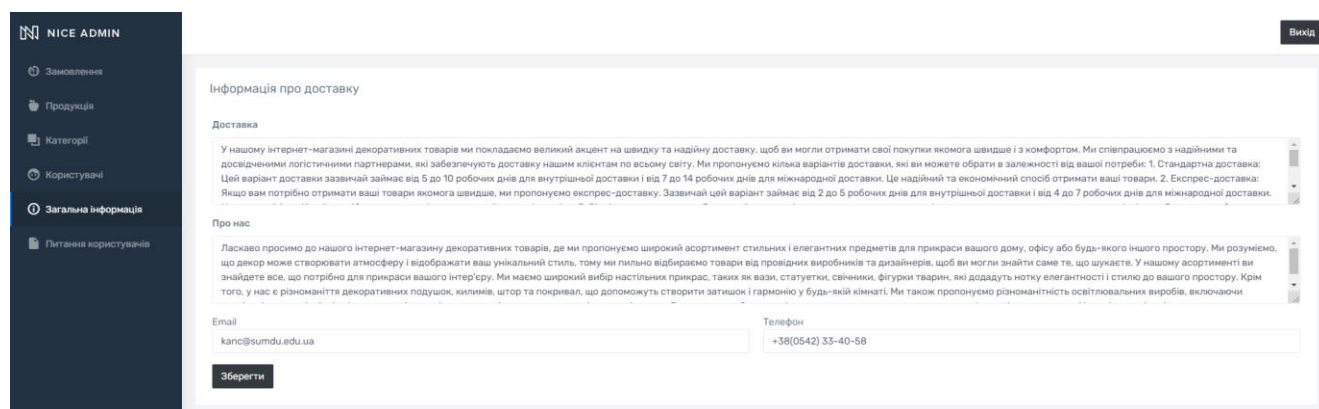
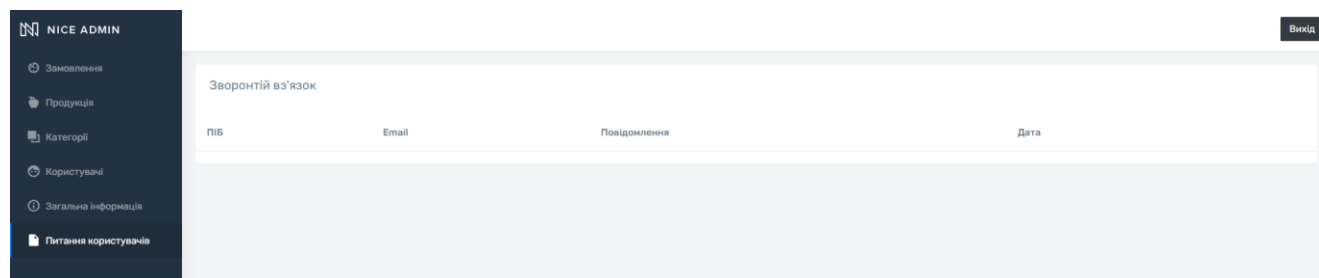


Рисунок 3.37 – Сторінка із доступними категоріями

На сторінці «Питання користувачів» можна переглянути повідомлення від зареєстрованих користувачів (рис.3.38).



The screenshot displays the NICE ADMIN interface. On the left is a dark sidebar with the following menu items: 'Замовлення', 'Продукція', 'Категорії', 'Користувачі', 'Загальна інформація', and 'Питання користувачів'. The main content area is titled 'Зворонтий зв'язок' and contains a table with the following headers: 'ПІБ', 'Email', 'Повідомлення', and 'Дата'. A 'Вихід' button is located in the top right corner.

ПІБ	Email	Повідомлення	Дата
-----	-------	--------------	------

Рисунок 3.38 – Сторінка «Питання користувачів»

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи бакалавра є розроблений web-додаток. Було проведено дослідження та аналіз основних аспектів впровадження інформаційних технологій у сфері онлайн-продажу садового інвентарю. Огляд додатків-аналогів дозволив виявити переваги та недоліки, що сприяли розробці більш універсального та зручного web-дodatка для підтримки діяльності магазину садового інвентарю. Під час виконання кваліфікаційної роботи була досліджена предметна область проекту. Був проведений аналіз аналогічних web-додатків, розроблені вимоги до проекту та календарний план, що сприяло моделюванню функціонування web-додатка та сформовано ТЗ.

Результати планування розробленого додатку представлені структурою задач для розробки продукту, організаційною структурою, матрицею відповідальності, а також визначені часові межі у вигляді часової шкали та діаграми Ганта. Було проаналізовано можливі ризики та визначено шляхи їх попередження або усунення.

Наступним кроком було розроблення прототипу та підготовка макета дизайну web-додатка. Була виконана верстка та реалізація програмної частини. До останніх етапів розробки відноситься наповнення web-додатка контентом, проведення тестування та перенесення web-додатка на сервер.

Результатом проекту є розроблений програмний продукт, спрямований на розвиток та поліпшення послуг у сфері аграрної сфери. Web-додаток розроблений з метою підвищити ефективність роботи інтернет магазину та зручного використання користувачем, зокрема оформлення замовлення та покупки садового інвентарю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Eric Lengyel: Foundations of Game Engine Development Volume 1 / Terathon Software LLC, 2019. – 200 с.
2. Jon Duckett: HTML and CSS: Design and Build Websites / Wiley, 2020. – 490 с.
3. Eloquent JavaScript: A Modern Introduction to Programming / Marijn Haverbeke, 2018. – 472 с.
4. Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability / New Riders, 2019. – 216 с.
5. Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship / Prentice Hall, 2021. – 464 с.
6. David Flanagan: JavaScript: The Definitive Guide / O'Reilly Media, 2020. – 706 с.
7. Jennifer Niederst Robbins: Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics / O'Reilly Media, 2018. – 808 с.
8. David Gassner: Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 / O'Reilly Media, 2018. – 810 с.
9. Brad Frost: Atomic Design / Brad Frost Web, 2019. – 210 с.
10. Garden Tools [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.gardentoolcompany.com/> (дата звернення: 05.04.2023).
11. Green Thumb Garden Supplies [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.greenthumbgardensupplies.com> (дата звернення: 05.04.2023).
12. The Garden Depot [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.thegardendepot.com.au> (дата звернення: 05.04.2023).
13. Addy Osmani: Learning JavaScript Design Patterns / O'Reilly Media, 2019. – 254 с.

14. Douglas Crockford: JavaScript: The Good Parts / O'Reilly Media, 2020. – 176 c.
15. Rebecca Murphey: JavaScript: Novice to Ninja / SitePoint, 2019. – 500 c.
16. Robin Nixon: Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 / O'Reilly Media, 2021. – 812 c.
17. Jon Duckett: JavaScript and jQuery: Interactive Front-End Web Development / Wiley, 2020. – 640 c.
18. Sandeep Panda: Mastering Django: Core / Packt Publishing, 2020. – 514 c.
19. Ethan Marcotte: Responsive Web Design / A Book Apart, 2021. – 160 c.
20. David Heinemeier Hansson, Jason Fried: Rework / Crown Business, 2020. – 288 c.
21. Brad Frost: Atomic Design / Brad Frost Web, 2019. – 210 c.
22. Jennifer Robbins: Web Design in a Nutshell / O'Reilly Media, 2021. – 746 c.
23. Terry Felke-Morris: Web Development and Design Foundations with HTML5 / Pearson, 2019. – 720 c.
24. Randy Connolly, Ricardo Hoar: Fundamentals of Web Development / Pearson, 2020. – 1248 c.

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ WEB-ДОДАТКУ

1.1 Призначення web-додатку

Web-додаток підтримки діяльності магазину садового інвентарю створений з метою надання онлайн-платформи, де покупці зможуть з легкістю переглядати, шукати та придбати різноманітний садовий інвентар. Цей додаток спрямований на полегшення процесу покупки та забезпечення високоякісного та зручного сервісу для своїх користувачів.

1.2 Мета створення web-додатку

Основною метою web-додатка підтримки діяльності магазину садового інвентарю є забезпечення клієнтам широкого набору якісних садових інструментів, пристроїв та матеріалів. Додаток ставить за мету надати зручну платформу, де покупці зможуть легко знайти потрібні товари, отримати детальну інформацію про них, здійснити замовлення та зручно оплатити їх. Магазин також має на меті забезпечити якісний та ефективний сервіс підтримки клієнтів, надаючи консультації та розв'язувати питання, що виникають під час покупки.

1.3 Цільова аудиторія

Web-додаток підтримки діяльності магазину садового інвентарю спрямований на широкий спектр клієнтів, які цікавляться садівництвом та мають потребу у садовому інструментарії та матеріалах. Цільова аудиторія містить професіоналів у галузі садівництва, сільськогосподарські підприємства, аматорських садівників та осіб, які просто цікавляться садовою тематикою.

Користувачі можуть мати різний рівень досвіду у садівництві та вік, але їх об'єднує спільна потреба у якісному та зручному садовому інвентарі.

2 ВИМОГИ ДО WEB-ДОДАТКУ

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Web-додаток підтримки діяльності магазину садового інвентарю потрібно розмістити на сервері, щоб забезпечити його доступність та користування користувачами. Це означає, що додаток буде завантажений і налаштований на сервері, щоб його можна було відвідати та використовувати через Інтернет. Це дозволить користувачам отримати доступ до функцій додатка, переглядати товари, робити покупки та отримувати підтримку безпосередньо через web-браузер. Розміщення додатка на сервері дозволяє забезпечити його постійну доступність та швидкий відгук на запити користувачів.

2.1.2 Вимоги до персоналу

Для ефективної роботи з web-додатком необхідний персонал, який буде відповідати певним вимогам та матиме необхідні навички і знання. Основними вимогами до персоналу є:

- Знання ОС Windows 10/11 та Windows Server 2012/2019, а також навички адміністрування систем і управління налаштуваннями користувачів. Це включає розуміння основних функцій операційної системи та здатність налагоджувати та керувати доступом користувачів до різних ресурсів.
- Вміння логічно мислити та швидко знаходити необхідну інформацію для розв'язування технічних проблем. Це включає здатність аналізувати причини проблем, розробляти стратегії усунення та знаходити оптимальні рішення.
- Навички писати чіткі технічні завдання (ТЗ) на основі отриманих заявок. Це включає вміння зрозуміти потреби замовника, описати вимоги до функціональності додатка та встановити чіткі терміни виконання завдання.

Ці вимоги допоможуть забезпечити належну роботу з web-додатком, забезпечити його ефективне функціонування, обробку заявок та зв'язок із замовниками.

2.1.3 Вимоги до збереження інформації

Вся інформація, яка використовується у web-додатку для підтримки діяльності магазину садового інвентарю, потрібна для зберігання у базі даних. Для цього використовується система управління базами даних MySQL. Ця система дозволяє створювати та управляти структурою бази даних, зберігати інформацію та здійснювати різноманітні операції з нею, такі як додавання, видалення, редагування та запити до даних. Вона забезпечує надійне збереження і доступ до інформації, що потрібна для ефективної роботи web-додатку та задоволення потреб користувачів.

2.1.4 Вимоги до розмежування доступу

Доступ розраховано на два типи користувачів:

- Адміністратор;
- Звичайні користувачі інформаційної системи.

Головна вимога до розмежування – адміністратор має можливість додавати, редагувати та видаляти інформацію на сайті. Робота ведеться через спеціальну панель.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

Структура web-додатку підтримки діяльності магазину садового інвентарю являє собою набір сторінок, які є пунктами головного меню.

Такими розділами є:

- Головна – на сторінці зображені головне меню та карусель з рекомендаціями інвентарю.
- Каталог – інформація про ціну кожного виду товару.
- Доставка.
- Про нас

2.2.2 Навігація

Навігація в web-додатку підтримки діяльності магазину садового інвентарю представлена виглядом меню, розташованого у шапці сторінки. Це меню містить пункти, за допомогою яких можна переходити на окремі сторінки зі відповідним вмістом. Кожен пункт меню відповідає певному функціоналу або розділу додатку. Користувачі можуть клацати на посилання в меню, щоб отримати доступ до відповідних сторінок та виконувати необхідні дії. Таким чином, навігація дозволяє зручно переміщатися по різних частинках додатка та використовувати його функціонал.

2.2.3 Наповнення web-додатку (контент)

Управління контентом буде вестися через адміністративну панель. До контенту відноситься усе, що пов'язане з тематикою садового інвентарю.

2.2.4 Дизайн та структура web-додатку

Інтерфейс сайту має забезпечувати легке розуміння структури представленої на ньому інформації та логічний перехід до різних розділів і сторінок. Для цього посилання на різні сторінки повинні бути виділені заголовками або кнопками, які відображають назви відповідних розділів. Коли користувач обирає певний пункт меню, на сайті повинна завантажуватися відповідна сторінка з інформацією, яка

відповідає цьому пункту. При цьому в блоці меню можуть розкриватися додаткові підрозділи або підменю, які відображають доступні підрозділи для обраного розділу. Це дозволяє користувачеві зручно орієнтуватися на сайті та знайти потрібну інформацію.

Прототип сторінок web-додатку представлений на рис. А.1.

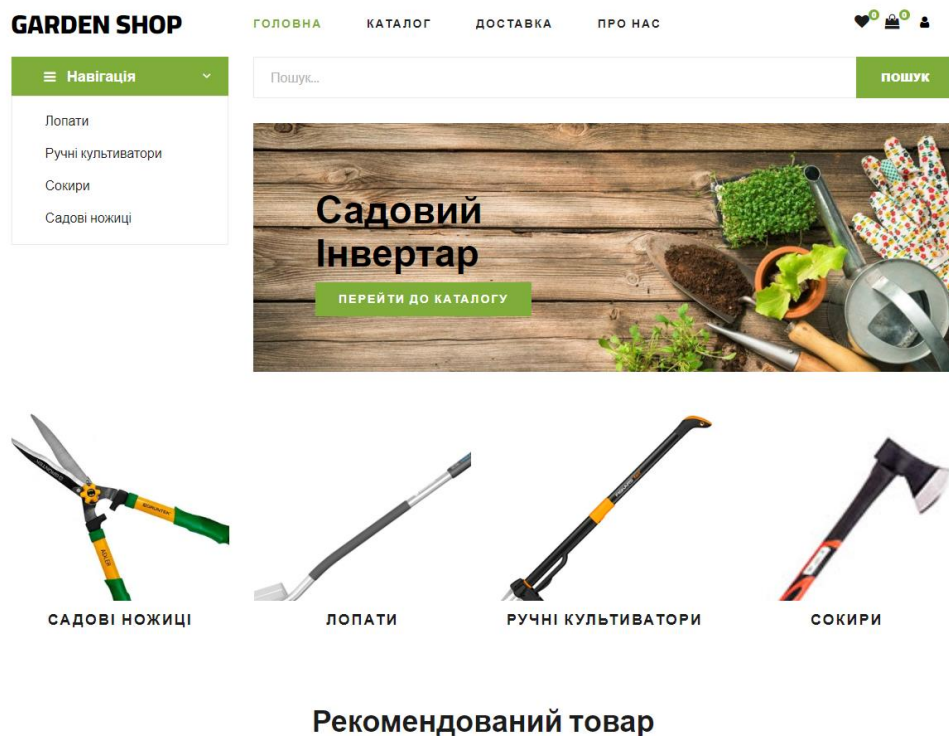


Рисунок А.1 – Прототип головної сторінки

2.2.5 Система навігації (карта web-додатку)

Карта веб-додатку підтримки діяльності магазину садового інвентарю зображена на рисунку А.2.



Рисунок А.2 – Карта web-додатку

2.3 Вимоги до функціонування web-додатку

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Перегляд основної інформації на сайті	Клієнт
UN-02	Перегляд загальної інформації інвентарю	Клієнт
UN-03	Перегляд та редагування даних у кошику при замовленні	Клієнт
UN-04	Можливість зворотного зв'язку	Клієнт
UN-05	Редагування даних	Адміністратор
UN-06	Перегляд рекомендованих товарів на сайті	Клієнт
UN-07	Оформлення різних типів замовлення	Клієнт
UN-08	Розрахунок орієнтовної вартості замовлення	Клієнт

2.3.2 Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

1. Web-додаток повинен складатись з двох частин, а саме користувацької та адміністративної.
2. Користувацька частина призначена для незареєстрованих користувачів які мають можливість користуватись сайтом, не виконуючи авторизацію.
3. Пошук інформації на сайті.
4. Перегляд інформації на сайті.
5. Реєстрація та авторизація користувачів.
6. Оформлення замовлення.
7. Вибір типу та адреси доставки замовлення.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність модуля оформлення замовлення	М	Надає можливість клієнту здійснити онлайн-замовлення продукції, що представлена на сайті
SR-02	Каталог товарів	S	Формує можливість перегляду товарів, що представлені для продажу на сайті
SR-03	База даних із замовленнями	М	Надає можливість відображати адміністратору інформацію про замовлення та їх поточний стан
SR-04	База даних з контентом	М	Відповідає за заповнення інформаційну систему контентом
SR-05	База даних з цінами на послуги	М	Надає можливість подальшого відображення цін на всі види послуг
SR-06	Наявність модуля зворотного зв'язку	С	Надає можливість клієнту зв'язатися з адміністратором
SR-07	Панель редагування	М	Відповідає за подальше наповнення та редагування сайту адміністратором

Умовні позначення в таблиці А.2:

- Must have (M) – вимоги, які повинні бути реалізовані в системі;
- Should have (S) – вимоги, які бажано мають бути виконані;
- Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до програмного забезпечення розроблення web-додатку

Реалізація web-додатку відбувається з використанням:

- HTML&CSS
- PHP 7.4.4 та його фреймворки
- MySQL 8.0

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення перегляду web-додатку

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам: веб-браузер: Firefox 3.5 і вище, або Opera 9.5 і вище, або Chrome 2 і вище.

3 СКЛАД І ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ WEB-ДОДАТКУ

Докладний опис етапів роботи зі створення web-додатку підтримки продажу кондитерських виробів наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Ініціалізація	5
1.1	Ознайомлення з предметної областю проекту	5
1.2	Аналіз конкурентів	3
1.3	Ідентифікація ідей	2
2	Планування	3
2.1	Підготовка матеріалу	2
2.2	Вивчення вимог проекту	2
2.3	Розробка календарного плану	2
2.4	Планування ризиків	1
3	Реалізація	40
3.1	Розробка прототипу	5
3.2	Підготовка макету дизайну сайту	5
3.3	Верстка	15
3.4	Робота над програмною частиною	15
3.5	Наповнення сайту контентом	1
3.6	Тестування	5
4	Завершення	2
4.1	Підготовка документації	1
4.1	Перенесення web-додатку на сервер	1

4 ВИМОГИ ДО СКЛАДУ Й ЗМІСТУ РОБІТ ІЗ ВВЕДЕННЯ WEB-ДОДАТКУ В ЕКСПЛУАТАЦІЮ

Розроблений програмний продукт спрямований на забезпечення росту та покращення підтримки діяльності магазину садового інвентарю. Web-додаток буде створений з метою забезпечення зручного користування споживачами. Можливості програмного продукту будуть цінними та результативними для споживачів з усієї України, оскільки в ньому буде реалізована можливість доставки замовлення та надання зручного набору можливостей.

ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

1.1 Розробка концепції проекту

1.1.1. Ідентифікація ідеї проекту.

Основною метою web-додатка підтримки діяльності магазину садового інвентарю є забезпечення клієнтам великого набору якісних садових інструментів, пристроїв та матеріалів. Додаток ставить за мету надати зручну платформу, де покупці зможуть легко знайти потрібні товари, отримати детальну інформацію про них, здійснити замовлення та зручно оплатити їх. Магазин також має на меті забезпечити якісний та ефективний сервіс підтримки клієнтів, надаючи консультації та розв'язувати питання, що виникають під час покупки.

1.1.2. Деталізація мети методом SMART

Багато успішних підприємців змогли побудувати своє підприємство завдяки ретельному аналізу та точному формулюванню своїх цілей. У цьому їм допомогла SMART-техніка постановки цілей, яка дозволила досягти успіху в продажах, сформуванню моделі поведінки та встановити стійкі правила для життєвої позиції. Розглянемо детальніше кожен пункт цієї техніки в таблиці Б.1, де надана докладна інформація про кожен складову SMART-підходу.

Таблиця Б.1 – Таблиця опису проекту за SMART методом

Specific (конкретна)	Розробка інформаційної системи, за допомогою якої можна буде виконувати поставлені цілі та задачі, та формуватися замовлення.
Measurable (вимірювана)	Прогрес та якість буде вимірюватися в кількості користувачах, замовленнях та кількості позитивних відгуків.

Продовження таблиці Б.1

Achievable (досяжна)	Данна інформаційна система буде розроблятися з використанням HTML, CSS, JavaScript, Vue.js та PHP, MySQL, Laravel.
Relevant (реалістична)	Для виконання проекту є всі необхідні засоби та знання.
Time-framed (обмежена у часі)	Для проекту визначений дедлайн та побудована Діаграма Ганта для відстеження часу виконання.

2 УПРАВЛІННЯ ВИМОГАМИ

2.1 Загальні вимоги до інформаційної системи

2.1.1 Вимоги до структури й функціонування інформаційної системи

Щоб забезпечити доступність інформаційної системи, планується реалізувати її у вигляді web-додатку, який буде доступний через Інтернет. Web-додаток буде складатися з різних взаємопов'язаних розділів, кожен з яких виконуватиме чітко визначені функції. Це дозволить забезпечити структурованість та зручність використання системи для користувачів.

2.1.2. Вимоги до персоналу

Для ефективної підтримки сайту та роботи з web-інтерфейсом системи керування сайтом, вимагатимуться загальні навички роботи з персональним комп'ютером і стандартним web-браузером. Це означає, що персонал повинен володіти базовими знаннями щодо використання комп'ютера, такими як робота з мишею, клавіатурою, виконання операцій копіювання, вставки, збереження файлів, а також вміти користуватися web-браузером для перегляду web-сторінок та виконання основних дій, таких як введення URL-адреси, перехід по посиланнях, заповнення форм тощо.

2.1.3. Вимоги до стилістичного оформлення сайту

Інтерфейс сайту повинен забезпечувати легке розуміння структури розміщеної на ньому інформації, а також логічний перехід між розділами та сторінками. Посилання на сторінки мають бути чіткими заголовками, що вказують на їх зміст. Коли користувач обирає один із пунктів меню, відповідна інформаційна сторінка має завантажуватися, а в блоці меню має відкриватися список підрозділів, які стосуються обраного пункту.

2.2 Вимоги до розділів

Web-додаток повинен складатися з наступних розділів:

- Товари
- Про нас–текстова сторінка;
- Доставка та оплата –текстова сторінка;
- Вхід/Реєстрація
- Особистий кабінет (посилання авторизованим користувачам)
- Кошик (з відображенням кількості та вартості розміщених товарів)

3 ОПИСАННЯ ФАЗИ РОЗРОБКИ ІТ-ПРОЕКТУ

3.1 Планування змісту структури робіт ІТ-проекту

WBS проекту - це розбиття проекту на конкретні результати, які повинні бути досягнуті для досягнення цілей проекту. Важливо розуміти, що в WBS збираються саме результати робіт, а не завдання, які потрібно виконати для отримання цих результатів. Вона є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль; наприклад, програма, проект чи договір.

Загальна WBS структура представлена на рисунку Б.1.

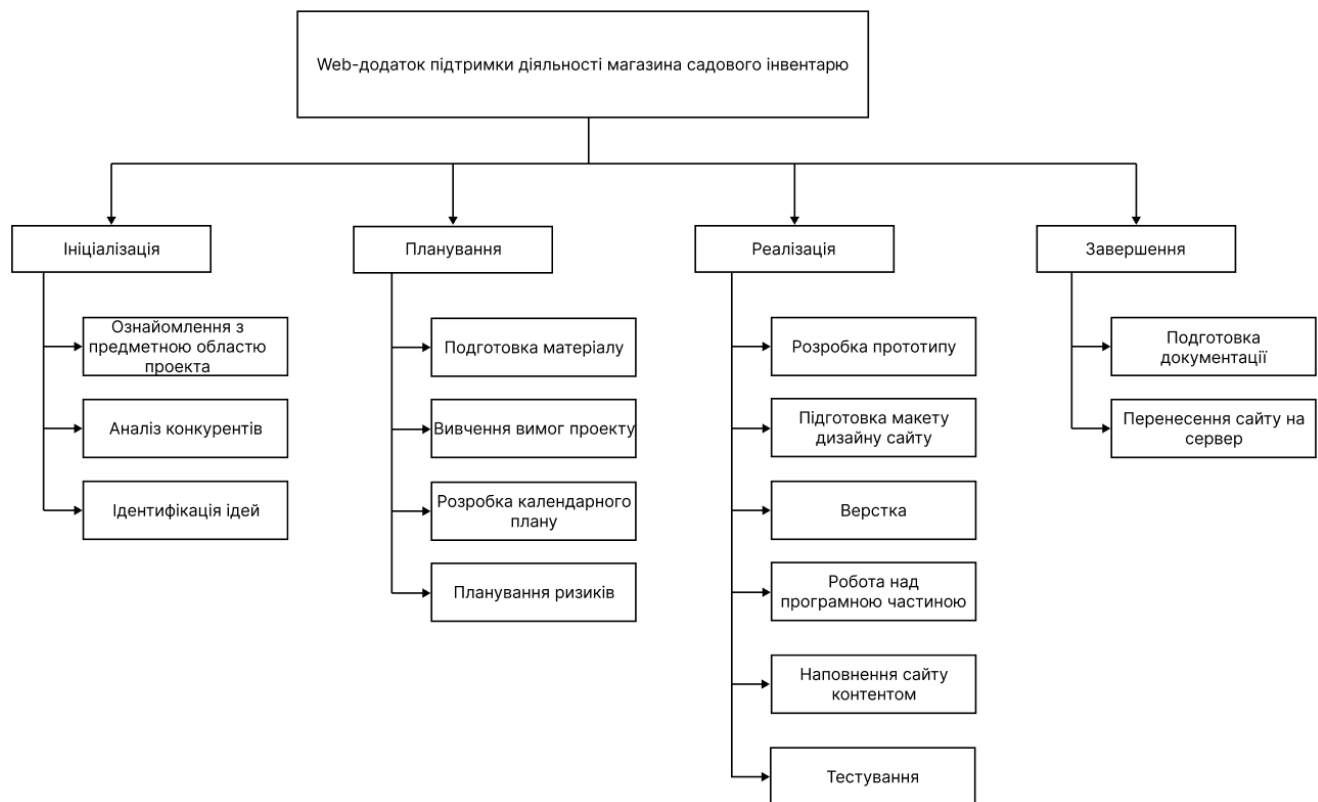


Рисунок Б.1 – WBS-структура проекту

3.2. Планування структури організації

Структура розподілу організації або OBS – це ієрархічна модель, що описує встановлені організаційні рамки для планування проектів, управління ресурсами, відстеження часу та витрат, розподілу витрат, звітування про доходи / прибуток та управління роботою.

Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту. На верхньому рівні OBS розташована команда проекту.

Ця структура стосується тільки внутрішньої організаційної структури проекту і не зачіпає відносин проектних груп чи учасників з батьківськими організаціями. Будується OBS аналогічно робочій структурі, а саме:

- На першому рівні відображається організаційна структура як єдиний елемент;
- На другому і нижчих рівнях триває поділ структури на основні організаційні елементи.

Загальна OBS структура представлена на рисунку Б.2.

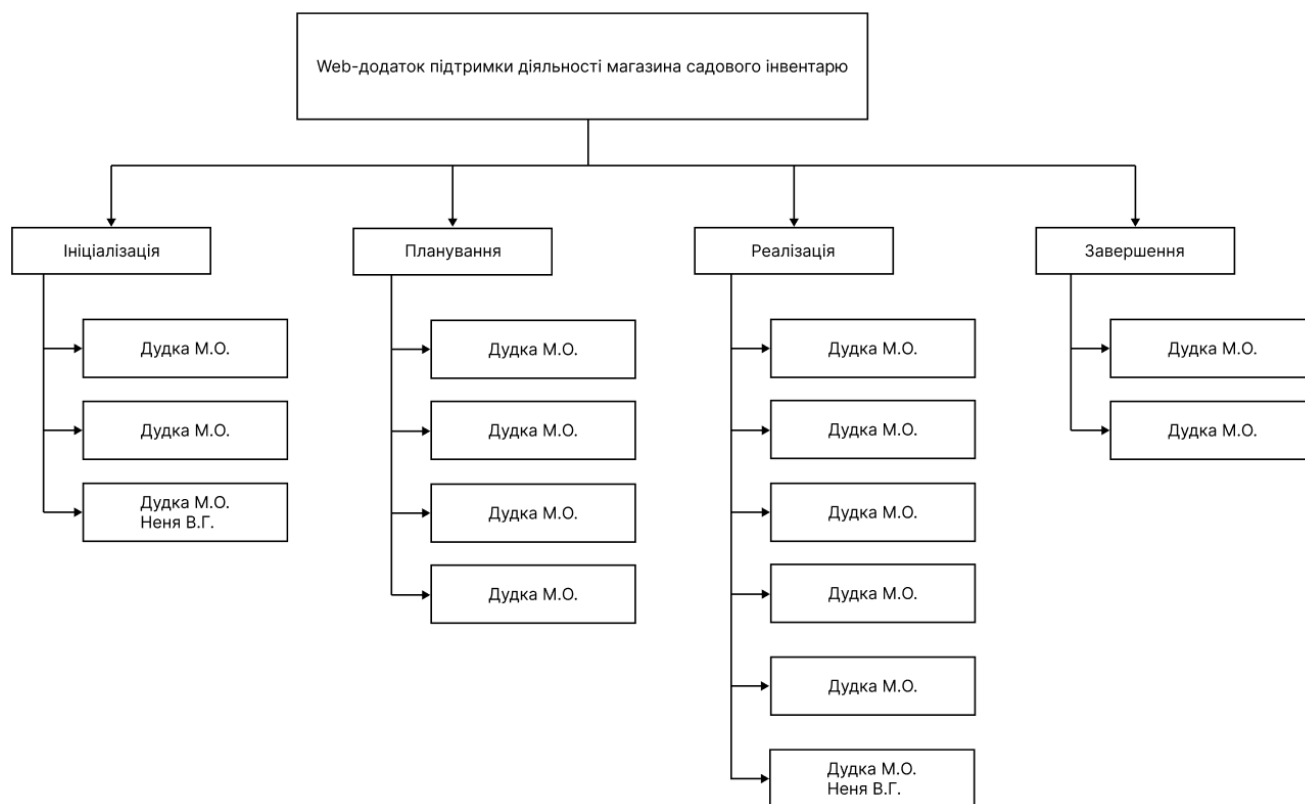


Рисунок Б.2 – OBS структура проекту

3.3. Побудова матриці відповідальності

На підставі OBS та WBS структур будується матриця відповідальності проекту. Матриця відповідальності закріплює за кожною елементарною роботою виконавця. Використовується для контролю відповідності розподілу ролей цілям проекту. На верхньому рівні розподіляються ролі та відповідальність по елементах OBS. На нижньому — фази проекту. Матриця відповідальності представлена в таблиці Б.2.

Таблиця Б.2 – Матриця відповідальності

Назва етапу		Дудка М.О.	Неня В.Г.
1	Ініціалізація	+	+
1.1	Ознайомлення з предметної областю проекту	+	-
1.2	Аналіз конкурентів	+	-
1.3	Ідентифікація ідей	+	+
2	Планування	+	-
2.1	Підготовка матеріалу	+	-
2.2	Вивчення вимог проекту	+	-
2.3	Розробка календарного плану	+	-
2.4	Планування ризиків	+	-
3	Реалізація	+	+
3.1	Розробка прототипу	+	-
3.2	Підготовка макету дизайну сайту	+	-
3.3	Верстка	+	-
3.4	Робота над програмною частиною	+	-
3.5	Наповнення сайту контентом	+	-
3.6	Тестування	+	+
4	Завершення	+	-
4.1	Підготовка документації	+	-
4.1	Перенесення сайту на сервер	+	-

3.4. Побудова календарного графіку виконання ІТ-проекту

Діаграма Ганта – це горизонтальна діаграма з тимчасової шкалою, яка використовується для ілюстрації плану робіт за проектом з прив'язкою до часу.

За допомогою діаграм Ганта керівники проектів і менеджери по продукту розбивають проекти на робочі завдання для зручності управління, підтримують порядок в роботі і роблять залежності між завданнями наочними.

Діаграми Ганта дозволяють спростити складові проекти. За допомогою цього засобу можна досить наочно і зручно для узагальнення представити велику кількість даних. Завдяки цій гістограмі велика кількість зацікавлених осіб, команд або їх учасників не стане проблемою для запису завдань, як і часті зміни обсягу роботи. Ще одна перевага використання діаграми Ганта полягає в тому, що вона дає загальне уявлення про проект в цілому, в тому числі про всі контрольних точках і терміни виконання. Діаграму Ганта можна уявити як ефективний засіб раннього попередження.

Розглянемо створену діаграму Ганта до заданої інформаційної системи (рис.Б.3-Б.4).

Task	Start Date	Days to Complete
Ініціалізація	06.Лют	5
Ознайомлення з предметної областю проекту	11.Лют	5
Аналіз конкурентів	16.Лют	3
Ідентифікація ідей	19.Лют	2
Планування	21.Лют	3
Підготовка матеріалу	24.Лют	2
Вивчення вимог проекту	26.Лют	2
Розробка календарного плану	28.Лют	2
Планування ризиків	02.Бер	1
Реалізація	03.Бер	40
Розробка прототипу	12.Кві	5
Підготовка макету дизайну сайту	17.Кві	5
Верстка	22.Кві	15
Робота над програмною частиною	07.Тра	15
Наповнення сайту контентом	22.Тра	1
Тестування	23.Тра	5
Завершення	28.Тра	2
Підготовка документації	30.Тра	1
Перенесення web-додатку на сервер	31.Тра	1

Рисунок Б.3 – Діаграма Ганта (інформація)

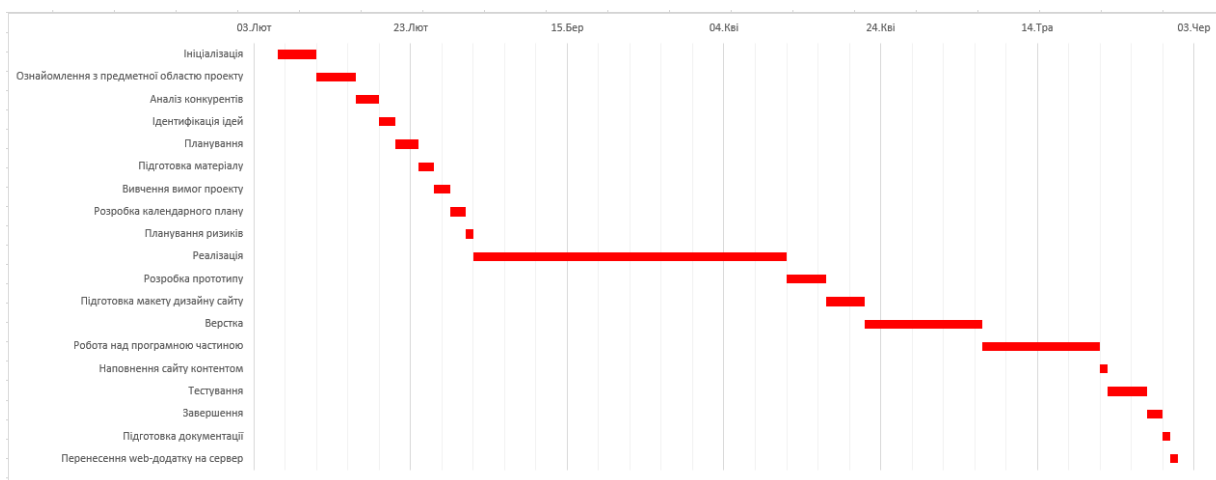


Рисунок Б.4 – Діаграма Ганта (графік)

4 УПРАВЛІННЯ РИЗИКАМИ ІТ-ПРОЕКТУ

Ризиком є ймовірна подія, яка у випадку її виникнення може як негативно, так і позитивно вплинути на конкретний проект. Управління ризиком – це процес зміни ризиків та реагування на події під час виконання проекту. При реалізації проекту важливою частиною є моніторинг ризиків. Отже, в даному випадку можна виділити деякі ризики.

Під час процесу аналізу для визначення числових значень ймовірності появи ступеня впливу, застосовувалася методика експертних оцінок. Виходячи цих оцінок можливо знайти ранг ризиків:

$$R = P * L,$$

де

- R – ранг ризику;
- P – ймовірність виникнення;
- L – ступінь впливу.

Шкала оцінки ризику може відповідати емпіричній шкалі оцінки ризику:

- 5 балів - критичний ризик (0,81 - 1);
- 4 бали - максимальний ризик (0,61 - 0,8);
- 3 бали - високий ризик (0,41 - 0,6);
- 2 бали - нормальний ризик (0,31 - 0,4); 1 бал - малий ризик (0 - 0,3).

Оцінки ризиків проекту наведено в табл. Б.4.

Таблиця Б.4 – Таблиця ризиків

№	Об'єкт ризику	Ризик	P	L	R
1	Час	Зміна пріоритету проекту	0,5	0,3	0,15
2	Якість	Системи, що не відповідають задачам, грубі помилки в алгоритмах процесів, критичні збої системи	0,1	0,4	0,04
3	Бюджет	Поява поза планових робіт по проекту	0,7	0,2	0,14
4	Трудові ресурси та їх кваліфікація	Неможливість участі в запланованих роботах по проекту необхідних співробітників зі сторони замовника та виконавця у зв'язку з відпусткою, відрядженням та ін.	0,1	0,1	0,01
5	Інтеграція	Не вірна інформація, щодо зовнішніх систем, з яких передбачена взаємодія в межах проекту	0,5	0,4	0,2
6	Ринок	Розширення функціональних характеристик програмних продуктів, які використовувалися замовником в межах мети проекту	0,1	0,05	0,005

Продовження таблиці Б.4

7	Ринок	Зміна вимог замовника, економічні зміни, посилення конкуренції, втрата позицій на ринку;	0,5	0,1	0,05
8	Бюджет	Зриви планів робіт; неправильна стратегія; персонал без потрібної кваліфікації; переплати за роботу/матеріали; непогоджені частини проекту; невірний кошторис.	0,5	0,2	0,1

ДОДАТОК В. КОД РЕАЛІЗАЦІЇ WEB-ДОДАТКУ

OrdersController.php

Контролер замовлень.

```

<?php

namespace App\Http\Controllers;

use App\Models\Order;
use App\Models\OrderProducts;
use App\Models\DeliveryTypes;
use App\Models\OrderStatus;
use App\Models\Category;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class OrdersController extends Controller
{
    /**
     * Show the form for creating a new resource.
     *
     */
    public function create()
    {
        if(isset($_COOKIE['cart_id'])) {
            \Cart::session($_COOKIE['cart_id']);
        }
        $data = \Cart::getContent();

        $sum = 0;

        foreach ($data as $value) {
            $sum += ($value['price'] * $value['quantity']);
        }
        return view('checkout', [
            'data' => $data,
            'sum' => $sum,
            'delivery_types' => DeliveryTypes::get(),
            'categories' => Category::get()
        ]);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     */
    public function store(Request $request)
    {
        $data = $request->all();
        $model = new Order();
        $order = $model->create($data);

        if(isset($_COOKIE['cart_id'])) {
            \Cart::session($_COOKIE['cart_id']);
        }
    }
}

```

```

$data = \Cart::getContent();

foreach ($data as $value) {
    OrderProducts::create([
        'orders_id' => $order['id'],
        'product_id' => $value['id'],
        'quantity' => $value['quantity']
    ]);
}

\Cart::clear();

return redirect('/thankyou');
}

public function edit($id)
{
    return view('admin.updateOrder', [
        'order' => Order::with('delivery')->find($id),
        'products' => OrderProducts::with('product.category')->where('orders_id', $id)->get(),
        'statuses' => OrderStatus::get()
    ]);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 */
public function update(Request $request, $id)
{
    $data = $request->all();
    $model = Order::find($id);
    $model->update($data);
    return response()->json();
}

/**
 * Remove the specified resource from storage.
 *
 */
public function destroy($id)
{
    Order::find($id)->delete();
    return response()->json();
}
}

```

ProductController.php

Контролер товару.

```

<?php

namespace App\Http\Controllers;

use App\Models\Product;
use App\Models\Category;
use Illuminate\Http\Request;

class ProductController extends Controller

```

```

{
protected $publicStorage = '/uploads/';
/**
 * Display a listing of the resource.
 *
 */
public function index()
{
    return view('admin.products', [
        'data' => Product::with('category')->get()
    ]);
}

/**
 * Show the form for creating a new resource.
 *
 */
public function create()
{
    return view('admin.createProduct', [
        'categories' => Category::get()
    ]);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 */
public function store(Request $request)
{
    $data = $request->all();
    if($request->file('photo')){
        $file = $request->file('photo');
        $filename = date('YmdHi').$file->getClientOriginalName();
        $file->move(public_path() . $this->publicStorage, $filename);
        $data['photo'] = $this->publicStorage . $filename;
    }
    $model = new Product();
    $model->create($data);
    return redirect('/dashboard/product');
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Product $product
 */
public function show($id)
{
    $data = Product::find($id);
    $recommendation = Product::where('recommendation', 1)->with('category')->limit(3)->get();
    return view('shop-single', [
        'data' => $data,
        'recommendation' => $recommendation,
        'categories' => Category::get()
    ]);
}

/**

```

```

* Show the form for editing the specified resource.
*
* @param \App\Models\Product $product
*/
public function edit($id)
{
    return view('admin.updateProduct', [
        'categories' => Category::get(),
        'product' => Product::find($id)
    ]);
}

/**
* Update the specified resource in storage.
*
* @param \Illuminate\Http\Request $request
* @param \App\Models\Product $product
*/
public function update(Request $request, $id)
{
    $data = $request->all();
    if($request->file('photo')){
        $file = $request->file('photo');
        $filename = date('YmdHi').$file->getClientOriginalName();
        $file->move(public_path() . $this->publicStorage, $filename);
        $data['photo'] = $this->publicStorage . $filename;
    }
    $model = Product::find($id);
    $model->update($data);
    return redirect('/dashboard/product');
}

/**
* Remove the specified resource from storage.
*
* @param \App\Models\Product $product
*/
public function destroy($id)
{
    Product::find($id)->delete();
    return response()->json();
}
}

```

CardController.php

Контролер кошику.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Product;
use App\Models\Category;

class CardController extends Controller
{

    public function cart() {

```



```

if(isset($_COOKIE['cart_id'])) {
    \Cart::session($_COOKIE['cart_id']);
}
$data = \Cart::getContent();

$sum = 0;

foreach ($data as $key => $value) {
    $sum += ($value['price'] * $value['quantity']);
}

return view('cart', [
    'data' => $data,
    'sum' => $sum,
    'categories' => Category::get()
]);
}

public function addToCart(Request $request) {

    $product = Product::find($request->id);

    if(!isset($_COOKIE['cart_id'])) {
        $uniqid = uniqid();
        setcookie('cart_id', $uniqid);
        $cart_id = $uniqid;
    } else {
        $cart_id = $_COOKIE['cart_id'];
    }

    \Cart::session($cart_id)->add([
        'id' => $product->id,
        'name' => $product->title,
        'price' => $product->price,
        'quantity' => (int) $request->quantity,
        'attributes' => [
            'photo' => $product->photo,
            'category_id' => $product->category_id
        ],
        'associatedModel' => $product
    ]);

    return response()->json(\Cart::getContent());
}

public function removeCart(Request $request) {
    $cart_id = $_COOKIE['cart_id'];
    \Cart::session($cart_id)->remove($request->id);
    return response()->json($request->id);
}

public function updateCart(Request $request) {
    $cart_id = $_COOKIE['cart_id'];
    \Cart::session($cart_id)->update($request->id, [
        'quantity' => $request->quantity
    ]);
    return response()->json();
}
}
}

```

Модель заомлень.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    use HasFactory;

    protected $table = 'orders';

    protected $fillable = [
        'name',
        'surname',
        'delivery_types_id',
        'postal_zip',
        'email',
        'phone',
        'note',
        'users_id',
        'sum',
        'order_status_id',
        'method_pay_id'
    ];

    public function status() {
        return $this->belongsTo('App\Models\OrderStatus', 'order_status_id');
    }

    public function delivery() {
        return $this->belongsTo('App\Models\DeliveryTypes', 'delivery_types_id');
    }

    public function pay() {
        return $this->belongsTo('App\Models\PayTypes', 'method_pay_id');
    }
}

```

Product.php

Модель товару.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;

    protected $table = 'product';

    protected $fillable = [
        'title',

```

```

        'description',
        'price',
        'photo',
        'recommendation',
        'category_id',
        'weight',
        'availability'
    ];

    protected $casts = [
        'created_at' => 'datetime:d.m.Y',
    ];

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

```

User.php

Модель користувачів.

```

<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
        'phone',
        'surname',
        'is_admin'
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**

```

```

* The attributes that should be cast.
*
* @var array<string, string>
*/
protected $casts = [
    'email_verified_at' => 'datetime',
    'created_at' => 'datetime:d.m.Y',
];

public function orders() {
    return $this->hasMany('App\Models\Order', 'orders_id');
}
}

```

welcome.blade.php

Шаблон головної сторінки.

```

<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>{{ config('app.name', 'Laravel') }}</title>
    @include('inc.header')
</head>

<body>
    @include('inc.head', [
        'categories' => $categories
    ])

    <section class="categories">
        <div class="container">
            <div class="row">
                <div class="categories__slider owl-carousel">
                    @foreach($categories as $key => $item)
                        <div class="col-lg-3">
                            <div class="categories__item set-bg" data-setbg="{{ $item['photo'] }}">
                                <h5><a href="/shop?category_id={{ $item['id'] }}">{{ $item['title'] }}</a></h5>
                            </div>
                        </div>
                    @endforeach
                </div>
            </div>
        </div>
    </section>

    <section class="featured spad">
        <div class="container">
            <div class="row">
                <div class="col-lg-12">
                    <div class="section-title">
                        <h2>Рекомендований товар</h2>
                    </div>
                </div>
            </div>
            <div class="row featured__filter">
                @foreach($data as $item)
                    <div class="col-lg-3 col-md-4 col-sm-6 mix oranges fresh-meat">
                        <div class="featured__item">
                            <a href="/product/{{ $item['id'] }}">

```

```

        
        <div class="featured__item__text">
            <h6>{{ $item['title'] }}</h6>
            <h5>{{ $item['price'] }}</h5>
        </div>
    </a>
</div>
</div>
@endforeach
</div>
</div>
</section>

@include('inc.footer', [
    'categories' => $categories
])
@include('inc.scripts')
</body>

</html>

```

cart.blade.php

Шаблон сторінки кошику.

```

<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>Кошик | {{ config('app.name', 'Laravel') }}</title>
    @include('inc.header')
</head>

<body>
    @include('inc.head', [
        'categories' => $categories
    ])

    <section class="shoping-cart spad">
        <div class="container">
            @if(count($data) > 0)
                <div class="row">
                    <div class="col-lg-8">
                        <div class="shoping__cart__table">
                            <table>
                                <thead>
                                    <tr>
                                        <th class="shoping__product">Товар</th>
                                        <th>Ціна</th>
                                        <th>Кількість</th>
                                        <th></th>
                                    </tr>
                                </thead>
                                <tbody>
                                    @foreach ($data as $item)
                                        <tr>
                                            <td class="shoping__cart__item">
                                                
                                                <h5>{{ $item['name'] }}</h5>
                                            </td>
                                            <td class="shoping__cart__price">

```

```

        {{ $item['price'] }}грн
    </td>
    <td class="shoping__cart__quantity">
        <div class="quantity">
            <div class="pro-qty">
                <input type="text" value="{{ $item['quantity'] }}">
            </div>
        </div>
    </td>
    <td class="shoping__cart__item__close">
        <span index="{{ $item['id'] }}" class="icon_close remove-cart"></span>
    </td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
<div class="col-lg-4">
    <div class="shoping__checkout">
        <h5>Всього</h5>
        <ul>
            <li>Ціна <span>{{ $sum }}грн</span></li>
        </ul>
        <a href="/checkout" class="primary-btn">Оформити замовлення</a>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="shoping__cart__btns">
            <a href="/shop" class="primary-btn cart-btn">ПРОДОВЖИТИ ПОКУПКИ</a>
        </div>
    </div>
</div>
@else
<div class="text-center">
    
</div>
@endif
</div>
</section>

@include('inc.footer', [
'categories' => $categories
])
@include('inc.scripts')
</body>

</html>

```

shop-single.blade.php

Шаблон сторінки з товаром.

```

<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>Каталог | {{ config('app.name', 'Laravel') }}</title>
    @include('inc.header')

```

```

</head>

<body>
  @include('inc.head', [
    'categories' => $categories
  ])

  <section class="product-details spad">
    <div class="container">
      <div class="row">
        <div class="col-lg-6 col-md-6">
          <div class="product__details__pic">
            <div class="product__details__pic__item">
              
            </div>
          </div>
        </div>
        <div class="col-lg-6 col-md-6">
          <div class="product__details__text">
            <h3>{{ $data['title'] }}</h3>
            <div class="product__details__price">{{ $data['price'] }} грн</div>
            <p>{{ $data['description'] }}</p>
            @if($data['availability'])
            <div class="product__details__quantity">
              <div class="quantity">
                <div class="pro-qty">
                  <input id="quality" type="text" value="1">
                </div>
              </div>
            </div>
            @endif
            <input id="index-product" type="hidden" value="{{ $data['id'] }}">
            @if($data['availability'])
            <a href="#" class="primary-btn buy-now">ДОДАТИ ДО КОШИКУ</a>
            @endif
            <a href="#" class="heart-icon wishlist-now"><span class="icon_heart_alt"></span></a>
            <ul>
              <li><b>Наявність</b> <span>{{ $data['availability'] ? 'на складі' : 'відсутній' }}</span></li>
              <li><b>Вага</b> <span>{{ $data['weight'] }} кг</span></li>
              <li><b>Поділитись</b>
                <div class="share">
                  <a href="#"><i class="fa fa-facebook"></i></a>
                  <a href="#"><i class="fa fa-twitter"></i></a>
                  <a href="#"><i class="fa fa-instagram"></i></a>
                  <a href="#"><i class="fa fa-pinterest"></i></a>
                </div>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </section>

  <section class="related-product">
    <div class="container">
      <div class="row">
        <div class="col-lg-12">
          <div class="section-title related__product__title">
            <h2>Рекомендовані товари</h2>
          </div>
        </div>
      </div>
    </div>
  </section>

```

```

        </div>
    </div>
</div>
<div class="row">
    @foreach ($recommendation as $item)
        @include('inc.product', [
            'item' => $item
        ])
    @endforeach
</div>
</div>
</section>

@include('inc.footer', [
    'categories' => $categories
])
@include('inc.scripts')

</body>

</html>

```

checkout.blade.php

Шаблон сторінки замовлення.

```

<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>Оформлення замовлення | {{ config('app.name', 'Laravel') }}</title>
    @include('inc.header')
</head>

<body>
    @include('inc.head', [
        'categories' => $categories
    ])

    <section class="checkout spad">
        <div class="container">
            <div class="checkout__form">
                <h4>Контактні дані</h4>
                <form method="POST" action="/orders">
                    <input type="hidden" name="users_id" value="{{ auth()->check() ? auth()->user()->id : " }}">
                    <input type="hidden" name="sum" value="{{ $sum }}">
                    <div class="row">
                        <div class="col-lg-8 col-md-6">
                            <div class="row">
                                <div class="col-lg-6">
                                    <div class="checkout__input">
                                        <p>Ім'я<span>*</span></p>
                                        <input type="text" name="name" value="{{ auth()->check() ? auth()->user()->name : " }}" required>
                                    </div>
                                </div>
                                <div class="col-lg-6">
                                    <div class="checkout__input">
                                        <p>Прізвище<span>*</span></p>
                                        <input type="text" name="surname" value="{{ auth()->check() ? auth()->user()->surname : " }}" required>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </section>

```



```

</div>
<div class="row">
  <div class="col-6">
    <div class="checkout__input">
      <p>Спосіб доставки<span>*</span></p>
      <select name="delivery_types_id" onchange="deliveryType()">
        @foreach($delivery_types as $item)
          <option value="{{ $item['id'] }}">{{ $item['title'] }}</option>
        @endforeach
      </select>
    </div>
  </div>
  <div class="col-6 postal_zip">
    <div class="checkout__input">
      <p>Відділення<span>*</span></p>
      <input type="text" name="postal_zip">
    </div>
  </div>
</div>

<div class="row">
  <div class="col-lg-6">
    <div class="checkout__input">
      <p>Телефон<span>*</span></p>
      <input type="text" name="phone" value="{{ auth()->check() ? auth()->user()->phone : '' }}" required>
    </div>
  </div>
  <div class="col-lg-6">
    <div class="checkout__input">
      <p>Email<span>*</span></p>
      <input type="text" name="email" value="{{ auth()->check() ? auth()->user()->email : '' }}" required>
    </div>
  </div>
</div>
<div class="checkout__input">
  <p>Коментар<span>*</span></p>
  <textarea name="" id="" cols="30" rows="5"></textarea>
</div>

<div class="col-lg-4 col-md-6">
  <div class="checkout__order">
    <h4>Ваше замовлення</h4>
    <div class="checkout__order_products">Товар</div>
    <ul>
      @foreach ($data as $item)
        <li>{{ $item['name'] }} <span>{{ $item['price'] }} грн</span></li>
      @endforeach
    </ul>
    <div class="checkout__order_total">Всього <span>{{ $sum }}</span></div>
    <p>Спосіб оплати</p>
    <div class="checkout__input__checkbox">
      <label for="payment">
        Карта банку
        <input name="method_pay" value="Карта банку" type="radio" id="payment">
        <span class="checkmark"></span>
      </label>
    </div>
    <div class="checkout__input__checkbox">
      <label for="paypal">
        Paypal

```

```

        <input name="method_pay" value="Paypal" type="radio" id="paypal">
        <span class="checkmark"></span>
    </label>
</div>
<div class="checkout__input__checkbox">
    <label for="cash">
        Готівка
        <input name="method_pay" value="Готівка" type="radio" id="cash">
        <span class="checkmark"></span>
    </label>
</div>
<button type="submit" class="site-btn">ЗАМОВИТИ</button>
</div>
</div>
</div>
</form>
</div>
</div>
</section>
@include('inc.footer', [
'categories' => $categories
])
@include('inc.scripts')
<script>
function deliveryType() {
    if ($('#delivery').val() == 3) {
        $('.postal_zip').hide()
    } else {
        $('.postal_zip').show()
    }
}
</script>
</body>

</html>

```

dashboard.blade.php

Шаблон головної сторінки адміністративної панелі.

```

<!DOCTYPE html>
<html dir="ltr" lang="en">

@include('inc.admin.head')

<link href="/admin/dist/chartist.min.css" rel="stylesheet">

<body>
    <div id="main-wrapper" data-navbarbg="skin6" data-theme="light" data-layout="vertical" data-sidebartype="full" data-
boxed-layout="full">
        @include('inc.admin.header')
        <div class="page-wrapper">
            <div class="container-fluid">
                <div class="row">
                    <div class="col-12">
                        <div class="card">
                            <div class="card-body">
                                <h4 class="card-title">Останні замовлення</h4>
                            </div>
                            <div class="table-responsive">
                                <table class="table table-hover">

```



```

        showArea: true,
        fullWidth: true,
        plugins: [
            Chartist.plugins.tooltip()
        ],
        axisY: {
            onlyInteger: true,
            scaleMinSpace: 40,
            offset: 20,
            labelInterpolationFnc: function(value) {
                return (value / 10) + 'k';
            }
        },
    });
    var chart = [chart];
});
</script>
</body>

</html>

```

updateOrder.blade.php

Шаблон головної сторінки з замовленням.

```

<!DOCTYPE html>
<html dir="ltr" lang="en">

@include('inc.admin.head')

<body>
    <div id="main-wrapper" data-navbarbg="skin6" data-theme="light" data-layout="vertical" data-sidebartype="full" data-
boxed-layout="full">
        @include('inc.admin.header')
        <div class="page-wrapper">
            <div class="container-fluid">
                <div class="card">
                    <div class="card-body">
                        <div class="container mb-5 mt-3">
                            <div class="row d-flex align-items-baseline">
                                <div class="col-xl-9">
                                    <p style="color: #7e8d9f;font-size: 20px;">Замовлення >> <strong>ID: #{{ $order['id']
}}</strong></p>
                                </div>
                                <div class="col-xl-3 float-end">
                                    <button onclick="printDiv()" class="btn btn-light text-capitalize border-0" data-mdb-ripple-
color="dark">
                                        <i class="fas fa-print text-primary"></i>Друк
                                    </button>
                                </div>
                            </div>
                        <hr>
                    </div>

                    <div class="container" id="order">
                        <div class="row">
                            <div class="col-4">
                                <ul class="list-unstyled">
                                    <li class="text-muted">ПІБ: <span style="color:#5d9fc5 ;">{{ $order['name'] }} {{
$order['surname'] }}</span></li>
                                    <li class="text-muted">Email: <span style="color:#5d9fc5 ;">{{ $order['email']
}}</span></li>
                                </ul>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <li class="text-muted">Телефон: <span style="color:#5d9fc5 ;">{{ $order['phone']
    }}</span></li>
    </ul>
</div>
<div class="col-4">
    <ul class="list-unstyled">
        <li class="text-muted">Спосіб доставки: <span style="color:#5d9fc5 ;">{{
$order['delivery']['title'] }}</span></li>
        <li class="text-muted">Спосіб оплати: <span style="color:#5d9fc5 ;">{{
$order['method_pay'] }}</span></li>
        <li class="text-muted">Відділення: <span style="color:#5d9fc5 ;">{{ $order['postal_zip']
    }}</span></li>
    </ul>
</div>
<div class="col-4">
    <ul class="list-unstyled">
        <li class="text-muted"><i class="fas fa-circle" style="color:#84B0CA ;"></i> <span
class="fw-bold">ID:</span>#{{ $order['id'] }}</li>
        <li class="text-muted"><i class="fas fa-circle" style="color:#84B0CA ;"></i> <span
class="fw-bold">Дата: </span>{{ $order['created_at'] }}</li>
        <li class="text-muted"><i class="fas fa-circle" style="color:#84B0CA ;"></i>
        <span class="me-1 fw-bold">Статус:</span>
        @if($order['order_status_id'] == 1)
        <span class="badge bg-primary text-black fw-bold">{{ $order['status']['title'] }}</span>
        @endif
        @if($order['order_status_id'] == 2)
        <span class="badge bg-success text-black fw-bold">{{ $order['status']['title'] }}</span>
        @endif
        @if($order['order_status_id'] == 3)
        <span class="badge bg-danger text-black fw-bold">{{ $order['status']['title'] }}</span>
        @endif
    </li>
    </ul>
</div>
</div>
<div class="row my-2 mx-1 justify-content-center">
    <table class="table table-striped table-borderless">
        <thead style="background-color:#84B0CA ;" class="text-white">
            <tr>
                <th scope="col">#</th>
                <th scope="col">Фото</th>
                <th scope="col">Назва</th>
                <th scope="col">Категорія</th>
                <th scope="col">Ціна</th>
                <th scope="col">Кількість</th>
            </tr>
        </thead>
        <tbody>
            @foreach($products as $key => $item)
            <tr>
                <th scope="row">{{ $key + 1 }}</th>
                <td></td>
                <td>{{ $item['product']['title'] }}</td>
                <td>{{ $item['product']['category']['title'] }}</td>
                <td>{{ $item['product']['price'] }}</td>
                <td>{{ $item['quantity'] }}</td>
            </tr>
            @endforeach
        </tbody>
    </table>

```

