ІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,

управління та адміністрування

Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: <u>Розробка веб-сайту для продажу одягу з реалізацією функції</u> розрахунку вартості замовлення

Виконала студентка групи <u>КН-5</u> <u>спеціальності 122 «Комп'ютерні науки»</u> <u>Теліженко Анастасія Олегівна</u>

Керівник <u>к.геогр.н., доцент</u> <u>Терещенко Тетяна Михайлівна</u>

Консультант

Рецензент <u>к.техн.н., доцент</u> Попов В.Л.

Одеса 2022

3MICT

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ	X
СИСТЕМ	7
1.1 Аналіз предметної області	7
1.2 Аналіз існуючих програмних систем	8
2 ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ	11
2.1 Функціональні та нефункціональні вимоги	11
2.2 Обґрунтування вибору операційної системи	11
2.3 Вибір середовища розробки	13
3.1 Реалізація АРІ	19
3.2 Реалізація GUI	20
3.3 Реалізація сервера	21
4 ПРОЕКТУВАННЯ WEB-ДОДАТКУ	26
4.1 Створення діаграми відношення сутностей	26
4.2 Створення діаграми варіантів використання системи	27
4.3 Створення діаграми діяльності	29
4.4 Створення діаграми послідовності	31
4.5 Створення макетів інтерфейсу	31
5 РЕАЛІЗАЦІЯ ІНТЕРНЕТ МАГАЗИНУ «DRAW STUDIO»	
5.1 середовище розробки	
5.2 Розробка серверної частини	41
ВИСНОВКИ	50
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	51
ДОДАТОК А	52

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- БД база даних
- ОС операційна система
- ПЗ програмне забезпечення
- API Application Programming Interface програмний інтерфейс

програми

IDE – Integrated Development Environment – інтегроване середовище розробки

- CSS Cascading Style Sheets Каскадні таблиці стилів
- JS Java Script
- Web World Wide Web Всесвітня мережа

ВСТУП

Для зручного пошуку інформації служать Web-сайти.

Веб-сайт – це інструмент, за допомогою якого можна залучити потенційних клієнтів і спростити початковий процес замовлення ваших послуг.

Потенційні клієнти за допомогою каталогів і пошукових систем за ключовими словами знайдуть сайт, ознайомляться з послугами та умовами і можуть захотіти стати вашими клієнтами.

Відвідувачам буде нескладно заповнити на сайті спеціальну форму-заявку і вказати в ній свої реквізити і побажання щодо послуги. Форма-заявка автоматично перешле по електронній пошті відповідальному менеджеру для обробки, уточнення інформації та її виконання.

На даний час із-за ситуації що відбувається в нашій країні, важко виконувати покупку виходячи в магазин та знайти ту річ яка дійсно буде подобатися. В цьому магазині буде колекція речей з патріотичними малюнками та надписами, а також малюнки на любий смак. Покупець може замовити власний дизайн ескізу або обрати вже готовий в магазині. Також можлива така функція коли замовник може відіслати нам свою річ щоб оновити її за допомогою малюнку.

Метою кваліфікаційної роботи є розробка інтернет-магазину для продажу кастомізованого одягу «Nasty Space».

Даний сайт повинен оптимально швидко, зручно надавати інформацію для користувача сайту. Інформація, посилання повинні бути представлені в зручному для користувача вигляді.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ

1.1 Аналіз предметної області

Кастомізація (ручний розпис одягу) дає можливість кожній людині відчути себе в ролі дизайнера та самостійно створити власну не повторну річ.

За допомогою кастомізації старі джинси(нові), футболки та інші речі знайдуть нове життя.

Кастом або кастомізація — це зміна зовнішнього вигляду за допомогою малюнків та надписів на будь якому одягу (адаптації на ваш смак). Змінити можна все що завгодно: одяг, взуття, аксесуари(сумки, рюкзаки). Унікальну річ можна отримати, якщо кастомізувати її самостійно або ж звернутися до людей, які займаються цим професійно, — до кастомайзерів.

Останнім часом кастомізація одягу набирає значної популярності. Хоча кастомізація зародилася ще у 70-х роках минулого століття в якості одного зі способів продемонструвати власний стиль. Молодь із Нью-Йорка, а саме представники хіп-хоп сфери, почали переробляти джинсові куртки під себе. Оскільки одяг у той час виготовляли великими тиражами. Паралельно з цим представники інших субкультур намагалися створити свій індивідуальний стиль. Наприклад, хіпі вирішили кастомізувати речі бусинами, вишивками й символами. Історія кастомізації одягу простежується і в британських панків, які теж підхопили цю ідею. В той час у представників кожного британського рок чи панк-гурту був свій стиль в одязі. Хтось обмежувався рваними елементами на футболках та джинсах, деякі доповнювали образ незвичними зачісками, а інші — кардинально змінювали гардероб і носили провокаційний макіяж. Створюючи власний зухвалий стиль.

1.2 Аналіз існуючих програмних систем

На даний час створено не так багато різноманітних сайтів, що надають можливість замовити малюнок з будь якої точки країни(міста) не виходячи з дому. Розглянемо на прикладі:

Сайт ЦЕХ (рис.1.1) це сайт для розпису одягу, та послуги хімчистки



Рисунок 1.1 – Інтерфейс сайту ЦЕХ

Основний недолік сайту – немає прикладів робіт(галереї), дизайн виконаний хаотично, навігація магазину не зручна.

Інтернет магазин Crafta – (рис.1.2) використовується для розміщення своїх робіт.



Рисунок 1.2 – Інтерфейс інтернет магазину Crafta

Застосунок Crafta надає можливіть художникам початківцям: розміщувати свої роботи. А покупцям легко орієнкуватися в ціні, опису, та загальної інформації товару. Сам магазин має дуже велику кількість товарів які можна обрати на свій смак, та знайти кастомізований одяг різних людей. Недоліками Smartify (рис.1.3) є те, що товар не придлягає поверненю, не вказана інформація про спосіб доставки та оплати

Сайт "Огонь" (рис.1.4) має різновидний розпис по предмтам та одягу.



Рисунок 1.3 – Інтерфейс сайту Огонь

недоліки сайту – навігація магазину не зручна, відсутня форма оплати, багато категорій розташовано хаотично, немає сортування по категоріям, відсутні описи товару.

Таким чином, під час виконання першого розділу бакалаврської роботи була обрана предметна область, в рамках якої розглянуті програмні аналоги і їх особливості. Переваги та недоліки розглянутих інтернет магазинів були взяті до уваги при розробці програмного продукту.

2 ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ

2.1 Функціональні та нефункціональні вимоги

Перед початком розробки застосунку були визначені функціональні та нефункціональні вимоги до нього.

До функціональних вимог належать:

- наявність БД товарів і клієнтів;
- можливість здійснити замовлення через форму.
- До нефункціональних вимог належать:
- браузер;
- підключення до Інтернету.

2.2 Обґрунтування вибору операційної системи

Web – додаток має багато переваг у порівнянні зі звичайним додатком, а саме:

- Не потребує установки. Для використання необхідно мати прилад з встановленим браузер і доступом в Інтернет.
- У разі оновлення додатка, його роблять лише на сервері, після чого усі користувачі мають доступ до оновленої версії.
- Web додаток дуже універсальний та практичний для користувача. Достатньо лише встановити додаток на сервер, з будь-якою ОС і клієнти зможуть користуватися з любого комп'ютера чи мобільного приладу, з різними ОС. А також з будь якого браузера: Mozilla Firefox, Opera, Google Chrome, Internet Explorer aбo Safari.
- Спрощується доступ до даних, завдяки тому що вони зберігаються в однієї БД.

Основними недоліками є:

• Обмежений доступ до функцій платформи, але у випадку інтернет магазина в них нема потреби.

• Базова продуктивність, але її достатньо для онлайн магазину.

Windows Server

Перевагами цієї операційної системи - це практичність, продуктивність та наявність широких можливостей. Через свою надійність Windows Server ідеально підходить для терміналів та файлових серверів.

В цієї ОС є такі недоліки:

- Не всі версії 32-розрядну архітектуру.
- Windows потребує набагато більших ресурсів, ніж будь-який інший аналог.
- Ця ОС є ліцензованим програмним забезпеченням.

Linux Server

Серверні ОС Linux мають велику популярність. Серед найбільш популярних ОС (що представляють це сімейство) можна виділити CentOS, Debian i Ubuntu.

CentOS – це безкоштовна операційна система для сервера зі швидким консольним менеджером Yellowdog Updater, Modified та більшістю популярних хостингових панелей. Головні переваги CentOS є те що можна відзначити велику кількість форумів і спільнот, що забезпечують всебічну підтримку користувачів цієї ОС. Недоліками можна вважати відсутність програм для роботи з репозиторіями.

Debian - це повністю універсальна оболонка, яка може використовуватися для роботи не тільки сервера, але і звичайного ПК. Debian вважається потужною та добре захищеною ОС. Головним її недоліком є відсутність регулярних оновлень.

Ubuntu відрізняється легким встановленням та налаштуванням, з якими впораються навіть новачки. Автоматичний інтерфейс Ubuntu значно полегшує

роботу з цією системою. Вибір Ubuntu буде актуальним за наявності скромного бюджету, а також відносно невеликого навантаження на сервер. Підсумовуючи, варто сказати, що універсальної ОС для серверів немає. Кожна має переваги та недоліки, які зазвичай пізнаються практично. Отже враховуючи особливості кожної ОС, для сервера інтернет-магазину обрала Ubuntu, тому що вона саме більше підходить для реалізації інтернет магазину.

2.3 Вибір середовища розробки

Інтегровані середовища розробки (IDE) та редактори коду (CE) – це програми, що використовуються для написання та редагування коду. Технічно можливо писати код у текстовому редакторі, але IDE та CE пропонують багато додаткових функцій, які призначені для спрощення процесу кодування.

IDE мають більше функцій, ніж редактори коду, але у деяких редакторах коду є можливість налаштування, подібно до IDE. Редактори коду часто мають такі функції, як виділення синтаксису, авто заповнення та зіставлення дужок. В свою чергу IDE об'єднують кілька інструментів розробника в єдиному графічному інтерфейсі користувача. Як мінімум, ці інструменти зазвичай складаються з редактора коду, компілятора чи інтерпретатора та відладчика.

У кожного програміста є індивідуальні переваги щодо IDE та редакторів коду. Тому можна знайти те що більше зручно для вас.

Найважливішими критеріями, які слід враховувати при оцінці ваших варіантів є:

• Вартість: є безкоштовні, і платні програми.

• Час навчання: потрібен час, щоб адаптуватися до нового інтерфейсу та запам'ятати поєднання клавіш.

• Функціональність і можливість налаштувань: програми мають різні функції, тому варто знати, які вам необхідні. Деякі популярні функції – це багатомовна підтримка, авто заповнення та інтеграція з Git та GitHub.

• Швидкість: потрібно враховувати як репутацію програми щодо швидкості, так і потужність вашого комп'ютера або пристрою для розміщення програми.

• Вимоги до пристрою: характеристики пристрою можуть вплинути на продуктивність програми.

• Сумісність з ОС: деякі IDE та редактори коду не завжди пропонують функціональність для усіх платформ. Під час перегляду варіантів пам'ятайте про сумісність з вашою ОС.

• Підтримка користувачів: програма відомої компанії-розробника програмного забезпечення може мати більш надійну підтримку користувачів, ніж менш популярна програма.

Rank	Change	IDE	Share	Trend
1	1	Visual Studio	25.31 %	+3.6 %
2	1	Eclipse	16.31 %	-1.1 %
3	44	Android Studio	11.39 %	-11.3 %
4	1	Visual Studio Code	8.74 %	+3.4 %
5	1	pyCharm	7.73 %	+2.5 %
6	1	IntelliJ	6.13 %	+1.3 %
7	444	NetBeans	5.3 %	-0.2 %

Рисунок 2.2 – Найпопулярніші IDE для мови JavaScript Розглянемо деякі з них, для подальшого порівняння особливостей: 1) Visual Studio Code Платформи: Windows, Linux, macOS Ціна: безкоштовно

Visual Studio Code — це безкоштовна HTML IDE від Microsoft. Цей редактор підтримує Typescript, C++, Java, Javascript, PHP, Python та інші. Visual Studio Code доступний не тільки для Windows, а також для macOS і навіть для Linux. Це дозволяє працювати на різних платформах з однаковим комфортом і функціональністю.

VSCode можна легко функціонально розширити за допомогою розширень. VS Code працює безпосередньо з Github і пропонує надзвичайно гарне підсвічування синтаксису для різних мов і розширений варіант завершення коду.

Плюси:

• Це відкритий код.

• Має багато функцій, таких як провідник рішень, контроль джерел, налагоджувача, область розширень.

- Підтримує термінал всередині вікна.
- Добре для розробки ядра .Net.

Мінуси:

- Треба звикнути до мінімалістичного інтерфейсу.
- Функціональність може бути недостатня для великих проектів.

2) Eclipse

Платформи: Windows / Linux / macOS / Solaris

Ціна: безкоштовно

Це безкоштовне середовище розробки з відкритим кодом, яке добре підходить як для початківців, так і для досвідчених розробників. Eclipse IDE працює з Java, C, C ++, Fortran, Javascript, PHP, Python, Ruby та іншими.

Окрім інструментів налагодження та підтримки систем Git/CVS, Eclipse поставляється з Java та інструментом плагінів. Функціональність Eclipse не така велика, як IntelliJ IDEA, але це середовище розробки поширюється з відкритим вихідним кодом.

Плюси:

- Можливість програмувати багатьма мовами.
- Значна гнучкість середовища завдяки модульності.
- Можливість інтеграції JUnit.
- Віддалене налагодження (при використанні JVM).
- Доповнення коду пропонується з мовного сервера.
- Ви можете перейти до оголошення змінних, класів і методів.

• Проекти, створені за допомогою MS test і xUnit, можуть запускатися безпосередньо в IDE.

Мінуси:

• Новачкам може бути важко зрозуміти різноманітність можливостей.

• Плагін керується спільнотою, тому немає гарантії, що він завжди працюватиме добре.

4) WebStorm

Платформи: Windows, Linux, macOS

Ціна: \$0-129/рік

WebStorm виділяється тим, що це повнофункціональна IDE JavaScript. Розробником цієї IDE є JetBrains.

Цей редактор JS чудово підходить для різних платформ, таких як React, Angular, Vue.js тощо. Його можна використовувати для налагодження скриптів вузлів і виконання тестів на вбудованому сервері.

За замовчуванням WebStorm налаштовано на автоматичне збереження файлів під час роботи з ними. Коли ви переходите на інший редактор Javascript без такої функціональності, ви починаєте відчувати, наскільки йому не вистачає після цієї IDE.

Плюси:

- Зручне авто заповнення для HTML, CSS i JavaScript.
- Перевірка помилок і просте налагодження коду.
- Вбудована інтеграція з системами контролю джерел, такими як GitHub.

- Гнучкість налаштувань.
- Досить велика кількість плагінів.
- Багато функцій.
- Немає необхідності шукати пакети та налаштовувати їх.
- Чудовий інструмент злиття.

• Розумне завершення коду, перевірка помилок на льоту, швидка навігація по коду та рефакторинг для JavaScript.

Мінуси:

• Завдяки широкій функціональності WebStorm є важким і ресурсомістким;

- Іноді при роботі над дуже великими проектами це з'їдає пам'ять;
- Притаманна всім IDE, вона повільна та ресурсомістка.
- Відносно складні налаштування.
- Платна IDE, що розповсюджується за підпискою.
- Повільна робота з великою кількістю проектів.

Для розробки диплому обрала середовище Visual Studio Code.

З ВИКОРИСТОВАНІ ТЕХНОЛОГІЇ

Веб-додаток — це прикладне програмне забезпечення, яке працює у веббраузері, на відміну від програм, які запускаються локально та в оригінальній операційній системі (ОС) пристрою. Веб-програми постачаються у всесвітній мережі користувачам із активним мережевим підключенням.

Архітектура або модель клієнт-сервер — це мережа прикладних програм, що розділяє завдання між клієнтами та серверами, які живуть в одній системі або мають спілкуватися через комп'ютерну мережу. Сервер-клієнт в основному покладається на відправку запиту до іншої програми для доступу до служби, доступної сервером. На цьому сервері працює кілька програм, які спільно використовують ресурси та розподіляють роботу між клієнтами.



Рисунок 3.1 – Архітектура клієнт-серверного додатку

Відношення клієнт-сервер відповідає шаблону запит-відповідь і повинно дотримуватися загальної процедури зв'язку, яка визначає мову, правила або шаблони діалогів, що використовуються. Зв'язок клієнт-сервер відповідає набору протоколів TCP або IP.

3.1 Реалізація АРІ

REST API — це API, який дотримується набору правил для взаємодії програми та служб один з одним. Оскільки він обмежений архітектурою REST, REST API називається RESTful API. API REST забезпечують гнучкий спосіб доступу до веб-сервісів без значних можливостей обробки.



Рисунок 3.2 – Модель REST API

Характеристики добре розробленого API:

– Гнучкість: REST API є гнучким з кількома типами викликів, як-от повернення різних форматів даних та структурні зміни за допомогою правильної реалізації гіпермедіа. Це дозволяє користувачам обмінюватися даними з клієнтами та серверами, навіть якщо вони розміщені на різних серверах.

– Адаптивний: REST API адаптується до будь-якої модифікації даних, які містяться в базі даних, навіть якщо вони розміщені на різних бек- та зовнішніх серверах. Оскільки це певною мірою залежить від кодів, це допомагає без проблем синхронізувати дані на веб-сайтах.

 Простота розуміння: оскільки REST використовує методи HTTP-дієслів (GET, POST, PUT або DELETE) для зв'язку, ці методи зрозумілі самі за собою.
 Крім того, архітектура REST допомагає підвищити продуктивність розробників, дозволяючи їм відображати інформацію на стороні клієнта та зберігати або маніпулювати даними на стороні сервера.

3.2 Реалізація GUI

Щоб реалізувати клієнтську частину додатку було обрано використовувати фреймворк React.

React (також відомий як React.js або ReactJS) — це безкоштовна бібліотека JavaScript із відкритим вихідним кодом для створення інтерфейсів користувача на основі компонентів інтерфейсу користувача. React можна використовувати як базу для розробки односторінкових, мобільних або серверних додатків із такими фреймворками, як Next.js. Однак React займається лише керуванням станом і відтворенням цього стану в DOM, тому створення додатків React зазвичай вимагає використання додаткових бібліотек для маршрутизації, а також певної функціональності на стороні клієнта.



Рисунок 3.3 – Віртуальне дерево компонентів

Віртуальний DOM

Важливою особливістю є використання віртуальної об'єктної моделі документа або віртуальної DOM. React створює кеш структури даних у пам'яті, обчислює отримані відмінності, а потім ефективно оновлює DOM, що відображається в браузері. Цей процес називається примиренням. Це дозволяє програмісту писати код так, ніби вся сторінка відображається під час кожної зміни, тоді як бібліотеки React відображають лише субкомпоненти, які фактично змінюються. Таке вибіркове відтворення забезпечує значне підвищення продуктивності.[12] Це заощаджує зусилля, пов'язані з перерахунком стилю CSS, макетом сторінки та рендерингом для всієї сторінки.

3.3 Реалізація сервера

Node.js — це кросплатформене середовище виконання JavaScript із відкритим вихідним кодом. Node.js дозволяє розробникам використовувати JavaScript для написання інструментів командного рядка та для сценаріїв на стороні сервера — запуск сценаріїв на стороні сервера для створення динамічного вмісту веб-сторінки, перш ніж сторінка буде відправлена у веббраузер користувача. Отже, Node.js являє собою парадигму «JavaScript всюди», що об'єднує розробку веб-додатків навколо однієї мови програмування, а не різних мов для сценаріїв на стороні сервера та клієнта.

Node.js має архітектуру, керовану подіями, з можливістю асинхронного введення-виведення. Ці варіанти дизайну спрямовані на оптимізацію пропускної здатності та масштабованості у веб-додатках з багатьма операціями введення/виводу, а також для веб-додатків реального часу (наприклад, комунікаційні програми в режимі реального часу та браузерні ігри).

База даних

База даних - це сукупність даних. База даних може бути такою ж простою, як текстовий файл зі списком імен. Або це може бути настільки ж складним, як велика, реляційна система управління базами даних, в комплекті з вбудованими інструментами, які допоможуть вам зберегти дані.

В якості базі даних обрана Mongoose DB.

Mongoose — це бібліотека моделювання об'єктів даних (ODM) для MongoDB та Node.js. Він керує зв'язками між даними, забезпечує перевірку схеми і використовується для перекладу між об'єктами в коді та представленням цих об'єктів у MongoDB.



Рисунок 3.4 – Зіставлення об'єктів між вузлом і MongoDB управляється через Mongoose.

MongoDB — це база даних документів NoSQL без схем. Це означає, що ви можете зберігати в ньому документи JSON, і структура цих документів може змінюватися, оскільки вона не застосовується, як бази даних SQL. Це одна з переваг використання NoSQL, оскільки вона прискорює розробку додатків і зменшує складність розгортання.

На (рис 3.5) зображено інтерфейс користувача. Плюсом є те що середовище виконано на англійській мові це дає змогу розширити свій словниковий запас. Але в той же час це і недолік тому що не всі знають добре англійську мову тому перший час користування буде важко користуватися нею

та на вивчення піде час що може збільшити час написання бази даних та здати вчасно проект. Звісно це на початку недолік, а потім перетворюється в плюси.

	* I Atlas App Services Charts	2. iž
DEPLOYMENT	STACY'S OR0-2022-05-24 > PROJECT 0	
rtabase	Database Deployments	
rta Lake PREVIEW	Q. Find a database deployment	+ Create
DATA SERVICES		
ggers		
to API	ardw-studio Connect View Monitoring Browse Collections ***	FREE SHAKED
ita Federation		
SECURITY		
twork Access	infontation of a craw-studio is y cuised Monitaring wild automatically resume when you connect to your cluster.	
ivanced	Visit the documentation for more info.	
rw On Atlas 🕓		
	ALCOUNT ALCOUNT THE TAX ALCOUNT ALCOUN	

Рисунок 3.5 – Інтерфейс користувача Mongoose.

На (рис 3.6) можна переглянути вже додані товари із певною інформацією про них.

Overview	Real Time	Metrics	Collections	Search	Profiler	Performance Advisor	Online Archive	Cmd
DATABASES: 1 COLLE	CTIONS: 2							
+ Crea	te Database	te sto	ST.products RAGE SIZE: 36KB TO nd Indexes	TAL DOCUMENTS: Schema A	9 INDEXES TOTA Inti-Patterns (3)	L SIZE: 86KB Aggregation Sea	roh Indexes o	
products		QUER	<pre>E { field: 'value YRESULTS: 1-9 OF 9id: objectId("Si name: "r-shirt" price: 1500 ing: "56412fe1-2: Size: "S" sex: "remale" mstyle: "Aquarel" isAvailable: truuV: 0</pre>	'') 22755909d7f2c431 3dd-45d7-8030-bs a" 2	Sice75f") S7ddbe6f6d.jpg"			
			_id: ObjectId("66 name: "1-Shirt" price: 1500 img: "dda74C29-9 size: "5" sex: "remale" material: "Cotto style: "Aquarel" isAvailable: trum v: 0	297559c9d7f2c431 c68-4213-9b15-9e n"	51ce761") d356a03ff1.jpg"			

Рисунок 3.6 – додані товари в Mongoose.

Postman

Postman — це можливість автоматизації тестових запитів. Один раз створюєте тест для запиту, і кожен раз, коли приходить відповідь, Postman автоматично тестує їх згідно з Вашими тест кейсами.

Дозволяє створювати і виконувати запити, документувати й моніторити Ваші сервіси в одному місці [9]

Для виконання елементарних запитів достатньо:

- 1. вибрати тип запиту
- 2. вбити запит у відповідне поле ...
- 3. ... або заповнити параметри через форму
- 4. натиснути кнопку "Send"

Якщо відкрити цей набір, перейти в "Методи запитів" і потім в "Запити GET", всі збережені дані відобразяться в центральній частині вікна Postman. Тепер натисніть "Відправити

C Filter History Collections	GET Request × + +++++++++++++++++++++++++++++++++	No Environment V O 🌣 Examples (0) 🕶
All Me Team	GET V https://postman-echo.com/get?test=123 Authorization Headers Body Pre-request Script Tests	Params Send V Save V
Auth: Digest Auth: Others Cookies	Type No Auth V	
Headers Request Methods ····	Body Cookies (1) Headers (14) Tests (6/6) Pretty Raw Preview JSON V Tests	Status: 200 OK Time: 1002 ms Size: 775 B
POST POST Raw Text POST POST Form Data PUT PUT Request	1 - { 2 - "args": { 3 "test": "123" 4 }, 5 - "heoders": { 6 "host": "postman-echo.com",	
PATCH PATCH Request DL. DELETE Request Utilities Utilities Utilities / Date and Time Utilities / Postman Collection	<pre>7 "occept": "*/*", 8 "occept-encoding1: "gzip, deflate", 9 "coche-encoding1: "no-coche", 10 "postmon-token": "d85cd5cc-26c1-42ee-9ec0-a9d0f8721512", 11 "user-agent": "PostmonRunt(me/6.2.5", 12 "x-forwarded-port": "483", 13 "x-forwarded-ports": "https" 14 }, 15 "unl": "https://postmon.echo.com/aet7test=123"</pre>	

Рисунок 3.7 – Приклад запиту

На (рис.3.8) показано тестування інформації вже для інтернет магазину Draw Studio

http:	ttp://localhost:4000/api/product 🖺 Save 🗸 🥖					
POS	POST					
Parar	ns Authorization Headers (9) Body • Pre-rec	uest Script Tests Settings			Cookies	
) no	ne 🧕 form-data 🔵 x-www-form-urlencoded 🌑 raw	binary GraphQL				
	КЕУ	VALUE	DESCRIPTION	000	Bulk Edit	
2	name	T-Shirt				
2	price	1500				
2	size	S				
2	sex	Female				
~	material	Cotton				
2	style	Aquarel				
	isAvailable	true				
	img	Tortul.jpg ×				
Respo					~	

Рисунок 3.8 – Приклад тестування

4 ПРОЕКТУВАННЯ WEB-ДОДАТКУ

Після проведення аналізу предметної області, програмних продуктів аналогів та вибору засобів розробки, необхідним етапом є проектування інформаційної системи. На цьому етапі необхідно створити UML-діаграми варіантів використання (прецедентів), активності і послідовності та розробити макети інтерфейсу застосунку.

Щоб почати розробляти базу даних потрібно визначити основні цілі, завдання та правила для вирішування проблеми, після чого приступити до проектування. Тому можемо сформулювати короткий опис завдання. Найменування завдання – розробка інтернет-магазину одягу.

Мета магазину – надання інформації про ексклюзивних товарів(ручного розпису фарбами), здійснювати покупки.

Функції:

- Повна інформація про товар.
- Інструкція щодо правильного використання товару.
- Калькулятор розрахунку ціни для замовлення індивідуального малюнку на одязі.
- Форма для замовлення товару.

4.1 Створення діаграми відношення сутностей

Діаграма відносин сутностей (ERD) - це візуальне уявлення бази даних, яке показує, як пов'язані елементи всередині. Діаграма складається з двох типів об'єктів — сутностей і відносин. Сутність у цьому контексті – це компонент даних із набору даних, що відображається у вигляді фігури на полотні. Відносини між сутностями представлені у вигляді зв'язків, які мають

спеціальні закінчення, які описують, як два елементи бази даних взаємодіють один з одним.

--Product Order PK **UniqueID** PK UniqueID H name name price surname secondname img size phone adress sex material FK product style isAvailable

На основі функцій які необхідно реалізувати була створена наступна діаграма бази даних:

Рисунок 4.1 – Діаграма бази даних додатку Draw Studio

4.2 Створення діаграми варіантів використання системи

Діаграма варіантів використання (англ. use-case diagram) - це діаграма, яка описує, функціонал програмної системи, що розробляється. [1]

Це графічне зображення, що у вигляді певних графічних позначень може показувати співвідношення між різними величинами, які порівнюються, та відношення між різними елементами (об'єктами, сутностями, процесами, поняттями тощо), які перебувають у залежності.

Тобто користувачі в системі поділяються на деякі групи, в залежності від якої вони отримують доступ до певних функцій системи.

Можна виділити такі групи користувачів в системі інтернет магазину:

- Адміністратори, група користувачів яким доступно редагування категорій, типів і товарів в магазині, а також додавання нових.
- Клієнт

Кожна з цих груп має свій набір функцій взаємодії з додатком:

Клієнти можуть:

- Переглянути товари
- Замовити товар заповнивши форму замовлення
- Розрахувати вартість малюнку на калькуляторі
- Замовити власний ескіз малюнку

Адміністратори можуть:

- Редагувати товари
- Додавати товари
- Видаляти товари

Також будь які користувачі можуть:

• Переглянути товари

Після виділення потрібних нам груп користувачів та способів взаємодії, для кожної була побудована наступна діаграма варіантів використання.

29



Рисунок 4.2 – Діаграма використання додатку Draw Studio

4.3 Створення діаграми діяльності

Діаграма активностей описує динамічні аспекти поведінки системи у вигляді блок-схеми, яка відображає бізнес-процеси, логіку процедур і потоки робіт — переходи від однієї діяльності до іншої. По суті, ми малюємо алгоритм дій (логіку поведінки) системи або взаємодії кількох систем.[2] Була створена діаграма що відображає діяльність користувача оформлення створення замовлення:



Рисунок 4.3 – Діаграма діяльності користувача

4.4 Створення діаграми послідовності

Діаграма послідовності використовується для демонстрації об'єктів які приймають участь у процесах системи, та послідовність взаємодій між ними.



Рисунок 4.4 – Діаграма послідовності

4.5 Створення макетів інтерфейсу

На етапі макет інтерфейсу додаток матеріалізується, а ідеї набувають контуру готового продукту.

Макети поділяються на два типи, низької та високої точності:

Макет низької точності – це грубі макети. Що відображують порядок, структуру та розташування елементів на екрані. Такими як (сітка, колір, шрифти і т.д.) Макет високої точності – це результати розробки дизайну інтерфейсу з такими як (колір, іконки, шрифти, сітка, фон). Макет у вигляді графічних файлів замовник отримає у форматі Photoshop, Figma та інші.

Для розробки макету було обрано середовище Figma.

Figma – це програма для веб-дизайнерів, за допомогою якої можна створювати не тільки прототипи, але й кінцеві інтерфейси сайтів та програм. Процес роботи в програмі інтуїтивно зрозумілий, а можливість спільної одночасної роботи над проектом приваблює все більше команд до реалізації прототипів саме у Figma.[3]

Плюси та мінуси Figma

До плюсів роботи у Figma можна віднести:

- Безкоштовна у випадку, якщо над проектом працюють не більше 2 користувачів.
- Створення та відображення макету в одній програмі
- Існує безліч вже готових шаблонів сайтів
- Для роботи з Figma потрібна нова версія браузера, а робоча версія не вимагає потужного і сучасного комп'ютера.
- Збереження версій проекту до 30 днів на безкоштовній версії

Мінуси Figma:

- Відсутність русифікації. Працювати у Figma доводиться англійською мовою.
- Хоча Figma працює офлайн, але щоб колеги побачили внесені зміни, потрібна синхронізація з хмарою.

В Figma з ліва є така панель з інструментами та елементами з якими можна взаємодіяти. Це дає зручно виділити той чи інший елемент не торкаючить на самий макет.



Рисунок 4.5 – Панель елементів

З права є панель яка відповідає за редагування виділеного елементу. Тут можна обрати колір, розмір, шрифт та інші налаштування для елементу



Рисунок 4.6 – Панель для редагування

Також є вкладка Inspect де програміст може подивитися CSS код для більш зручного написання сайту.

Properties		
Width	259px	
Height	346px	
Тор	95px	
Left	650px	
Blend	Pass through	
Code		
CSS ~	\leftrightarrow =	
/* Black Friday 2020 _ Figure Graphic Tee _ SHEIN USA 1 */		
position:	absolute;	
width: 259	px;	
left: 650p	орх, х;	
top: 95px;		
<pre>background: url(Black Friday 2020 _ Figure Graphic Tee _ SHEIN USA.jpg);</pre>		

Рисунок 4.7 – Панель з кодом CSS

В результаті проектування магазину Draw Studio були створені три основні сторінки: головна сторінка з товарами(рис. 4.8), форма замовлення (рис 4.9), а також сторінка з калькулятором (рис 4.10).

Головна сторінка магазина доступна всім користувачам. На ній розміщені всі товари які є в магазині, а також функції сортування для швидкого та зручного пошуку товару.



Рисунок 4.8 – Головна сторінка інтернет магазину

Коли ми натискаємо на товар з'являється така форма з інформацією про товар та формою для замовлення

	NEW A The Best Online Sale	RRIVALS s to Shop This Weekend	
	ФУТЕОЛКА		Suile
315.00		РОЗМІР: 5 жіноча МАТЕРІАЛ: хлопок СТИЛЬ: акварельний	
313,00		ІНСТРУКЦІЯ: -прасувати без пару -прати не вище 40с -не викручувати руками Є в наявності ЦІНА: \$15.00/400 uah	.515.00
-	(ім'я: Прізвище: (Побатькові: (номер телефону:	Відділення нової пошти: Оплата: е нова пошта замовити Ехіт Х	Smile
\$15.00	\$13.00	\$15.00	\$15.00

Рисунок 4.9 – Форма інформації та замовлення товару

На сторінці **ЗАМОВИТИ** розташований калькулятор розрахунку ціни для ескізу. Це дасть замовнику дізнатися приблизну або точну ціну ескізу для замовлення на власному одязі або на одязі який є в наявносі.

Nasly Space		@fonouna	WMaraanu	🗟 Замовити
Trending 2022 Draw Studio Oper resel spolers search oper corolaei war olice a do opprine casie Shap rese	Amelouin			Smile
	NEW ARRIVALS The Best Online Sales to Shop This We	ekend		
	калькулятор ціни			
	PO3MIP + 10+10 0 20x20 0 30x30 0	(50>)		
	ВИД * (дутболка) (динисовка) * (динисов (руутта)	>		
	СТИЛІ (гровий) © мультиции * (гровий) © аніми)	# #		
	ЕСКІЗ (готовий) Свласний	>		
	РЕЗУЛЬТАТ			
	Nastly Space			
	Diperca Taiposo 😵 🕑 💩 x(380)(83164023) natiy spacegymel.com 🙆 <table-cell></table-cell>			
	Anastania Telizhenko 7022			

Рисунок 4.10 – Сторінка калькулятора розрахунку ціни ескізу

5 РЕАЛІЗАЦІЯ ІНТЕРНЕТ МАГАЗИНУ «DRAW STUDIO»

Щоб почати розробку потрібно встановити все необхідне ПО яке буде використовуватись для створення магазину: IDE VSCode, середовище розробки. Та деякі корисні розширення до нього.

Платформа NodeJS і ExpressJS для створення сервера.

Mongoose - Насамперед розглянемо MongoDB. MongoDB - це база даних, що зберігає ваші дані як документи.[4]

Express - є популярним веб-фреймворком, написаним на JavaScript і працюючим усередині середовища виконання node.js. Цей модуль висвітлює деякі ключові переваги цього фреймворку, встановлення середовища розробки та виконання основних завдань веб-розроблення та розгортання.[5]

АРІ - програмістам спрощує створення коду. це набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою).[6]

5.1 середовище розробки

Для розробки інтернет магазину було обрано середовище розробки VSCode.

Visual Studio Code — це безкоштовна IDE від Microsoft. Цей редактор підтримує різні мови програмування. [7]

VSCode можна легко функціонально розширити за допомогою розширень. VS Code працює безпосередньо з Github і пропонує надзвичайно гарне підсвічування синтаксису для різних мов і розширений варіант завершення коду.



Рисунок 5.1 – Інтерфейс VSCode

Розширення в VSCode дають можливість працювати з різноманітними мовами програмування. Для програміста це дуже зручно адже не потрібно втановлювати різні програмні середовища для мов програмування. Все знаходитья в одній програмі.

Також розширення допомагають зменшити час на написання коду, або зробити красиві іконки для файлів які можна подивитися на (рис.5.1). Ще дуже зручна функція Live Share вона дає змогу підключатися до іншого користувача та працювати над проектом в парі.



Рисунок 5.2 – Встановленні розширення

5.2 Розробка серверної частини

Створюємо спершу файлову структуру сервера, та всі необхідні пакети. В кожній директорії буде зберігатись певні частини сервера.



Рисунок 5.3 – Файлова система сервера



Рисунок 5.4 – Список пакетів

Створимо навігацію, використовуючи маршрути, клієнтська частина додатку зможе взаємодіяти з сервером, загружати нові дані, або отримати якусь інформацію.

JS index.	js ×
server >	src > routes > JS index.js >
1	<pre>const Router = require('express').Router;</pre>
2	<pre>const productRouter = require('./product-router');</pre>
	<pre>const orderRouter = require('./order-router');</pre>
5	<pre>const router = new Router();</pre>
	<pre>router.use('/product', productRouter);</pre>
	<pre>router.use('/order', orderRouter);</pre>
9	
10	<pre>module.exports = router;</pre>
11	

JS orde	r-router.js ×
server 2	> src > routes > JS order-router.js >
1	<pre>const Router = require('express').Router;</pre>
	const router = new Router();
	<pre>const OrderController = require('/controllers/order-controller');</pre>
	router.post('/', OrderController.addOrder);
	router.get('/', OrderController.getAllOrders);
	<pre>router.get('/:id', OrderController.getOneOrder);</pre>
	router.delete('/:id', OrderController.deleteOrder);
	<pre>module.exports = router;</pre>
11	

Рисунок 5.5 – Маршрути сервера

5.3 Розробка клієнтської частини

Щоб створити клієнтську частину додатку починемо з файлової структури та установки необхідних пакетів.



Рисунок 5.6 – Файлова система клієнта



Рисунок 5.7 – Список пакетів для клієнта

Структура клієнтської частини має свої директорії для зберігання певних фрагментів коду:

- models інтерфейси
- арі запити на сервер
- routes список маршрутів додатку
- components всі реакт компоненти з яких створений додаток.
- pages набір сторінок, кожна з яких використовує деякі компоненти

В папці components знаходяться створений додаток, він в свою чергу поділяється на ще папки в ких зберігається код до певного елементу.



Рисунок 5.8 – Файлова система папки components.

Нам необхідний такий код під назвою Navbar.jsx :

```
import React from 'react';
import { NavLink } from 'react-router-dom';
import styles from './Navbar.module.css';
import logo from '../../assets/img/logo.svg';
import homeIcon from '../../assets/img/nav-home.svg';
import orderIcon from '../../assets/img/nav-order.svg';
import sketchIcon from '../../assets/img/nav-sketch.svg';
const Navbar = () => {
    return (
        <div className={styles.navbar}>
            <img src={logo} className={styles.logo}></img>
            <nav className={styles.nav}>
                <div href="" className={styles.nav_item}>
                    <img src={homeIcon} className={styles.icon}></img>
                    <NavLink className={styles.label} to="main">
                        Головна
                    </NavLink>
                </div>
                <div href="" className={styles.nav_item}>
                    <img src={orderIcon} className={styles.icon}></img>
                    <NavLink className={styles.label} to="order">
                        Замовити
                    </NavLink>
                </div>
                <div href="" className={styles.nav_item}>
                    <img src={sketchIcon} className={styles.icon}></img>
                    <NavLink className={styles.label} to="sketch">
                        Ескізи
                    </NavLink>
                </div>
            </nav>
        </div>
   );
};
export default Navbar;
```

Jsx розширення в файлі дає нам можливість працювати з кодом html в js. Це набагато спрощує написання коду, робить його зручнішим та зрозумілішим.

Тепер цей код можна стилізувати за допомогою CSS, виглядає це таким чином:

```
.navbar {
    background-color: #a7aaae;
```

```
height: 71px;
   display: flex;
    justify-content: space-between;
    align-items: center;
}
.logo {
   margin-left: 10vw;
    padding-top: 10px;
   height: 45px;
}
.nav {
   display: flex;
    justify-content: space-evenly;
   align-items: center;
   width: 30%;
   margin-right: 42px;
}
.label {
   text-decoration: none;
   font-family: 'Roboto';
   font-style: normal;
   font-weight: 400;
   font-size: 1rem;
    line-height: 19px;
    letter-spacing: 0.11em;
    color: #493232;
    border-bottom: 1px solid #493232;
}
.nav_item {
   display: flex;
    justify-content: space-between;
}
/* icon */
.icon {
   width: 20px;
   height: 20px;
   margin-right: 4px;
}
```

Ось що вийшло в результаті написання нами цього коду

Nasty Space

🖗 Головна 📄 Замовити 🎾 Ескізи

Рисунок 5.9 – Навігація сайту

Ще дуже важливим моментом є те що ми зображення імпортуємо на початку коду, присвоюємо ім'я, це дає змогу зробити код більш зрозумілішим та швидко знайти елемент та звернутися до нього.

☐ Заметки о выпуске: 1.68.1
client > src > components > navbar > 🌺 Navbar.jsx >
1 import React from 'react';
<pre>2 import { NavLink } from 'react-router-dom';</pre>
<pre>3 import styles from './Navbar.module.css';</pre>
<pre>4 import logo from '//assets/img/logo.svg';</pre>
5 import homeIcon from '//assets/img/nav-home.svg';
6 import orderIcon from '//assets/img/nav-order.svg';
<pre>7 import sketchIcon from '//assets/img/nav-sketch.svg';</pre>
8

Рисунок 5.10 – Імпорт зображення

Після того як ми доповнили код ми можемо його зберегти на Github.

GitHub — один з найбільших вебсервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git i розроблений на Ruby on Rails i Erlang компанією GitHub, Inc (panime Logical Awesome).[8]

Сервіс безкоштовний для проєктів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проєктів пропонуються різні платні тарифні плани.



Рисунок 5.11 – GitHub сторінка

За допомогою певних команд ми завантажуємо зміни на GitHub та це матиме вигляді коментаря з всими змінами.



Рисунок 5.12 – Додані зміни на GitHub

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра був створений інтернет магазин «Draw Studio» для продажу унікальних речей з ручним розписом спеціальними фарбами по тканині. Для цього був проведений аналіз предметної області та порівняльний аналіз існуючих програм-аналогів у Draw Studio. В якості програмного засобу розробки було обрано середовище VSCode тому що, що це середовище дає змогу працювати з різними мовами одночасно та все знаходиться в одному місці за допомогою спеціальних розширень. Для розробки дизайну інтерфейсу було обрано середовище Figma. Це онлайн середовище яке дає змогу багатьом користувачам працювати разом над одним проектом. Зміни завжди зберігаються автоматично тому це дає гарантію що все збережеться в разі якихось збоїв з мережею. Було описано всі технології які використовувалися. На етапі проектування були створені UML-діаграми варіантів використання (прецедентів), активності і послідовності та розроблені макети інтерфейсу застосунку. Тому можна сказати що головна ідея інтернет магазину була виконана. Користувачі зможуть замовити одяг з оригінальним принтом в любий куточок України. А також запропонувати власний ескіз та замовити його

В роботі представлені можливості застосунку та поетапний приклад його використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chart of precedents [Електронний ресурс].

Режим доступу: <u>https://uk.wikipedia.org/wiki</u>

2. Activity chart [Електронний ресурс]

Режим доступу:<u>https://evergreens.com.ua/ua/articles/uml</u>

diagrams.html#:~:text=%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0 %B0%D0%BC%D0%B0

3. UI / UX analysis on the example of a prototype in Figma and basic principles [Електронний ресурс]

Режим доступу: <u>https://habr.com/ru/post/508028/</u>

4. Introduction to Mongoose for MongoDB and Node.js [Електронний ресурс]

Режим доступу: <u>https://code.tutsplus.com/uk/articles/an-introduction-to-</u> mongoose-for-mongodb-and-nodejs--cms-29527

5. Express [Електронний ресурс]

Режим доступу: <u>https://expressjs.com/uk/</u>

6. What is an API [Електронний ресурс]

Режим доступу: <u>https://marketer.ua/ua/shho-take-api/#:~:text=</u>

7. Visual Studio Code [Електронний ресурс]

Режим доступу: https://uk.wikipedia.org/wiki/Visual_Studio_Code#:~:text=Visual%20Studio

8. GitHub [Електронний ресурс]

Режим доступу: <u>https://uk.wikipedia.org/wiki/GitHub#:~:text=GitHub.</u>

9. Postman [Електронний ресурс]

Режим доступу: <u>https://www.quality-assurance-group.com/znajomtes-</u> postman-dlya-testuvalnyka-must-have/

ДОДАТОК А

Вихідний код програми

```
const mongoose = require('mongoose');
const dotenv = require('dotenv');
dotenv.config();
const express = require('express');
const fileUpload = require('express-fileupload');
const cors = require('cors');
const path = require('path');
const router = require('./routes/index');
const PORT = process.env.PORT || 4000;
const app = express();
app.use(express.json());
app.use(fileUpload({}));
app.use(express.static(path.resolve(__dirname, '..', 'public')));
app.use(cors());
app.use('/api', router);
const start = async () => {
   try {
        await mongoose.connect(
            'mongodb+srv://Stacy:boZY1X1F6Kt7rURF@draw-
studio.gqedw.mongodb.net/?retryWrites=true&w=majority'
        );
        app.listen(PORT, () => {
            console.log(`App is running at http://localhost:${PORT}`);
        });
    } catch (error) {
        console.log(error);
    }
};
start();
```

import React from 'react';

import './App.css';

import AppRouter from './components/AppRouter';

import FirstScreen from './components/first-screen/FirstScreen';

import Footer from './components/footer/footer';

import Navbar from './components/navbar/Navbar';

```
function App() {
```

return (

```
<div className="App">
```

<header>

<Navbar />

<FirstScreen />

</header>

<main>

<AppRouter />

</main>

<footer>

<Footer />

</footer>

</div>

```
);
```

}

```
export default App;
import React from 'react';
import { NavLink } from 'react-router-dom';
```

```
import styles from './Navbar.module.css';
import logo from '../../assets/img/logo.svg';
import homeIcon from '../../assets/img/nav-home.svg';
import orderIcon from '../../assets/img/nav-order.svg';
import sketchIcon from '../../assets/img/nav-sketch.svg';
```

```
const Navbar = () => {
    return (
        <div className={styles.navbar}>
            <img src={logo} className={styles.logo}></img>
            <nav className={styles.nav}>
                <div href="" className={styles.nav_item}>
                    <img src={homeIcon} className={styles.icon}></img>
                    <NavLink className={styles.label} to="main">
                        Головна
                    </NavLink>
                </div>
                <div href="" className={styles.nav_item}>
                    <img src={orderIcon} className={styles.icon}></img>
                    <NavLink className={styles.label} to="order">
                        Замовити
                    </NavLink>
                </div>
                <div href="" className={styles.nav_item}>
                    <img src={sketchIcon} className={styles.icon}></img>
                    <NavLink className={styles.label} to="sketch">
```

```
Ескізи
```

```
</NavLink>
                </div>
            </nav>
        </div>
    );
};
export default Navbar;
import React from 'react';
import styles from './modal.module.css';
const Modal = ({
    name,
    img,
    size,
    sex,
    material,
    style,
    isAvailable,
    price,
}) => {
    return (
        <div className={styles.card}>
            <h2>{name}</h2>
            <div className={styles.content}>
                <img className={styles.img} src={img} alt="t-shirt photo" />
```

```
<div className={styles.content_text}>
```

```
Size:{size}
      Sex: {sex}
      Material: {material}
      Style: {style}
      instruction:
      прасувати без пару
          прати не вище 40С
          не викручувати руками
      {isAvailable ? (
          <span className={styles.yes}>YES</span>
      ) : (
          <span className={styles.no}>NO</span>
      )}
      Price {price}
   </div>
</div>
<div className={styles.form}>
   <div className={styles.inputs}>
      <input
          type="text"
          placeholder="name"
          className={styles.input_name}
      />
      <input
          type="text"
```

```
placeholder="surname"
```

```
className={styles.input_name}
        />
        <input
            type="text"
            placeholder="secondname"
            className={styles.input_name}
        />
        <input
            type="text"
            placeholder="phone +(000)-000-00-00"
            className={styles.input_name}
        />
        <input
            type="text"
            placeholder="New post office:"
            className={styles.input_name}
        />
    </div>
    Оплата: Нова пошта
    <button className={styles.button_buy}>Buy</button>
    <button className={styles.button_exit}>Exit</button>
</div>
```

);

</div>

};

```
export default Modal;
import React, { useEffect, useState } from 'react';
import Product from '.../product/product';
import styles from './product-list.module.css';
const ProductList = () => {
    const [products, setProducts] = useState([
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$500.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$1500.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$5000.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
```

58

```
name: 'T-Shirt',
            price: '$500.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$1500.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$5000.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$500.00',
        },
        {
            img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
            name: 'T-Shirt',
            price: '$1500.00',
        },
        {
```

59

```
60
           img:
'https://i.pinimg.com/736x/2d/aa/e0/2daae037e8100de649e2256160ce87c3.jpg',
           name: 'T-Shirt',
           price: '$5000.00',
       },
   ]);
   return (
       <div className={styles.shop}>
           <h1 className={styles.title}>new arrivals</h1>
           The Best Online Sales to Shop This Weekend
           <div className={styles.container}>
               <div className={styles.products}>
                  {products.map((product, i) => {
                      return (
                          <Product
                              key={i}
                              img={product.img}
                              name={product.name}
                              price={product.price}
                          />
                      );
                  })}
               </div>
           </div>
```

```
</div>
);
};
export default ProductList;
```