

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«Веб-ресурс для компанії 'BEZDNA' з використанням алгоритму  
онлайн-конструктора макету»**

**Завідувач випускаючої кафедри**

**Керівник роботи**

**Студентки групи ІІІ – 61**

**Довбиш А.С.**

**Проценко О.Б.**

**Музика К.Р.**

**СУМИ 2020**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**

**до випускної роботи**

студентки четвертого курсу, групи ІН-61 спеціальності “Інформатика” денної форми навчання Музики Ксенії Русланівни.

**Тема: «Веб-ресурс для компанії 'BEZDNA' з використанням алгоритму онлайн-конструктора макету»**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2020 г.

**Зміст пояснювальної записки:** 1) інформаційний огляд; 2) вибір засобів реалізації, які будуть використовуватись при розробці продукту; 3) практична реалізація.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2020 г.

Керівник випускної роботи \_\_\_\_\_ Проценко О.Б.

Завдання прийняв до виконання \_\_\_\_\_ Музика К.Р.

## РЕФЕРАТ

**Записка:** 62 стор., 29 рис., 1 додаток, 24 джерела.

**Об’єкт дослідження** — Веб-ресурс для компанії 'BEZDNA' з використанням алгоритму онлайн-конструктора макету.

**Мета роботи** — Збільшення кількості потенційних покупців та продажів компанії «BEZDNA», популяризація zero-waste руху шляхом використання інтернет-магазину.

**Результати** — було досліджено актуальність використання екологічно чистих товарів на прикладі продукції сумського бренду “Bezdna”, який займається розробкою, створенням та продажем еко-торбинок; визначено необхідність створення та розробки інтернет-магазину для подальшого розвитку бренду, як представника малого бізнесу; проаналізовано інформацію щодо способів створення такого інтернет-магазину, сучасних тенденцій дизайну та функціоналу сайтів. Результатом став веб-ресурс для компанії 'BEZDNA' з використанням алгоритму онлайн-конструктора макету створений за допомогою Vue.js.

ВЕБ-РЕСУРС, ІНТЕРЕНТ-МАГАЗИН, ЕКО-ТОРБИНКА,  
ШОПЕР, VUE.JS, SQL ТА NO SQL СИСТЕМИ УПРАВЛІННЯ  
БАЗАМИ ДАНИХ.

## ЗМІСТ

ВСТУП.....	4
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	5
1.1 Аналіз предметної області та аналогів .....	5
1.2 Актуальність створення інтернет-магазину .....	10
1.3 Постановка задачі .....	11
2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ.....	13
2.1 Вибір фреймворків.....	13
2.2 Вибір системи управління базами даних.....	15
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ .....	19
3.1 Структурно-функціональне моделювання процесу розробки веб-ресурсу для компанії “BEZDNA”.....	19
3.2 Моделювання варіантів використання веб-ресурсу для компанії “BEZDNA”.....	22
3.3 Інформаційна модель.....	24
3.4 Підготовка середовища розробки .....	27
3.5 Результат реалізації веб-ресурсу.....	31
ВИСНОВКИ .....	35
СПИСОК ЛІТЕРАТУРИ.....	36
ДОДАТОК.....	38

## ВСТУП

“2020 Can’t Get Any Worse” – що означає “2020 рік не може бути гіршим” – стала головною тезою цього року. Причиною популярності згаданої фрази є пандемія коронавірусу SARS-CoV-2, яка за два з половиною місяці поширилася на всі континенти і перевернула людський світ.

Здається, що настав кінець звичного життя людства: усі змушені ізолюватися; більшість компаній, малий та середній бізнес вимушено не працюють або функціонують мінімально; кількість безробітних стрімко зросла – в Україні станом на 24 квітня 2020 року показник безробіття на 36% вищий у порівнянні з цією ж датою минулого року. Але саме завдяки розвиненим технологіям людство за такий короткий час змогло перейти на віддалену роботу. Бізнес, який раніше обходився без веб-ресурсу, нині вимушений переходити в онлайн-простір, аби подолати дистанцію до споживача, спричинену карантинном. Таким чином бізнес почав у прискореному темпі надавати свої послуги або доступ до власних товарів через веб-ресурси. Інтернет-магазини, односторінкові сайти, блоги заповнюють простір інтернету, пропонуючи різноманітні продукти та сервіси, у тому числі еко-товари.

Ось чому тема екології стає вкрай важливою. Через відсутність туристів, зупинку виробництв та громадського транспорту тощо міста “очищуються”. Соціальні мережі переповнені фотографіями та відео, які демонструють несподівано позитивний вплив пандемії на природу. Жителі Уханя, наприклад, вперше за довгі роки побачили чисте небо та дихали свіжим повітрям; канали Венеції стали прозорими. Це наочно продемонструвало людям наш жахливий вплив на довкілля. Через це напрям zero-waste – рух за розумне споживання – став особливо популярним.

Отже, актуальними проблемами сьогодення є використання інтернет-ресурсів для організації роботи малого та середнього бізнесу, його зв'язку зі споживачем, а також популяризація напрямку zero-waste.

## **1 ІНФОРМАЦІЙНИЙ ОГЛЯД**

### **1.1 Аналіз предметної області та аналогів**

На сьогоднішній день люди завдяки інтернету отримують доступ до величезної кількості сервісів через різноманітні веб-ресурси. Купівля продуктів харчування чи авіаквитків, замовлення різноманітних документів чи квитків, оренда готелів чи авто, продаж чого завгодно, онлайн навчання та онлайн викладання, сервіси для перегляду фільмів та ще багато чого - все це здійснюється через різноманітні додатки. Тож очевидно, що задля старту та розвитку свого бізнесу в сучасному світі його веб-представлення на просторах Інтернету просто необхідно. Що ж таке веб-сайт?

Веб-ресурс, або веб-сайт - це сторінка або кілька сторінок, які знаходяться в мережі Інтернет, зв'язані логічно та мають унікальну адресу. Такі сторінки можуть включати текстову, графічну інформацію, різноманітні гіперпосилання або мультимедіа-компоненти. Вибір веб-сайту має вирішальне значення для створення мережевої частини будь-якого бізнесу. Зараз в Інтернеті існує майже 1,7 мільярда веб-сайтів, що відрізняються типом вмісту, спеціалізацією, особливостями використання. [1]

На Інтернет-просторах можна віднайти різні класифікацію сайтів за різними ознаками. Нижче приведені основні їх типи:

- сайт-візитка;
- корпоративний сайт;
- сайт-сервіс;
- сайт-каталог;
- інтернет-магазин;
- лендинг;
- форум;

- блог;
- соціальна мережа.

Детально ознайомившись із кожним з них було вирішено, що інтернет-магазин слугуватиме ідеальним рішенням для поставленого завдання. По-перше, об'єм інтернет продажів збільшується з кожним роком і в Україні він ще може продовжуватися досить довго, тож зараз найкращий час аби продублювати свій бізнес у Інтернеті. По-друге, інтернет-покупки - це форма електронної комерції, яка дозволяє споживачам обирати та купувати товари чи послуги у продавця через Інтернет за допомогою веб-браузера чи додатку, що робить життя покупця легшим. Звичайний інтернет-магазин дозволяє клієнту не лише переглядати асортимент товарів або послуг, а також надає інформацію про технічні та інші характеристики товарів та ціни, умови доставки та оплати, термін дії та гарантії щодо продукту.

Перевагами інтернет-магазину є:

- можливість придбати будь-який товар або деяку послугу;
- немає прив'язки до міста чи країни, що збільшує кількість потенційних покупців;
- доступний цілодобово;
- можливість порівняння товарів.

Єдиним недоліком є наявність шахраїв та той факт, що товар неможливо оглянути перед покупкою, але так як більшість магазинів дозволяють оплатити товар після його отримання та огляду, то дана проблема не є актуальною.

Результатом виконання проекту буде впроваджений інтернет-магазин для компанії “BEZDNA”, використання якого підвищить рівень конкурентоспроможності компанії на ринку.

Перед початком розробки інтернет-магазину було проведено дослідження вже існуючих веб-ресурсів за схожою тематикою. В ході аналізу було визначено сильні і слабкі сторони інших сайтів, які допомогли

сформувати бачення функціоналу та інтерфейсу майбутнього інтернет-магазину.

Перший аналог – сайт [tk-ekosumka.com.ua](http://tk-ekosumka.com.ua) українських еко-сумок (рис. 1.1) [2]. Дизайн сайту не виглядає сучасним та привабливим, при цьому є дуже інформативним, але на ньому розміщено забагато тексту та посередньої інформації, що відштовхує покупців, бо сайт повинен бути максимально інтуїтивно-зрозумілий для користувача. Перевагою даного сайту є наявність горизонтального головного меню, футера з контактною інформацією та можливість замовити власний принт на еко-сумці.

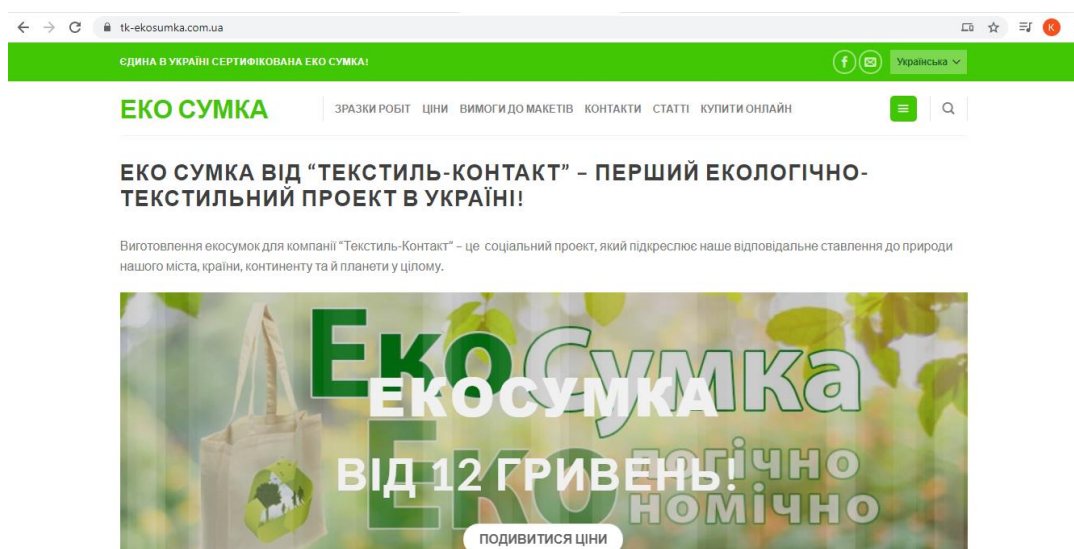


Рисунок 1.1 – Головна сторінка сайту [tk-ekosumka.com.ua](http://tk-ekosumka.com.ua)

Наступний аналог – це сайт <https://shop.theytoldme.com/> (рис. 1.2) [3]. Надзвичайно цікавий сайт, який приваблює користувачів мінімалістичністю. Привернув увагу до цього сайту не лише дизайн та простота, а ще й можливість підтримати команду “They told me” донатом. Великою перевагою даного інтернет-магазину є той факт, що користувачеві є необхідним реєструватися на сайті, аби замовити еко-сумку. Не сподобалося, що головна сторінка вміщує в себе всю інформацію про магазин та продукти, що змушує користувача багато скролити, аби дістатися певного продукту, до того ж відсутня функція, яка б повернула користувача одразу до початку сторінки в незалежності від того в якій саме частині сторінки він/вона знаходиться.



Порівняно із попереднім аналогом даний сайт виглядає більш приваблим та є набагато зручнішим у використанні, але не вистачає можливості отримати шопер з власним написом чи малюнком.

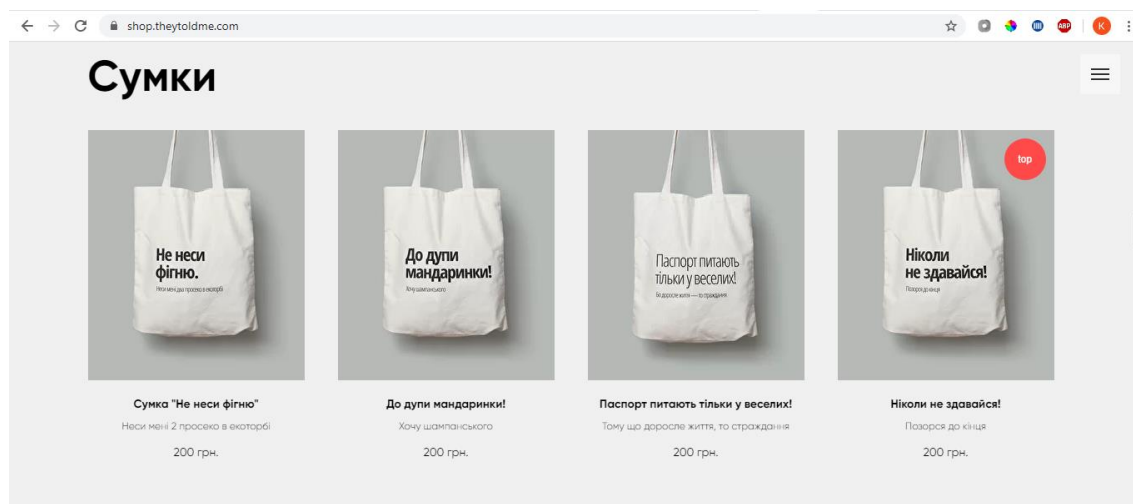


Рисунок 1.2 – Розділ «Сумки» інтернет-магазину shop.theytoldme.com

Останнім інтернет-магазином, який було розглянуто, став сайт <https://4fresh.ru/> (рис. 1.3) [4].

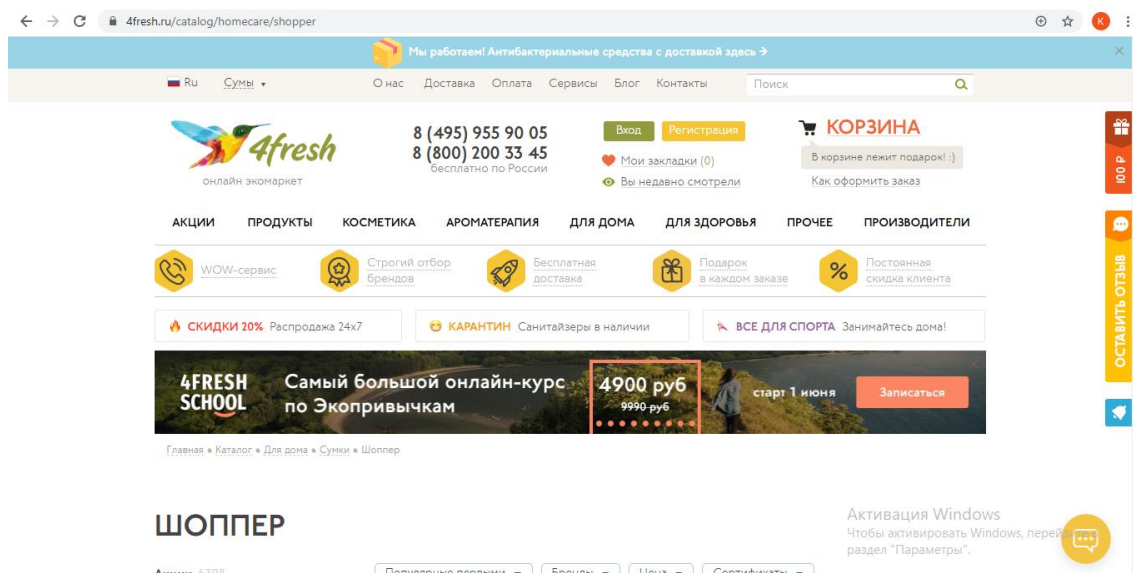


Рисунок 1.3 – Головна сторінка інтернет-магазину 4fresh.ru

Сайт не можна назвати зручним у використанні через складне меню. Інтернет-магазин містить велику кількість цікавих та корисних функцій, але через те, що всі вони розташовані у хаотичному порядку – це збиває

користувача зі шляху обирання товару та його покупки. До того ж сайт містить величезну кількість різноманітних іконок, картинок і тд. (рис. 1.4), які через свою яскравість не вписуються в основний дизайн сайту та відволікають потенційного покупця, тобто він/вона просто губиться серед різномаїття.

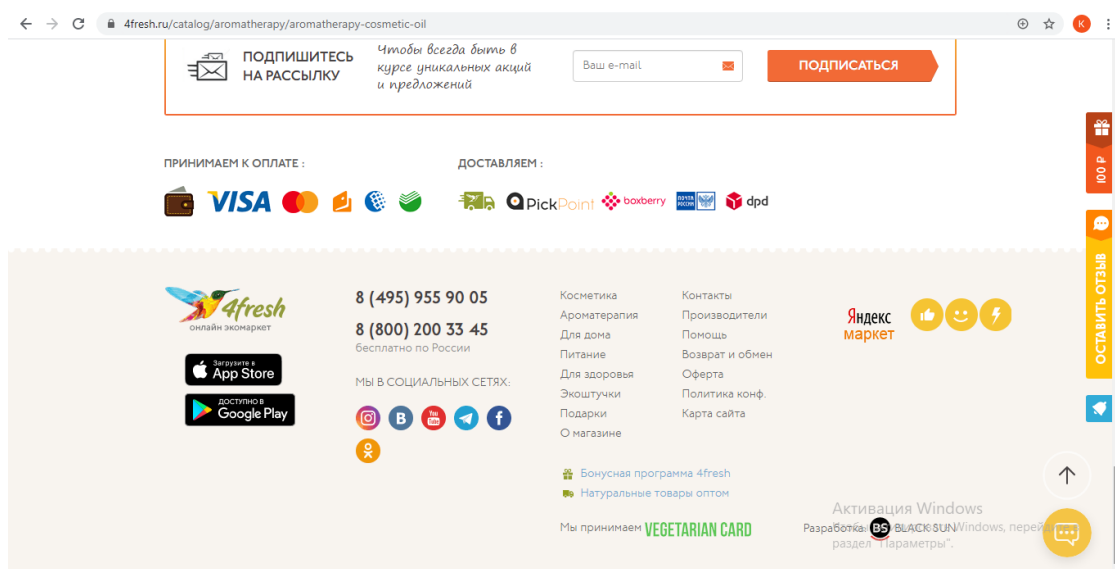


Рисунок 1.4 – Футер інтернет-магазину 4fresh.ru

Ознайомившись з популярними інтернет-магазинами, що займаються еко-товарами було отримано декілька важливих висновків, які в подальшому були враховані при розробці власного веб-ресурсу.

По-перше, мінімалістичний максимально простий дизайн є важливим пунктом при розробці веб-сайту, так як привабливим та зручним можна залучити більше потенційних покупців.

По-друге, зручно використовувати горизонтальне меню та розміщувати картинки великих розмірів на початковій сторінці сайту, на яких зображена актуальна, важлива інформація про акції, новинки і тому подібне.

По-третє, зручно користуватися інтернет-магазином, який не потребує попередньої реєстрації для оформлення замовлення, та має функцію нанесення власного принта на виріб.

Перераховані вище пункти були взяті до уваги та використані при формуванні технічного завдання та розробці інтернет-магазину.

## **1.2 Актуальність створення інтернет-магазину**

У порівнянні з минулим, людство живе у прискореному темпі і має набагато більше як можливостей, так і потреб. Ростають великі міста, жителям яких все важче встигати задовільняти свої потреби та організувати свій побут. Як наслідок попит народив пропозицію і різноманітні бізнеси були вимушені вийти в Інтернет-простір. Тож тепер завдяки веб-ресурсами ми маємо можливість зекономити час на відвідуванні магазинів у пошуках сукні або на поїзді до аеропорту з метою купити авіаквитки і віддати цей час своїй родині, самоосвіті, загалом, використати його з більшою користю та задоволенням.

Розробка інтернет-магазину має на меті розміщення інформації щодо товарів у мережі Інтернет, які можна переглядати та купувати. Це вкрай необхідно, якщо виробник регіональний та його продукція не повсякденного вжитку. Збільшити продажі, а відповідно і виробництво можна, якщо донести інформацію про свій товар до максимальної кількості потенційних покупців. У нашому випадку клієнтська аудиторія - це соціально відповідальна та активна молодь. І знайти її легше і швидше у мережі Інтернет.

Під час невеличкого дослідження було виявлено, що:

- приблизний товарообіг послуг та товарів на ринку e-commerce на сьогодні складає близько 2.91 трильярда доларів. На 2021 рік прогнозується його зростання до 3.56 триліона доларів [5];
- 31% українських користувачів Інтернету хоча б один раз робили покупку в онлайні;
- 64% користувачів мобільних пристроїв воліють купувати через мобільні додатки. Приблизно 52% вважають, що браузері в цьому плані зручніші [6];

- ближче до 2040 року на ринок електронної комерції буде приходиться приблизно 95% всієї торгівлі у світі;
- 58% людей роблять покупки онлайн, бо інтернет-магазини доступні цілодобово;
- 85% покупців бажають досліджувати продукт онлайн перед покупкою;
- 95% покупців передивляються онлайн-огляди, які допомагають більше дізнатися про продукт;
- 23% покупців відмовляться від покупки, якщо для цього буде необхідно створити акаунт (zareєstrуватися);
- 33% людей віддають перевагу компаніям, які на їхню думку турбуються про стан навколишнього середовища [7].

Щодо інтернет-торгівлі в Україні, то товарообіг на ринку e-commerce в Україні сьогодні становить приблизно 65 млрд грн. 40% українських користувачів мережі Інтернет робили онлайн покупку хоча б раз у житті.

Згідно зі статистикою та даними компанії GfK Ukraine об'єм онлайн-торгівлі в Україні зріс приблизно у 2 рази за останні 7 років. До того ж в Україні нараховується близько 20 мільйонів постійних інтернет-користувачів, 36% яких є постійними інтернет-покупцями. Як наслідок отримуємо приблизно 7 млн людей, які у більшості є жителями великих міст [21].

Відповідно до досліджень GfK, найбільш активними онлайн-покупцями - є жителі Києва та області.. Найчастіше купують в Мережі люди з повною вищою освітою у віці 26-35 років.

### **1.3 Постановка задачі**

Метою кваліфікаційної роботи бакалавра є створення інтернет-магазину, який надасть можливість користувачам ознайомитися з брендом “Bezдна”, каталогом його продукції, вибрати та замовити еко-торбинки у режимі онлайн. Експлуатація результату роботи повинно збільшити число потенційних покупців, та як наслідок збільшити дохід компанії.

Основними задачами для виконання поставленого завдання є: аналіз подібних веб-ресурсів, аналіз необхідного функціоналу веб-сайту, його клієтської та адміністративної частин, створення бази даних, розробка мокапів та зовнішнього вигляду сайту, наповнення інтернет-магазину контентом, мануальне тестування. Цей інтернет-магазин готується конкретно для фірми “BEZDNA”. Планується, що веб-ресурс буде направлений на клієнтів не лише із Сумської області, а й з усієї України.

Отже, необхідно виконати наступні кроки для успішного створення інтернет-магазину:

- провести пошук та проаналізувати інформацію згідно теми роботи;
- проаналізувати подібні веб-ресурси, виділити їх переваги та недоліки;
- обрати найбільш зручні технічні та програмні засоби для реалізації веб-ресурсу;
- розробити сучасний дизайн веб-сайту, який буде відповідати основним тенденціям UI/UX дизайну;
- розробити адміністративну та клієнтську частини сайту;
- наповнити інтернет-магазин контентом;
- протестувати веб-сайт;

Сайт повинен бути зручним, сучасним та інтуїтивно зрозумілим у використанні. Для досягнення цієї мети було обрано відповідний дизайн та інтерфейс веб-ресурсу.

Для адміністрування інтернет-магазину буде використовуватися панель управління. Адміністратор інтернет-магазину матиме можливість додавати, видаляти, редагувати контент сайту, а також отримуватиме запити від користувачів, які бажають зробити замовлення.

## 2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

### 2.1 Вибір фреймворків

Після обговорення та отримання завдання, необхідно було вирішити, яким шляхом найкраще реалізувати проект, та що саме для цього використовувати. Тисячі сайтів створюються кожного дня і велика їх кількість написана за допомогою конструкторів сайтів, найвідомішими з яких є WIX, WordPress та uKit. Не зважаючи на переваги, конструктори сайтів мають ряд недоліків: обмежений дизайн та функціонал, користувач не є власником сайту, доступ до найбільш корисних функцій надається лише через платну підписку і тд. Як відомо, написання будь-якого веб-ресурсу складається з використання декількох мов програмування, кожна з яких має свої специфічні функції та напрямки. Загалом ці мови поділяються на фронтенд - відповідає за зовнішній вигляд сайту та бекенд – відповідає за серверну частину.

Одним із найпопулярніших інструментів, що спрощує створення інтерфейсу є фреймворк Bootstrap. Фреймворк – це набір інструментів, які слугують для прискорення розробки веб-ресурсу. Тобто фреймворк - це скелет, який включає в себе велику кількість бібліотек, що спрощують розробку сайту. Фреймворки можуть відрізнятися своїми функціями в залежності від завдань, які поставлені перед ними. Їх основним завданням є надати програмісту можливість приділяти увагу більш унікальним задачам, а не витратити час на всім відомі основоположні задачі та функції [8]. Bootstrap - це безкоштовна CSS-інфраструктура, яка орієнтована на адаптивну розробку. Він містить CSS-шаблони на основі JavaScript, а також форми, кнопки, елементи навігації та інші частини інтерфейсу [9].

Переваги використання Bootstrap:

- адаптивність;
- крос-браузерність;

- зрозумілий код;
- ідентичність стилів.

Недоліки використання Bootstrap:

- шаблонність;
- відсутність підтримки у старих браузерах;
- всі сайти зроблені за допомогою даного фреймворку схожі один на один.

За допомогою вищеописаного фреймворку буде побудовано каркас інтернет-магазину. Але лише привабливого зовнішнього вигляду недостатньо. Тут на допомогу приходить JavaScript, а саме Node.js - це серверна платформа для роботи з JavaScript. JavaScript виконує дію на стороні клієнта, а Node.js - на сервері. Node.js вміє працювати з зовнішніми бібліотеками, викликати команди з коду на JavaScript і виконувати роль веб-сервера [10].

Також буде використано Vue.js - це веб-фреймворк для розробки призначених для користувача інтерфейсів на мові програмування JavaScript [11]. Відповідно до статистики за 2019 рік, 40.5% розробників вже використовує Vue.js і 34.2% хотіли б вивчити його в якості нового інструменту [12].



Рисунок 2.1 – Статистика використання фреймворків за 2019 рік

Центральна концепція Vue.js - це концепція компонент (компонент - це екземпляр Vue з попередньо встановленими опціями), тобто невеликих частин інтерфейсу користувача, які можна використовувати декілька разів. Таким чином, і весь додаток складається з різних частин-компонентів. Один компонент може включатив в себе кілька інших компонентів, тобто використовується ієрархія вкладеності. Кожен з таких компонент може містити в собі HTML, CSS і JavaScript-код, який властивий тільки даному компоненту і відповідає за роботу тільки цього компонента (рис. 2.1) [13].

Основними перевагами використання фреймворку Vue.js є:

- невеликий розмір фреймворку;
- простота;
- доступна, детальна та зрозуміла документація;
- простота інтеграції платформи з існуючими додатками;
- можливість запускати програму із браузеру;
- налаштовуваність.

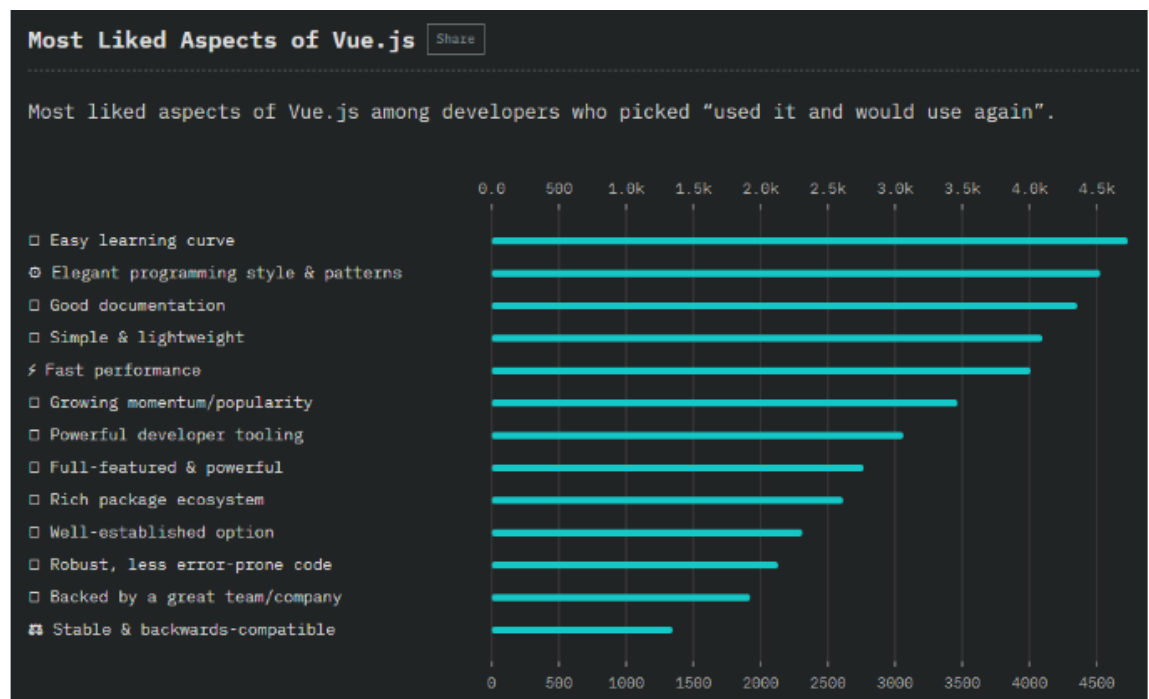


Рисунок 2.2 – Найулюбленіші аспекти у користуванні Vue.js [22]

## 2.2 Вибір системи управління базами даних



Коли справа доходить до вибору системи управління базою даних, то найважче обрати між реляційними (SQL) та нереляційними (NoSQL) базами даних [15]. Не зважаючи на те, що обидва варіанти є популярними, було розглянуто переваги і недоліки обох СУБД:

– Мова

Бази даних SQL являються більш обмежувальними, бо потребують визначення структури даних заздалегідь, ще до початку роботи з ними. На відміну від SQL, NoSQL має неструктуровану схему даних, які можуть зберігатися у документах, або за допомогою графів, або за допомогою пари ключ-значення;

– Масштабованість

У більшості баз даних SQL вони масштабуються по вертикалі, а NoSQL по горизонталі;

– Співтовариство

Не зважаючи на те, що популярність NoSQL швидко зростає, його спільнота не така велика у порівнянні із співтовариством SQL, який має тисячів чатів та форумів, на яких експерти дають поради, щодо використання кращих практик SQL;

– Структура

Основою баз даних SQL є таблиці, а основою NoSQL баз даних можуть бути документи, пари ключ-значення, графи чи навіть стовпці. (рис. 2.3) [14].

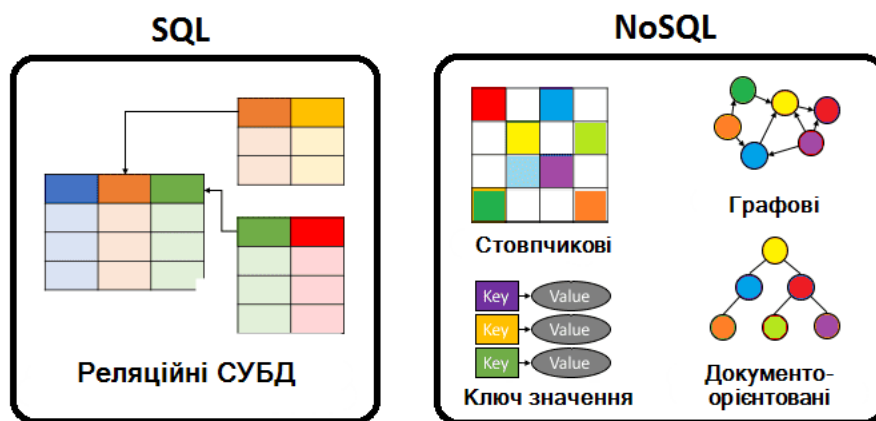


Рисунок 2.3 – Графічне представлення SQL та NoSQL СУБД [24]

Згідно статистичних досліджень проведених у 2019 році NoSQL СУБД користується майже 40% організацій. Щодо популярності СУБД, то MySQL безперечно домінує з 38,9% використання, за ним йде MongoDB з 24,6%, PostgreSQL з 17,4%, Redis з 8,4% і Cassandra з 3,0% (рис. 2.4) [16].

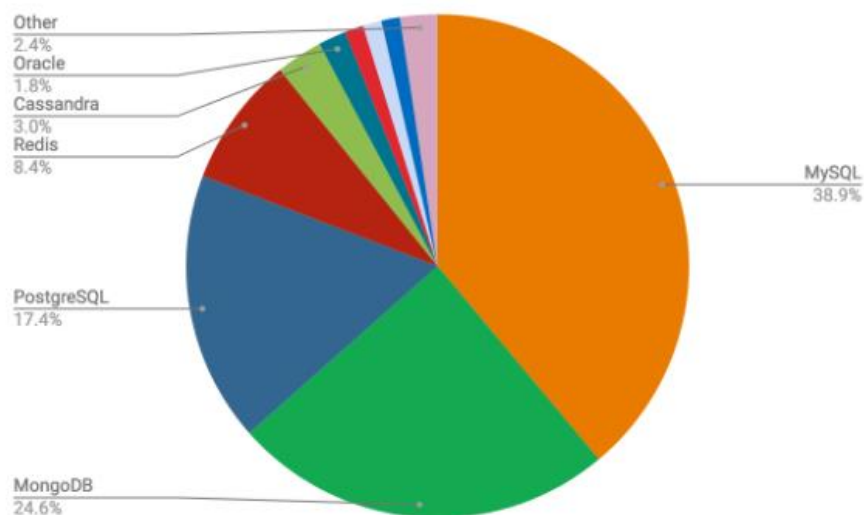


Рисунок 2.4 – Діаграма використання СУБД

MongoDB є NoSQL системою управління базою даних і як видно з діаграми набирає популярності. MongoDB - це крос-платформна СУБД, яка забезпечує високу продуктивність і легку масштабованість. Дана СУБД має в основі не звичні таблиці, а колекції та документи, які зберігають в собі дані у форматі ключ-значення.

Документоорієнтовані БД зручно використовувати для задач, де необхідно зберігати інформацію впорядковано, але можна обійтися без

величезних кількості зв'язків між даними. Також схема не є невід'ємною частиною документу, тобто кожен документ може мати будь-яку кількість полів, що і відрізняє MongoDB та власне будь-які NoSQL СУБД від реляційних, бо при спробі зберегти різномірні дані отримаємо пусті поля.

На рисунку 2.5 зображені еквіваленти, які використовуються у MongoDB у порівнянні із RDBMS (Relational Database Management System).

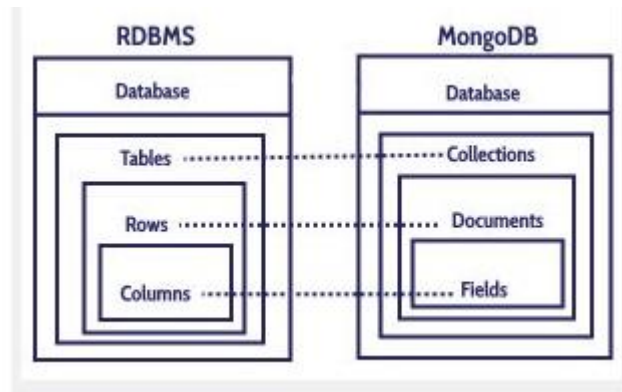


Рисунок 2.5 – Відповідність між RDBMS та MongoDB

Основні переваги використання MongoDB [23] :

- Досить легко встановити та налаштувати MongoDB;
- MongoDB є безкоштовною;
- Звичні join операції не підтримуються MongoDB, так як зв'язок між даними відсутній;
- Мова запитів документів більш проста та зрозуміла для користування у порівнянні з SQL запитам;
- Всі дані зберігаються у форматі документів у вигляді JSON файлів;
- MongoDB є більш продуктивною у порівнянні із реляційними базами даних.

Для розробки був обраний Visual Studio Code - редактор коду, розроблений Microsoft. Він оснащений доступним набором інструментів для редагування та налагодження. Редактор легко інтегрується з іншими сервісами. Його персональні властивості також легко розширити [17].

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 3.1 Структурно-функціональне моделювання процесу розробки веб-ресурсу для компанії “BEZDNA”

IDEF0 – це нотація графічного моделювання, яка використовується для створення функціональної моделі, що відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, які пов'язують ці функції. Стандарт IDEF0 (Integration Definition for Function Modeling) затверджений в США в 1993 році в якості стандарту обробки інформації. Методологія IDEF0 являється одним із найпопулярніших підходів для опису бізнес-процесів.

До особливостей можна віднести:

- використання контекстної діаграми  
контекстна діаграма – діаграма, яка знаходиться у вершині, де об'єкт моделювання поданий у вигляді одного блоку зі стрілками, які показують взаємозв'язок об'єкта із зовнішніми чинниками;
- декомпозиція  
нотація IDEF0 надає можливість використовувати декомпозицію процесу до необхідного рівня деталізації, тобто розбивати основний процес на його підпроцеси. Дочірня діаграма буде охоплювати ту саму область, що і головний процес, але описує її більш детально. Всі стрілки основного процесу при декомпозиції переносяться на дочірні процеси у виді граничних стрілок; стрілок.
- домінування  
при декомпозиції блоки моделі розташовуються по діагоналі від лівого верхнього кута до нижнього правого кута. Таким чином

показується як впливає один блок на інший - це і мається на увазі під домінуванням;

– 4 типи стрілок

Всього існує 4 типи стрілок: Вхід, Вихід, Механізми, Управління.

Розглянемо функціональну модель процесу розробки веб-ресурсу для компанії «BEZDNA» (рис. 3.1).



Рисунок 3.1 – Контекстна діаграма

Діаграма складається з блоку та стрілок. Блок діяльності відповідає за конкретний процес. Стрілками позначають об'єкти, які впливають на процес, або є його результатом. Стрілки, в залежності від свого напрямку і положенню відносно блоку, мають різий зміст:

- вхідні (стрілки, які входять у блок з лівої сторони) - ставлять певне завдання, надають дані, які використовуються процесом, щоб створити те, що з'явиться на виході;
- вихідні (стрілки, які виходять за правого боку блоку) - виводять результат діяльності вироблені процесом;
- керуючі (зверху вниз) - механізми управління, тобто будь-які правила, інструкції, положення, які ставлять певні умови для проходження певного процесу;

- механізми (знизу до верху) – визначають, що саме використовується для проходження процесу, тобто ресурси для виконання.

В даному випадку вхідними стрілками процесу «Розробка веб-ресурсу для компанії «BEZDNA» є замовлення і контент. У результаті буде отриманий інтернет-магазин «BEZDNA». Керуючим механізмом є керівник та вимоги замовника. Виконавиця ,СУБД та фреймворки - є механізмами процесу.

Далі необхідно виконати декомпозицію контекстної діаграми, яка призначена для деталізації функцій і формується при розбитті контекстної діаграми на підзадачі, що описують кожен дочірній блок і їх взаємодію між собою [18].

Основний процес буде поділено на наступні блоки під час декомпозиції першого рівня:

- створення БД;
- розробка макету;
- реалізація компонент сайту;
- тестування.

Діаграма декомпозиції першого рівня показана на рисунку 3.2.

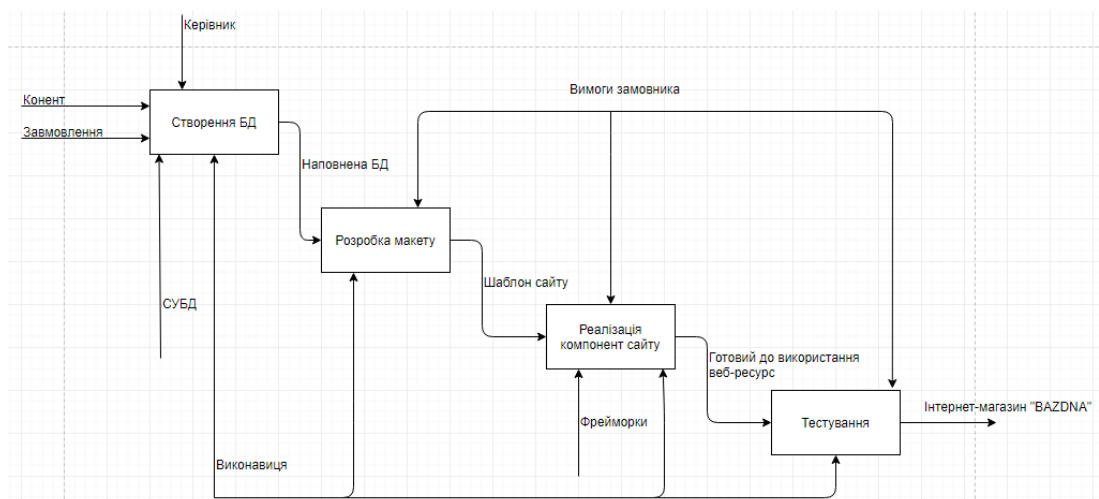


Рисунок 3.2 – Діаграма декомпозиції першого рівня

Вхідною стрілкою до блоку «Створення БД» є «Замовлення» та «Контент»; вихідною – «Наповнена БД»; стрілка контролю – «Керівник»; стрілками механізмів – «Виконавиця» та «СУБД».

Вхідною стрілкою до блоку «Розробка макету» є «Наповнена БД»; вихідною – «Шаблон сайту»; стрілки контролю – «Вимоги замовника» стрілками механізмів – «Виконавиця».

Вхідними стрілками до блоку «Реалізація компонент сайту» є «Шаблон сайту»; вихідною – «Готовий до використання веб-ресурс»; стрілки контролю – «Вимоги замовника»; стрілками механізмів – «Виконавиця» та «Фреймоврки».

Вхідною стрілкою до блоку «Тестування» є «Готовий до використання веб-ресурс»; вихідною – «Інтернет-магазин “BEZDNA”»; стрілка контролю – «Вимоги замовника»; стрілками механізмів – «Виконавиця».

### **3.2 Моделювання варіантів використання веб-ресурсу для компанії “BEZDNA”**

Для того, аби у повній мірі зрозуміти всі можливі варіанти роботи з розробленим веб-ресурсом була побудована діаграма варіантів використання (рис. 3.3).

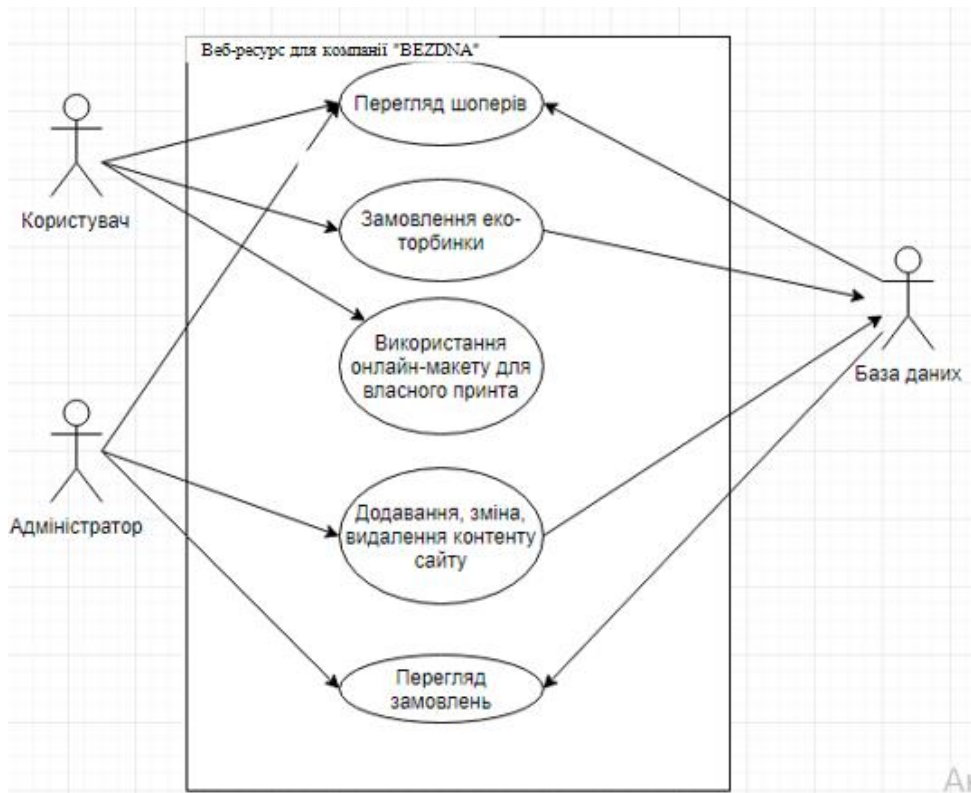


Рисунок 3.3 – Діаграма варіантів використання

Суть даної діаграми полягає в тому, що спроектована система подається у вигляді безлічі сутностей або акторів, які за допомогою варіантів використання взаємодіють із системою. При цьому актором називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на систему. У свою чергу, варіант використання служить для опису послуг, які система надає актору. Тобто кожен варіант використання визначає певну послідовність дій, які виконує система при взаємодії актора з нею. Інформація про те, як саме буде реалізована дана взаємодія, не входить до діаграми [19].

Для веб-ресурсу компанії “BEZDNA” було визначено декілька акторів:

- Адміністратор – людина, яка керує сайтом та його контентом, приймає заявки користувачів про власний принт на виробі (в даній ситуації цю роль буде виконувати власник компанії “BEZDNA”);



- Користувач – клієнт, який перейшов на сторінку інтернет-магазину та отримав можливість ознайомитися з усією необхідною інформацією про компанію, товари та послуги, що вона надає, та замовити шопер з власним чи готовим малюнком;

- База даних – база даних, яка включає всю необхідну інформацію щодо інтернет-магазину, зберігає, видаляє, оновлює необхідну інформацію.

Варіантами використання для веб-ресурсу для компанії “BEZDNA” є:

- ВВ Використання онлайн-макету для власного принту – ВВ дає можливість переглянути як саме буде виглядати принт на шопері;

- ВВ Замовлення еко-торбинки – ВВ дозволяє клієнту обрати будь-який шопер з існуючого переліку, додати до кошика та замовити;

- ВВ Перегляд шоперів – ВВ дозволяє дізнатися про шопери, які є в наявності;

- ВВ Додавання, зміна, видалення контенту сайту – ВВ дозволяє адміністратору додавати, видаляти чи змінювати інформацію про товари, які знаходяться на сайті;

- ВВ Перегляд замовлень – ВВ надає адміністратору можливість бачити замовлення на адмін-панелі.

### **3.3 Інформаційна модель**

Як правило, робота над створенням сайту починається з оформлення прототипу, його докладної структури. Найпростіший спосіб це зробити - намалювати ручкою на папері. Ось неповний список проблем і завдань, які допомагає вирішити прототип:

- візуалізація ідеї, уявлення про зовнішній вигляд майбутнього сайту на ранніх стадіях;

- можливість внести зміни і фіналізувати бачення ціною мінімальних витрат;

- можливість більш точної оцінки термінів і бюджету повного обсягу робіт;
- розуміння перспектив використання і розвитку сайту, раннє обговорення гіпотез [20].

Були розроблені наступні прототипи сторінок для інтернет-магазину «BEZDNA». Попередньо із амовником було обговорено той факт, що в ході розробки деякі деталі може бути видозмінено з його згоди.

У кожному прототипі показаний вигляд головного меню та футера. Згідно з прототипу (рис.3.4) головна сторінка окрім головного меню та футеру міститиме баннер з новинками товарів, акціями та новинами. Нижче буде знаходитись слайдер із популярними товарами та кнопка, яка перенаправить користувача до магазину, де будуть знаходитись всі товари компанії «BEZDNA».

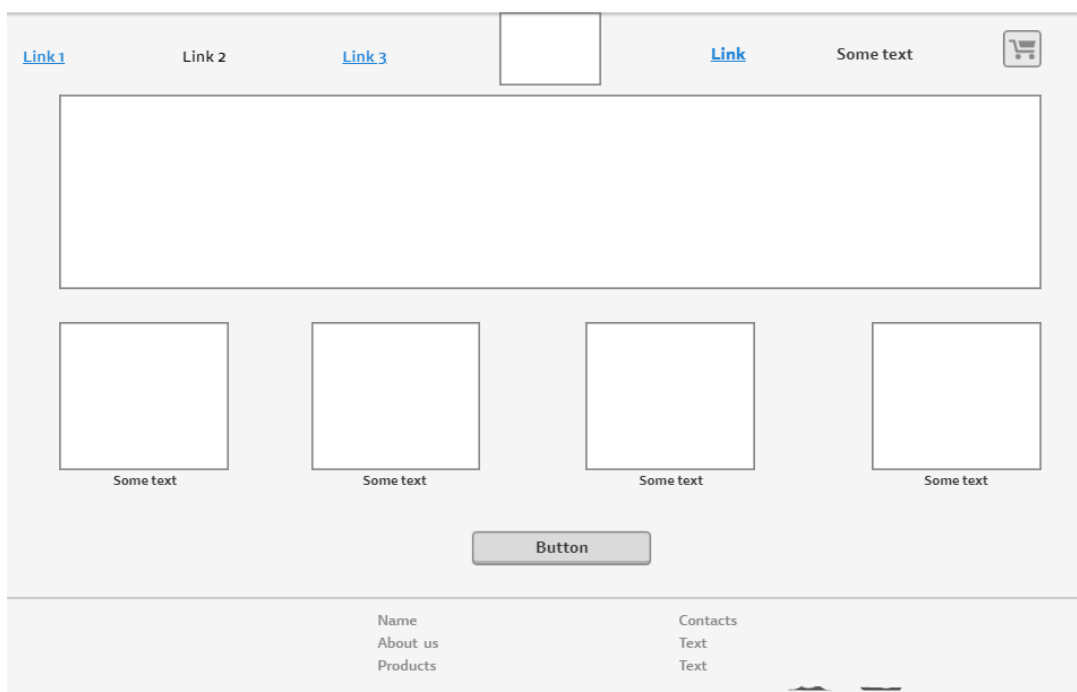


Рисунок 3.4 - Прототип головної сторінки

Сторінка товарів міститиме всі наявні товари з назвою та ціною кожного. Також буде можливість перемикати сітку товарів із трьох в ряд до одного в ряд, таким чином детальний опис товару буде краще видно.

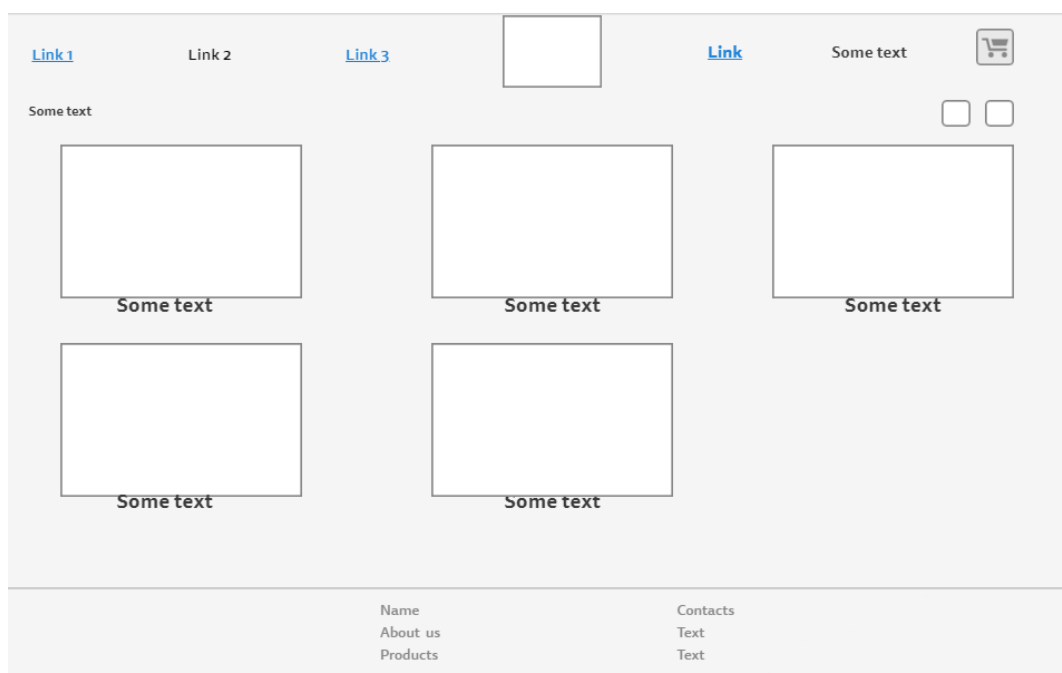


Рисунок 3.5 - Прототип сторінки товарів

На сторінці кошику користувачу буде виведена картинка товару, його назва, ціна, кількість та загальна сума товару.

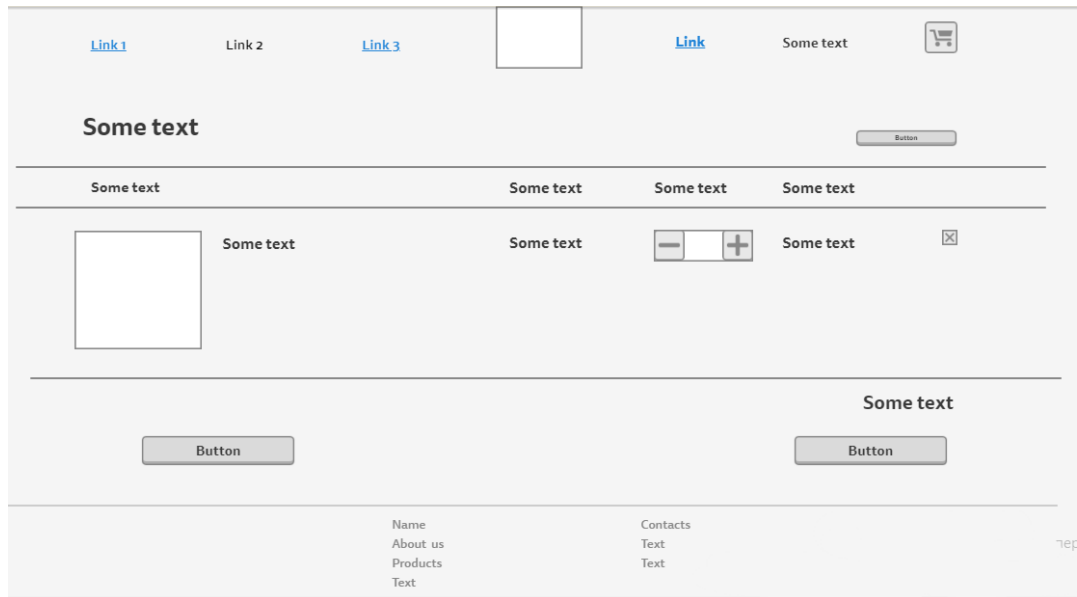


Рисунок 3.6 - Прототип сторінки кошику

На даному веб-ресурсі застосована блочна структура побудови сайту, яка найбільш поширена структура в Інтернеті. Тобто в даному випадку кожна сторінка може посилатися на декілька інших, з нею пов'язаних.

### 3.4 Підготовка середовища розробки

Для того, щоб розпочати безпосередньо створення сайту, необхідно виконати наступні кроки:

- встановити VS Code;
- встановити Node.js;
- виконати інсталяцію за допомогою пакентого менеджера при базових залежностей проекту, таких як express, mongoose і тд;
- підключити базу даних .

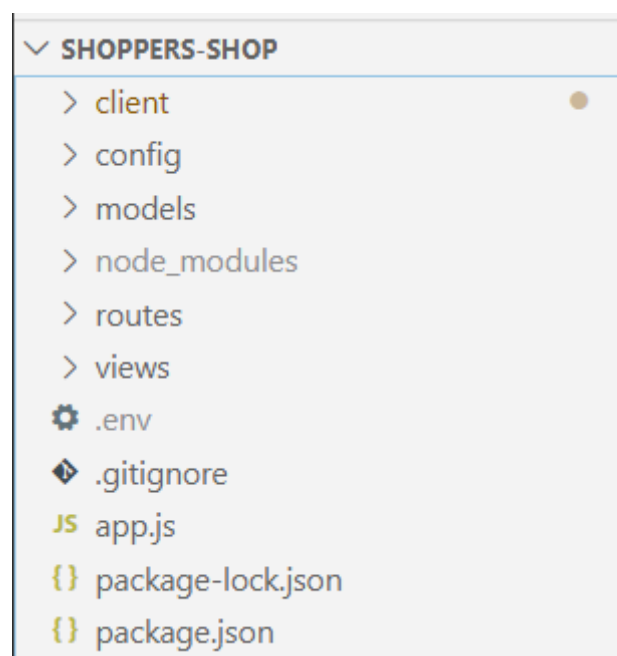
Інсталяція редактору не займає багато часу. Для цього необхідно перейти на офіційну сторінку Visual Studio Code та завантажити редактор коду VS Code.

Для того, щоб встановити Node.js необхідно виконати наступні кроки. Необхідно завантажити поточну версію з офіційного сайту. Перевірити факт, що Node.js встановлений можна за допомогою команди `node -v`, написану у консолі. Для автоматизації установки і оновлення пакетів, як правило, застосовуються систему управління пакетами або менеджери. Безпосередньо в Node.js для цієї мети використовується пакетний менеджер NPM (Node Package Manager). NPM за замовчуванням встановлюється разом з Node.js, тому нічого додатково встановлювати не потрібно.

Будуть використані наступні команди npm:

- `npm init` - створює `package.json`, який визначає параметри проекту;
- `npm install [module]` – завантажує готові модулі, які можна використовувати у розробці, наприклад, `express`, `mongoose`, `dotenv` та інші;
- `npm install --save-dev` - встановлює залежності для розробки;
- `npm install -g @vue/cli` - ініціалізація проекту (клієнт);
- `npm run dev` – дозволяє запустити проект у режимі розробки;
- `npm express install --save save express` – встановлює веб-фреймворк Express, який є локальним сервером.

Структура проекту зображена на рисунку 3.7.



### Рисунок 3.7 – Структура проекту “BEZDNA”

Так як Vue.js складається з компонент, то в цілому спрощену структуру проекту можна зобразити у вигляді ієрархії (рис 3.8).

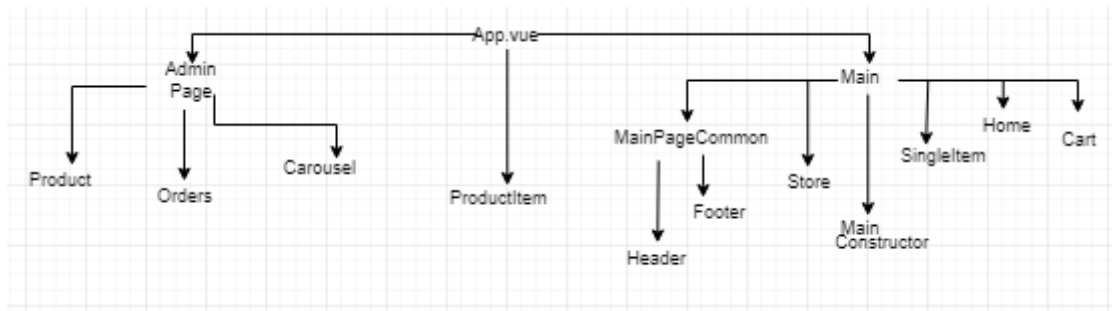


Рисунок 3.8 – Ієрархія компонентів проекту “BEZDNA”

Управління базою даних здійснюється через адмін-панель. Сама база даних, виглядає наступним чином (рис. 3.9):

- products: містить детальну інформацію про кожний шопер;
- maincarousels: містить інформацію про блоки каруселі;
- orders: містить інформацію про замовлення.

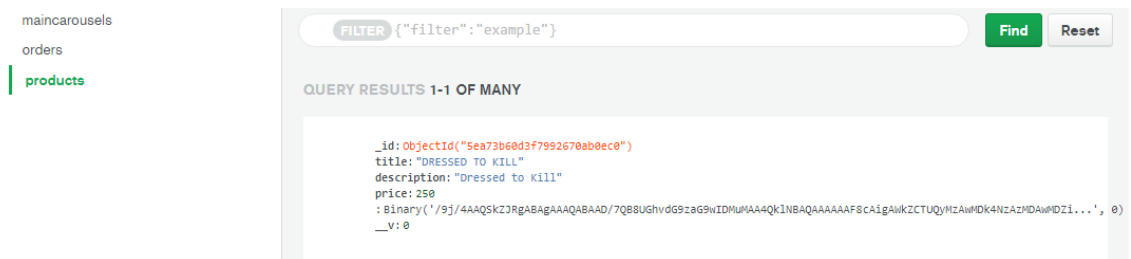


Рисунок 3.9 – Структура бази даних

Завдяки фреймворку Bootstrap була знайдена тема, що підійшла для тематики інтернет-магазину. Після редагування основної теми, внесення деяких коректив з боку компанії, було отримано результат, що цілком задовільнив замовника. Видозмінені CSS-елементи були додані до відповідних компонентів, у результаті чого була сформована верстка та основний зовнішній вигляд проекту.

Одним із основних файлів є App.vue, тому що саме він є кореневим глобальним компонентом, який вміщує в собі всі глобальні налаштування,

наприклад, стилі. App.vue слугує контейнером для рендерингу всіх дочірніх компонент усередині себе. Фрагмент файлу App.vue наведено на рисунку 4.4.

```

193 </style>
194 <script>
195 import Header from "./components/Header"
196 import Footer from "./components/Footer"
197 import ConfirmPopup from "./components/Cart/ConfirmPopup"
198 import "materialize-css"
199 import "vue-loading-overlay/dist/vue-loading.css"
200
201 export default {
202   components: { ConfirmPopup, Footer, Header },
203 }
204 </script>
---
```

Рисунок 3.9 – Частина компоненти App.vue

Також була створена примітивна адмін-панель (рис. 3.10 – 3.11), яка дозволяє управляти контентом веб-ресурсу.

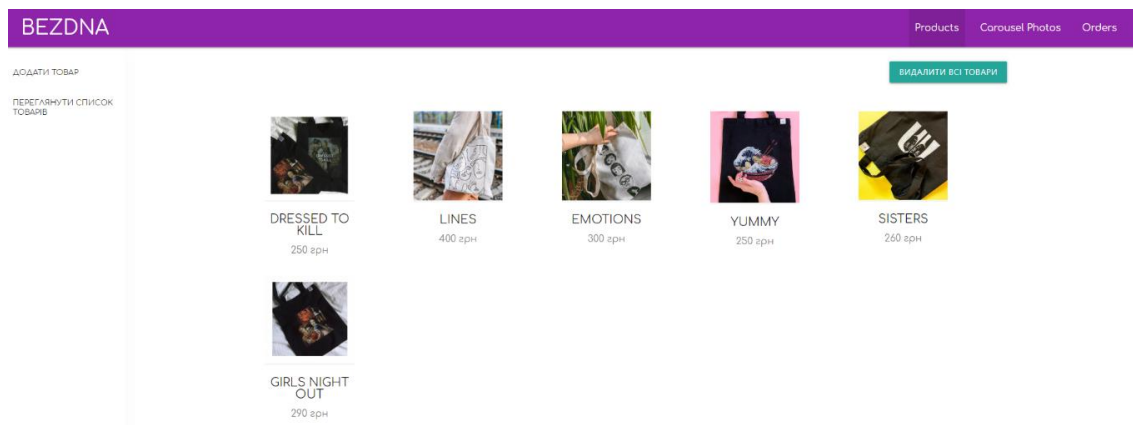


Рисунок 3.10 – Перегляд існуючих товарів на панелі адміністратора

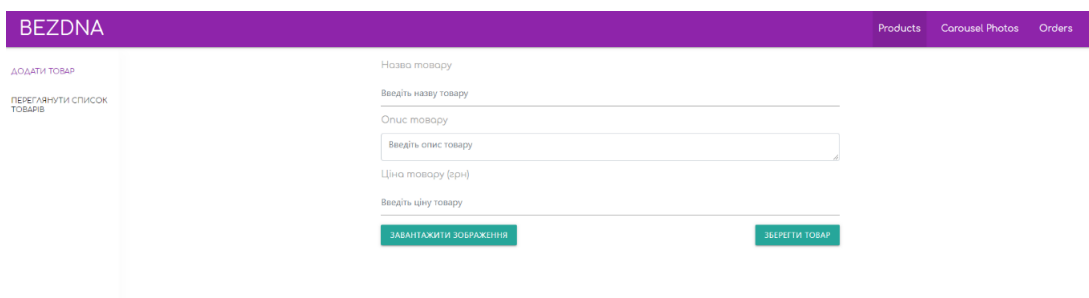


Рисунок 3.11 – Можливість додати товар на панелі адміністратора

### 3.5 Результат реалізації веб-ресурсу

У результаті реалізації на головній сторінці інтернет-магазину “BEZDNA” (рис. 3.12) користувач може переглянути популярні товари з опцією перегляду та додавання до кошику, або основні новини та інформацію про акції.

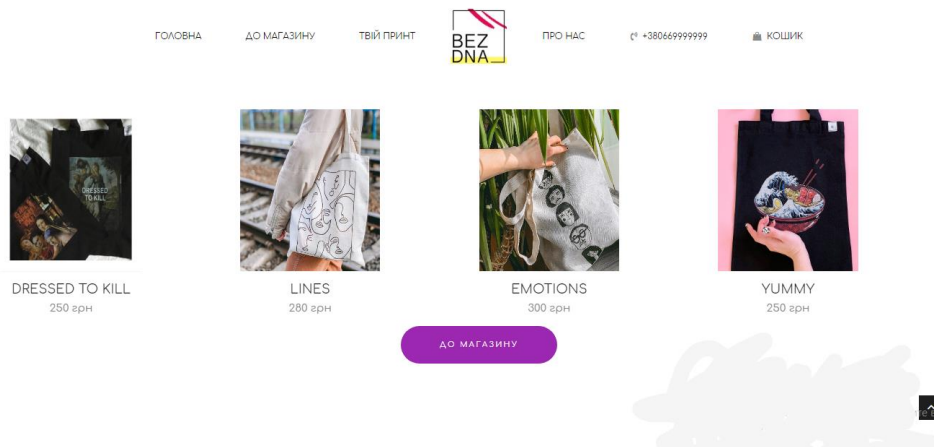
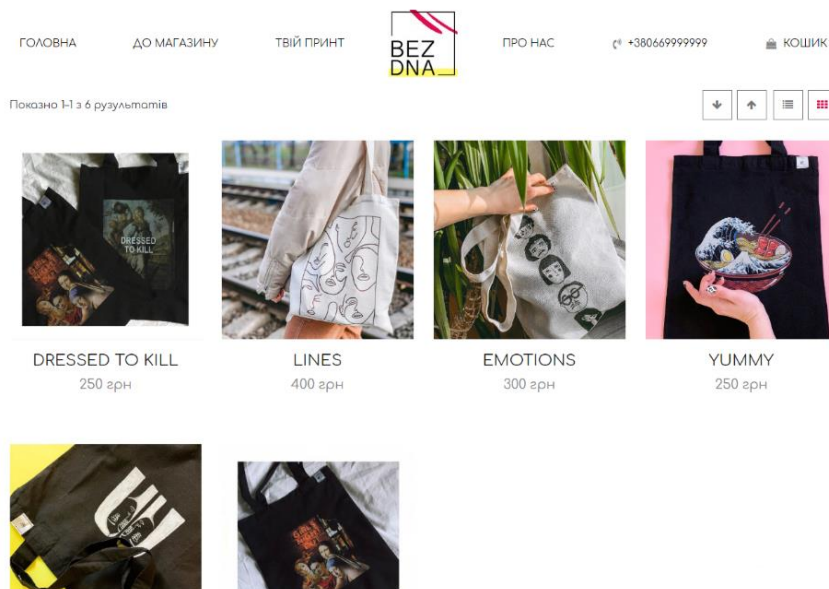


Рисунок 3.12 – Головна сторінка

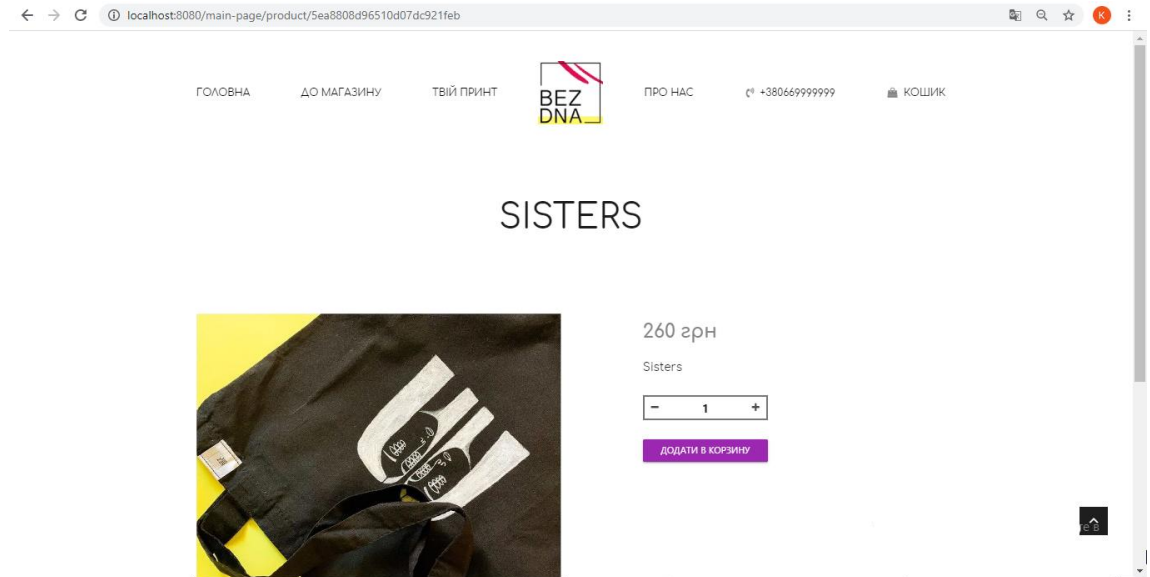
Наступною сторінкою є сторінка «До магазину» (рис. 3.13). На даній сторінці користувач має можливість переглянути товари або додати його до кошику, натиснувши на іконку «кошик». Також є можливість змінювати формат відображення товарів із трьох у ряд до одного.





### Рисунок 3.13 – Сторінка «До магазину»

Натиснувни на перегляд товару , а саме на іконку пошуку, яка з’являється при наведенн на товар, користувачу буде показана сторінка конкретного шоперу з можливістю додати до кошику різну кількість товару, а також переглянути деталі товару (рис. 3.14).



### Рисунок 3.14 – Сторінка з інформацією про шопер

Додавши товар до кошику, буде відображено повідомлення, який саме шопер був доданий до кошику (рис. 3.15).



### Рисунок 3.15 – Повідомлення про доданий до кошику товар

При наведенні на «Кошик» у стрічці головного меню, відображається картка з інформацією про кількість товару та сумму замовлення (рис. 3.16).



Рисунок 3.16 – Картка з інформацією про замовлення

На сторінці «Кошик» (рис. 3.17) користувач має змогу переглядати додані до кошика товари, а саме: зображення та назву шоперу, ціну однієї еко-сумки, змінювати кількість обраного товару. Відповідно до цього буде розрахована загальна сума замовлення. Після натискання кнопки «Оформити замовлення» користувачу буде надана можливість заповнити деталі замовлення.

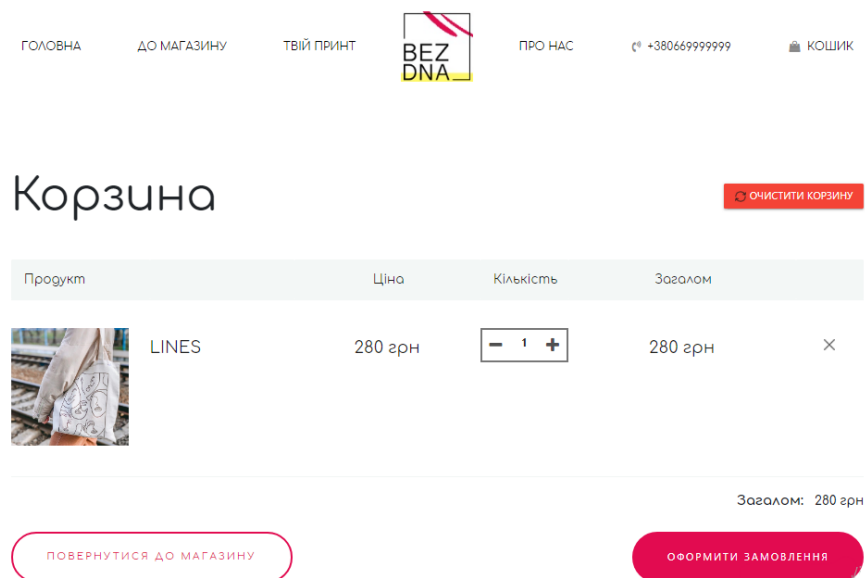


Рисунок 3.17 – Сторінка «Кошик»

Після натискання кнопки «Оформити замовлення» користувачу буде необхідно заповнити деталі замовлення (рис. 3.18).

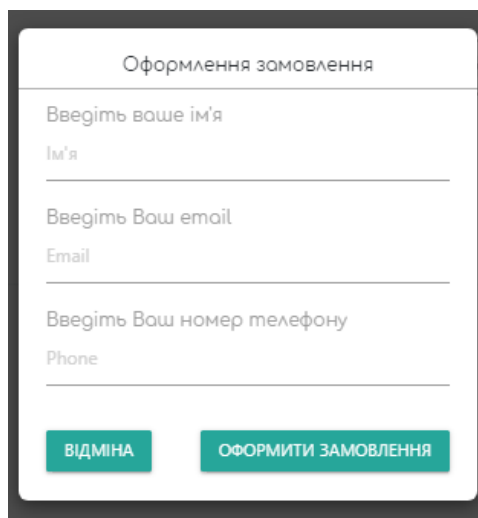


Рисунок 3.18 – Деталі замолення

Результат використання форми оформлення замовлення у вигляді повідомлення на електронній пошті зображено на рисунку 3.19. У повідомленні наведено номер замовлення, ПІБ користувача, що замовив шопер, його/її номер телефону, електронна пошта, та сума замовлення. Також наведені телефон та електронна пошта, за якими можна зв'язатися з компанією «BEZDNA».

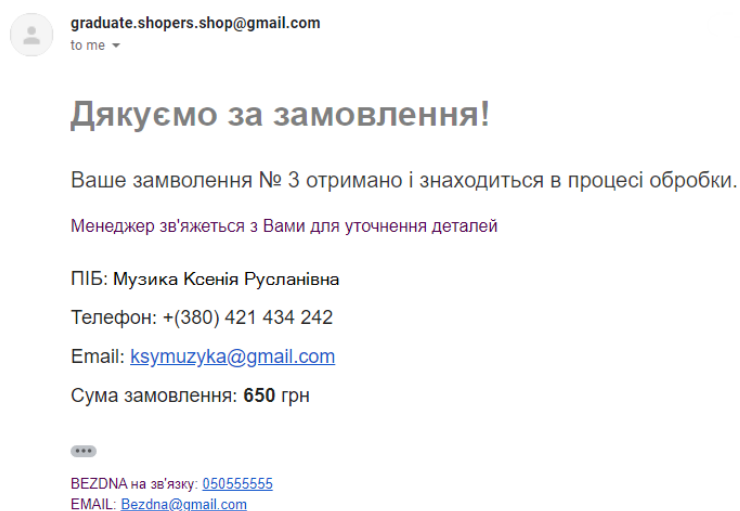


Рисунок 3.19 – Результат роботи оформлення замовлення

На сторінці «Мій принт» (рис. 3.20) міститься онлайн-конструктор макету, який дозволяє користувачу приміряти, як саме буде виглядати деяке

зображення чи надпис на шопері. Задля зручності була додана можливість обирати шопери різного однотонного кольору.



Рисунок 3.20 – Сторінка «Мій принт»

## ВИСНОВКИ

У ході виконання даної роботи було створено веб-ресурс для компанії 'BEZDNA' з використанням алгоритму онлайн-конструктора макету. На сьогоднішній день будь-який успішний бізнес повинен мати свою власну веб-сторінку, аби бути доступним для більшої кількості користувачів, що збільшить попит на продукцію чи послуги та прибуток компанії.

У ході підготовки до виконання проекту було досліджено актуальність використання екологічно чистих товарів, а саме шоперів власного виробництва, проаналізовано ресурси, які займають рекламою або продажем товарів схожої направленості, виділено їхні недоліки та переваги. У результаті аналізу було визначено необхідність створення інтернет-магазину для подальшого розвитку бренду "BEZDNA" як представника малого бізнесу.

Перед початком розробки разом із замовником було складено технічне завдання, з огляду на яке було обрано найбільш зручні методи та інструменти для розробки, а саме:

- Фронт-енд фреймворк Bootstrap

- Фреймворк Vue.js для створення користувальницьких інтерфейсів
- NoSQL система управління базами даних MongoDB

За допомогою фреймворку Bootstrap була обрана тема для сайту, яка відповідає основним сучасним тенденціям у сфері UI/UX дизайну. Створивши клієнтську та адміністративну частини сайту його розробку можна вважати завершеною. Готовий веб-ресурс для компанії містить велику кількість компонент, які ієрархічно вкладаються одна в одну та відповідають за певний функціонал чи частину сторінки. Після наповнення сайту контентом був отриманий готовий продукт, який було протестовано задля виявлення та виправлення дефектів. Тож у кінці процесу був отриманий готовий до експлуатації інтернет-магазин, який надасть користувачам можливість ознайомитись з брендом “Bezдна”, каталогом його продукції, вибрати та замовити еко-торбинки у режимі онлайн.

## СПИСОК ЛІТЕРАТУРИ

1. MOSSEO [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mosseo.ru/seo-slovar/veb-resurs/>
2. Інтернет-магазин Еко сумка від «Текстиль - Контакт» [Електронний ресурс] – Режим доступу до ресурсу: [https://www. tk-ekosumka.com.ua/](https://www.tk-ekosumka.com.ua/)
3. Інтернет-магазин – «TheyToldMe» [Електронний ресурс] – Режим доступу до ресурсу: <https://shop.theytoldme.com/>
4. 4fresh – Онлайн екомаркет [Електронний ресурс] – Режим доступу до ресурсу: <https://4fresh.ru/>
5. AG.Marketing [Електронний ресурс] – Режим доступу до ресурсу: <https://ag.marketing/elektrona-komerciya-v-ukraini-2018/>
6. LEMARBET – SEO + CRO Agency [Електронний ресурс] – Режим доступу до ресурсу: <https://lemarbet.com/ua/razvitie-internet-magazina/interesnye-fakty-ob-internet-torgovle/>

7. ECOMHUB – Информационный портал электронной коммерции [Электронный ресурс] – Режим доступа до ресурсу: <https://ecomhub.ru/samyj-polnyj-obzor-statistiki-jelektronnoj-torgovli-v-mire-za-2019/>
8. Community by timeweb [Электронный ресурс] – Режим доступа до ресурсу: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>
9. ИТШЕФ [Электронный ресурс] – Режим доступа до ресурсу: <https://itchief.ru/bootstrap/introduction>
10. Руководство по Node.js для начинающих [Электронный ресурс] – Режим доступа до ресурсу: <https://frontender.info/absolute-beginners-guide-to-nodejs/>
11. What is Vue.js? [Электронный ресурс] – Режим доступа до ресурсу: <https://vuejs.org/v2/guide/>
12. Front End Frameworks [Электронный ресурс] – Режим доступа до ресурсу: <https://2019.stateofjs.com/front-end-frameworks/>
13. SQL против NoSQL [Электронный ресурс] – Режим доступа до ресурсу: <https://tproger.ru/translations/sql-vs-nosql/>
14. An Overivew of Vue.js [Электронный ресурс] – Режим доступа до ресурсу: <https://flaviocopes.com/vue-introduction/>
15. NoSQL новая методология разработки нереляционных баз данных / М. Фаулер, П. Садаладж // Издавництво Диалектика-Вильямс Москва – 2017.
16. Тенденции баз данных в 2019 – SQL против NoSQL, Лучшие бази данных [Электронный ресурс] – Режим доступа до ресурсу: <https://andreyex.ru/bazy-dannyh/tendentsii-baz-dannyh-v-2019-sql-protiv-nosql-luchshie-bazy-dannyh-ispolzovanie-odnoj-ili-neskolkih-baz-dannyh/>
17. Visual Studio Code [Электронный ресурс] – Режим доступа до ресурсу: <https://code.visualstudio.com/>

18. Функциональное моделирование на базе стандарта IDEF0: метод. указания / сост. Д.Ю. Киселев, Ю.В. Киселев, А.В. Вавилин. // Самара: Изд-во СГАУ, 2014. – 20 с.
19. Современные методы описания функциональных требований к системам / А. Коберн // Санкт-Петербург: Изд-во ЛОРИ, 2016.
20. Прототипирование сайта. Фундамент вашего продвижения в Интернете [Электронный ресурс] – Режим доступа до ресурсу: <https://artklen.ru/blog/post46>
21. Z Україна [Электронный ресурс] – Режим доступа до ресурсу: <https://zet.in.ua/>
22. Pros and Cons of Vue.js Framework [Электронный ресурс] – Режим доступа до ресурсу: <https://medium.com/@ronak8036/pros-and-cons-of-the-vue-js-framework-8015dcbc05ef>
23. Advantages of Mongo DB [Электронный ресурс] – Режим доступа до ресурсу: [https://acodez.in/mongodb-nosql-database/#Advantages\\_of\\_MongoDB](https://acodez.in/mongodb-nosql-database/#Advantages_of_MongoDB)
24. Какие бывают типы NoSQL СУБД DB [Электронный ресурс] – Режим доступа до ресурсу: <https://www.bigdataschool.ru/wiki/nosql>

## ДОДАТОК

```
<template>
  <div id="app">
    <router-view />
  </div>
</template>

<style lang="scss">
@import url("https://fonts.googleapis.com/css2?family=Comfortaa:wght@400;500;600;700&display=swap");
@import url("https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.4.0/css/bootstrap-grid.min.css.map");
@import "~materialize-css";
*:focus {
  outline: none;
}
html,
body,
div,
```

span,  
applet,  
object,  
iframe,  
h1,  
h2,  
h3,  
h4,  
h5,  
h6,  
p,  
blockquote,  
pre,  
a,  
abbr,  
acronym,  
address,  
big,  
cite,  
code,  
del,  
dfn,  
em,  
img,  
ins,  
kbd,  
q,  
s,  
samp,  
small,  
strike,  
strong,  
sub,  
sup,  
tt,  
var,  
b,  
u,  
i,  
center,  
dl,  
dt,  
dd,  
ol,  
ul,  
li,  
fieldset,  
form,  
label,



```
legend,  
table,  
caption,  
tbody,  
tfoot,  
thead,  
tr,  
th,  
td,  
article,  
aside,  
canvas,  
details,  
embed,  
figure,  
figcaption,  
footer,  
header,  
hgroup,  
menu,  
nav,  
output,  
ruby,  
section,  
summary,  
time,  
mark,  
audio,  
video {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    vertical-align: baseline;  
}  
  
/* HTML5 display-role reset for older browsers */  
article,  
aside,  
details,  
figcaption,  
figure,  
footer,  
header,  
hgroup,  
menu,  
nav,  
section {  
    display: block;
```

```
}  
  
body {  
  line-height: 1;  
}  
  
ul {  
  list-style: none;  
}  
  
blockquote,  
q {  
  quotes: none;  
}  
  
blockquote:before,  
blockquote:after,  
q:before,  
q:after {  
  content: "";  
}  
  
button:focus {  
  outline: none;  
}  
  
table {  
  border-collapse: collapse;  
  border-spacing: 0;  
}  
  
select {  
  -webkit-appearance: none;  
  -moz-appearance: none;  
  -webkit-user-select: none;  
  -moz-user-select: none;  
  -ms-user-select: none;  
  user-select: none;  
}  
  
select::-ms-expand {  
  display: none;  
}  
  
img {  
  max-width: 100%;  
  height: auto;  
}
```

```

iframe {
  width: 100%;
}
a {
  &:hover {
    text-decoration: none;
  }
}
body {
  font-family: "Comfortaa", cursive;
}
.back-top {
  width: 45px;
  height: 45px;
  background-color: #1e1e1e;
  right: 40px;
  bottom: 70px;
  z-index: 9;
  display: none;
  font-size: 24px;
  color: #fefefe;
  position: fixed;
  text-align: center;
  line-height: 45px;
  cursor: pointer;
  &.show {
    display: block !important;
  }
}
</style>
<script>
import Header from "./components/Header"
import Footer from "./components/Footer"
import ConfirmPopup from "./components/Cart/ConfirmPopup"
import "materialize-css"
import "vue-loading-overlay/dist/vue-loading.css"

export default {
  components: { ConfirmPopup, Footer, Header },
}
</script>

<template>
  <div
    class="item"
    :class="{
      list: listView,
      grid: !listView,
      carousel: isCarousel,

```

```

        editPage: isEditPage,
    }"
>
<div class="product">
  <div class="images">
    <span>
      
    </span>
    <div class="button-group">
      <button
        @click="
          isEditPage
            ? $emit('selectProduct', productData._id)
            : $emit('addToCart', { productData, quantity: 1 })
          "
        class="custom-btn"
        :class="{ pink: !isEditPage, green: isEditPage }"
      >
        <i
          class="fa"
          :class="{
            'fa-shopping-bag': !isEditPage,
            'fa-pencil': isEditPage,
          }"
          aria-hidden="true"
        >></i>
      </button>
      <button
        @click="
          isEditPage
            ? $emit('deleteProduct', productData._id)
            : $emit('selectProduct', productData._id)
          "
        class="custom-btn"
        :class="{ pink: !isEditPage, 'red darken-2': isEditPage }"
      >
        <i
          class="fa fa-search"
          :class="{ 'fa-search': !isEditPage, 'fa-trash-o': isEditPage }"
          aria-hidden="true"
        >></i>
      </button>
    </div>
  </div>
  <div class="info-product">
    <a href="#" class="title">{{ productData.title }}</a>
    <span class="price">{{ productData.price + " грн" }}</span>
  </div>
</div>

```

```

    </div>
</template>

```

```

<script>
export default {
  name: "Product-item",
  props: {
    productData: {
      type: Object,
    },
    listView: {
      type: Boolean,
      default: null,
    },
    isCarousel: {
      type: Boolean,
      default: false,
    },
    isEditPage: {
      type: Boolean,
      default: false,
    },
  },
  computed: {
    decodedImage: function () {
      return (new Image()).src = `data:image/jpg;base64,${new Buffer(
        this.productData.image.data,
        "base64"
      ).toString("base64")}`;
    },
  },
}
</script>

```

```

<style scoped lang="scss">
.item {
  &.editPage {
    .product {
      width: 150px !important;
      min-height: 100%;
      margin-bottom: 0;
      &:hover {
        .button-group {
          opacity: 1;
          visibility: visible;
          display: flex;
          align-items: center;
          justify-content: center;
          button,

```

```
    a {
      display: flex;
      justify-content: center;
      align-items: center;
      width: 35px;
      height: 35px;
      line-height: normal;
      margin-bottom: 0;
      letter-spacing: normal;
    }
  }
}
.info-product {
  .title {
    font-size: 20px;
  }
  .price {
    font-size: 16px;
  }
}
.images {
  img {
    min-height: 100%;
    max-height: 100%;
  }
}
}
}
&.carousel {
  .product {
    width: 259px !important;
  }
}
&.grid {
  .product {
    width: 100%;
    img {
      width: 100%;
    }
  }
}
}
&.list {
  .product {
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 0;
    .title {
```

```

        display: block;
        font-size: 30px;
    }
    &:hover {
        .images {
            box-shadow: none;
            img {
                opacity: 0.5;
            }
        }
        .button-group {
            opacity: 1;
            visibility: visible;
        }
    }
    .images {
        width: 40%;
        height: 320px;
        background-color: transparent;
    }
    .info-product {
        width: 60%;
        padding: 0 10px 0 70px;
        text-align: left;
        left: auto;
        -ms-transform: translateX(0);
        transform: translateX(0);
    }
    }
    }
}
.product {
    position: relative;
    margin-bottom: 80px;
    width: 259px;
    min-height: 378px;
    &:hover {
        cursor: pointer;
        .images {
            box-shadow: 0 13px 19px 1px rgba(119, 119, 119, 0.24);
            img {
                opacity: 0.5;
            }
        }
        .button-group {
            opacity: 1;
            visibility: visible;
        }
    }
}

```

```
.info-product {
  display: flex;
  flex-direction: column;
  text-align: center;
  .title {
    font-size: 24px;
    color: #1e1e1e;
    margin-bottom: 14px;
  }
  .price {
    font-size: 20px;
    color: #878787;
  }
}

.images {
  background-color: #f3f7f6;
  position: relative;
  display: -ms-flexbox;
  display: flex;
  margin-bottom: 20px;
  -ms-flex-align: center;
  align-items: center;
  -ms-flex-pack: center;
  justify-content: center;
  text-align: center;
  transition: 0.2s linear;
  max-height: 300px;
  a {
    max-height: 300px;
    transition: 0.2s linear;
  }
  img {
    transition: 0.2s linear;
    min-height: 300px;
    max-height: 300px;
    width: auto;
    height: 100%;
    margin: 0 auto;
  }
}

.button-group {
  top: 50%;
  left: 50%;
  width: 100%;
  -ms-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
  position: absolute;
  opacity: 0;
}
```



```

    visibility: hidden;
    transition: 0.2s linear;
    button {
      cursor: pointer;
    }
  }
}
.custom-btn {
  font-size: 14px;
  letter-spacing: 3px;
  text-transform: uppercase;
  display: inline-block;
  transition: all 0.3s ease;
  border: none;
  text-align: center;
  color: #fff;
  background-color: #e20b50;
  border-radius: 40px;
  margin-bottom: 72px;
  width: 70px;
  height: 70px;
  padding: 0;
  line-height: 70px;
  &:not(:last-child) {
    margin-right: 10px;
  }
  &.pink {
    color: #fff;
    background-color: #e20b50;
  }
}
}
</style>

<template>
  
</template>

<script>
export default {
  name: "TopCarousel",
  data() {
    return {}
  },
  props: {
    slide: {
      type: Object,
      required: true,
    },
  },
  computed: {

```

```

    decodedImage: function () {
      return (new Image().src = `data:image/jpg;base64,${new Buffer(
        this.slide.image.data,
        "base64"
      ).toString("base64")}`)
    },
  },
}
}
</script>

<style scoped lang="scss">
img {
  max-width: 60%;
}
</style>

<template>
  <tr>
    <td class="product-thumbnail">
      
    </td>
    <td class="product-name">{{ item.title }}</td>
    <td class="product-price text-center">
      <span class="price">{{ item.price }} грн</span>
    </td>
    <td class="product-quantity text-center">
      <div class="quantity">
        <span
          style="outline: none;"
          :style="{visibility: quantity !== 1 ? 'visible' : 'hidden'}"
          @click="
            quantity > 1 && quantity !== 0 ? quantity-- : false, updatePrice()
          "
          class="minus"
        >
          <i class="fa fa-minus" aria-hidden="true"></i>
        </span>
        <input v-model="productQuantity" />
        <span
          @click="quantity > 0 ? quantity++ : false, updatePrice()"
          class="plus"
        >
          <i class="fa fa-plus" aria-hidden="true"></i>
        </span>
      </div>
    </td>
    <td class="product-price text-center" data-title="Total">
      <span class="price">{{ totalPrice }} грн</span>
    </td>
  </tr>
</template>

```

```

<td class="product-remove text-right">
  
</td>
</tr>
</template>

<script>
export default {
  name: "TableRow",
  data() {
    return {
      quantity: 1,
    }
  },
  props: {
    item: Object,
  },
  computed: {
    productQuantity: function() {
      return this.quantity = this.item.quantity
    },
    totalPrice: function () {
      return this.item.price * this.quantity
    },
    decodedImage: function () {
      return (new Image().src = `data:image/jpg;base64,${new Buffer(
        this.item.image.data,
        "base64"
      ).toString("base64")}`)
    },
  },
  created() {
    if (this.item.quantity) {
      this.quantity = this.item.quantity
    }
  },
  methods: {
    async updatePrice() {
      if (this.quantity > 0) {
        await this.$store.dispatch('updateProductQuantity', {
          product: this.item,
          quantity: this.quantity,
          productTotalPrice: this.totalPrice
        })
        await this.$store.dispatch('updateTotalPrice')
      }
    }
  }
}

```

```

    }
  },
},
}
</script>

<style scoped lang="scss">
.cart {
  margin-top: 130px;

  &-top {
    display: flex;
    align-items: center;
    justify-content: space-between;
    margin-bottom: 55px;
    h3 {
      font-size: 65px;
    }
    span {
      padding-top: 22px;
      display: inline-block;

      a {
        font-size: 12px;
        color: #1e1e1e;
        letter-spacing: 3px;
        text-transform: uppercase;

        img {
          margin-right: 5px;
          vertical-align: bottom;
        }
      }
    }
  }
}

.bottom-table {
  margin-top: 35px;
  display: flex;
  justify-content: space-between;

  .custom-btn {
    padding: 28px 50px;
    font-size: 14px;
    letter-spacing: 3px;
    text-transform: uppercase;
    display: inline-block;
    transition: all 0.3s ease;
    border: none;
  }
}

```

```
text-align: center;
color: #fff;
background-color: #e20b50;
border-radius: 40px;

&:hover {
  background-color: lighten(#e20b50, 10);
}

&.outline {
  background-color: #ffffff;
  color: #e20b50;
  border: 2px solid #e20b50;

  &:hover {
    color: #ffffff;
    border: 2px solid transparent;
    background-color: #e20b50;
  }
}
}
}

.table-product {
  width: 200px;
}

.product-name,
.product-price,
.product-remove {
  padding-top: 57px;
}

.product-quantity {
  width: 113px;
}

.product-remove {
  display: flex;
  justify-content: center;
}

.product-name {
  max-width: 196px;
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}
```

```
.product-remove {
  img {
    cursor: pointer;
  }
}

table {
  border-collapse: collapse;
  border-spacing: 0;
  width: 100%;

  .quantity {
    padding: 10px 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
    border: 2px solid #777;

    i {
      color: transparentize(#000, 0.3);
      vertical-align: middle;
    }

    .minus {
      cursor: pointer;
      display: inline-block;
      padding: 0 0 0 11px;
    }

    .plus {
      cursor: pointer;
      display: inline-block;
      padding: 0 11px 0 0;
    }

    input {
      font-weight: bold;
      width: 42px;
      font-size: 16px;
      color: #1e1e1e;
      text-align: center;
      border: none;
      background-color: transparent;
      padding: 0 10px;
      outline: none;
      user-select: none;
      height: 15px;

      &:focus {
```

```

        outline: none;
    }

    &:active {
        outline: none;
    }
}

tr {
    border-bottom: 1px solid #f3f7f6;
}

td {
    font-size: 18px;
    color: #1e1e1e;
    vertical-align: top;
    text-align: left;
    padding: 40px 0;

    &.text-center {
        text-align: center;
    }
}

thead {
    background-color: #f3f7f6;

    td {
        padding: 20px 15px 21px 19px;
    }
}

tbody {
    td {
        font-size: 24px;
    }
}
}
</style>

<template>
<div class="popup-wrapper">
<div class="popup-container">
<div class="popup-top">
<div class="title">
<h3 v-if="confirmOrderData.created">
    {{ `Замовлення № ${confirmOrderData.orderNumber} підтверджено` }}

```

```

    </h3>
    <h3 v-else>Оформлення замовлення</h3>
  </div>
</div>
<div class="popup-content">
  <form v-if="!confirmOrderData.created" @submit.prevent="submitOrder">
    <div class="input-group">
      <label for="name">Введіть ваше ім'я</label>
      <input
        v-model="formData.name"
        id="name"
        type="text"
        minlength="3"
        required
        placeholder="Ім'я"
      />
    </div>
    <div class="input-group">
      <label for="email">Введіть Ваш email</label>
      <input
        v-model="formData.email"
        id="email"
        type="email"
        required
        placeholder="Email"
      />
    </div>
    <div class="input-group">
      <label for="email">Введіть Ваш номер телефону</label>
      <input
        v-model="formData.phone"
        v-mask="' +(380) ### ### ###'"
        id="phone"
        type="text"
        required
        placeholder="Phone"
      />
    </div>
    <div class="bottom-buttons">
      <button @click="cancel" class="btn">Відміна</button>
      <button type="submit" class="btn">Оформити замовлення</button>
    </div>
  </form>
  <div v-else class="confirmed">
    <h4>Дякуємо за замовлення</h4>
    <span>З вами зв'яжуться найближчим часом</span>
    <div class="buttons">
      <button @click="cancel" class="btn">Закрити вікно</button>
      <button

```



```

        @click="
            cancel(), $router.push({ name: 'store', query: { page: 1 } })
        "
        class="btn"
    >
        Продовжити покупки
    </button>
</div>
</div>
</div>
</div>
</div>
</div>
</template>

<script>
export default {
  name: "ConfirmPopup",
  props: {
    confirmOrderData: {
      type: Object,
    },
  },
  data() {
    return {
      formData: {
        name: "",
        email: "",
        phone: "",
      },
    }
  },
  methods: {
    cancel() {
      this.formData.name = this.formData.email = this.formData.phone = ""
      this.$emit("closePopup")
    },
    async submitOrder() {
      if (
        this.formData.name.trim() &&
        this.formData.email.trim() &&
        this.formData.phone.trim()
      ) {
        await this.$store.dispatch('updateTotalPrice')
        await this.$store.dispatch("createOrder", {
          name: this.formData.name,
          email: this.formData.email,
          phone: this.formData.phone,
        })
      }
    }
  }
}

```

```

    },
  },
}
</script>

```

```

<style scoped lang="scss">
.popup-wrapper {
  position: fixed;
  z-index: 99999;
  top: 0;
  left: 0;
  height: 100vh;
  width: 100vw;
  background: transparentize(#000, 0.3);
  display: flex;
  align-items: center;
  justify-content: center;
}
.popup-container {
  display: flex;
  flex-direction: column;
  /*justify-content: space-between;*/
  width: 400px;
  /*height: 300px;*/
  background: #fff;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.5);
  border-radius: 8px;
  padding: 24px 0 10px 0;
  z-index: 9999;
}
.bottom-buttons {
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 15px 0 0 0;
}
.popup-top {
  display: flex;
  justify-content: center;
  border-bottom: 1px solid rgba(0, 0, 0, 0.3);
  /*padding: 0px 24px;*/
  padding-bottom: 10px;
}
.popup-content {
  padding: 15px 24px 15px;
  /*border-bottom: 1px solid rgba(0, 0, 0, 0.3);*/
  .input-group {
    margin-bottom: 1rem;
    &:last-child {

```

```

        margin-bottom: 0;
    }
}
input {
    margin-top: 0.5rem;
    border-radius: 4px;
    border: 1px solid rgba(0, 0, 0, 0.3);
    padding: 10px;
    width: 100%;
}
}
.confirmed {
    text-align: center;
    h4 {
        text-align: center;
        font-weight: bold;
        margin: 20px;
    }
    span {
        display: inline-flex;
        margin-bottom: 20px;
    }
    .buttons {
        display: flex;
        align-items: center;
        justify-content: space-between;
    }
    button {
        font-size: 13px;
    }
}
</style>

<template>
<div class="card">
    <div class="card-top">
        <div class="order-number">Замовлення № {{ order.orderNumber }}</div>
        <div class="creation-date">Дата створення: {{ creationDate }}</div>
    </div>
    <div class="card-main">
        <div class="person-info">
            <span class="name">Ім'я: {{ order.userName }}</span>
            <span class="email">Email: {{ order.userEmail }}</span>
            <span class="phone">Телефон: {{ order.userPhone }}</span>
        </div>
        <ul>
            <li v-for="product of order.products">
                <div class="id">
                    ID продукту:

```

```

    <router-link
      target="_blank"
      :to="{
        path: `/main-page/product/${product.id}`,
        query: {
          isOrderLink: true,
        },
      }"
    >
      {{ product.id }}
    </router-link>
  </div>
  <div class="quantity">Кількість: {{ product.quantity }}</div>
</li>
</ul>
<div class="total">
  <span>Загальна вартість: </span>
  <span>{{ order.totalPrice }} грн</span>
</div>
<div class="status">
  <span>Статус замовлення: </span>
  <span>{{ orderStatus }}</span>
</div>
</div>
<div class="bottom-buttons">
  <button
    @click="$emit('updateOrderStatus', { newStatus: 'confirmed', order })"
    v-if="order.type === 'new'"
    class="btn"
  >
    Підтвердити
  </button>
  <button
    @click="$emit('updateOrderStatus', { newStatus: 'done', order })"
    v-if="order.type === 'confirmed'"
    class="btn"
  >
    Виконано
  </button>
  <button
    v-if="order.type !== 'rejected' && order.type !== 'done'"
    @click="$emit('updateOrderStatus', { newStatus: 'rejected', order })"
    class="btn red darken-1"
  >
    Відкликати
  </button>
</div>
</div>
</template>

```

```

<script>
export default {
  name: "OrderCard",
  props: {
    order: {
      type: Object,
    },
  },
  computed: {
    orderStatus: function () {
      switch (this.order.type) {
        case "all":
          return
        case "new":
          return "Нове замовлення"
        case 'rejected':
          return 'Відкликано'
        case 'done':
          return 'Виконано'
        case 'confirmed':
          return 'Підтверджено'
      }
    },
    creationDate: function () {
      return new Date(this.order.creationDate).toLocaleDateString("ukr")
    },
  },
}
</script>

```

```

<style scoped lang="scss">
.card {
  padding: 10px 0 10px 10px;
  &-top {
    display: flex;
    justify-content: space-between;
    align-items: center;
  }
  &-main {
    display: flex;
    flex-direction: column;
    justify-content: center;
    .person-info {
      display: flex;
      flex-direction: column;
      justify-content: center;
      span {
        display: flex;

```



```

    justify-content: space-between;
    margin-top: 10px;
  }
  .bottom-buttons {
    display: flex;
    justify-content: space-between;
    align-items: flex-end;
    flex: 1;
    padding-top: 10px;
    padding-right: 10px;
  }
</style>
import Vue from "vue"
import VueRouter from "vue-router"

Vue.use(VueRouter)

const routes = [
  {
    path: "/",
    redirect: "/main-page/home",
  },
  {
    path: "/main-page",
    component: () => import("../views/Main"),
    children: [
      {
        path: "home",
        component: () => import("../views/Main-Page/Home"),
      },
      {
        path: "cart",
        name: "cart",
        component: () => import("../views/Main-Page/Cart"),
      },
      {
        path: "store",
        name: "store",
        component: () => import("../views/Main-Page/Store"),
      },
      {
        path: "product/:id",
        name: "single-product",
        component: () => import("../views/Main-Page/SingleItem"),
      },
      {
        path: 'constructor',
        name: 'constructor',
        component: () => import('../views/Main-Page/Constructor')
      }
    ]
  }
]

```

```

    }
  ],
},
{
  path: "/admin",
  name: "admin",
  component: () => import("../views/AdminPage"),
  redirect: "/admin/products",
  children: [
    {
      path: "products",
      name: "products",
      component: () => import("../views/Admin/Products"),
      redirect: "/admin/products/add-product",
      children: [
        {
          path: "add-product",
          name: "add-product",
          component: () =>
            import("../components/Admin/Products/ProductCardEdit"),
        },
        {
          path: "products-list",
          name: "products-list",
          component: () =>
            import("../components/Admin/Products/AllProductsList"),
        },
        {
          path: "edit/:id",
          name: "product-edit",
          component: () =>
            import("../components/Admin/Products/ProductCardEdit"),
        },
      ],
    },
  ],
},
{
  path: "carousel",
  name: "carousel",
  component: () => import("../views/Admin/CarouselPhotos"),
  redirect: "/admin/carousel/add-photo",
  children: [
    {
      path: "add-photo",
      name: "add-photo",
      component: () =>
        import("../components/Admin/Carousel/Main-carousel-images"),
    },
    {

```



```
        path: "photos-list",
        name: "photos-list",
        component: () => import("../components/Admin/Carousel/PhotosList"),
      },
    ],
  },
  {
    path: "orders",
    name: "orders",
    component: () => import("../views/Admin/Orders"),
  },
  {
    path: "preview",
    name: "created-product-preview",
    component: () => import("../components/Admin/Products/ProductPreview"),
  },
  {
    path: "upload-carousel-images",
    name: "carousel-images",
    component: () =>
      import("../components/Admin/Carousel/Main-carousel-images"),
  },
],
},
]
```