

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ КАТАЛОГУ
ТОВАРІ ДЛЯ ТВОРЧОСТІ

SOFTWARE IMPLEMENTATION OF AN ONLINE
CATALOGUE FOR CREATIVITY GOODS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-120сп

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ДОН А.Р.

(ПРИЗВИЩЕ та ініціали)

Керівник ГОЛУБ Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ПОЛЯКОВ М.О.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ДОН Анастасії Романівни
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація інтернет каталогу товарів для творчості.
Software implementation of an online catalogue for creativity goods

керівник проєкту (роботи) к.т.н., ГОЛУБ Тетяна Василівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “04” квітня 2022 року № 87

2. Строк подання студентом проєкту (роботи) 01 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз предметної області.
2. Проєктування та реалізація програмного забезпечення. 3. Програмування. 4. Тестування та експлуатація застосунку.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ГОЛУБ Т.В., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання « 04 » квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

Підпис Анастасія ДОН
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Підпис Тетяна ГОЛУБ
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 67 с., 30 рис., 3 табл., 3 дод., 10 джерел.

ТОВАРИ, БАЗА ДАНИХ, ВЕБКАТАЛОГ, VUE.JS, MONGODB

Об'єктом роботи є перегляд товарів для творчості. Предметом роботи є програмне забезпечення для перегляду товарів для творчості. Метою створення додатку є розробка програмного продукту, призначеного для перегляду товарів у магазині.

Матеріали, методи та технічні засоби: мова програмування Vue.js, система керування базами даних MongoDB.

Результати. Створено каталог товарів магазину для творчості. Це скоротить час вибору продукту перед покупкою в самому магазині. Покупець зможе переглянути повну інформацію про товар.

Висновки. Розроблена програма дозволяє переглядати всю доступну інформацію про товар, який представлений у магазині.

Галузь використання. Перегляд товарів для творчості представлених в магазині.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 67 pages, 30 figures, 3 tables, 3 appendixes, 10 sources.

PRODUCTS, DATABASE, WEB CATALOG, VUE.JS, MONGODB

The object of the work is the review of goods for creativity. The subject of the work is software for viewing goods for creativity. The purpose of creating the application is to develop a software product designed for browsing products in the store.

Materials, methods and technical means: Vue.js programming language, MongoDB database management system.

Results. A store product catalog has been created for creativity. This will reduce the time of product selection before purchasing in the store itself. The buyer will be able to view complete information about the product.

Conclusions. The developed program allows viewing all available information about the product presented in the store.

Field of use. Review of products for creativity presented in the store.

Перелік скорочень та умовних познач	7
Вступ	8
1. Аналіз предметної області	9
1.1.Особливості магазину товарів для творчості	9
1.2.Аналіз видів сайтів для представлення магазину	10
1.3.Постановка завдання	11
2. Проєктування та реалізація програмного забезпечення	13
2.1.Обґрунтування вибору мови програмування	13
2.2.Використовувані моделі баз даних	16
2.3.Обґрунтування вибору середовища розробки системи	19
2.4.Середовище функціонування системи	20
3. Програмування	23
3.1.Структурна схема програми	23
3.2.Основні рішення щодо реалізації компонентів системи	23
3.2.1.Розробка модулів системи	23
3.2.2 Реалізація програми	26
4. Тестування та експлуатація програми	31
4.1.Призначення і умови використання програми	31
4.2.Тестування програми	33
Висновки	38
Перелік джерел посилення	39
Додаток А Технічне завдання	40
Додаток Б Текст програми	43
Додаток В Слайди презентації	60

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

HTML – HyperText Markup Language;

IE – Internet Explorer;

JS – JavaScript;

БД – база даних;

СКБД – система керування базами даних;

ODM – об’єктно-документний відображувач.

ВСТУП

Сьогодні Інтернет є найпопулярнішою технологією для просування свого бізнесу. Так, кожен може легко відвідати ту чи іншу сторінку в Інтернеті, ввівши адресу сайту або натиснувши посилання.

Створення вебсайту включає не тільки розробку красивого оригінального дизайну сторінки. Розробка сайту також вимагає правильного вибору кольірних схем, шрифтів, зручної навігації та інших елементів.

Вебкаталог — це спосіб організації та категоризації гіперпосилань вашого сайту. Як приклад вебкаталог «e-Katalog» в якому представлено різні товари від побутових приладів до посуду. Покупець може знайти товар, який йому потрібен, і порівняти ціни і наявність в магазинах в міста, де він хоче зробити покупку.

Категорії сайту базується на вмісті всього сайту, а не окремих сторінок чи ключових слів. Відповідно до принципів каталогу, сайти можуть бути віднесені до однієї або кількох категорій або підкатегорій.

Вебкаталоги забезпечують трафік у потрібному напрямку і наповнюють ваш сайт односторонніми посиланнями. Це допомагає просувати ваш сайт.

Це підвищує популярність ваших посилань і покращує результати пошуку. Вебкаталоги створюють списки для альтернативних посилань, які можна купити за гроші.

Метою створення додатку є розробка програмного продукту, призначеного для перегляду товарів у магазині.

Дана робота присвячена розробці програмного продукту, призначеного для перегляду товарів у магазині.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості магазину товарів для творчості

Метою вебсайту та додатку є розробка онлайн-каталогу магазинів, що спеціалізуються на товарах для творчості.

Основна ідея - створити каталог, який допоможе користувачам краще познайомитися з магазинами та знайти швидкі відповіді на свої запитання.

Магазин - це заклад або частина закладу зі спеціальними приміщеннями для роздрібного продажу товарів і надання послуг покупцям. Магазин з торговельним залом, підсобним приміщенням і приміщеннями для приймання, зберігання та підготовки товарів до продажу є складом.

"ArtHouse" - це магазин, різноманітних фарб та різноманітних матеріалів для творчості. В "ArtHouse" є все необхідне, від пензлів до підрамників. У магазині представлений широкий асортимент фарб різних виробників, наприклад:

- Rosa - один з провідних в країні виробників професійних товарів для малювання, хобі та декору. Вони розпочали виробництво в 1996 році. Компанія почала з виробництва полотен для живопису ручної роботи.

- Royal Talens - провідний виробник розкішних товарів для художників, заснований у 1899 році Мартіном Таленсом, банкіром, який не лише знався на фінансах, а й мав пристрасть до живопису. Отримавши високу оцінку від голландської королівської сім'ї, він вирішив надати своїй продукції королівський статус.

- Classico - це тепло сонця, барвисті італійські пейзажі, нескінченно надихаючий темперамент кольору та безмежна яскравість.

Магазин має широкий асортимент товарів і потребує втілення додаткових інструментів для зручності його функціонування. Таким інструментом є вебкаталог.

1.2 Аналіз видів сайтів для представлення магазину

Головна ідея — створити сайт, який допоможе користувачам ближче познайомитися з магазином, знайти швидко відповіді на питання, які їх цікавлять, дізнатися про наявність певних речей, які присутні в магазині.

В даний час всі сайти прийнято розділяти на наступні види:

- візитна картка;
- landing page;
- персональний;
- корпоративний;
- інтернет-магазин;
- каталог [1].

Візитна картка - це невеликий інтернет-ресурс, зазвичай не більше 10-15 сторінок, що містить контактну інформацію про конкретну компанію, а також перелік і опис послуг, що надаються..

Landing Page - це новий формат вебсайтів. Це вебсторінка на піддомені або домені, основна мета якої - спонукати користувачів до цільової дії.

Персональний - це сайт-візитка, але він не орієнтований на вирішення бізнес-завдань. Зазвичай створення персонального сайту замовляють приватні особи, щоб надати конкретну інформацію про себе, свої хобі та улюблені заняття.

Корпоративний - такі вебресурси призначені для повноцінних бізнес-структур. Окрім стандартних розділів, сайти компаній містять більш детальну інформацію про товари, які вони продають, та послуги, які вони пропонують.

Інтернет-магазин — є одним з найпоширеніших вебресурсів для продажу на сьогоднішній день. Він є розширеною версією онлайн-каталогу, але, на відміну від нього, пропонує віртуальний кошик і можливість замовлення товарів за допомогою різних способів оплати. При цьому умови

гарантії, доставки, оплати та обміну варіюються в залежності від обраного ІМ, але завжди відповідають нормам і вимогам чинного законодавства.

Інтернет-каталог – зручна і доступна подача конкретної інформації у великому обсязі. Зазвичай має вигляд каталогу. На відміну від переліку товарів, розташованих на інших інтернет-ресурсах, характеризується більш детальною розробкою, з обов’язковою функцією пошук по сайту.

Онлайн-каталоги наразі є більш корисним ресурсом, де покупці можуть вибрати товари, прочитати необхідну інформацію та вирішити, який магазин є найкращим місцем для покупки. Оскільки в каталозі доступно більше інформації, люди можуть дізнатися більше про товар, його особливості та недоліки.

Покупець, заходячи на сайт, зможе побачити весь асортимент товару, ціну і наявність того чи іншого товару в магазині.

Каталог " ArtHouse " представляє конкретні товари, представлені в магазині. Там представлені різні бренди фарб, пензлів і інших товарів, але вони зібрані в одному магазині і покупцю буде зручно вибирати що йому треба.

1.3 Постановка завдання

Після оцінки цільової області та вибору найкращого типу вебсайту для представлення магазину, було вирішено створити програмне забезпечення з наступним переліком функцій:

- навігація: для того, щоб користувачам було зручно користуватися програмним забезпеченням. Вони не повинні губитися в структурі або відволікатися на сторонні матеріали;
- дизайн: вебсайт повинен мати сучасний та естетично привабливий вигляд;
- інтерфейс повинен бути зроблений так, щоб він легко сприймався очима, з приємними шрифтами та кольоровими схемами;

- інтерфейс повинен бути мінімалістичним, максимально простим та інтуїтивно зрозумілим;
- вебкаталог повинен бути розроблений з урахуванням максимальної зручності для користувача;
- відвідувачі повинні мати можливість бачити товари, які їм подобаються;
- адаптивність: сайт повинен коректно відображатися в різних браузерах і на мобільних пристроях;
- повнота інформації та оригінальність контенту: текст і зображення на сайті повинні відповідати всім поставленим перед ним завданням. Тексту має бути мінімум і він має бути цікавим для відвідувачів. Текст повинен бути структурованим і легко читатися з екрану.

Основними вимогами до розробленої системи є наступні

- вимагати перегляду оригінальних форм;
- вебсайт та додаток повинні забезпечувати можливість переходу від однієї форми до іншої;
- вебсайти та додатки повинні мати можливість працювати без необхідності встановлення програмного забезпечення;
- вебсайт та додаток повинні мати зручний для користувача інтерфейс, і слід приділяти максимальну увагу всім користувачам.

Основними вимогами до надійності програми є:

- сайт та додаток повинні не давати можливість користувачеві введення даних в комірки з вихідними даними;
- зручний та швидкий пошук потрібної інформації;
- сайт та додаток повинні конкретно інтерпретувати і зберігати інформацію;
- для повернення на головну форму сайту повинна бути кнопка “ArtHouse”;
- для закриття форми повинна бути кнопка «Вихід» (тобто повне закриття форми).

2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Обґрунтування вибору мови програмування

Веброзробка зазвичай реалізується наступними мовами. Vue.js, React та Angular (табл. 2.1).

Таблиця 2.1 – Вибір мови розробки програми

Функції	Vue.js	React	Angular
Синтаксиз	Простий у використанні та зрозумілий	JSX (JavaScript XML) для опису компонентів.	HTML-подібний шаблонний синтаксис.
Розмір	Невеликий розмір, що дозволяє швидку завантаження	Невеликий розмір, можливість додатково оптимізувати розмір з допомогою Webpack	Більший розмір, особливо залежно від використання додаткових модулів
Продуктивність	Швидка відповідь завдяки прямому зв'язку з DOM.	Швидка відповідь завдяки використанню віртуального DOM.	Швидка відповідь завдяки використанню віртуального DOM

Продовження таблиця 2.1 – Вибір мови розробки програми

Навчання	Легкий для вивчення та початку роботи.	Потребує деякого часу та знань, особливо щодо концепцій віртуального DOM.	Вимагає певного часу та знань, особливо щодо концепцій модульності та ін'єкції v залежностей.
----------	--	---	---

React - це декларативна, ефективна та гнучка бібліотека JavaScript для побудови користувацьких інтерфейсів. [2].

Angular - це фреймворк інтерфейсу з відкритим вихідним кодом, написаний на TypeScript, розроблений компанією Google і керований Angular Team, спільнотою приватних розробників та інших компаній [3].

Vue - це прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від монолітних фреймворків, Vue розроблений для поступового впровадження. Його ядро вирішує проблеми переважно на рівні подання, що дозволяє легко інтегруватися з іншими бібліотеками та існуючими проектами. [4].

Front-end - клієнтська частина користувацького інтерфейсу для програмної та апаратної частин сервісу

Vue.js був обраний як мова програмування для створення front-end.

В ході роботи надавала перевагу HTML, мові гіпертекстової розмітки, та каскадним таблицям стилів CSS, іноді використовуючи JavaScript для пришвидшення процесу.

HTML5 (англ. HyperText Markup Language, version 5) - мова для структурування та представлення вмісту всесвітньої павутини. HTML5 була створена як єдина мова розмітки, яка надає можливість поєднувати синтаксичні норми HTML и XHTML.

Веббраузер отримує HTML-документ з вебсервера або локальної пам'яті і перетворює його на мультимедійну вебсторінку. HTML семантично визначає структуру вебсторінки і спочатку містить сигнали про зовнішній вигляд документа [5].

CSS - спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних [6].

JavaScript (JS) - це динамічна, об'єктно-орієнтована мова прототипування; це реалізація стандарту ECMAScript [7].

Найчастіше використовується для кодування вебсторінок, дозволяючи клієнтській стороні (пристрою кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно зв'язуватися з сервером і змінювати структуру і зовнішній вигляд вебсторінки..

Для написання серверного коду з використанням JavaScript було обрано програмну платформу Node.js.

Node.js - це платформа з відкритим вихідним кодом для запуску високопродуктивних мережевих додатків, написаних на JavaScript. Досі JavaScript використовувався для обробки даних у браузері користувача. node.js дозволяє запускати JavaScript-скрипти на сервері та надсилати результати користувачеві [8].

Node.js має наступні особливості:

- асинхронна однопотокова модель виконання запитів;
- безшовне введення/виведення;
- модульна система CommonJS;
- рушій JavaScript Google V8.

Для створення мобільного додатку бренду «ArtHouse» використано Android.

Основні переваги:

- Android перевершує iOS від Apple у багатьох функціях, таких як перегляд веб-сторінок та інтеграція з сервісами Google. Крім того, на відміну від iOS, Android є відкритою платформою, що дозволяє більшій кількості розробників реалізовувати функції;

- спочатку було заборонено встановлювати додатки з "неперевіраних джерел", таких як карти пам'яті, але це обмеження можна відключити в налаштуваннях пристрою стандартним способом, а оскільки додатки можна встановлювати на смартфони та планшети, які не підключені до Інтернету, це дозволяє будь-кому безкоштовно створювати та тестувати Android-додатки на власних пристроях;

- підтримує різні апаратні платформи, включаючи Android, ARM, MIPS і x86.

Недоліки:

- деякі пристрої Android використовують сервіси Google для надсилання автентифікаційної інформації, наприклад, інформації про пересування користувача, на сервер у режимі реального часу;

- для доступу до сервісів Google, включаючи Google Play, необхідно використовувати спеціальний додаток, який виробник мобільного телефону має право встановити на ваш мобільний телефон після підписання контракту з Google.

2.2 Використовувані моделі баз даних

Сімейство нереляційних моделей даних можна розділити на чотири типи: Тип "ключ-значення", документно-орієнтований тип, стовпчиковий тип та тип графів. Проаналізувавши кожен тип, для інформації про товари було обрано документно-орієнтовану базу даних.

Завдяки своїй гнучкості та підтримці змінних схем, вони переважно використовуються для управління напівструктурованими даними. Документно-орієнтовані бази даних дуже схожі на ключові значення в деяких своїх реалізаціях. Однак, їхньою одиницею є документ

Ви можете написати будь-який документ, будь-який JSON і абсолютно будь-який набір даних. Це дуже часто використовується, наприклад, при створенні каталогу, де кожен товар має різні властивості.

Наприклад, профілі користувачів. Хтось згадує свій улюблений фільм, чи улюблену страву. У таких випадках, щоб не записувати все в одне поле, ви записуєте все в документно-орієнтований JSON.

В якості СУБД було обрано MongoDB. MongoDB - це кросплатформенна, документно-орієнтована система управління базами даних з відкритим вихідним кодом.

У той час як традиційні реляційні бази даних зберігають однотипні об'єкти в жорсткій структурі, документно-орієнтовані бази даних зберігають колекції, які можуть містити різні об'єкти з різною структурою та різними наборами властивостей.

Відповідний процес відбору проб (count, group, MapReduce), вставки, зміни та видалення. Між об'єктами немає зв'язку, і об'єкт може лише зберігати інші об'єкти у своїх властивостях. Підтримуються як унікальні, так і складені індекси. Індеси можна застосовувати до властивостей вкладених об'єктів. [9].

Швидкість роботи MongoDB проявляється в процесі вставки даних, який відбувається дуже швидко. До речі, не абстрагуючись від механізму зберігання, формат зберігання також сумісний зі способом передачі об'єктів по мережі, використовуючи індекс для вибору об'єкта, знаходження його розташування і повернення клієнту фрагмента файлу певної довжини. Плюси. Відсутність абстракції від storage engine.

В MongoDB для кожного документа унікальний ідентифікатор, який має назву `_id`. Якщо не вказати його значення, він буде згенерований автоматично.

Унікальний ідентифікатор - це не автоінкрементоване поле, а унікальне 12-байтне число, згенероване клієнтом. По-перше, немає проблем із синхронізацією реплік. Це означає, що вставка може виконуватися незалежно на двох різних машинах, не викликаючи конфліктів. Крім того, після перебудови бази даних пошукові системи не будуть використовувати старі посилання для пошуку нових статей.

Спроектowana документно-орієнтовану модель зображена на рисунку 2.1.

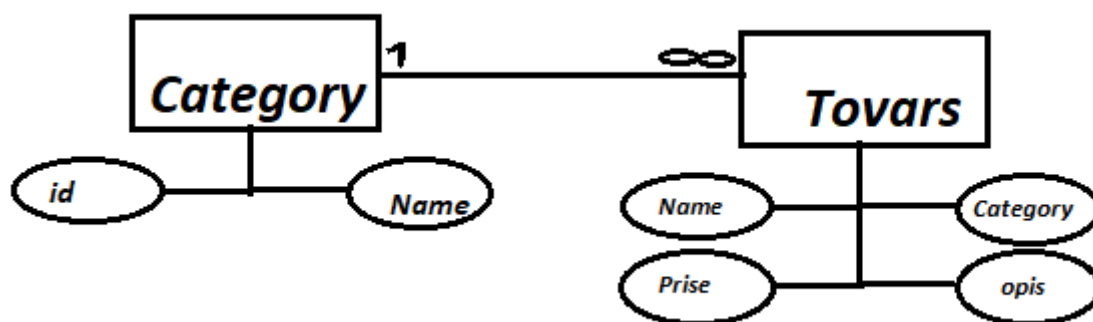


Рисунок 2.1 – Документно-орієнтовану модель бази даних

Документ «Tovar» (id, Name, Price, opis) зберігає інформацію про товари інтернет-каталогу. Опис документа «Tovar» представлений в таблиці 2.2.

Таблиця 2.2 Опис документа «Tovar»

Назва ключа	Тип значення	Опис
id	String	Забезпечує унікальність товару
Name	String	Назва товару
Price	String	Ціна товару
opis	String	Опис товару

Документ «Categories» (id, type) зберігає інформацію про категорії інтернет-каталогу. Опис документа «Categories» представлений в таблиці 2.3.

Таблиця 2.3 Опис документа «Categories»

Назва ключа	Тип значення	Опис
id	String	Забезпечує унікальність товару
type	String	Тип товару

База даних зберігає повну інформацію про назви товарів, їх характеристику, ціну, категорії товарів

2.3 Обґрунтування вибору середовища розробки системи

Програмний продукт розроблено за допомогою середовища розробки Visual Studio Code.

Visual Studio Code - це інструмент для створення програм для сучасних веб-додатків та хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний для платформ Windows, Linux та OS X..

Visual Studio Code базується на результатах розробки вільного проекту Atom, що розповсюджується на GitHub. Зокрема, Visual Studio Code є плагіном Atom Shell, що використовує движок браузера Chromium та Node.js. Зауважимо, що на сайті Visual Studio Code не згадується про використання вільного проекту Atom ні в прес-релізі, ні в офіційному блозі.

Основні можливості і переваги програми

- Visual Studio Code підтримує роботу з TypeScript, JavaScript, Node.js і Mono;
- має вбудований відладчик та командний рядок;
- підтримує майже всі мови програмування;
- має вбудовану бібліотеку елементів коду;
- автодоповнення при введенні коду;
- до бібліотеки можна додавати власні частини;
- підсвічування синтаксису;
- одночасна роботи з декількома проектами;
- підтримка багатовіконний і двохпанельний режимів;
- розширення функціоналу за допомогою плагінів;
- інтеграція з Visual Studio Team Services, GitHub і GIT;
- вбудовані інструменти для тестування, створення, пакування та розгортання додатків;

- публікація створених програмних продуктів в Microsoft Azure (за посередництвом Visual Studio Team Services);
- інтегрована довідкова система;
- командна робота над проектами;
- широкий набір налаштувань і багатоплатформлена.

2.4 Середовище функціонування системи

Розроблений програмний продукт може працювати в будь-якому браузері.

Браузер - програмне забезпечення, зазвичай для комп'ютерів або інших електронних пристроїв, підключених до Інтернету, яке дозволяє користувачам маніпулювати текстом, зображеннями та іншою інформацією на гіпертекстових вебсторінках. Текст і зображення можуть містити посилання на інші вебсторінки на цьому або інших вебсайтах. Браузери гіперпосилань дозволяють користувачам швидко і легко отримувати доступ до інформації на багатьох вебсторінках.

Станом на квітень 2019 року для серфінгу в Інтернеті доступна низка додатків (рисунок 2.2). Найпопулярніші з них:

- Google Chrome,
- Safari,
- Internet Explorer та Edge,
- Mozilla Firefox,
- Opera.

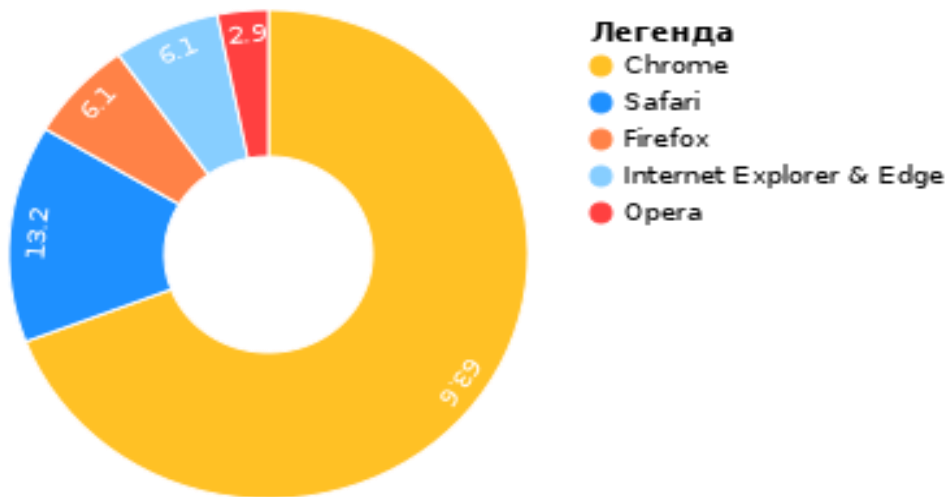


Рисунок 2.2- Частка браузерів на світовому ринку станом на 2019 року

Кожен браузер інтерпретує гіпертекст по-різному. Це пов'язано з різною реалізацією механізмів відображення і створенням спеціальних тегів для розширення функціональності стандартного HTML. Наприклад, один і той самий текст по-різному відображається в Internet Explorer і Safari.

Google Chrome - це безкоштовний веб-браузер, розроблений компанією Google на основі браузера Chromium з відкритим вихідним кодом та іншого програмного забезпечення з відкритим вихідним кодом.

Safari - це браузер, розроблений компанією Apple і доступний для операційних систем Mac OS X та iOS. Він стабільно посідає четверте місце за кількістю користувачів (частка ринку в травні 2013 року - 7,89%). Через низьку частку пристроїв Apple в Україні його частка серед браузерів значно нижча за середньосвітовий показник.

Windows Internet Explorer 9 (скорочено IE9) - версія Microsoft Internet Explorer, випущена 14 березня 2011 р. IE9 включає більшість специфікацій CSS3, закруглені межі, рендеринг SVG, колірні профілі ICC і швидший рендеринг JavaScript. IE9 також забезпечує апаратне прискорення обробки графіки за допомогою Direct2D.

Mozilla Firefox - безкоштовний браузер з відкритим вихідним кодом, заснований на ядрі Gecko (проект компанії Mozilla).

Опера - це веб-браузер, розроблений норвезькою компанією Opera Software і вперше випущений в 1994 році групою дослідників з Telenor в Норвегії. З 1995 року, коли була створена перша версія, до 2013 року Опера включала додаткові функції (електронна пошта та багато інших сервісів) для підключення до Інтернету, але після переходу з пропрієтарного движка браузера на движок Blink з відкритим вихідним кодом, додаткові функції були перенесені з браузера в пов'язаний з ним браузер..

3 ПРОГРАМУВАННЯ

3.1 Структурна схема програми

Клієнт і сервер взаємодіють за архітектурою REST. Кожній одиниці інформації відповідає певне з'єднання. Дія над даними визначається методами HTTP-запиту (GET, POST, PUT, DELETE).

Приклад взаємодії клієнта і сервера зображено на рисунку 3.1.



Рисунок 3.1 – Приклад взаємодії клієнта і сервера

Структурна схема сервера зображена на рисунку 3.2.

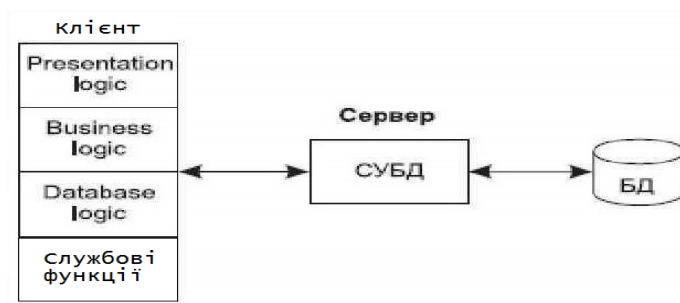


Рисунок 3.2 – Структурна схема сервера

3.2 Основні рішення щодо реалізації компонентів системи

3.2.1 Розробка модулів системи

Для найкращої відповідності вимогам до системи, що створюється, було обрано REST-архітектуру web-додатка (рисунок 3.3).

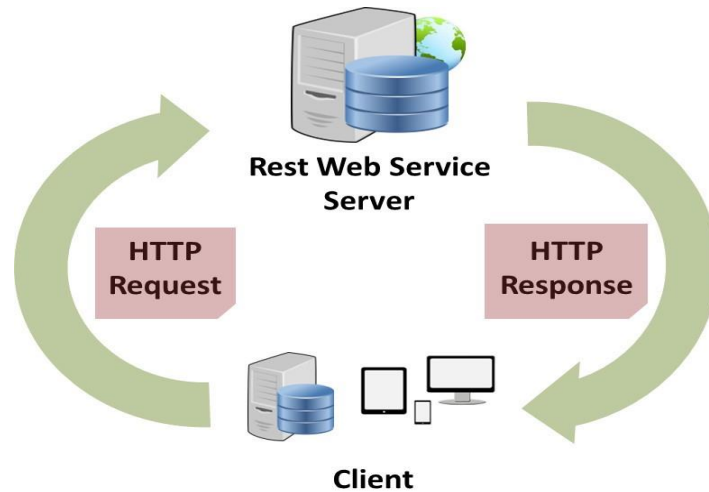


Рисунок 3.3 - REST-архітектуру web-додатка

У цій архітектурі серверна та клієнтська частини повністю розділені. Сервер зберігає та оновлює дані, в той час як клієнт відображає дані в інтерфейсі користувача та реагує на дії, що виконуються цим інтерфейсом. Такий поділ дозволяє кожному компоненту розвиватися незалежно.

Клієнт-сервер - це обчислювальна або мережева архітектура, в якій завдання і мережеві навантаження розподіляються між постачальниками послуг, які називаються серверами, і споживачами послуг, які називаються клієнтами.

Сервери - це додатки, які надають певні послуги іншим додаткам і відповідають на запити клієнтів на певні типи ресурсів.

Клієнт - це програма, яка використовує послуги, що надаються серверною програмою.

Іншими словами, клієнт і сервер - це лише ролі, які виконує програма. Один і той самий додаток може бути і клієнтом, і сервером одночасно. Фізично клієнт і сервер - це програмне забезпечення.

Взаємодія зазвичай відбувається через комп'ютерні мережі за допомогою мережевих протоколів, іноді між сервером і клієнтом відсутня інформація про стан, коли клієнт і сервер знаходяться на одному комп'ютері, тобто кожен запит містить всю інформацію, необхідну для його обробки, а дані про стан сесії зберігаються на клієнті і надсилаються з кожним запитом, що підвищує масштабованість.

Всі компоненти REST-архітектури підтримують однорідні інтерфейси. Це зменшує зв'язок між компонентами та послугами, які вони надають, і полегшує зміну компонентів у разі необхідності.

Обчислювальна модель клієнт-сервер є основним підходом до розподілених обчислень. У цій моделі додатки розділені і розміщені на різних платформах для підвищення ефективності. Зазвичай це означає, що додатки для відображення даних розміщуються на терміналі користувача, а додатки для управління даними - на сервері, як показано на рисунку 3.4. Таке розділення усуває недоліки персональних комп'ютерів, пов'язані з низькою обчислювальною потужністю та надійністю, і дозволяє відокремити персональні комп'ютери один від одного.

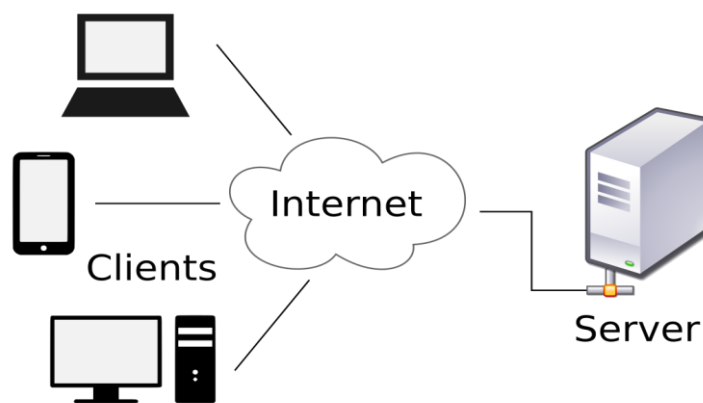


Рисунок 3.4 - Схема архітектури клієнт-сервер

HTTP розшифровується як Hyper Text Transfer Protocol - протокол передачі даних, який використовується для передачі гіпертекстових документів у комп'ютерних мережах [10].

HTTP належить до протоколів моделі OSI 7-го прикладного рівня.

Основне призначення протоколу HTTP - передача вебсторінок (текстових файлів з розміткою HTML), але інші файли, пов'язані з вебсторінками (зображення і додатки), також можуть бути успішно передані за допомогою HTTP (в цьому випадку HTTP конкурує з більш складним FTP) HTTP.

AJAX - це не окрема технологія, а концепція, яка використовує кілька взаємопов'язаних технологій, а методологія розробки користувацького інтерфейсу AJAX поєднує в собі кілька основних методів і технологій.:

- використання DHTML для динамічної зміни вмісту сторінки;
- XMLHttpRequest можна використовувати для запиту до сервера "на льоту" без перезавантаження всієї сторінки;
- альтернативний метод — динамічне підвантаження коду JavaScript в тег <SCRIPT> з використанням DOM, що здійснюється із використанням формату JSON);
- динамічне створення дочірніх фреймів

Ці підходи можуть створити більш зручний веб-інтерфейс на сторінках веб-сайту, які вимагають активної взаємодії з користувачем. Браузер перезавантажує веб-сторінку. Браузери не перезавантажують веб-сторінки і не відправляють дані на сервер без візуального підтвердження (якщо тільки ви не хочете самі побачити процес з'єднання з сервером). Ajax став популярним завдяки розвитку таких веб-сайтів, як Gmail, Google Maps і Google Suggest. Після того, як Google взяв активну участь у розробці таких веб-сайтів, як Gmail, Google Maps і Google Suggest, ринок усвідомив ефективність цієї технології, коли Google почав активно використовувати AJAX при створенні сайтів. Створення цих веб-сайтів підтвердило ефективність цієї техніки.

3.2.2 Реалізація програми

Каталог магазину "ArtHouse" дає можливість користувачам переглядати наявні товари в магазині, їх опис та характеристику. Коли клієнт заходить на сайт, перше що він бачить це головна сторінка, яка представляє собою верхню панель з меню сайта, назвою магазину, інформацію про цей магазин, контактну інформацію та посилання на завантаження додатку.

Блок інформації про магазин містить короткі візитні повідомлення щодо спрямованості роботи.

```

<div>

<h1 style="font-family: Pompadur; font-size: 550%" > <span> ArtHouse</span> </h1>
<i>
<h4 style="font-family: Pompadur; font-size: 120%"> <br> Усі, хто заходить до цього магазину,
будуть зачаровані великим вибором товарів.<br><span> Тут ви знайдете все потрібне для
творчості </span></h4>
<h4 style="font-family: Pompadur; font-size: 120%"> <br> У ArtHouse ви також
знайдете:<br><span> художні фарби, полотна, пензлі, рамки, олівці, дошки та багато
іншого!</span></h4>
</i>
</div>

```

Блок контактної інформації містить адресу фактичного магазину, номер телефона директора, поштову адресу та посилання на завантаження додатку .

```

<div class="col-md">
<div class="ftco-footer-widgit mb-4">
<div class="block-23 mb-3">
<ul>
<li><a href="#"><span class="icon icon-star"></span><span class="text">г.Запорожжє ул.Правды
61а</span></a></li>
<li><a href="#"><span class="icon icon-phone"></span><span
class="text">+380990473895</span></a></li>
<li><a href="#"><span class="icon icon-envelope"></span><span
class="text">ArtHouse.com</span></a></li>

<a href="./assets/ArtHouse-debug.3948440.2.apk"> ArtHouse </a>
</ul>
</div>
</div>
</div>

```

Маню сайту включає в себе вибів категорії товарів, а саме: художні фарби, пензлі, полотна в рулоні, подарункові набори, розпис тканини, аксесуари художника.

Блок категорій товарів.

```

<div class="collapse navbar-collapse" id="ftco-nav">
<ul class="navbar-nav ml-auto">
<li class="nav-item"><router-link to="/about" class="nav-link">Головна</router-link></li>
<li class="nav-item"><router-link to="/all" class="nav-link">Товари</router-link></li>
<!-- <li class="nav-item"><a href="index.html" class="nav-link">Товари</a></li> -->
<li class="nav-item dropdown active">
<a class="nav-link dropdown-toggle" href="index.html" id="dropdown04" data-toggle="dropdown"
aria-haspopup="true" aria-expanded="false">Категорії</a>
<div class="dropdown-menu" aria-labelledby="dropdown04">
<router-link class="dropdown-item" to="/paints">Художні фарби</router-link>
<router-link class="dropdown-item" to="/brushes">Пензлі</router-link>
<router-link class="dropdown-item" to="/holstsinarul">Полотна в рулоні </router-link>
<router-link class="dropdown-item" to="/holst">Полотно на підрамнику </router-link>
<router-link class="dropdown-item" to="/presents">Подарункові набори </router-link>
<router-link class="dropdown-item" to="/painting">Розпис тканини </router-link>
<router-link class="dropdown-item" to="/Accessories">Акcesуари художника </router-link>
</div>

```

Метод сортування категорії працює таким чином, що при виборі певної категорії, в БД перевіряється `_id Category` і виводить конкретні товари які підходять під цю категорію

```

•
filter: function(){
let resultList = ''
resultList = this.tovars.filter((element) => {
switch(this.cat) {
case 'paints' : return element.Category == 'Краски художественные';
case 'brushes' : return element.Category == 'Кисти';
case 'holstsinarul' : return element.Category == 'Холсты в рулоне';
case 'holst' : return element.Category == 'Холст на подрамнике';
case 'presents' : return element.Category == 'Подарочные наборы';
case 'painting' : return element.Category == 'Роспись ткани';
case 'Accessories' : return element.Category == 'Аксессуары художника';
default: return element.Category != '';
}
});

```

Користувач також може вісортувати товари по ціні і в певній категорії, так і по всіх товарах.

Блок сортировки по ціні.

```
<div class="dropdown-menu" aria-labelledby="dropdown04">
  <router-link class="dropdown-item" :to="`/${currentCategory}/50`" >до 50 грн</router-link>
  <router-link class="dropdown-item" :to="`/${currentCategory}/200`" >від 50 грн до 200
  грн</router-link>
  <router-link class="dropdown-item" :to="`/${currentCategory}/500`" >від 200 грн до 500
  грн</router-link>
  <router-link class="dropdown-item" :to="`/${currentCategory}/1000`" >від 500 грн до 1000
  грн</router-link>
  <router-link class="dropdown-item" :to="`/${currentCategory}/2000`" >від 1000 грн до 2000
  грн</router-link>
  <router-link class="dropdown-item" :to="`/${currentCategory}/5000`" >від 2000 грн до 5000
  грн</router-link>
</div>
```

Метод сортировки по ціні працює таким чином, що при виборі певної цінового діапазону , в БД перевіряється `_id Price` і виводить конкретні товари які підходять під цей діапазон .

```
if (this.price > 0)
{
  resultList = resultList.filter((element) => {
    switch(this.price) {
      case '50' : return ((element.Price < Number(this.price)) && (element.Price > 0)); break;
      case '200' : return ((element.Price < Number(this.price)) && (element.Price > 50)); break;
      case '500' : return ((element.Price < Number(this.price)) && (element.Price > 200)); break;
      case '1000' : return ((element.Price < Number(this.price)) && (element.Price > 500)); break;
      case '2000' : return ((element.Price < Number(this.price)) && (element.Price > 1000)); break;
      case '5000' : return ((element.Price < Number(this.price)) && (element.Price > 2000)); break;
      default: return true;
    }
  })
}
```

Підключення до БД відбувається наступним чином.

```
const MongoClient = require("mongodb").MongoClient;
```

```

const url = "mongodb://*** /";
const mongoClient = new MongoClient(url);
async function run() {
  try {
    await mongoClient.connect();
    const db = mongoClient.db("admin");
    const result = await db.command({ ping: 1 });
    console.log("Підключено");
    console.log(result);
  } catch (err) {
    console.log("Виникла помилка");
    console.log(err);
  } finally {
    await mongoClient.close();
    console.log("Відключено");
  }
}
run().catch(console.error);

```

Інформація про назву, ціну, наявність в опис товару зберігається в JSON файлі

```

{
  id: 3,
  "Name": "Набір масляних фарб, Ладога, 8*18 мл ",
  "Price": "388",
  "Category": "Художні фарби",
  "photo": "assets/images/lo.jpg",
  "opis": "До складу набору масляних фарб Ладога входить 8 кольорів: білила цинкові (100), кадмій жовтий (201), крапlak червоний (317), ультрамарин (501), блакитна ФЦ (500), зелена ФЦ (703), умбра палена (408), газова сажка (801).",
  "opis1": "Наявність: На складі ",
  "opis2": "Набір у картонній коробці.",
  "opis3": "Виробництво: ЗХК 'Ладога'",
  "opis4": "Об'єм/вага: 18 мл ",
  "opis5": "Квітів: 8 кольорів ",
},

```

В ході роботи було реалізовані всі завдання для даного програмного продукту.

4 ТЕСТУВАННЯ ТА ЕКСПЛУАТАЦІЯ ПРОГРАМИ

4.1 Призначення і умови використання програми

Створюваний вебдодаток призначений для перегляду товарів наявних в магазині.

Вебдодаток створений в середовищі програмування Visual Studio Code.

База даних створена в СКБД MongoDB.

Робота з базою даних на сервері здійснюється за допомогою ODM Mongoose.

Мінімальні системні вимоги середовища (хостингу) для належного функціонування сервера вебдодатку:

- CPU: Intel Core i5;
- RAM: 2 Gb;
- HDD/SSD: 1Gb (мінімально, для початкового запуску), але при розростанні, збільшенні кількості користувачів, тем та відповідей, для зберігання бази даних форуму може знадобитися чималий дисковий простір;
- вебсервер: Apache 2 або Nginx;
- СКБД: MongoDB;
- порти: 80 (HTTP), 27018 (стандартний порт MongoDB);
- програмне забезпечення: середовище Node.js, менеджер пакетів NPM.

Створена система має розподілену архітектуру, тобто складається з клієнта та сервера, між якими йде обмін інформацією. Клієнт, на відміну від сайтів-аналогів, має зручний, неперевантажений інтерфейс, мінімалістичний дизайн та інтуїтивно зрозумілу навігацію. Сервер забезпечує стабільну роботу клієнта.

Сайт має наступні функції:

- зручна навігація по розділам і темам;
- можливість перегляду інформації про товари;
- перегляд товарів за різними фільтраціями.

Для виконання програми програмісту потрібно середовище Node.js, менеджер пакетів NPM та програмне забезпечення для роботи з Git'ом.

Після створення посилання на репозиторій з вихідним кодом програміст може скопіювати його в локальний репозиторій за допомогою консольної команди `git clone 'link_to_repository'` або візуального інтерфейсу, в залежності від програмного забезпечення, яке використовується для роботи з Git'ом на комп'ютері .

Папка з колекціями документів бази даних називається `tovars`, повинна розміщуватись на одному сервері з серверною частиною додатку.

Щоб запустити проект на локальному сервері для доопрацювання слід:

- за допомогою терміналу перейти до каталогу проекту, в якому міститься файл `package.json`;
- ввести команду `npm run start`;
- в терміналі з'явиться адресу порта, на якому працює потрібний сайт.

Для завантаження нових пакетів використовується команда `npm install «назва_пакету»`, для видалення — `npm uninstall «назва_пакету»`.

Скомпілювати проект для розміщення на хостингу можна командою `npm run build`, після чого всі необхідні файли з'являться в каталозі `dist`.

Зупиняється виконання проекту на локальному сервері комбінацією `ctrl + C` при фокусі на терміналі.

Вхідними даними для додатку є колекції в базі даних, що зберігають інформацію про всі теми, розділи, відповіді на форумі та користувачів. Вихідними даними є візуальне представлення цих даних на клієнтській частині додатку. Дані між клієнтом і сервером передаються в тілі HTTP запитів/відповідей в вигляді строки формату JSON .

Роутери з `index.js` отримують HTTP-запити від клієнту і пересилають їх на обробники `middlewares`, які перенаправляють дані у функції з `actions` для роботи з базою даних, використовуючи моделі, описані у каталозі `model`, визначені бібліотекою `Mongoose`. Інформація про користувачів, проекти, налаштування знаходиться в базі даних `MongoDB`. При необхідності

робиться перевірка вхідних даних перед роботою з БД за допомогою validator.

Програма реалізує наступні функції:

- перегляд основної інформації про характеристику товарів;
- не дає можливість користувачеві вводити нову інформацію в інтернет-каталог;
- конкретно інтерпретує і зберігає інформацію.
- адаптивність: сайт правильно відображається в різних браузерах, а також на мобільних пристроях;
- інтерфейс мінімалістичний, максимально простий та інтуїтивно зрозумілий.

Завантажувальний файл програми має назву ArtHouse.exe. База даних має назву ArtHouse.mdb.

Вхідними даними до програми є файл бази даних ArtHouse.mdb.

Вихідними даними програми являється візуальні екранні форми.

4.2 Тестування програми

Створений додаток призначений для перегляду клієнтом потрібної інформації про товар.

Для роботи програми потрібен персональний комп'ютер з доступом до Інтернету, а апаратна платформа повинна відповідати мінімальним системним вимогам, щоб браузер під управлінням будь-якої операційної системи міг користуватися сайтом.

Цей додаток повинен відповідати мінімальним системним вимогам до персонального комп'ютера з доступом до Інтернету, апаратної платформи під управлінням будь-якої операційної системи та браузера для використання сайту.

Для користування сайтом розробники переходять за відповідними посиланнями, після чого відкривається сторінка онлайн-каталогу(рисунок 4.1).



Рисунок 4.1 – Сторінка інтернет-каталогу

Після відкриття інтернет-каталогу, користувач може оглянути товари, які є у магазині «ArtHouse». Перш за все на сайті існує фільтрація за категорією, щоб користувачу було більш зручніше проглядати товари магазину (рис.4.2 – 4.6).

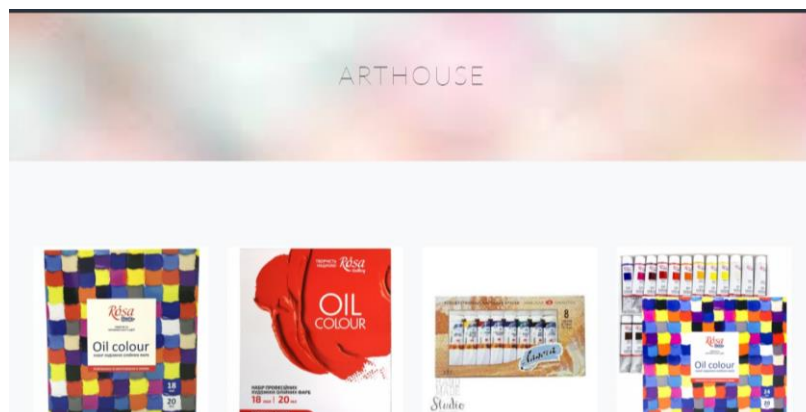


Рисунок 4.2 – Фільтрація «Художні фарби»

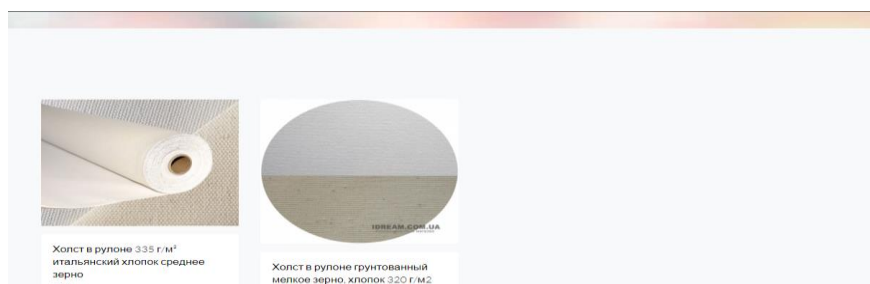


Рисунок 4.3 – Фільтрація «Полотна в рулоні»

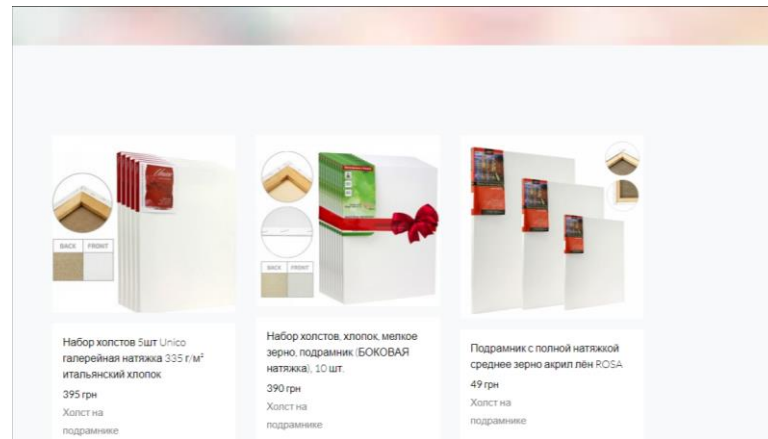


Рисунок 4.4 – Фільтрація «Полотно на підрамнику»

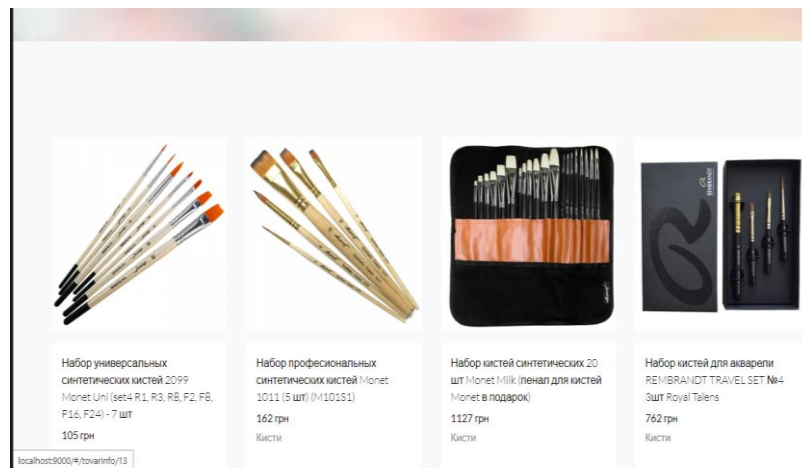


Рисунок 4.5– Фільтрація «Кисти»

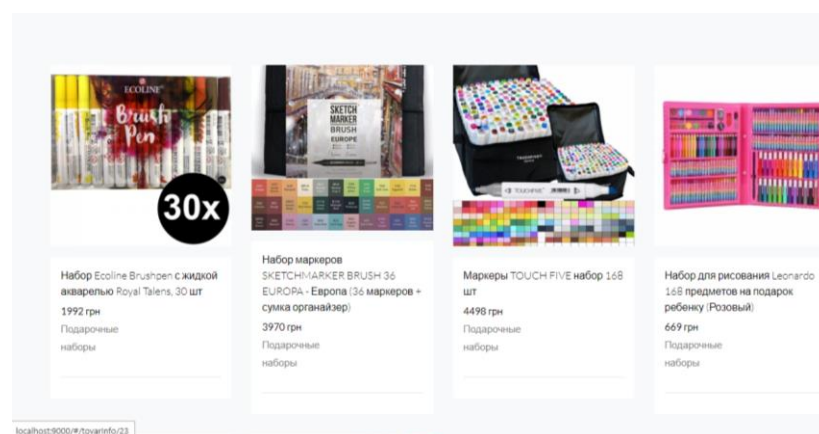


Рисунок 4.6– Фільтрація «Подарункові набори»

Ще на сайті представлення фільтрація за ціною(рисунок 4.7).Покупець може вибрати той діапазон ціни який йому потрібен.

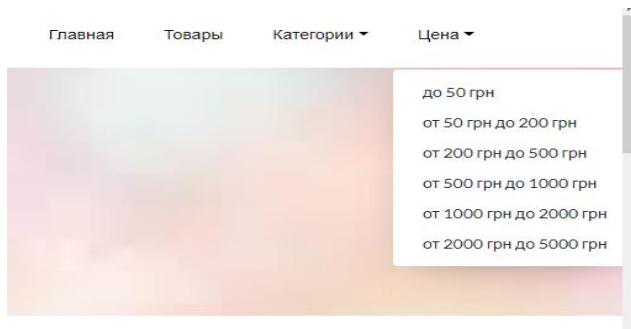


Рисунок 4.7– Фільтрація за ціною

На рисунку 4.8 зображено логотип магазину.



Рисунок 4.8 – Логотип магазину

На рисунках 4.9 - 4.11 представлено інтерфейс мобільного додатку

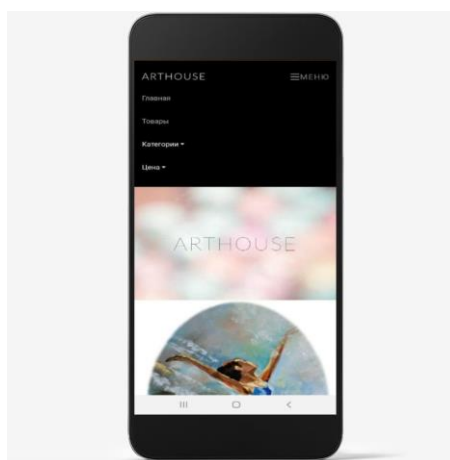


Рисунок 4.9 – Головне меню



Рисунок 4.10 – Сторінка товарів

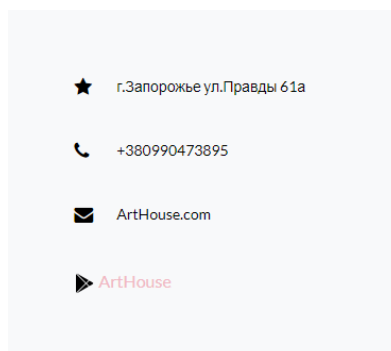


Рисунок 4.11 –Контактна інформація

Виконане тестування роботи каталогу та програми відбулось успішно.
Всі функції працюють коректно.

ВИСНОВКИ

При виконанні дипломної роботи була детально вивчена та проаналізована предметна область, яка стосується роботи з даними інтернет-каталогу.

Було проведено аналіз наявного програмного забезпечення, що пропонується на ринку, для розробки сайтів, які спеціалізуються на презентації товарів магазину покупцям в мережі інтернет.

У даній дипломній роботі розроблена інформаційна система для перегляду, пошуку даних, необхідних для роботи магазину товарів для творчості.

Програмне забезпечення було розроблено за допомогою JavaScript фреймворку Vue.js в середовищі Visual Studio Code.

Програмне забезпечення призначене для роботи на ПК та мобільних пристроях під управлінням Microsoft Windows 10. Програмне забезпечення має зручний інтерфейс, який створює можливість максимальної зручності для некваліфікованих кінцевих користувачів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Види сайтів / [Електронний ресурс]. – Режим доступу: <https://apri-code.com/uk/razrabotka/vidy-sajtov/>
2. Посібник: знайомство з React / [Електронний ресурс]. – Режим доступу: <https://uk.legacy.reactjs.org/tutorial/tutorial.html>
3. Angular vs Vue.js: що обрати / [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/columns/angular-vs-vuejs/>
4. The Progressive JavaScript Framework / [Electronic resource]. – Access mode: <https://vuejs.org/>
5. HTML5 – що це? / [Електронний ресурс]. – Режим доступу: <https://armedsoft.com/ua/blog/html5-shcho-ce>
6. CSS Підручник / [Електронний ресурс]. – Режим доступу: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>
7. What Is Javascript (JS)? / [Electronic resource]. – Access mode: <https://aws.amazon.com/ru/what-is/javascript/>
8. About Node.js / [Electronic resource]. – Access mode: <https://nodejs.org/uk/about>
9. Build the next big thing/ [Electronic resource]. – Access mode: <https://www.mongodb.com/>
10. Різниця між HTTP та HTTPS / [Електронний ресурс]. – Режим доступу: <https://hostiq.ua/wiki/ukr/http-https/>

ДОДАТОК А
Технічне завдання

Вступ

Сьогодні Інтернет є найпопулярнішою технологією для просування свого бізнесу. Так, кожен може легко відвідати ту чи іншу сторінку в Інтернеті, ввівши адресу сайту або натиснувши посилання.

Вебкаталог — це спосіб організації та категоризації гіперпосилань вашого сайту. Як приклад вебкаталог «e-Katalog» в якому представлено різні товари від побутових приладів до посуду. Покупець може знайти товар, який йому потрібен, і порівняти ціни і наявність в магазинах в міста, де він хоче зробити покупку.

A.1 Підстави для розробки

Розробка в межах дипломної кваліфікаційної роботи реалізується на підставі наказу № 87 від 4 квітня 2023 р. за Національним університетом «Запорізька політехніка» за темою «Програмна реалізація інтернет каталогу товарів для творчості».

A.2 Призначення розробки

Метою вебсайту та додатку є розробка онлайн-каталогу магазинів, що спеціалізуються на товарах для творчості

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Функції, які виконуватиме програма за результатами розробки

- перегляд основної інформації про характеристику товарів;
- не дає можливість користувачеві вводити нову інформацію в інтернет-каталог;
- конкретно інтерпретує і зберігає інформацію.

- адаптивність: сайт правильно відображається в різних браузерах, а також на мобільних пристроях;
- інтерфейс мінімалістичний, максимально простий та інтуїтивно зрозумілий.

A.3.2 Умови експлуатації

Для роботи програми потрібен персональний комп'ютер з доступом до Інтернету, апаратна платформа повинна працювати під управлінням будь-якої операційної системи і відповідати мінімальним системним вимогам браузера, який буде використовуватися на вебсайті.

A.3.3 Вимоги до складу та параметрів технічних засобів

Мінімальні системні вимоги середовища (хостингу) для належного функціонування сервера вебдодатку:

- CPU: Intel Core i5;
- RAM: 2 Gb;
- HDD/SSD: 1Gb (мінімально, для початкового запуску), але при розростанні, збільшенні кількості користувачів, тем та відповідей, для зберігання бази даних форуму може знадобитися чималий дисковий простір;

A.4 Порядок контролю та приймання

Остаточна перевірка отриманих результатів виконання роботи здійснюється під час захисту дипломної роботи.

ДОДАТОК Б
Текст програми

Index.html

```
<!DOCTYPE html>
<html>

<head>
<title>ArtHouse</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=Lato:100,300,400,700,900"
rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Roboto+Condensed:300,400,700"
rel="stylesheet">

<link rel="stylesheet" href="/assets/css/open-iconic-bootstrap.min.css">
<link rel="stylesheet" href="/assets/css/animate.css">

<link rel="stylesheet" href="/assets/css/owl.carousel.min.css">
<link rel="stylesheet" href="/assets/css/owl.theme.default.min.css">
<link rel="stylesheet" href="/assets/css/magnific-popup.css">

<link rel="stylesheet" href="/assets/css/aos.css">

<link rel="stylesheet" href="/assets/css/ionicons.min.css">

<link rel="stylesheet" href="/assets/css/bootstrap-datepicker.css">
<link rel="stylesheet" href="/assets/css/jquery.timepicker.css">

<link rel="stylesheet" href="/assets/css/flaticon.css">
<link rel="stylesheet" href="/assets/css/icomoon.css">
<link rel="stylesheet" href="/assets/css/style.css">
<script src="/assets/js/tovars.js"></script>
```

```
</head>
```

```
<body>
```

```
<div id="app"></div>
```

```
<section class="ftco-section-parallax" >
```

```
<div class="parallax-img d-flex align-items-center">
```

```
<div class="container">
```

```
<div class="row d-flex justify-content-center py-5">
```

```
<div class="col-md-7 text-center heading-section ">
```

```
<h1 class="big">ArtHouse</h1>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<footer class="ftco-footer bg-light ftco-section">
```

```
<div class="container">
```

```
<div class="row mb-5">
```

```
<div class="col-md">
```

```
<div class="ftco-footer-widget mb-4">
```

```
<h2 class="ftco-heading-2">ArtHouse</h2>
```

```
<ul class="ftco-footer-social list-unstyled float-md-left float-lft mt-5">
```

```
<li class=""><a href="#"><span class="icon-twitter"></span></a></li>
```

```
<li class=""><a href="#"><span class="icon-facebook"></span></a></li>
```

```
<li class=""><a href="#"><span class="icon-instagram"></span></a></li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<div class="col-md">
```

```
<div class="ftco-footer-widget mb-4 ml-md-5">
```

```
<h2 class="ftco-heading-2">Menu</h2>
```

```

<ul class="list-unstyled">
<li><a href="#" class="py-2 d-block">Shop</a></li>
<li><a href="#" class="py-2 d-block">About</a></li>
<li><a href="#" class="py-2 d-block">Journal</a></li>
<li><a href="#" class="py-2 d-block">Contact Us</a></li>
</ul>
</div>
</div>
<div class="col-md-4">
<div class="ftco-footer-widget mb-4">
<h2 class="ftco-heading-2">Help</h2>
<div class="d-flex">
<ul class="list-unstyled mr-l-5 pr-l-3 mr-4">
<li><a href="#" class="py-2 d-block">Shipping Information</a></li>
<li><a href="#" class="py-2 d-block">Returns & Exchange</a></li>
<li><a href="#" class="py-2 d-block">Terms & Conditions</a></li>
<li><a href="#" class="py-2 d-block">Privacy Policy</a></li>
</ul>
<ul class="list-unstyled">
<li><a href="#" class="py-2 d-block">FAQs</a></li>
<li><a href="#" class="py-2 d-block">Contact</a></li>
</ul>
</div>
</div>
</div>
<div class="col-md">
<div class="ftco-footer-widget mb-4">

<div class="block-23 mb-3">
<ul>

<li><a href="#"><span class="icon icon-phone"></span><span
class="text">+380990473895</span></a></li>
<li><a href="#"><span class="icon icon-envelope"></span><span
class="text">ArtHouse.com</span></a></li>
</ul>

```

```

</div>
</div>
</div>
</div>
<div class="row">
<div class="col-md-12 text-center">

</div>
</div>
</div>
</footer>
<div id="ftco-loader" class="show fullscreen"><svg class="circular" width="48px"
height="48px"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-
width="4" stroke="#eeeeee"/><circle class="path" cx="24" cy="24" r="22" fill="none"
stroke-width="4" stroke-miterlimit="10" stroke="#F96D00"/></svg></div>

</div>

</body>
<script src="./assets/js/jquery.min.js"></script>
<script src="./assets/js/jquery-migrate-3.0.1.min.js"></script>
<script src="./assets/js/popper.min.js"></script>
<script src="./assets/js/bootstrap.min.js"></script>
<script src="./assets/js/jquery.easing.1.3.js"></script>
<script src="./assets/js/jquery.waypoints.min.js"></script>
<script src="./assets/js/jquery.stellar.min.js"></script>
<script src="./assets/js/owl.carousel.min.js"></script>
<script src="./assets/js/jquery.magnific-popup.min.js"></script>
<script src="./assets/js/aos.js"></script>
<script src="./assets/js/jquery.animateNumber.min.js"></script>
<script src="./assets/js/bootstrap-datepicker.js"></script>
<script src="./assets/js/scrollax.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3MaE7SQ74_uJiUL
gl1JY0H2s&sensor=false"></script>
<script src="./assets/js/google-map.js"></script>

```

```
<script src="./assets/js/main.js"></script>  
<script src="./assets/js/tovars.js"></script>  
<script src="main.js"></script>  
</html>  
Tovars.vue  
  
<template>  
  <section class="ftco-section bg-light">  
    <div class="container-fluid">  
      <div class="row">  
        <div class="col-sm col-md-6 col-lg-3" v-for="item in filtertovars" v-bind:key="item.id">  
          <div class="product">  
            <router-link :to="/tovarInfo/'+item.id"></router-link>  
            <pre></pre>  
            <div class="text py-3 px-3" >  
              <h3>{{item.Name}} </h3>  
              <div class="d-flex">  
                <div class="pricing">  
                  <p class="price">{{item.Price}} грн</p>  
                  <p class="categori"><span class="mr-2 price-dc">{{item.Category}}</span></p>  
                </div>  
              </div>  
            <hr>  
          </div>  
        </div>  
      </div>  
    </section>  
  </template>
```

```

<script>
import Vue from 'vue'
import axios from 'axios'
import VueAxios from 'vue-axios'

export default {
  props: ['cat','price'],
  data: function() {
    return {
      id: 0,
      name: "",
      Category: "",
      tovars: clothes,

    };
  },
  mounted: function(){

    this.tovars = clothes;

  },
  computed: {
    filtertovars () {
      return this.filter();
    }
  },
  beforeRouteUpdate (to, from, next) {
    console.log('beforeRouteUpdate', to.params.price)
    this.filter();
    next()
  },
  methods: {
    filter: function(){

```

```

let resultList = ''
resultList = this.tovars.filter((element) => {
switch(this.cat) {
case 'paints' : return element.Category == 'Краски художественные';
case 'brushes' : return element.Category == 'Кисти';
case 'holstsirul' : return element.Category == 'Холсты в рулоне';
case 'holst' : return element.Category == 'Холст на подрамнике';
case 'presents' : return element.Category == 'Подарочные наборы';
default: return element.Category != '';
}
});
if (this.price > 0)
{
resultList = resultList.filter((element) => {
switch(this.price) {
case '50' : return ((element.Price < Number(this.price)) && (element.Price > 0)); break;
case '200' : return ((element.Price < Number(this.price)) && (element.Price > 50)); break;
case '500' : return ((element.Price < Number(this.price)) && (element.Price > 200)); break;
case '1000' : return ((element.Price < Number(this.price)) && (element.Price > 500));
break;
case '2000' : return ((element.Price < Number(this.price)) && (element.Price > 1000));
break;
default: return true;
}
});
}
return resultList;
},
},
}
</script>
<style scoped>
.fade-enter-active, .fade-leave-active {
transition: opacity .5s;
}
.fade-enter, .fade-leave-to {

```

```
opacity: 0;
```

```
}
```

```
</style>
```

```
Main.js
```

```
import Vue from 'vue'
```

```
import VueRouter from 'vue-router'
```

```
import App from './components/App.vue'
```

```
import Tovars from './components/Tovars.vue'
```

```
import About from './components/About.vue'
```

```
import TovarInfo from './components/tovarInfo.vue'
```

```
require('velocity-animate');
```

```
const routes = [
```

```
{ path: '/about', component: About},
```

```
{ path: '/tovarInfo/:id', component: TovarInfo, meta: { requiresAuth: true }, },
```

```
{ path: '/:cat?/:price?', component: Tovars, meta: { requiresAuth: true }, props: true },
```

```
]
```

```
const router = new VueRouter({
```

```
  routes
```

```
})
```

```
Vue.use(VueRouter)
```

```
new Vue({
```

```
  render: h => h(App),
```

```
  el: '#app',
```

```
  router
```

```
  // store
```

```
})
```

```
Store.js
```

```

import Vue from 'vue'
import Vuex from 'vuex'
import createPersistedState from 'vuex-persistedstate'

Vue.use(Vuex)

const store = new Vuex.Store({
  plugins: [createPersistedState()],
  state: {
    count: 0,
    user: null,
    currentstyle: ''
  },
  mutations: {
    setCount: (state, count) => state.count = count,
    setUser: (state, user) => state.user = user,
    setCurrentstyle: (state, currentstyle) => state.currentstyle = currentstyle,
  },
  getters: {
    getCount: (state) =>
    {
      return state.count
    },
    getUser: (state) =>
    {
      return state.user
    },
    getCurrentstyle: (state) =>
    {
      return state.currentstyle
    }
  }
})

export default store;
tovarinfo.vue
<template>

```

```
<div>
```

```
<pre></pre>
```

```

```

```
<div color="black" >
```

```
<h4>{{tovar.Name}}</h4>
```

```
<hr style=" border: none; border-top: 1px dotted black;">
```

```
<h5>
```

```
<p >{{tovar.opis1}}</p>
```

```
<hr style=" border: none; border-top: 1px dotted black;">
```

```
<p>{{tovar.opis}}</p>
```

```
<p >{{tovar.opis2}}</p>
```

```
<p >{{tovar.opis3}}</p>
```

```
<p >{{tovar.opis4}}</p>
```

```
<p >{{tovar.opis5}}</p> </h5>
```

```
<h4>Цена:{{tovar.Price}}грн</h4>
```

```
<
```

```
</div>
```

```
</div>
```

```
<!-- </div> -->
```

```
</template>
```

```
<script>
```

```
import tovars from '../assets/js/tovars.js'
```

```
import Vue from 'vue'
```

```
import axios from 'axios'
```

```
import VueAxios from 'vue-axios'
```

```
import modal from './_modal.vue'
```

```
Vue.use(VueAxios, axios)
```

```
export default {
  components: {
    modal
  },
  // props: {
  //   id: "",
  // },
  data: function() {
    return {
      tovar : {},
      showModal: false
    };
  },
  computed: {
    // getCount () {
    //   return this.$store.getters['getCount']
    // }
  },
  mounted: function(){
    //alert(clothes);
    for(let i = 0; i < clothes.length; i++){
      if(clothes[i].id == this.$route.params.id) this.tovar = clothes[i];
    }
    // tovars.forEach(item => {
    //   alert(item.id);
    //   alert(this.$route.params.id);
    //   if(item.id == this.$route.params.id) this.tovar = item;
    // })
    // let foundStudent = this.tovar.find((element) => {
    //   return element.id == id;
    // });
    // this.id = foundStudent.id;
    // this.name = foundStudent.name
```

```
},
```

```

methods: {
  submitFile() {
    let formData = new FormData();
    formData.append('file', this.file);
    formData.append(' tovar', this.tovar.id);
    console.log('>> formData >> ', formData);

    // You should have a server side REST API
    axios.post('http://localhost:9000/uploadphoto',
      formData, {
        headers: {
          'Content-Type': 'multipart/form-data'
        }
      })
      .then(function () {
        console.log('SUCCESS!!');
      })
      .catch(function () {
        console.log('FAILURE!!');
      });
  },
  handleFileUpload() {
    this.file = this.$refs.file.files[0];
    console.log('>>>> 1st element in files array >>>> ', this.file);
  }
}
}
}
</script>

```

App.vue

```
<template>
```

```
<div>
```

```
<nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark ftco-navbar-light
ftco-navbar-light-2" id="ftco-navbar">
  oi-menu"></span>Меню
</button>
```

```
<div class="collapse navbar-collapse" id="ftco-nav">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item"><router-link to="/about" class="nav-link">Главная</router-link
    ></li>
    <li class="nav-item"><router-link to="/all" class="nav-link">Товары</router-link></li>
    <!-- <li class="nav-item"><a href="index.html" class="nav-link">Товары</a></li> -->
    <li class="nav-item dropdown active">
      <a class="nav-link dropdown-toggle" href="index.html" id="dropdown04" data-
      toggle="dropdown" aria-haspopup="true" aria-expanded="false">Категории</a>
      <div class="dropdown-menu" aria-labelledby="dropdown04">
        >
```

```
<router-link class="dropdown-item" to="/holst">Холст на подрамнике</router-link>
<router-link class="dropdown-item" to="/presents">Подарочные наборы</router-link>
</div>
</li>
<li class="nav-item dropdown active">
  <a class="nav-link dropdown-toggle" id="dropdown04" data-toggle="dropdown" aria-
  haspopup="true" aria-expanded="false">Цена</a>
  <div class="dropdown-menu" aria-labelledby="dropdown04">
```

```
<router-link class="dropdown-item" :to="`/${currentCategory}/500`">от 200 грн до 500
грн</router-link>
<router-link
```

```
-image: url('../assets/images/zast1.jpg');">
<div class="container">
  <div class="row no-gutters slider-text align-items-center justify-content-center">
    <div class="col-md-9 text-center">
      <h1 class="mb-0 bread">ArtHouse</h1>
```

```
<!-- <p class="breadcrumbs"><span class="mr-2"><a
href="index.html">Товары</a></span></p> -->
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<router-view></router-view>
```

```
</div>
```

```
</template>
```

```
<script>
```

```
export default {
```

```
data () {
```

```
return {
```

```
  immediate: true,
```

```
  handler (to, from) {
```

```
    if (to.params && to.params.cat) {
```

```
      this.currentCategory = to.params.cat
```

```
    }
```

```
  }
```

```
}
```

```
}
```

```
}
```

```
</script>
```

```
Package.json
```

```
{
```

```
  "name": "first",
```

```
  "version": "1.0.0",
```

```
  "description": "",
```

```
  "scripts": {
```

```
    "start": "webpack-dev-server",
```

```
    "test": "echo \"Error: no test specified\" && exit 1"
```

```
  },
```

```

"author": "",
"license": "ISC",
"devDependencies": {
  "webpack": "^4.25.1",
  "webpack-dev-server": "^3.1.10"
},
"dependencies": {
  "axios": "^0.18.0",
  "css-loader": "^1.0.1",
  "img-loader": "^3.0.1",
  "multer": "^1.4.1",
  "style-loader": "^0.23.1",
  "velocity-animate": "^1.5.2",
  "vue": "^2.5.17",
  "vue-animate-number": "^0.4.2",
  "vue-axios": "^2.1.4",
  "vue-loader": "^15.4.2",
  "vue-router": "^3.0.1",
  "vue-template-compiler": "^2.5.17",
  "vue-wysiwyg": "^1.7.2",
  "vuex": "^3.0.1",
  "vuex-persistedstate": "^2.5.4",
  "webpack-cli": "^3.1.2"
}
}
this.file = this.$refs.file.files[0];
console.log('>>>> 1st element in files array >>>> ', this.file);
}
}
}
</script>

```

App.vue

<template>

<div>

```
<nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark ftco-navbar-light  
ftco-navbar-light-2" id="ftco-navbar">  
oi-menu"></span>Меню  
</button>
```

```
<div class="collapse navbar-collapse" id="ftco-nav">  
<ul class="navbar-nav ml-auto">  
<li class="nav-item"><router-link to="/about" class
```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ КАТАЛОГУ ТОВАРІ ДЛЯ ТВОРЧОСТІ

Дипломна кваліфікаційна робота

Виконала: студентка групи КНТ-120сп

Анастасія ДОН

Керівник:

Тетяна ГОЛУБ

2023

Рисунок В.1 – Слайд № 1

Об'єкт, предмет, мета роботи

- Об'єктом роботи є перегляд товарів для творчості.
- Предметом роботи є програмне забезпечення для перегляду товарів для творчості.
- Метою створення додатку є розробка програмного продукту, призначеного для перегляду товарів у магазині.

Рисунок В.2 – Слайд № 2

Завдання роботи

- аналіз предметної області;
- визначення вимог до розробки;
- проектування програмної реалізації інтернет каталогу товарів для творчості
- програмна реалізація інтернет каталогу товарів для творчості



Рисунок В.3 – Слайд № 3

Аналіз предметної області

Категорії	Візитна картка	Корпоративний	Інтернет-магазин	Інтернет-каталог
Індивідуальний дизайн	+	+	+	+
Форма зворотного зв'язку	+	+	+	+
Пошук по сайту	-	+	+	+
Каталог товарів	-	-	+	+
Сторінка товару	-	-	+	+
Пошук по каталогу	-	-	+	+
Фотогалерея	-	-	+	+
Корзина	-	-	+	-

Рисунок В.4 – Слайд № 4

Вибір засобів розробки

Функції	Vue.js	React	Angular
Синтаксис	Простий у викорис- танні та зрозумілий	JSX (JavaScript XML) для опису компонентів.	HTML-подібний шаблонний синтаксис.
Розмір	Невеликий розмір, що дозволяє швидку завантаження	Невеликий розмір, можливість додатково оптимізувати розмір з допомогою Webpack	Більший розмір, особливо залежно від використання додаткових модулів
Продуктивні- сть	Швидка відповідь завдяки прямому зв'язку з DOM.	Швидка відповідь завдяки використанню віртуального DOM.	Швидка відповідь завдяки використанню віртуального DOM

Рисунок В.5 – Слайд № 5

Документно-орієнтована модель бази даних

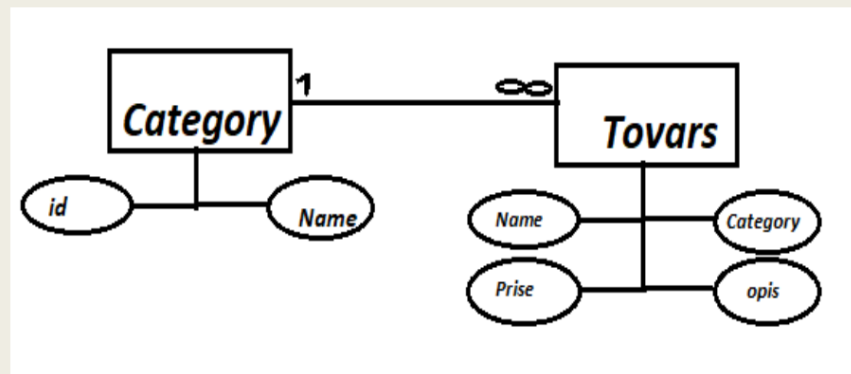


Рисунок В.6 – Слайд № 6

Опис документа «Tovar»

Назва ключа	Тип значення	Опис
id	String	Забезпечує унікальність товару
Name	String	Назва товару
Price	String	Ціна товару
opis	String	Опис товару

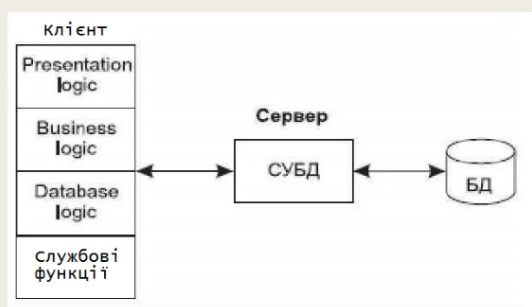
Опис документа «Categories»

Назва ключа	Тип значення	Опис
id	String	Забезпечує унікальність товару
type	String	Тип товару

Рисунок В.7 – Слайд № 7

Структурна схема програми

Структурна схема сервера



Взаємодія клієнта і сервера

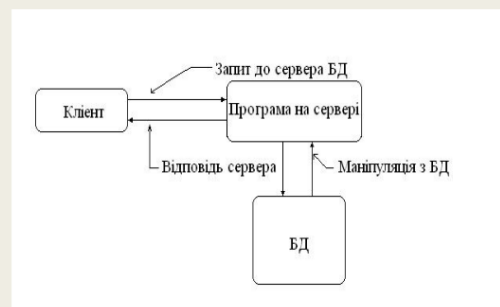


Рисунок В.8 – Слайд № 8

Екранні форми



Рисунок В.9 – Слайд № 9

Екранні форми

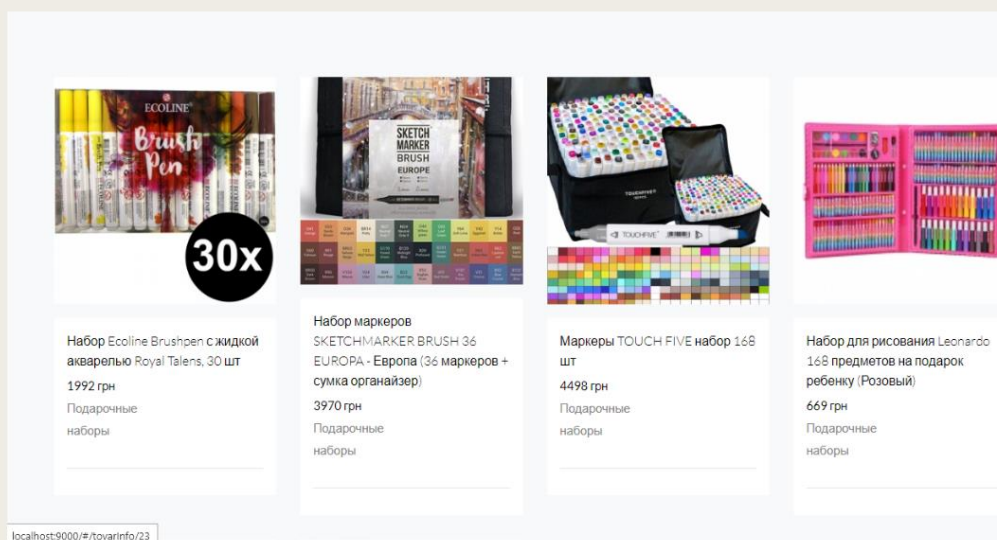


Рисунок В.10 – Слайд № 10

Екранні форми

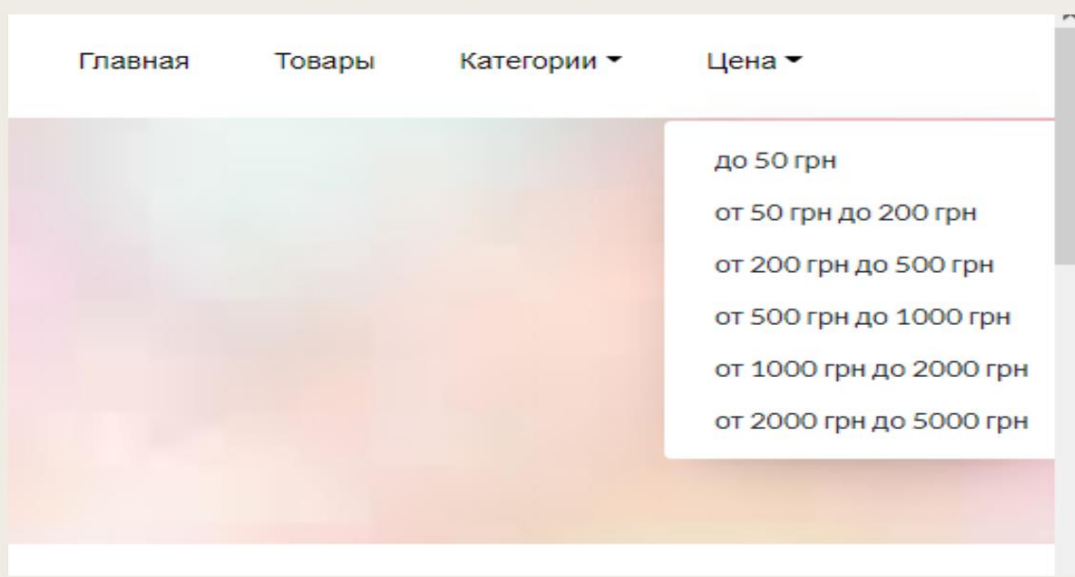


Рисунок В.11 – Слайд № 11

Екранні форми

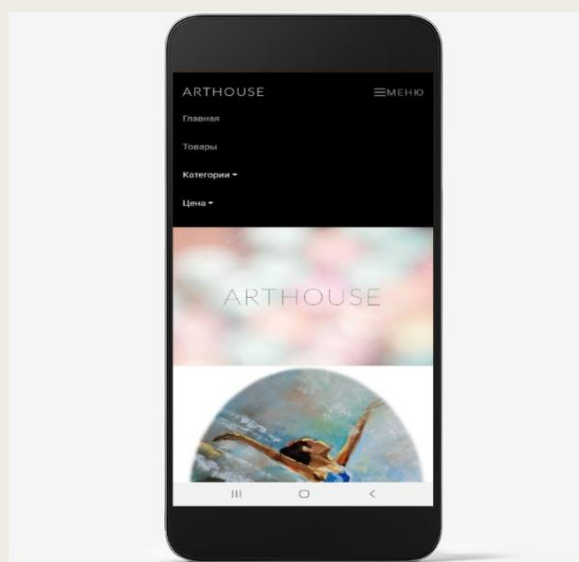


Рисунок В.12 – Слайд № 12

Висновки

У даній дипломній роботі розроблена інформаційна система для перегляду, пошуку даних, необхідних для роботи магазину товарів для творчості . Програмне забезпечення розроблене у середовищі Visual Studio Code за допомогою фреймворка JavaScript Vue.js. Програма працює на ПК під керуванням операційних систем Microsoft Windows 10, а також на будь-якому мобільному пристрої. Програма має зручний, максимально орієнтований на некваліфікованого кінцевого користувача інтерфейс.

Рисунок В.13 – Слайд № 13