

РЕФЕРАТ

Кваліфікаційна робота на здобуття освітнього ступеню «бакалавр» за спеціальністю 121 – Інженерія програмного забезпечення. Тернопільський національний технічний університет ім. Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПс-43, 2023 рік. Пояснювальна записка до кваліфікаційної роботи на здобуття освітнього ступеню «бакалавр» містить: 59 с., 23 рис., 8 табл., 4 додатка.

Об'єкт дослідження – інформація про товар та послуги меблевого інтернет магазину.

Мета роботи – автоматизація торгівлі меблевою продукцією з метою покращення обслуговування потенційних покупців; полегшення пошуку, редагування, введення нових даних, їх сортування та виведення інформації за умовою тощо.

Метод дослідження – це опис предметної області через опрацювання та аналіз потреб та документації меблевого підприємства та магазину; проектування бази даних із використанням методів нормалізації; написання і відлагодження сайту із клієнтською та адміністративною частиною.

Отримані результати – розроблено та протестовано сайт інтернет магазину для продажу меблевої продукції.

Меблі, інтернет магазин, база даних, HTML, PHP, JS, сортування, пошук, реєстрація.

ANNOTATION

Qualification work for obtaining an educational degree "bachelor" in specialty 121 - Engineering of program support. Ternopil National Technical University. Ivan Puluj, Faculty of Computer and Information Systems and Gram Engineering, Department of Gram Engineering, group CPs-43, 2023. The explanatory note to the qualification work for the educational degree "bachelor" contains: 59 p.23 pic. 8 table. 4 applications.

The object of research is information about the goods and services of the furniture online store.

The purpose of the work is to automate the trade in furniture products in order to improve the service of potential buyers; facilitating search, editing, entering new data, sorting them and displaying information on condition, etc.

Research method – description of the subject area by working out and analyzing the needs and documentation of the furniture company and store; database design using normalization methods; writing and debugging a site with client and administrative parts.

The results obtained – the website of an online store for the sale of furniture products was developed and tested.

Furniture, online store, database, HTML, PHP, JS, sorting, search, registration.

ЗМІСТ

РЕФЕРАТ	4
ANNOTATION.....	5
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
1 АНАЛІТИЧНА ЧАСТИНА	9
1.1 Опис об'єкта автоматизації.....	9
1.2 Аналіз подібних проектних рішень.....	10
2 ПРОЕКТНА ЧАСТИНА	12
2.1 Постановка завдання.....	12
2.2 Проектування бази даних.....	12
2.3 Проектування інформаційної системи.....	17
3 ПРАКТИЧНА ЧАСТИНА	22
3.1 Проектування інтерфейсу користувача	22
3.2 Опис програмних модулів.....	29
3.3 Опис результатів тестування	32
4 БЕЗПЕКА ЖИТТЕДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	41
4.1 Ергономічний та дизайнерський аналіз верстату, установки, обладнання.....	41
4.2 Естетичне оформлення та ергономічне дослідження робочого місця	42
ВИСНОВКИ.....	45
ПЕРЕЛІК ПОСИЛАНЬ	46
ДОДАТКИ.....	49
Додаток А.....	50
Додаток Б.....	51
Додаток В	56
Додаток Г	59

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – програмне забезпечення

ОС – операційна система

БД – база даних

СУБД – система управління базами даних

PHP – Hypertext Preprocessor

HTML – Hyper Text Markup Language

CSS – Cascading Style Sheets

JS – JavaScript

ІК – інтерфейс користувача

АІС – автоматизована інформаційна система

ВСТУП

У теперішній час виробляти меблеву продукцію стало значно простіше. І велика кількість новостворених та класичних підприємств стикається із проблемою збуту своєї продукції. Для фабрики дрібносерійного виробництва основною можливістю конкурувати з гігантами ринку є інтернет-комерція, яка може зрівняти шанси на отримання замовлень. Саме тому, перенесення продажів у мережу Інтернет є актуальним та доцільним.

Метою проекту є створення інтернет магазину для продажу меблевої продукції. Розроблений веб сайт повинен надавати відвідувачам можливість переглядати та купляти потрібні їм товари, а також дозволяти адміністраторам зручно керувати замовленнями, переліком виробів та їх категоріями тощо. У відповідності до вимог та наявної інформації потрібно створити базу даних для обслуговування інтернет-ресурсу, забезпечити швидкість, зручність роботи та захист інформації.

У першому розділі проекту зроблено описано безпосередньо об'єкт автоматизації, виділено головні завдання, що стоять перед мережевою торгівлею; проведено аналіз подібних проектних рішень, проаналізовано їх сильні та слабкі сторони, зроблено відповідні висновки щодо змісту та оформлення майбутнього сайту.

У другому розділі поставлено завдання, спроектовано базу даних та відповідну інформаційну систему.

У третьому розділі описано процес проектування інтерфейсу користувача, роботу програмних модулів та наведено деякі результати тестування сайту різними методиками.

Останній розділ містить деякі вказівки щодо охорони праці та створення ергономічного робочого місця потенційних користувачів.

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Опис об'єкта автоматизації

Головна мета будь-якого підприємства – це заробіток і чим він більший – тим більше можливостей для подальшого розвитку. У зв'язку з тим, що сьогодні значна більшість людей може собі дозволити електронні прилади, які дозволяють вільно користуватися інтернетом у будь-який момент та з будь-якого місця, відповідно росте попит на створення інтернет сторінок для взаємодії підприємства із користувачами.

Основною перевагою інтернет сторінок є економія часу. Швидше за все кожен стикався з ситуацією, коли необхідно витратити час, щоб дістатися до потрібного місця, знайти консультанта чи відповідальну особу, дочекатися своєї черги. Це все – витрати часу, якого при сьогоднішньому ритмі життя і так не вистачає.

Проектування та написання веб сайтів для відкритої онлайн торгівлі – це один з найкращих способів заробітку, який використовують більшість сучасних торговельних мереж та й окремих магазинів. Використання таких сайтів – це і хороша реклама, і значна економія коштів, які мали б бути виділені на утримання персоналу у торгових залах та на складах; оренду приміщень тощо. Отже, вартість товару в інтернет магазинах, зазвичай, нижча, що дозволяє збільшити товарооборот, і, відповідно, прибуток.

Інтернет магазини дозволяють користувачу переглянути весь доступний каталог товарів, які є у наявності на складах магазину. Щоб полегшити пошук необхідного товару доступна низка способів фільтрації та різного вигляду панелі пошуку. Головне для споживача – знайти товар за необхідними критеріями. Ними можуть бути назва товару, ціна, колірна гама, матеріал, з якого товар виготовлено тощо.

Більшість інтернет магазинів для представлення каталогів товару виводять їх «сіткою» [1–3]. При такому способі всі товари розбиваються на блоки, та

виводяться один за одним. Кожен блок, зазвичай, складається з наступних компонент: зображення товару, заголовок, ціна, іноді кнопка для здійснення купівлі. При переході на сторінку товару користувачу надається зовсім інше представлення товару – залишаються старі компоненти, але також з'являються нові: опис товару, додаткові зображення та інші.

Перш за все користувач помічає об'єкти, які рухаються. Через цей людський фактор доцільно створити певну компоненту, яка буде прокручуватися автоматично – привертаючи тим самим увагу відвідувача сторінки. У бокових панелях часто можна помітити стрічки, які прокручуються методом який називається «Слайдер». У таких стрічках має міститися інформація, яка повинна зацікавити потенційного відвідувача сайту.

Інтернет ресурси дозволяють користувачам реєструватися у базі даних сайту, подібна функція дозволяє збільшити доступний функціонал користувачеві. Також при реєстрації, адміністратор може надавати певні привілеї тим чи іншим користувачам за їх унікальним логіном.

Необхідним елементом також є форма зворотнього зв'язку. Завдяки ній користувачі можуть залишати свої відгуки про товар, або ж отримати певну консультацію з експлуатації, або ж повернення товару назад.

У дипломному проекті буде створено сайт для інтернет магазину продажу меблів.

1.2 Аналіз подібних проектних рішень

Станом на сьогодні у мережі Інтернет є безліч інтернет магазинів, які активно відвідуються та використовуються покупцями. Подібна кількість сприяє хорошій конкуренції між магазинами. Для того, хто планує створити власний сайт, такі сторінки є джерелом для аналізу позитивних та негативних сторін програмного продукту.

Розглянемо подібні проектні рішення, що реалізують інтернет магазини.

Інтернет магазин «Rozetka» (<https://rozetka.com.ua/>) [1] представляє широкий асортимент товарів. Головна сторінка містить слайдер з новинами, а також надає можливість пошуку товару за широким спектром фільтрації.

Інтерфейс сайту є зручним. Шукати необхідні товари легко і на це затрачається мінімальна кількість часу. Головною перевагою інтернет магазину «Розетка» є його зрозумілість та зручність у використанні для будь-якого користувача. Особливих недоліків сайт не має і вважається одним із кращих в своєму роді.

Сайт створений за допомогою відомої Content management system (системи керування контентом) – WordPress.

Інтернет магазин «Комфі» (<https://comfy.ua/>) [2] аналогічно надає користувачу доступ до широкої палітри асортименту товарів. Також доступна панель для зручного пошуку в асортименті.

Дизайн сайту складається з доволі приємних кольорів: зелений, помаранчевий та білий. Стиль дизайну – мінімалізм. Більша частина наповнення знаходиться в контейнерах, які просто обгорнуті рамками. Дизайн хоч і не має явних проблем, але і не є сильною стороною сайту.

Головною перевагою є широкий асортимент побутової техніки, а недоліком – зовнішній вигляд. Сайт написано за допомогою HTML5, CSS3, JavaScript та PHP.

Інтернет магазин «Фотос» (<https://f.ua/>) [3] дозволяє користувачеві переглянути асортимент доступних товарів має зручне меню навігації. Зовнішній вигляд сайту нагадує вище згаданий інтернет магазин «Розетка». Різниця між цими сайтами лише в кольоровій палітрі використаній для елементів сторінки.

Сильною стороною сайту є також великий асортимент товарів і непогані можливості для сортування товарів за певними критеріями. Інтернет магазин «Фотос» має всі необхідні елементи, які необхідні для продажу товарів.

Як і вищеописані, сайт створений за допомогою CMS WordPress, JavaScript, CSS3, HTML5. Недоліком є неоригінальність.

2 ПРОЕКТНА ЧАСТИНА

2.1 Постановка завдання

Для організації роботи зі збуту меблевої продукції; покращення взаємодії між потенційними покупцями та представниками торгівлі та виведення діяльності у всесвітню мережу, розробити сайт «Інтернет магазин для продажу меблевої продукції».

Вхідними даними додатку є інформація про товар, новини, дані кожного користувача; вихідні дані аналогічні, проте до вихідної інформації додається формування накладних та інших супутніх документів щодо здійснення купівлі товару користувачем.

На сайті необхідно організувати систему адміністрування та передбачити ролі користувачам сайту.

Веб-додаток має виводити перелік товарів у зручному вигляді, виконувати їх пошук та фільтрацію. Робота сайту повинна забезпечувати виконання наступних функцій:

- пошук, фільтрацію, виведення даних;
- можливість здійснення покупки товару;
- панель адміністратора для ведення сайту.

Передбачити забезпечення цілісності та коректності введених даних методами автоматичного внесення даних та їх перевірки, застосування атрибутів автоінкременту тощо.

2.2 Проектування бази даних

Предметною областю є автоматизація процесу продажу меблевих товарів. Цей проект зручний та потрібний для сфери торгівлі, оскільки він значно

полегшує та пришвидшує процес здійснення замовлення. Інтернет магазин меблевої продукції не обов'язково мусить бути вузькоспеціалізованим лише на продаж меблів. Профільність магазину легко змінюється введенням в базу необхідних товарів та змін категорій, що відповідають іншій сфері.

База даних буде використовуватися для обліку товарів, які присутні на складі.

Використовувати базу даних буде адміністратор сайту, який матиме змогу безпосередньо вилучати товар з бази, додавати його, змінювати категорії тощо. Він матиме доступ до усієї бази даних.

Користувачі матимуть доступ лише до функцій сортування, фільтрування та пошуку даних, а також можливість здійснювати замовлення товару.

Після проведення аналізу предметної області ми визначимо основні інформаційні об'єкти, які називаються сутностями, вони потрібні для проектування бази даних вебсайту. Після виділення кожної сутності їй присвоюємо ім'я, яке повинно наштовхувати на особливості використання сутності, та бути зрозумілим для користувача.

Документування вибраних сутностей полягає у підготовці детального визначення кожної сутності, а також існуючі для неї псевдоніми і опис особливостей використання.

Вся інформація про типи сутностей бази даних, псевдоніми та особливості використання, їх опис представлено у таблиці 2.1 [4, 5, 8].

Таблиця 2.1 – Опис сутностей бази даних «db_shop»

Ім'я сутності	Опис	Псевдоніми	Особливості використання
1	2	3	4
Кошик	Дані про товар користувача, які додані до кошику	card	Використовують користувачі сайту
Категорії	Список категорій товару, який присутній на сайті	category	Використовується для прив'язки категорії товару
Товар	Список товарів для продажу	table_products	Використовується для виведення інформації про товар на сторінку сайту
Користувачі	Список користувачів	users	Зберігає дані про користувачів сайту

Для зберігання даних у програмі створено базу даних «db_shop». Для роботи з базою даних у проекті було використано методи підключення PHP до бази даних MySQL [16, 17]. Вони дозволяють працювати з базою даних за допомогою стандартних класів бібліотек та інструментів класів PHP-MySQL і мови SQL.

База даних містить чотири таблиці. Таблиця «cart» містить інформацію про товар, який додав до кошика клієнт. Замовлення товарів буде здійснюватися із використанням даних з цієї таблиці. Загальна структура таблиці наведено у таблиці 2.2.

Таблиця 2.2 – Структура таблиці «card»

№	Поле	Тип	Атрибути	Опис
1	cart_id	Число (int)	Унікальний, ключ	Ідентифікатор запису
2	cart_id_products	Число (int)	не порожній	Ідентифікатор товару в корзині
3	cart_price	Число (int)	не порожній	Ціна
4	cart_count	Число (int)	не порожній	Кількість товару
5	cart_datatime	Дата (date)	не порожній	Дата додавання товару до корзини
6	cart_ip	Стрічка (varchar)	Не порожній	Дані користувача

Таблиця «categorija» містить всі категорії, які потрібні для здійснення сортування товару. Структура таблиці наведена у таблиці 2.3.

Таблиця 2.3 – Структура таблиці «categorija»

№	Поле	Тип	Розмір	Опис
1	id	Число (int)	Унікальний, не порожній	Ідентифікатор запису
2	type	Стрічка (varchar)	не порожній	Назва яка ідентифікує категорію

Структура таблиці «table_products» наведена у таблиці 2.4.

Таблиця 2.4 – Структура таблиці «table_products».

№	Поле	Тип	Розмір	Опис
1	2	3	4	5
1	products_id	Число (Int)	Унікальний, не порожній	Ідентифікатор запису
2	title	Стрічка (varchar)	Не порожній	Назва товару
3	price	Число (Int)	Не порожній	Ціна товару
4	brand	Стрічка (Varchar)	Не порожній	Категорія товару
5	mini_description	Стрічка (Text)	Не порожній	Міні-опис товару
6	image	Стрічка (Varchar)	Не порожній	Посилання на зображення
7	description	Стрічка (Text)	Не порожній	Повний опис товару
8	datetime	Дата і час (Datetime)	Не порожній	Дата додавання товару
9	new	Число (Int)	Не порожній	Чи товар є новинкою
10	leader	Число (Int)	Не порожній	Чи товар є лідером продажу
11	sale	Число (Int)	Не порожній	Чи товар в категорії «Розпродаж»
12	visible	Число (Int)	Не порожній	Видимість товару
13	count	Число (Int)	Не порожній	Кількість на складі
14	tupe_tovara	Стрічка (Varchar)	Порожній	Тип товару
15	brand_id	Число (Int)	Порожній	Ідентифікація бренду
16	vote	Число (Int)	Не порожній	Кі-сть голосів
17	votes	Число (Float)	Не порожній	Оцінка

Таблиця «users» зберігає дані про користувачів для їх ідентифікації та верифікації на сайті. Структура таблиць наведена у таблиці 2.5.

Таблиця 2.5 – Структура таблиці «users»

№	Поле	Тип	Атрибути	Опис
1	User_id	Число (int)	Унікальний, не порожній	Ідентифікатор запису
2	login	Стрічка (varchar)	не порожній	e-mail адреса користувача
3	password	Стрічка (varchar)	не порожній	Пароль

При створенні бази необхідно унеможливити помилки, пов'язані з внутрішньою її організацією. Для цього в таблицях створено ключове поле – унікальний ідентифікатор «id». Завдяки цьому є можливість ідентифікувати кожен запис окремо для здійснення додавання, видалення чи оновлення.

Для ідентифікатора задано атрибут “AutoIncrement”, що дозволяє спростити використання бази даних, ізолюючи користувача від задання унікальних ідентифікаторів для кортежів [7, 8].

У реляційній базі даних зв'язки дозволяють уникнути повторення даних. Структура зав'язків наведена у таблиці 2.6, а діаграма «Сутність-зв'язок» у додатку А.

Таблиця 2.6 – Структура зв'язків таблиць у базі даних «db_shop»

Назва таблиці 1	Назва зв'язку	Назва таблиці 2	Тип зв'язку
category	має	type_tovar	1:М
cart	має	user	1:М

Під час проектування та створення бази даних виникає необхідність створення такої схеми, яка б містила таблиці та деяку задану кількість атрибутів певного типу і одночасно мінімізувала дублювання даних цих таблиць, реалізувала можливість спрощення процедури опрацювання та оновлення даних в таблицях. Щоб виконати таке завдання було створено процес нормалізації початкових відношень. Результатом такого використання є перетворення до першої, другої, третьої форм будь якого початкового відношення [4]. Атрибути усіх відношень прості і мають лише одне значення, тому відношення знаходиться у першій нормальній формі. Зауважимо, що вони приведені до другої нормальної форми, оскільки знаходяться в першій нормальній формі і кожний неключовий атрибут функціонально-повно залежить від ключа. Відношення нормалізовані до третьої нормальної форми, бо усі їх неключові атрибути нетранзитивно залежать від первинного ключа.

Створення усіх таблиць відбувається одним запитом (код запису зображено на рисунку 2.1):

```

CREATE TABLE [Users](
[Id] INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
[login] NVARCHAR (50) NOT NULL,
[password] NVARCHAR (50) NOT NULL)
GO
CREATE TABLE [category](
[Id] INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
[type] NVARCHAR (50) NULL,
[brand] INT NOT NULL)
Go
CREATE TABLE [cart](
[Id] INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
[cart_id_products] int NOT NULL,
[cart_price] int NOT NULL,
[cart_count] int NOT NULL,
[cart_datatime] date NOT NULL,
[cart_ip] int NOT NULL)
Go
CREATE TABLE [table_products](
[Id] INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
[title] varchar (255) NOT NULL,
[price] int NOT NULL,
[brand] varchar (255) NOT NULL,
[image] varchar (255) NOT NULL,
[description] varchar (255) NOT NULL)

```

Рисунок 2.1 – Код створення таблиць бази даниї «db_shop»

2.3 Проектування інформаційної системи

Одним із найважливіших етапів при створенні програмного забезпечення є проектування. Хорошим помічником у проектуванні інформаційної системи є уніфікована мова моделювання. Проектування інтернет магазину продажу меблевої продукції здійснено з Unified Modeling Language (уніфікована мова моделювання – вона ж UML). UML може застосовуватися для будь-якого етапу життєвого циклу розробки прикладних програм [9].

UML є широко застосовується, це стандарт, що за допомогою графічних позначень створює абстрактної моделі системи, вони називається UML-моделлю. UML був створений для визначення та візуалізації, проектування, а також документування в основному програмних систем.

Для проектування інформаційної системи, потрібно побудувати діаграму прецедентів (Use case Diagram), послідовності (Sequence Diagram), та діаграму класів (Class Diagram).

Діаграма прецедентів – діаграма, яка показує відношення в системі між акторами та прецедентами. Суть даної діаграми: проєктована система представляється у вигляді необмеженої кількості сутностей чи акторів, які взаємодіють із системою за допомогою так званих варіантів використання [10].

Для створення діаграми прецедентів інтернет магазину продажу меблевої продукції визначено три види акторів «Адміністратор» «Користувач», «Авторизований користувач». Головною функцією для акторів «Користувач» та «Авторизований користувач» є можливість використання навігаційного меню на усіх сторінках сайту, на яких він може переглядати інформацію та використовувати функцію пошуку.

Для актора «Авторизований користувач» є доступ до написання коментарів під будь який доступний товар на сайті, а також можливість оцінити його. Актор «Адміністратор» має абсолютний доступ до джерела сайту та бази даних, що дозволяє йому керувати наповненням сайту та інформацією в базі даних [10, 11]. Діаграму прецедентів для інтернет магазину продажу меблевої продукції наведено на рисунку 2.2.

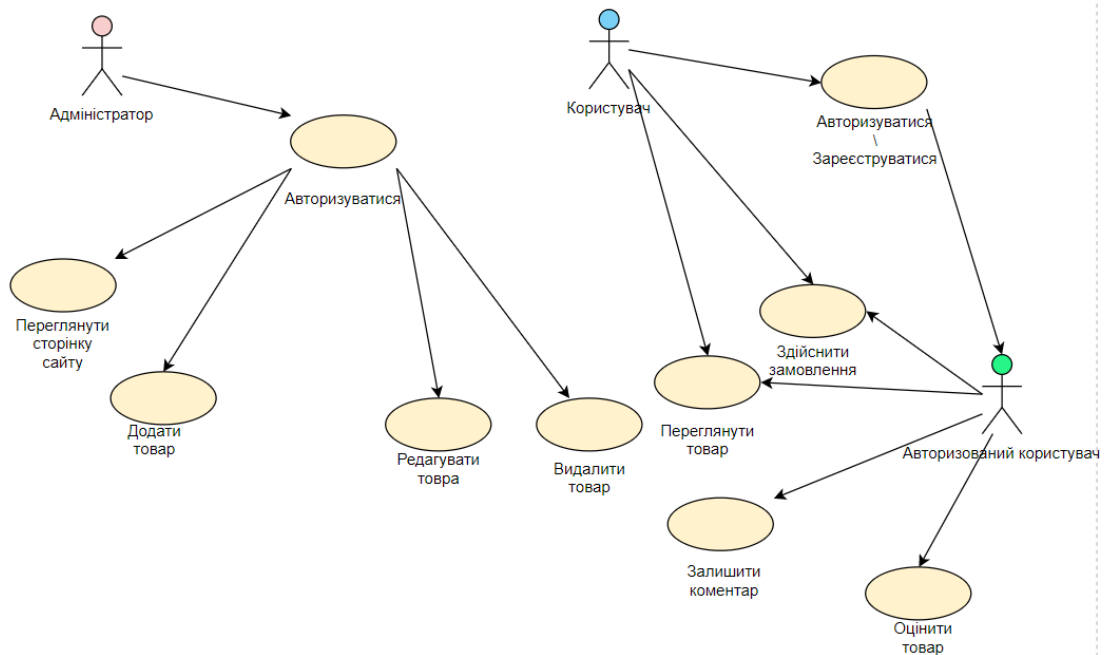


Рисунок 2.2 – Діаграма прецедентів інтернет магазину продажу меблевої продукції

Після створення діаграми прецедентів було побудовано дівграму класів, яка складається з класів прямокутних форм та зв'язків між ними. Класи відображаються у вигляді прямокутників, які можуть бути поділені на дві або три частини. Верхня частина прямокутника містить ім'я класу, друга частина містить список атрибутів, а третя частина – список операцій класу. Друга і третя частини прямокутника можуть бути пропущені. Діаграма класів використовується для представлення статичної структури системної моделі у термінах класів об'єктно-орієнтованого програмування.

Діаграма класів є графом, де вершини представлені елементами типу «класифікатор», які взаємодіють за допомогою різних типів структурних зв'язків. Важливо зазначити, що діаграма класів може також містити інтерфейсти, пакети, відносини, а також окремі екземпляри, такі як об'єкти та зв'язки. Ця діаграма є графічним відображенням структурних зв'язків органічної моделі системи, які залишаються незмінними незалежно від часу.

Користувач розпочинає взаємодію з сайтом з головної сторінки, яка взаємодіє з базою даних, підключаючи її до сайту. Інформація, яка відображається

на деяких сторінках сайту, отримується з бази даних. База даних взаємодіє з усіма основними сторінками сайту, постачаючи їм необхідну інформацію, а також виконуючи функцію пошуку, яка реалізована на сторінках. Діаграму класів для інтернет магазину продажу меблевої продукції зображена на рисунку 2.3.

Тепер побудуємо діаграму послідовності. Вона служить для наочного відображення взаємодії об'єктів протягом часу. Діаграма послідовності відображає лише об'єкти, які безпосередньо беруть участь у взаємодії, не відображаючи можливих статичних асоціацій з іншими об'єктами. При побудові цієї діаграми кожному об'єкту сценарію надається лінія життя, яка відображає послідовність подій від його створення до знищення. Графічно кожен об'єкт зображується прямокутником, розташованим у верхній частині його лінії життя. У середині прямокутника записуються ім'я об'єкта та ім'я класу, розділені двокрапкою.

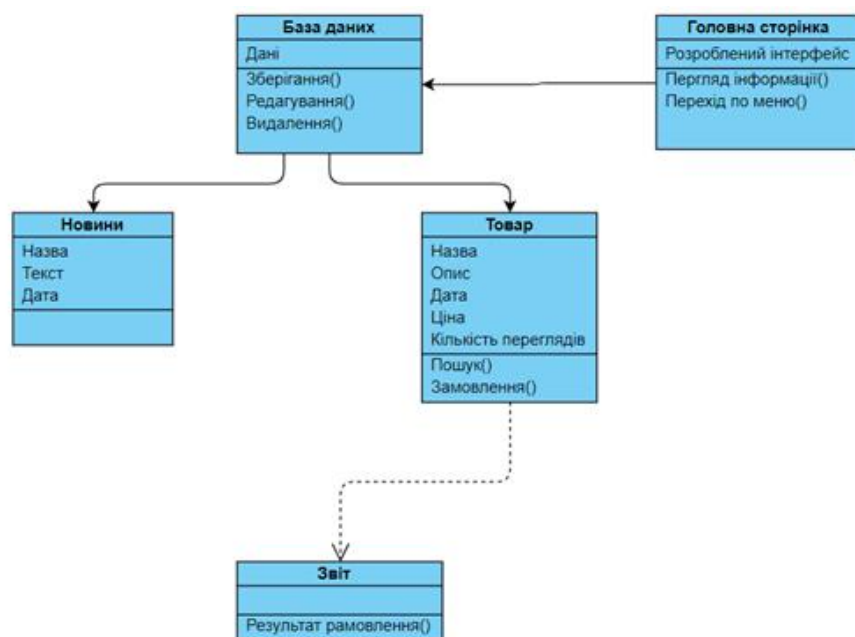


Рисунок 2.3 – Діаграма класів інтернет магазину продажу меблевої продукції

Діаграма послідовності відображає процес роботи функції пошуку протоколів на сайті. Для здійснення пошуку, користувач повинен ввести дані, які використовуються для пошуку. Сторінка пошуку створює критерій пошуку, перевіряє його та передає до пошукової системи. Пошукова система звертається до бази даних для отримання списку протоколів. Після отримання списку система перевіряє, чи відповідає критерій пошуку будь-якому елементу зі списку, і виводить результати пошуку. Якщо в списку є відповідний запис, його відображають користувачу. В іншому випадку – система повідомить користувача про відсутність інформації по заданому критерію. Діаграму послідовностей для сценарію пошуку на сайті «інтернет магазину продажу меблевої продукції» зображено на рисунку 2.4.

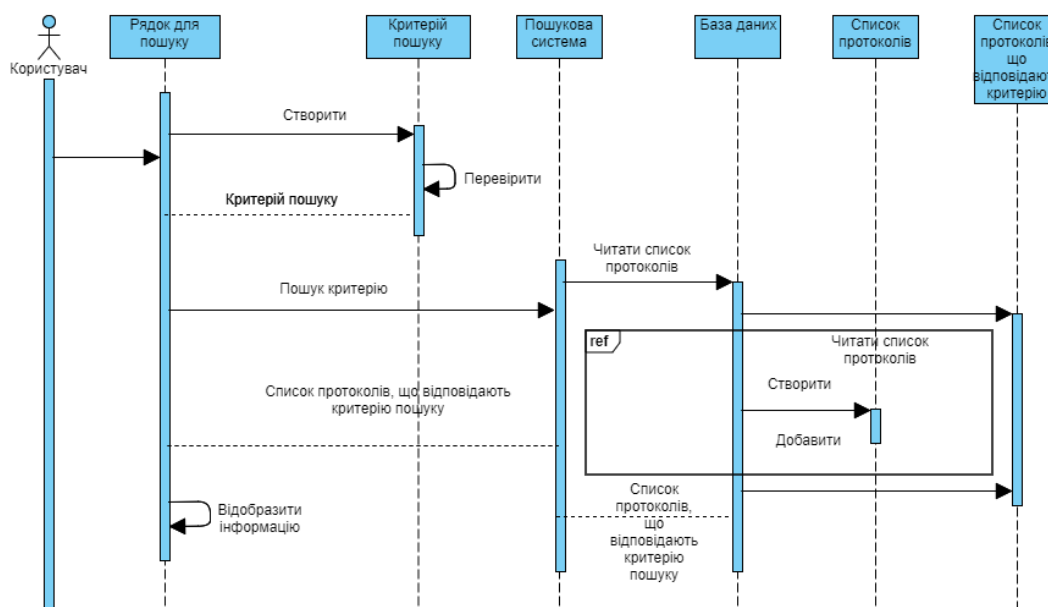


Рисунок 2.4 – Діаграма послідовності для сценарію пошуку товару

На даному етапі з використанням уніфікованої мови моделювання (UML) було спроектовано та побудовано діаграму прецедентів для сайту «Інтернет магазин продажу меблевої продукції», після цього також були створені діаграми класів та послідовності. В результаті була спроектована інформаційна система для розроблюваного сайту.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Проектування інтерфейсу користувача

Під час створення сайту необхідно врахувати вимоги користувачів до інтерфейсу, що залежать від декількох факторів:

- Можливостей, які прямо пов'язані з інтерфейсом користувача;
- часу відгуку, тобто швидкість реакції інтерфейсу на дії користувача;
- надійності інтерфейсу, яка впливає на стабільність та відсутність помилок.

Ці фактори мають велике значення. Можливості інтерфейсу користувача повинні повністю відображати функціональні можливості програми. Час відгуку інтерфейсу має бути мінімальним, щоб користувач не витрачав багато часу на очікування виконання дій. Окрім того, необхідною умовою є надійність ІК – здатність зберігати протягом тривалого часу функціональні елементи інтерфейсу.

Основу обміну інформацією між користувачем та системою складають діалоги. Інтерфейс програми повинен бути зручним та інтуїтивно зрозумілим. Ніякі дії користувача не повинні призвести до помилок в системі. У разі виникнення непередбачуваних станів, користувач повинен отримати зрозуміле повідомлення про помилку, можливі причини її виникнення та перелік рекомендованих подальші дій для її усунення [22].

Сайт меблевого магазину складається з п'яти сторінок, три з яких об'єднані між собою гіперпосиланнями. Кожна зі сторінок (крім авторизації, «Про нас» та «Контакти») має типову структуру, що включає в себе «header» – заголовкову частину; «footer» – підвальна частина, яка традиційно містить основну інформацію про меблевий магазин, адресу, посилання на популярні соцмережі та інформацію про розробників та основну частину сторінки, що містить контент – всю потрібну користувачеві інформацію.

При завантаженні користувач потраплятиме на сторінку авторизації (див. рис.3.1.). На рисунку наведено один із макетів авторизації користувача на сайті.

Логічним є наявність підказок в інтерфейсі користувача, які дозволяють інтуїтивно зрозуміло вводити дані у потрібні поля. При введенні невірних даних (логіну чи пароля) буде виведено повідомлення про помилку та пропозицію повторного введення даних.

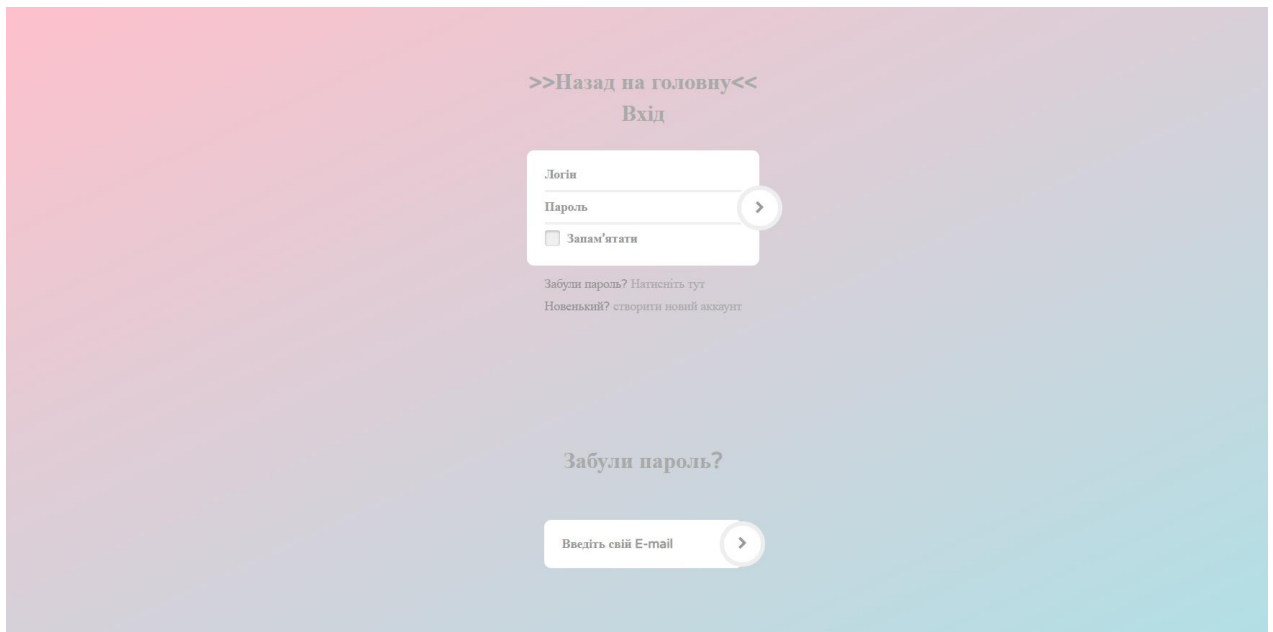


Рисунок 3.1 – Вікно «Вхід в аккаунт»

Після успішної авторизації система перенаправляє користувача на головну сторінку ресурсу (див. рис. 3.2). На цій сторінці розміщена інформація про товари у вигляді «сітки», яка включатиме в себе фото товару, його ціну, лічильник переглядів тощо. Окрім того, на сторінці є вертикальне меню, пунктами якого є всі можливі види сортування для зручності користувача (вид товару, назва фірми-виробника). Також є горизонтальне меню із пунктами:

- головна;
- новини;
- лідери продажу;
- розпродаж;

кожен з яких є посиланням на відповідну сторінку.

Кнопка «Корзина» перенаправляє користувача на сторінку, де знаходиться весь асортимент товару, який перед цим покладено до кошику (див. рис. 3.3)[12, 13, 14].

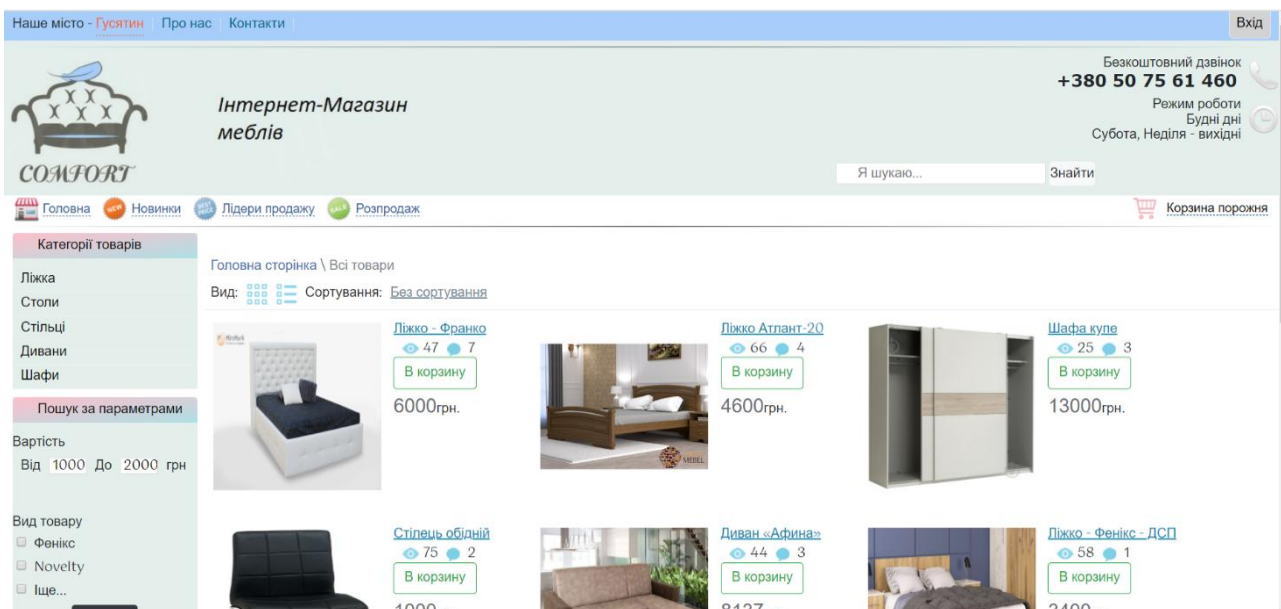


Рисунок 3.2 – Головна сторінка сайту

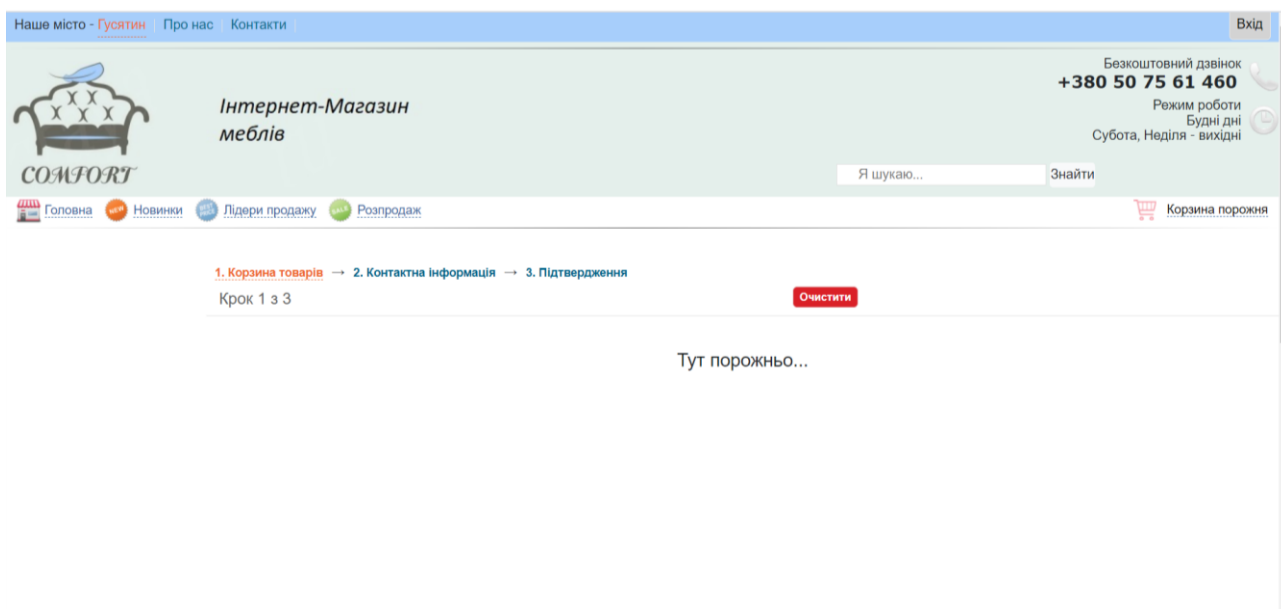


Рисунок 3.3 – Вікно «Корзина»

Також на сайті є можливість перейти на сторінку в якій розміщені товари, які належать певній категорії:

- Новинки;

- Лідери продажу;
- Розпродаж.

Всі ці сторінки аналогічні головній, але їх наповнення відрізняється згідно тому, під яку категорію підпадають ті чи інші записи (див. рис. 3.4, 3.5, 3.6) [17].

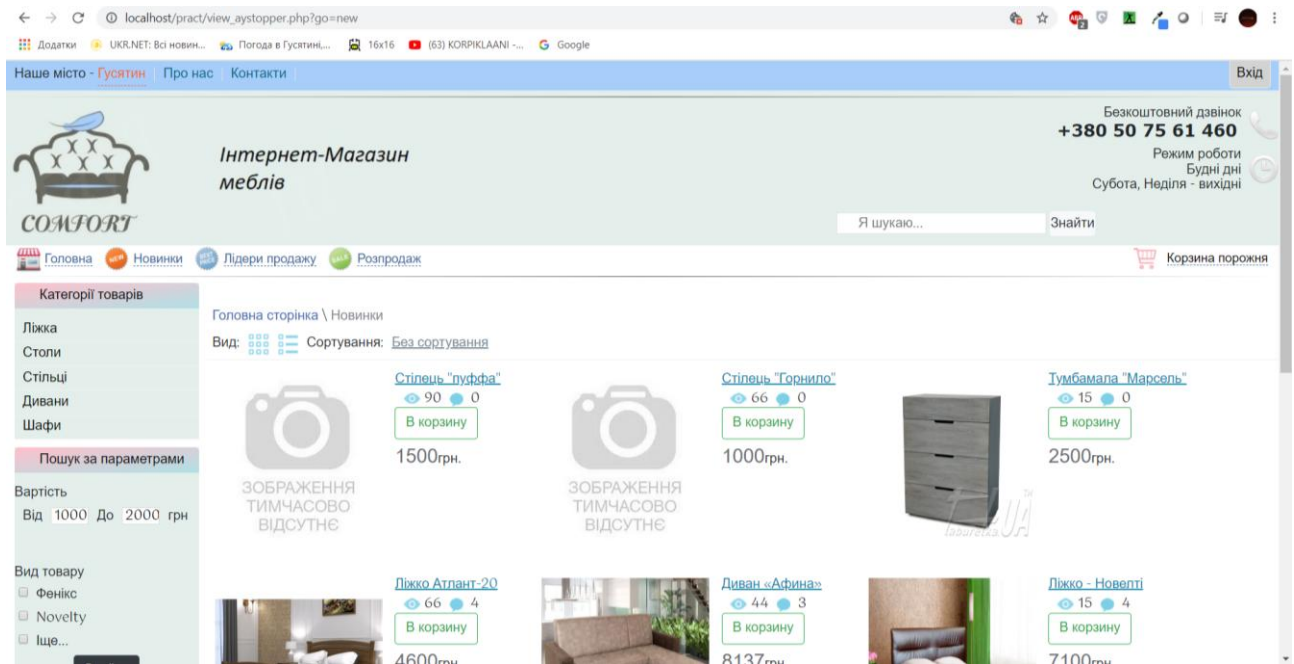


Рисунок 3.4 – Сторінка «Новинки»

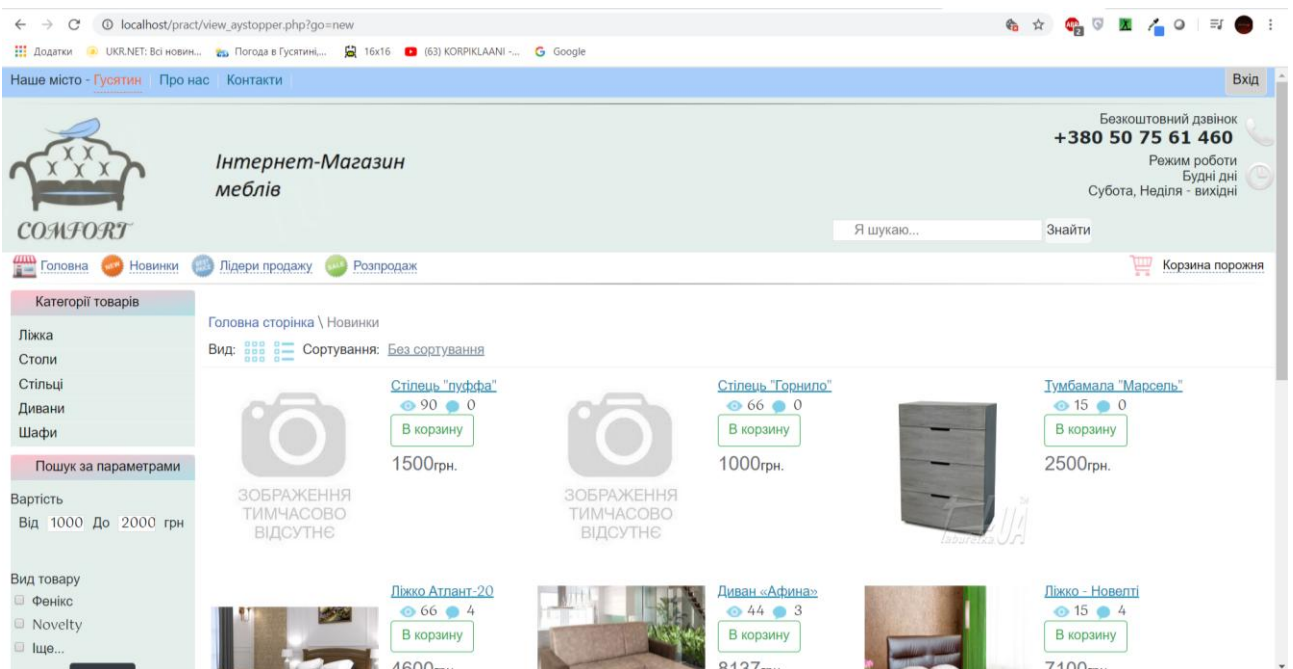


Рисунок 3.5 – Сторінка «Лідери продажу»

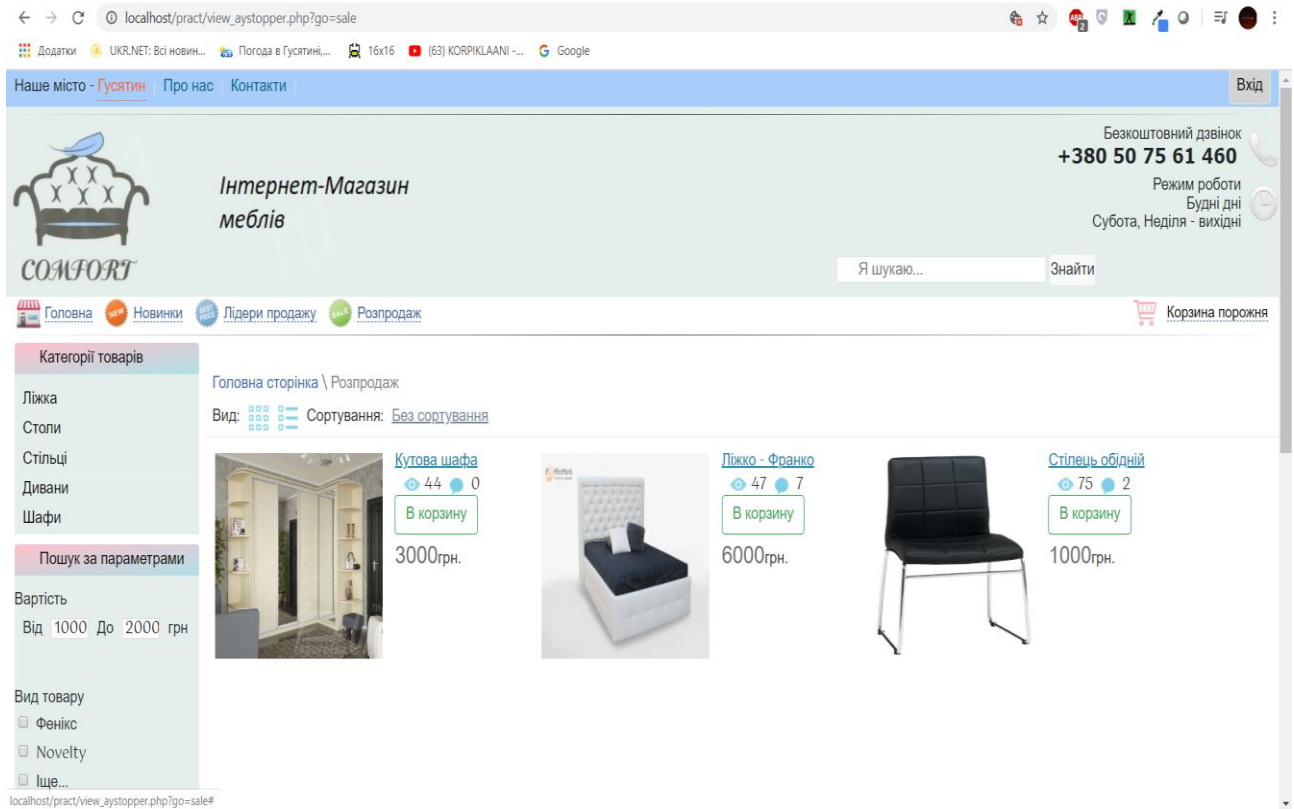


Рисунок 3.6 – Сторінка «Розпродаж»

Всі ці сторінки, насправді, є однією сторінкою але з різним GET-параметром завдяки якому і відбувається виведення товарів, які підпадають під ті чи інші критерії [17].

Вкладка «Про нас» дозволяє користувачеві ознайомитися з діяльністю підприємства, її історією та сьогоденням (див. рис. 3.7). Вкладка «Контакти» дозволяє перехід на сторінку, де описано можливі способи зв'язку з підприємством (див. рис. 3.8) [15].

Дизайн сторінок витримано у мінімалістичному стилі та спокійних блідих тонах, що є ергономічним та не відволікає користувача від основного завдання – розгляду та вибору продукції.

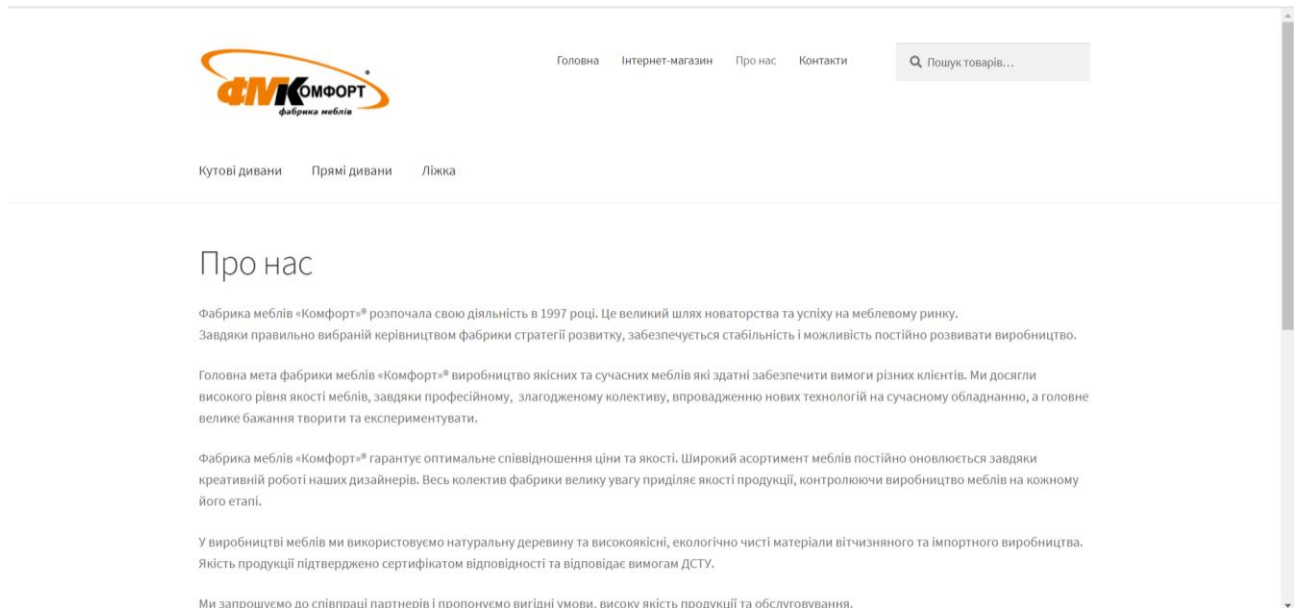


Рисунок 3.7 – Сторінка «Про нас»

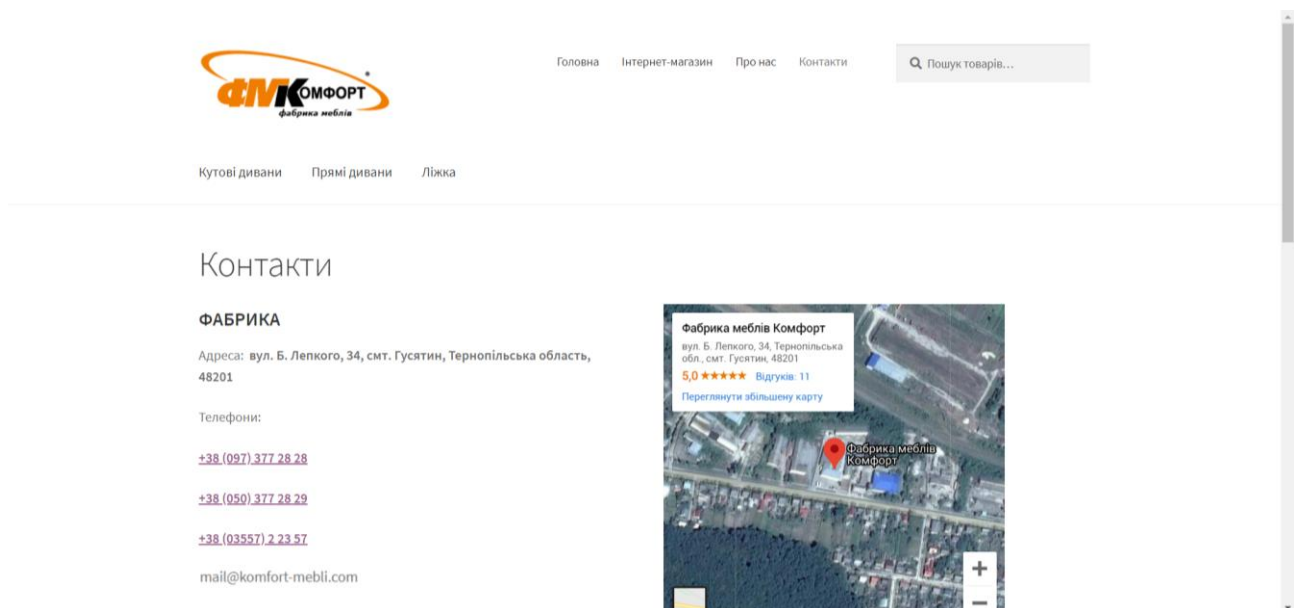


Рисунок 3.8 – Сторінка «Контакти»

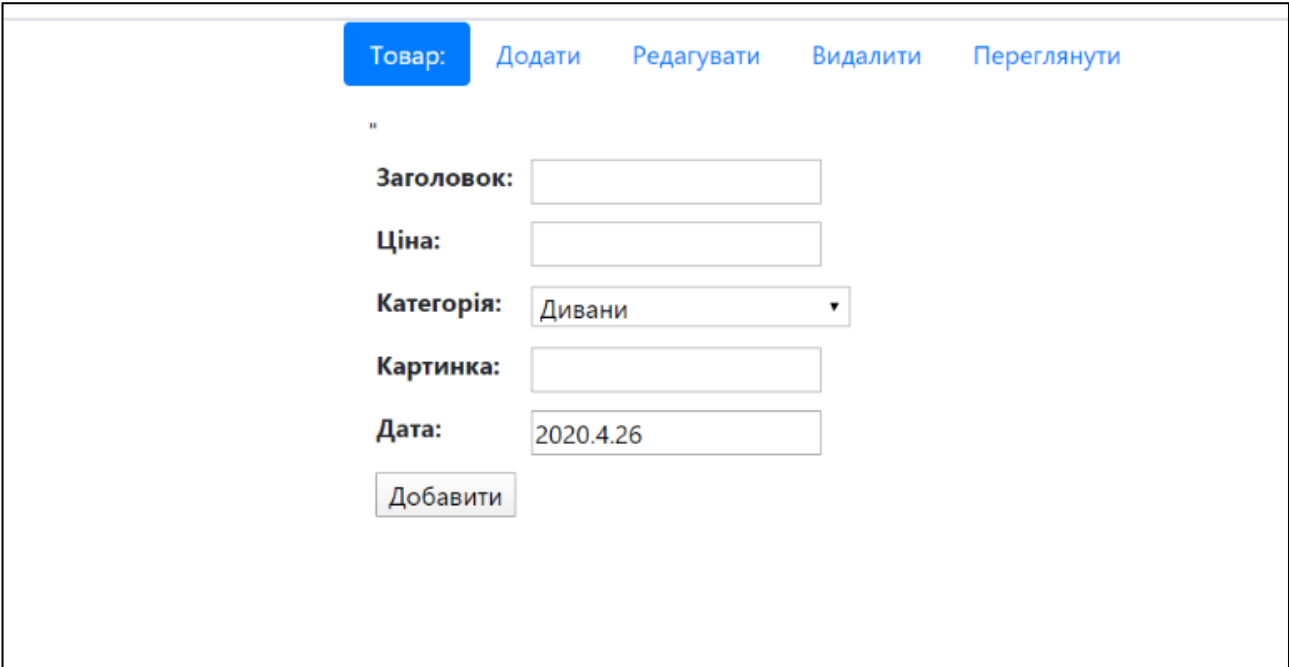
Якщо авторизований користувач – адміністратор, то після авторизації він отримає доступ до сторінки з меню, у якому можна переходити між пунктами:

- Товар;
- Додати;
- Редагувати;
- Видалити;

- Переглянути.

Ці пункти призначені для роботи з базою даних (таблицею, де містяться товари). Меню адміністратора виконану стилі ультра-мінімалізму [15].

Щоб додати товар без безпосереднього втручання в базу даних було використано форму в якій заповнюються поля, а дані з них після натиснення на кнопку «Добавити», завдяки SQL-запиту з ключовим словом «INSERT INTO», надсилаються в поля таблиці БД (див. рис. 3.9) [18, 19, 20].



The screenshot shows a web interface for adding a product. At the top, there is a navigation menu with buttons: "Товар:" (highlighted in blue), "Додати", "Редагувати", "Видалити", and "Переглянути". Below the menu, there is a form with the following fields:

- Заголовок:** [text input field]
- Ціна:** [text input field]
- Категорія:** [dropdown menu with "Дивани" selected]
- Картинка:** [text input field]
- Дата:** [text input field with "2020.4.26" entered]

At the bottom of the form is a button labeled "Добавити".

Рисунок 3.9– Сторінка адміністратора (форма «Додати»)

Для того, щоб видалити товар з бази даних на форму адміністратора «Видалити» виводиться список всіх товарів з їхніми Id. Вибравши необхідний товар зі списку та натиснувши кнопку «Видалити» товар буде видалено з бази даних, за допомогою SQL-запиту з ключовим словом «DELETE» (див. рис. 3.10).

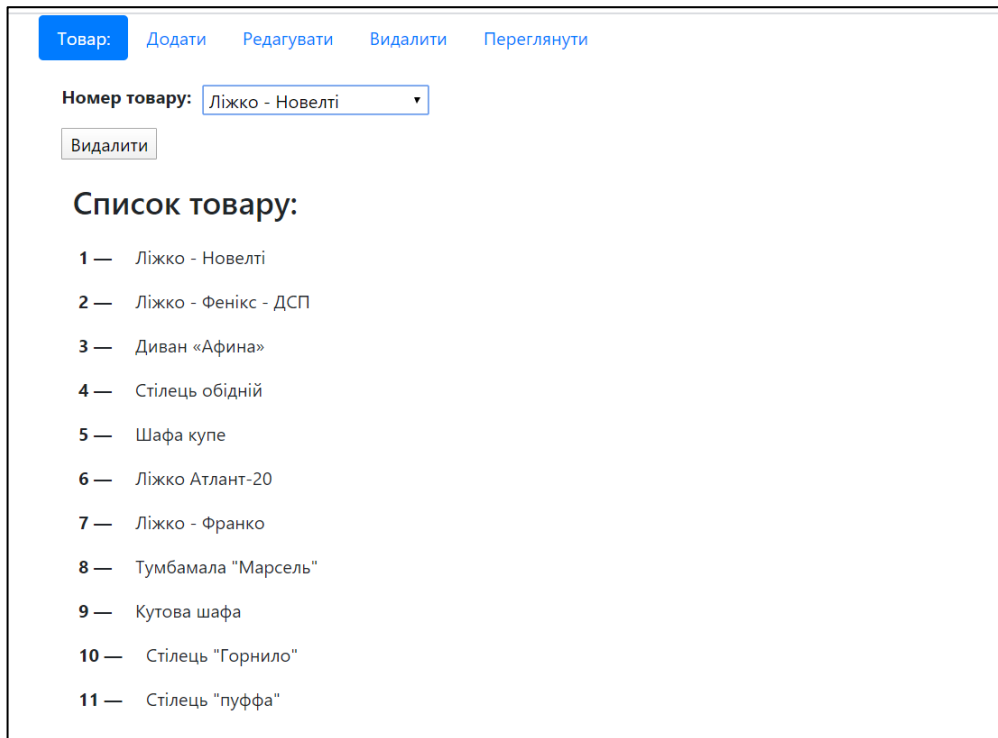


Рисунок 3.10 – Сторінка адміністратора (форма «Видалити»)

Сторінка адміністратора є хорошим прийомом для пришвидшення роботи адміністратора сайту на відмінну від безпосереднього втручання в базу даних.

3.2 Опис програмних модулів

Для розробки сайту обрано фреймворк Bootstrap. Він включає містить набір інструментів та шаблонів для створення сайтів і веб-додатків. Bootstrap включає в себе HTML і CSS шаблони оформлення, веб-форми, кнопки, мітки, блоки навігації та інші компоненти веб-інтерфейсу. Це дозволяє прискорити використання набутих в процесі навчання знань для розробки веб-сайту.

Для операцій з базою даних (БД) реалізовано PHP-скрипти. З їх допомогою встановлюється підключення до БД та перевіряється наявність таблиць. Код для встановлення зв'язку з базою даних зображено на рисунку 3.11.

```

<?php
define("host", "localhost");
define("user", "root");
define("password", "");
define("db_name", "db_shop");
$db_connect = mysqli_connect(HOST, USER, PASSWORD);
mysqli_select_db($db_connect,DB_NAME);

mysqli_query ($db_connect,"set_client='utf8'");
mysqli_query ($db_connect,"set character_set_results='utf8'");
mysqli_query ($db_connect,"set
collation_connection='utf8_general_ci'");
mysqli_query ($db_connect,"SET NAMES utf8");

$link = mysqli_connect('localhost','root','','db_shop');

if (mysqli_connect_errno())
{
    echo "Помилка в підключені бази даних
('mysqli_connect_errno().'): mysqli_connect_error();
    exit();
}

?>

```

Рисунок 3.11 – Код для встановлення зв'язку з базою даних

Для керування таблицями написано SQL-запити з параметрами, що дозволяє перевіряти дані, які підставляються в запит.

У результаті виконання SQL-запиту формується масив результатів по типу ключ-значення. На потрібній сторінці масив опрацьовується та виводяться дані з бази даних. Функція для формування головної сторінки відображено на рисунку 3.2.

```

$result = mysqli_query($link, "SELECT * FROM table_products WHERE visible='1'
ORDER BY $sorting");
if (mysqli_num_rows($result) > 0) {
    $row = mysqli_fetch_Array($result);
    do {
        if ($row["image"] != "" /*&&
file_exists("../uploads_images/".$row["image"].*/) {
            $img_path = $row["image"];
            $max_width = 200;
            $max_height = 200;
            list($width, $height) = getimagesize($img_path);
            $ratioh = $max_height / $height;
            $ratiow = $max_width / $width;
            $ratio = min($ratioh, $ratiow);
            $width = intval($ratio * $width);
            $height = intval($ratio * $height);
        } else {
            $img_path = "../img/no-image.png";
            $width = 110;
            $height = 200;
        }
        echo '
<li>
    <div id="image" class="block-images-grid">
        <a href="#"></a>
    </div>
    <p class="style-title-grid"><a href="">' . $row["title"] .
'</a></p>
    <ul class="reviews-and-counts-grid">
        <li><p>' . $row["view"] .
'</p></li>
        <li><p>' . $row["comment"] .
'</p></li>
    </ul>
    <input type="submit" name="submit" class="btn btn-outline-success"
value="В корзину"/>
    <p class="style-price-grid"><strong>' . $row["price"] .
'</strong>грн.</p>
    <div class="mini-features">
    </div>
    </li>
';
    } while ($row = mysqli_fetch_Array($result));
}

```

Рисунок 3.12 – Функція для формування головної сторінки

Модуль опрацювання коментарів та міток карти реалізовано за допомогою асинхронних запитів аґах. Це дозволяє у випадку отримання результатів уникнути перезавантаження всієї сторінки, відбувається лише оновлення заданого блоку. Також, у випадку асинхронного запиту до сервера, сторінка не приймає статус «102 - Processing», тому користувач може успішно продовжувати користування сайтом (див. рис. 3.13).

```

$(document).ready(function() {
    //Add new comment
    $("#ComSubm").click(function (e) {

        e.preventDefault();

        var post_id = $("#hidden_postid").val();
        var name = $("#name").val();
        var email = $("#email").val();
        var text = $("#text").val();
        jQuery.ajax({
            type: "POST", // Method
            url: "saveComment.php", //url
            dataType:"text", // Data type
            data: {
                hidden_postid: post_id,
                name: name,
                email: email,
                text: text,
            },
            success:function(response){
                $('#commentDiv').load(document.URL + '
#commentDiv');

                $("#text").val('');//Clear
                $("#name").val('');
                $("#email").val('');
            },
            error:function (xhr, ajaxOptions, thrownError){
                alert(thrownError); //Error msg
            }
        });
    });
});

```

Рисунок 3.13 – Код програми для асинхронного додання коментарів

3.3 Опис результатів тестування

У процесі розробки сайту, перед його розміщенням на хостингу, важливо провести перевірку працездатності функціоналу.

Тестування сайту включає перевірку відповідності його технічному завданню та перевірку функціональності. Залежно від вихідного технічного завдання, в процесі тестування можуть проводитися наступні перевірки:

- перегляд сайту на різних моніторах;
- перегляд сайту в різних браузерях;
- перевірка працездатності гіперпосилань на сайті;
- аналіз usability;
- перевірка html/css.

Розглянемо три типи тестів для веб-сайту, що був розроблений:

- ручне тестування, також відоме як ad hoc-тестування. Його результатом є складання Bug report, який містить відомості про виявлені помилки;
- автоматичне тестування;
- функціональне тестування.

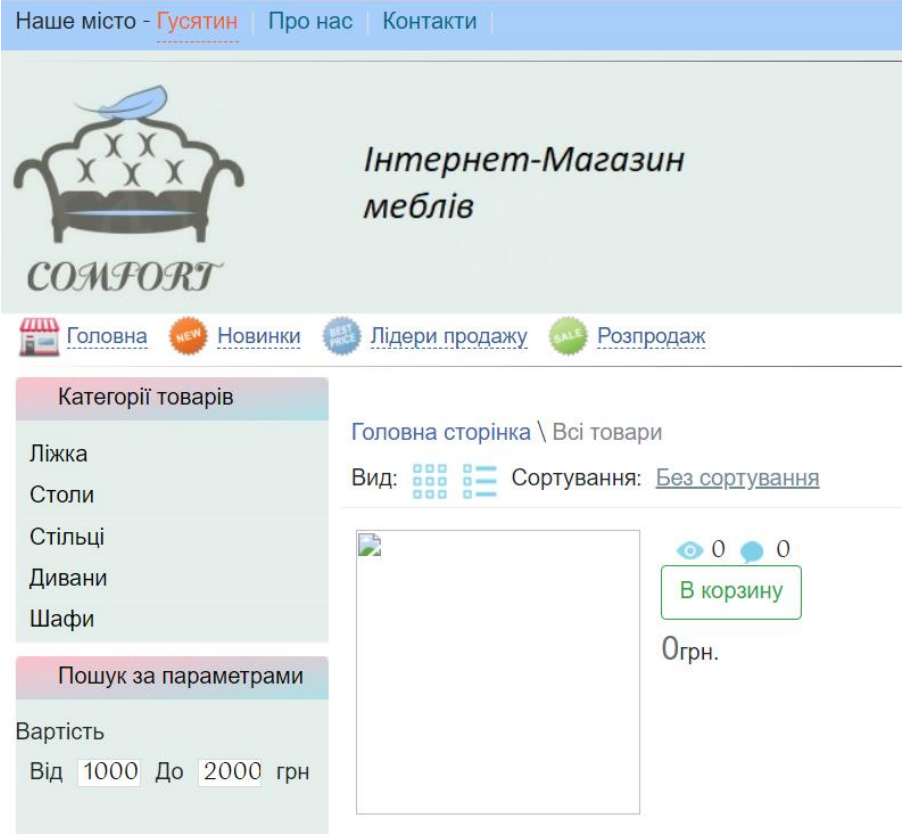
Ad hoc-тестування, або інтуїтивне тестування, є формою тестування, що проводиться без тест-плану та документації. Воно базується на передбаченні помилок та особистому досвіді тестувальника.

Результатом ручного тестування є створення Bug report, який містить опис виявлених проблем. Структура Bug report наведена у таблиці 3.1.

Таблиця 3.1 – Структура Bug-report

Інтернет магазин меблів	
1	2
Короткий опис (Summary)	При додаванні товару за допомогою форми адміністратора дані не вводилися коректно (всі поля були порожніми або рівні 0)
Проект (Project)	Інтернет магазин меблів
Компонент додатку (Component)	Наповнення сайту, GUI
Номер версії (Version)	Версія v 1.0.2
Серйозність (Severity)	S3 Значний (Major)
Пріоритет (Priority)	P2 Середній (Medium)
Автор (Author)	Дишлевий Павло
Призначення (Assigned To)	Developer – Дишлевий Павло
Середовище	
ОС / Сервіс Пак і т.д. / Браузер + версія / ...	Windows 10 Pro 2019, Google Chrome 74.0.3729.157 (64 біт), Mozilla Firefox 66.0.3 (64 біт)

Продовження таблиці 3.1

1	2
Опис	
Кроки відтворення (Steps to Reproduce)	1. Заповнюємо форма «додати товар» 2. Перехід на головну сторінку сайту 3. Контент відображається з порожніми або 0-вими полями
Фактичний результат (Result)	Доданий товар завдяки формі адміністратора відображається з порожніми полями.
Очікуваний результат (Expected Result)	Доданий товар завдяки формі адміністратора відображається коректно.
Додатково	
Прикріплений файл (Attachment)	 <p>The screenshot shows the homepage of an online furniture store named 'COMFORT'. The header includes navigation links: 'Наше місто - Гусятин', 'Про нас', and 'Контакти'. Below the header is a banner with the store's logo (a sofa) and the text 'Інтернет-Магазин меблів'. A navigation bar contains icons for 'Головна', 'Новинки', 'Лідери продажу', and 'Розпродаж'. The main content area is titled 'Категорії товарів' and lists categories: Ліжка, Столи, Стільці, Дивани, and Шафи. Below this is a search section 'Пошук за параметрами' with a price filter set from '1000' to '2000' UAH. The search results area shows 'Головна сторінка \ Всі товари' and 'Вид: [grid icon] [list icon] Сортування: Без сортування'. A product placeholder is visible with a 'В корзину' button and '0' items in the cart.</p>

Помилка була виправлена за допомогою редагування SQL-запиту, в якому було допущено синтаксичну помилку, яка видавала попередження але продовжувала працювати (див. Лістинг 3.4).

Лістинг 3.4 – SQL-запит на додавання товару

```
$sql = "INSERT INTO `table_products`(`products_id`, `title`, `price`,  
`brand`, `image`, `datetime`) VALUES (NULL, `title`, `price`, `brand`,  
`image`, `datetime`);"
```

Далі проведемо тести розробленого сайту за допомогою функціонально тестування.

Функціональне тестування перевіряє чи реалізовані функціональні вимоги, тобто можливості ПЗ в певних умовах вирішувати завдання, потрібні користувачам.

Функціональні вимоги визначають, що саме робить продукт, які завдання вирішує.

Функціональні вимоги включають в себе:

- функціональна придатність;
- точність;
- можливість до взаємодії;
- відповідність стандартам та правилам;
- захищеність.

При тестуванні веб додатку розглянемо усі функції, методи та компоненти, які реалізовані в системі на відповідність.

Результати тестування системи функціональним методом наведені в таблиці 3.2.

Таблиця 3.2 – Створення набору test-case

ID	Тип	Опис	Очікувано	Реально	Pass/ Fail
01	positive	Перевірка сайту на кросбраузерність	У всіх браузерах сайт повинен мати однаковий вигляд	В різних браузерах сайт має однаковий вигляд	Pass
02	positive	Перевірка валідності посилань на панелі навігації	При натисканні на посилання здійснюється перехід на певну сторінку	При натисканні на посилання відбувається перехід на очікувану сторінку	Pass
03	negative	Перевірка на однаковий вигляд фотографій різних браузерів	У різних браузерах фотографії повинні відображатись однаково	У різних браузерах фотографії не однакової якості	Fail
04	positive	Перевірка на слайдера зображень правильне виконання	Слайдер повинен перемикає фотографії у правильній послідовності і змінювати номер фотографій	Слайдер перемикає фотографії у правильній послідовності і змінює номер фотографій	Pass

Для перевірки кросбраузерності сайту, було проведено огляд веб-сторінок у різних інтернет-браузерах та їх версіях. В процесі перевірки було обрано два браузери, що представляють різні платформи та варіанти веб-переглядачів: Microsoft Edge (див. рис. 3.14) та Google Chrome (див. рис. 3.15). У результаті, впевнилися у коректності відображення інформації різними браузерами.

Головним недоліком сайту є те, що він не є адаптований під пристрої з різними розширеннями – користувачі смартфонів та планшетів не зможуть користуватися сайтом через «викревлений» інтерфейс (див. рис 3.15) [18].

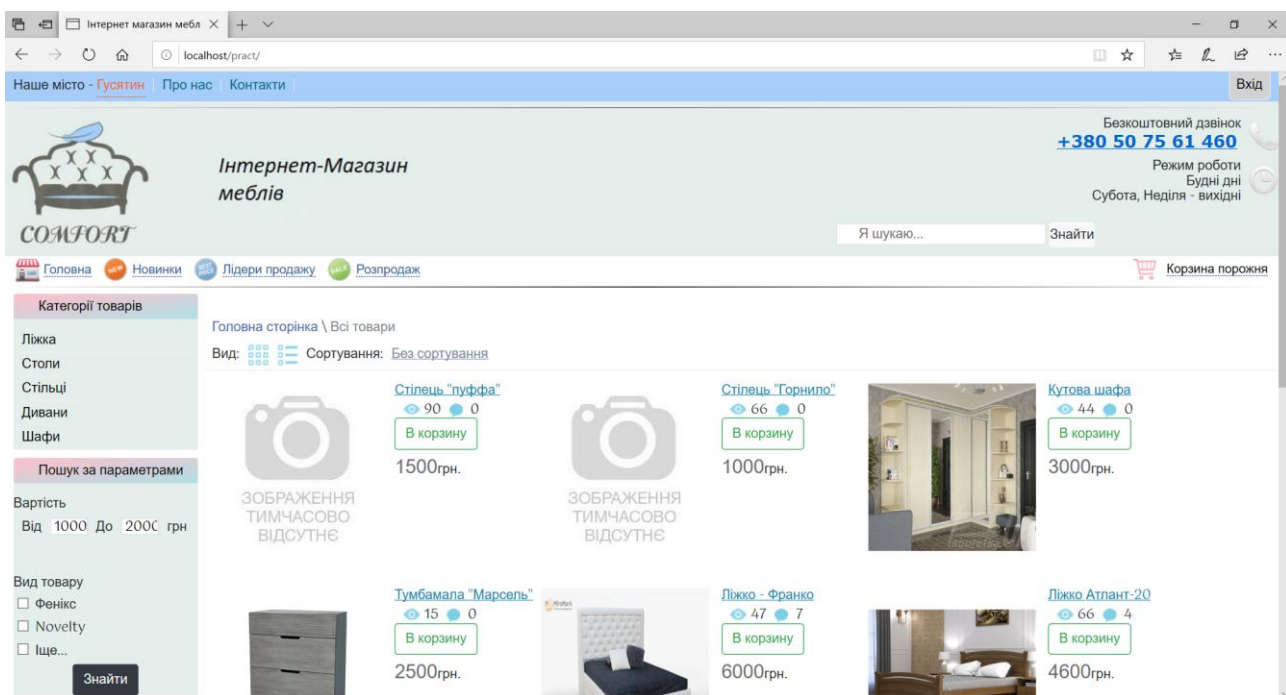


Рисунок 3.14 – Вигляд сторінки сайту у браузері Microsoft Edge

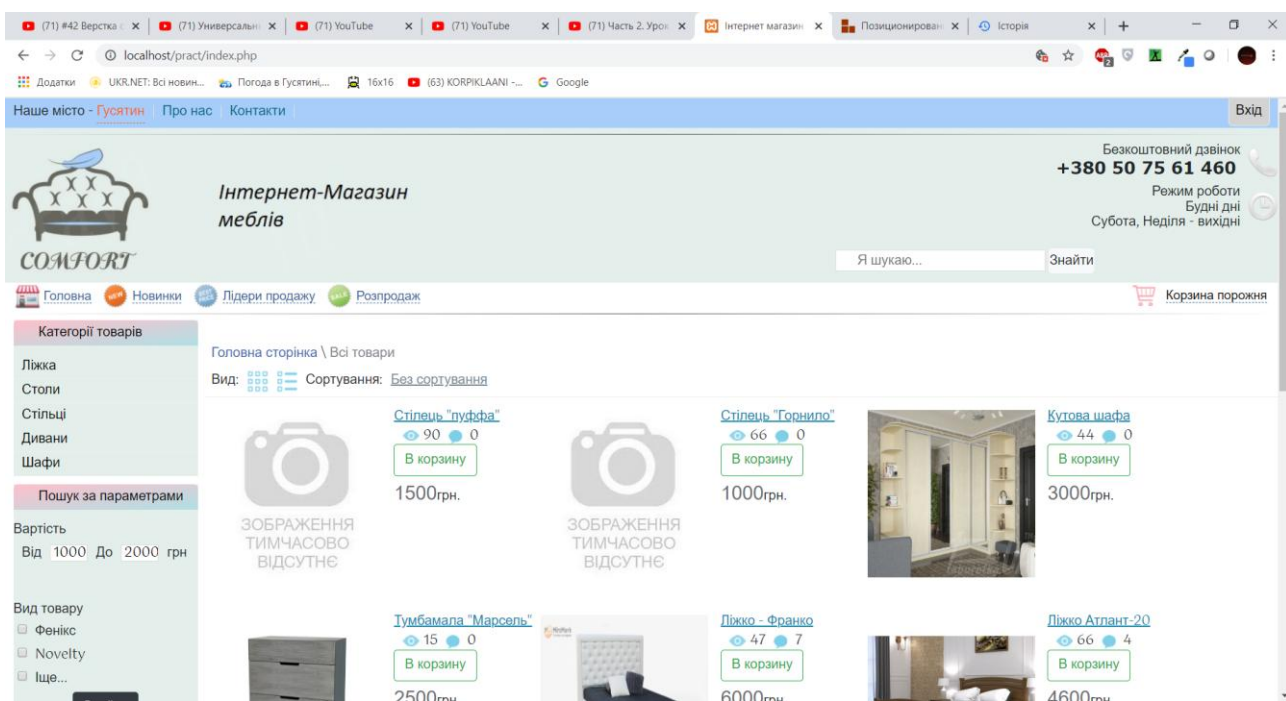


Рисунок 3.15 – Вигляд сторінки сайту у браузері Google Chrome

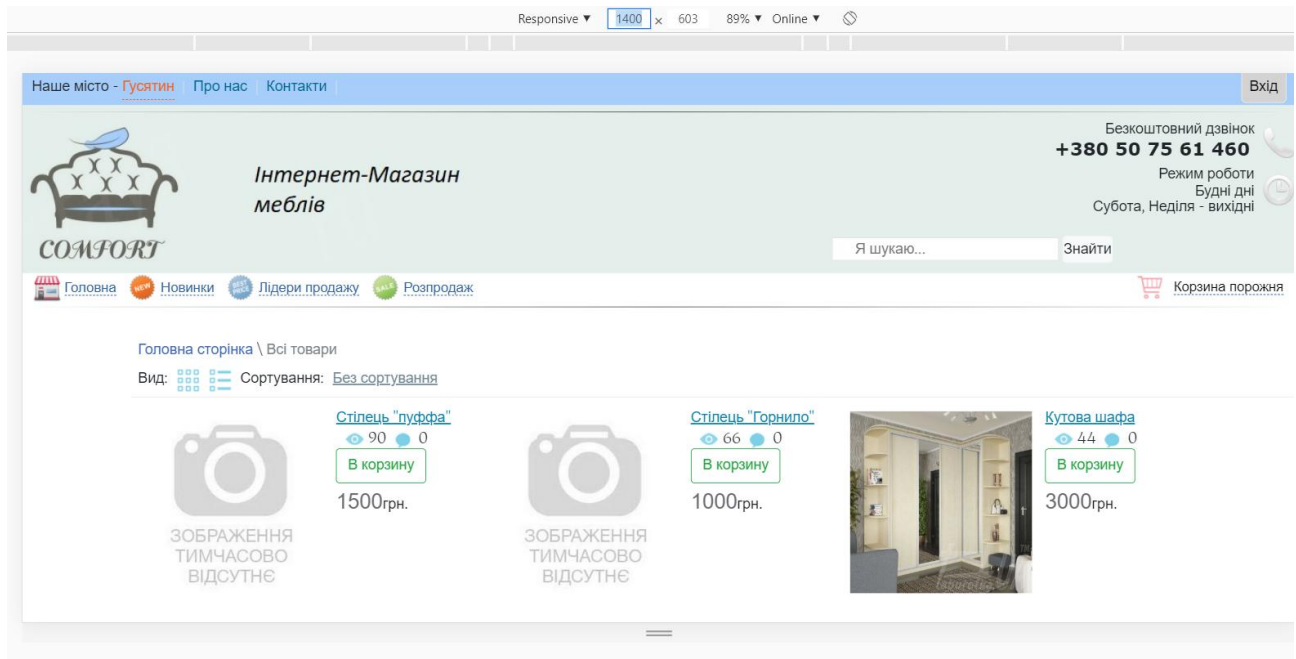


Рисунок 3.16 – Вигляд сторінки (ширина девайса 1400 пікселів)

Впродовж розробки додаток обов'язково має перевірятися засобами автоматичного тестування, оскільки не завжди можливо визначити всі проблеми та баги вручну. Для тестування додатку використаємо Selenium IDE.

Selenium – це інструмент для автоматизованого тестування веб-додатків. Застосування Selenium дозволяє автоматизувати будь-які інші повсякденні дії, які зазвичай виконуються через веб-браузер.

Selenium IDE є розробкою як розширення для браузера Mozilla Firefox, тому його можна встановити двома способами, перший спосіб – це через вкладку розширень самого браузера, або другий спосіб з офіційного сайту SeleniumHQ.

Працювати з середовищем Selenium IDE, не складає великих труднощів, усі функціональні можливості інтуїтивно зрозумілі та доступні навіть тестувальнику початківцю. У першу чергу починаємо запис логування усіх тестів, які будуть виконуватися, натиснувши кнопку початку запису. Після чого переходимо на сайт та виконуємо необхідні маніпуляції по функціоналу сторінок.

Отже, виконаємо набір автоматизованих тестів для перевірки автоізації на сайт (див. рис. 3.17 та 3.18).

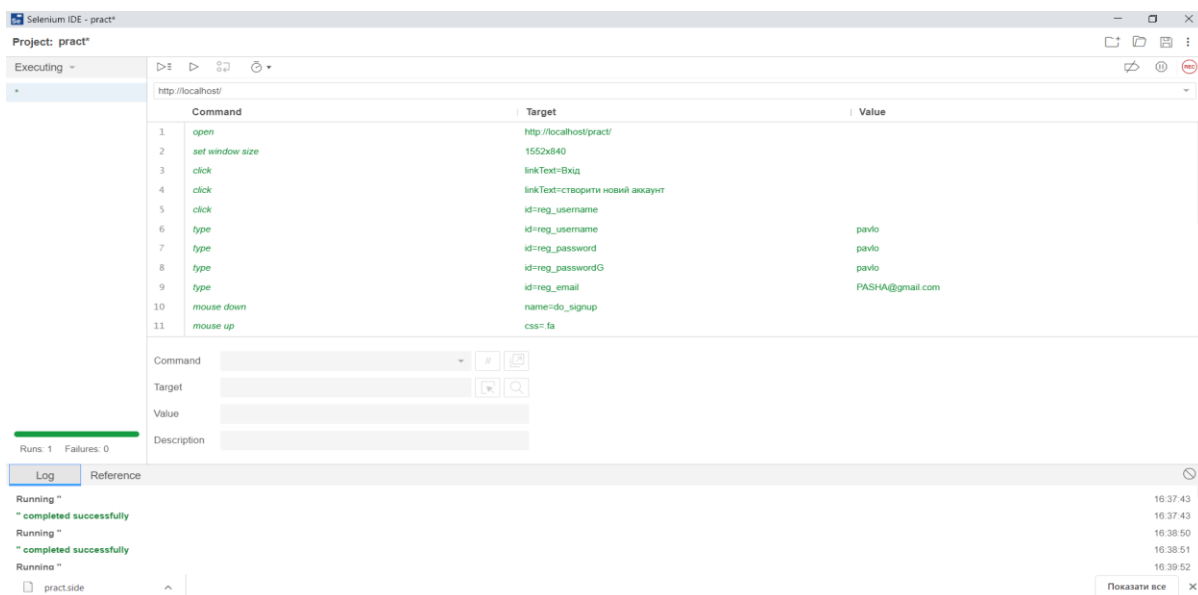


Рисунок 3.17 – Послідовність команд тесту

19. type on id=input-search with value Шафа ОК	16:51:06
20. click on css=.btn-light ОК	16:51:07
21. click on linkText=Головна ОК	16:51:07
22. click on linkText=Вийти? ОК	16:51:09
" completed successfully	16:51:12

Рисунок 3.18 – Журнал результатів тесту

Виконавши низку різних наборів тестів, Selenium IDE показав декілька дрібних помилок, пов'язаних із некоректними гіперпосиланнями та одну критичну помилку з некоректною URL адресою. Це пов'язано з тим, що автоматичне тестування сайту відбувалося локально, без адреси хостингу та домену сайту [21, 22].

У процесі тестування також було здійснено такі перевірки:

- Перевірка коректного відображення кольорів на сайті при різних налаштуваннях колірної палітри монітора, а також перевірка відображення шрифтів, анімації та графічних зображень;
- Перевірка властивостей кожної сторінки сайту, таких як заголовки, ключові слова, описи та інші мета-теги;

- Перевірка відповідності змісту кожної сторінки вихідним матеріалам замовника та правопису на кожній сторінці;
- Перевірка валідності верстки за допомогою сервісу W3C Validator (див. рис. 3.19).

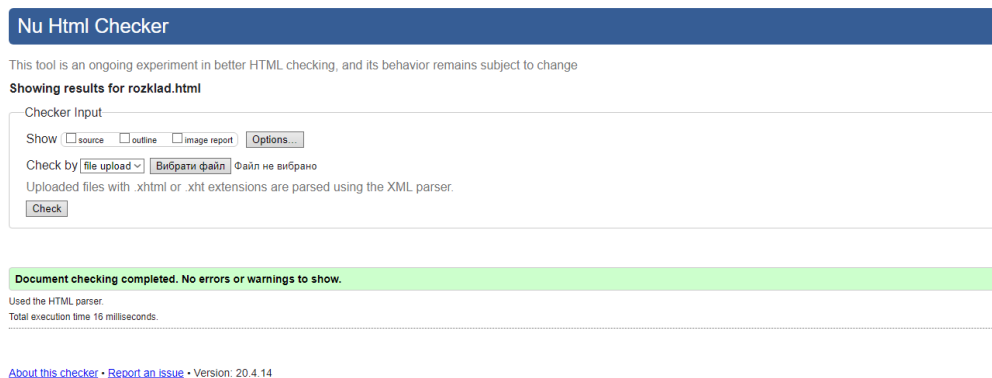


Рисунок 3.19 – Перевірка валідності верстки

Один з принципів практичного програмування полягає в тому, що жодна програма не досягає бажаних результатів після першої спроби трансляції та виконання [23, 24, 25]. Таким чином, під час всіх проведених тестів було виявлено певні недоліки в різних структурних частинах програми, зокрема, при проектуванні користувацького інтерфейсу GUI (графічний інтерфейс користувача), функціональному функціонуванні та написанні програмного коду. Однак, останній тест показав, що більшість програмного коду продукту написана належним чином і практично не містить помилок.

Завдяки розробці тестів та їх проведенню над сайтом було успішно усунуто виявлені недоліки та помилки, і була досягнута стабільна робота додатку.

4 БЕЗПЕКА ЖИТТЕДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Ергономічний та дизайнерський аналіз верстату, установки, обладнання

Станом на 2023 рік сучасна електронна техніка присутня в кожному домі і у чималих екземплярах. Її кількість та популярність зумовлена багатьма факторами, та основні з них – це оптимізація часових затрат на ту чи іншу працю, можливість паралельно виконувати кілька процесів одночасно. Також вагомим чинником є комфорт. Та на кожну дію є протидія. Ні для кого не є секретом позитивні сторони персональних комп'ютерів: можливість пошуку інформації та її опрацювання в об'ємах, недопустимих людині, але за це необхідно розплачуватися згубною дією на організм користувача з боку комп'ютера.

Ергономічний та дизайнерський аналіз верстатів, установок та обладнання має вирішальне значення для забезпечення безпеки життєдіяльності працівників у різних сферах, таких як промисловість, виробництво, лабораторії та інші робочі середовища. Ці аналізи допомагають ідентифікувати та усунути потенційні небезпеки, пов'язані з використанням технічних засобів.

Під час ергономічного аналізу враховуються фізіологічні та психологічні фактори, які впливають на працездатність та комфорт працівників. Оцінюються параметри, такі як розташування робочих місць, висота столів і стільців, положення елементів управління, освітлення, шум та вібрація. Мета полягає у забезпеченні оптимального співвідношення між працею та фізичними можливостями працівників, зменшенні втоми та ризику виникнення професійних захворювань.

Дизайнерський аналіз спрямований на створення функціональних, безпечних та естетично привабливих рішень для верстатів, установок та обладнання. Враховуються аспекти, такі як простота використання, зручність обслуговування, відповідність нормативним вимогам, ефективне використання простору та матеріалів, а також можливість максимального захисту від потенційних небезпек.

Щоденна робота людини за комп'ютером при недотриманні принципів ергономіки, вимог санітарії та режиму роботи, може призвести до «ергономічних» захворювань. Такий термін ввели медики через появу нових недуг, які пов'язані, зокрема, із впливом комп'ютерів та периферійних пристроїв на здоров'я тих, хто з ними працює. Зокрема, за даними медичних досліджень Всесвітньої організації охорони здоров'я (ВООЗ), у користувачів ПК виявлено нові види захворювань: синдром «комп'ютерного стресу оператора»; травми повторних навантажень (накопичування й акумулювання нездужання); фотоепілептичні приступи тощо [26].

4.2 Естетичне оформлення та ергономічне дослідження робочого місця оператора

Основне навантаження під час роботи за комп'ютером припадає на очі. Їх стомлюваність багато в чому залежить не тільки від якості зображення на екрані, але і від загальної освітленості приміщення. Згідно з гігієнічними нормами освітленість на поверхні стола і клавіатурі повинна бути не менше 300 люкс, а вертикальна освітленість екрана – всього 100-250 люкс. Дослідження фізіологів і гігієністів переконливо довели, що і напівтемрява, і занадто висока освітленість екрана призводять до швидкого зорового стомлення. Також одним із факторів псування зорового апарату є «завмирання користувача». Під час роботи з комп'ютером користувач дуже часто не помічає, що вона забуває кліпати очима. У результаті цього відбувається пересихання захисної плівки на очах, що несе за собою псування зору [27].

Розміщувати комп'ютер рекомендується так, щоб світло (природне або штучне) падало збоку, краще зліва, це позбавить від тіней і допоможе знизити освітленість екрана. Як джерело освітлення рекомендується застосовувати

люмінесцентні лампи типу ЛБ зі світильниками серії ЛПО36 з екранізованими ґратами. Лампи розжарювання краще використовувати для місцевого освітлення зони робочого документа (клавіатури, книги, зошити). А настільна лампа, навпаки, повинна мати щільний абажур, який спрямовує світло прямо в зону робочого документа.

Пил, що осідає на екрані, знижує його освітленість, погіршує видимість зображення і сприяє накопиченню статичної електрики. Тому перед початком роботи, ще до включення комп'ютера, слід протирати екран м'якою неворсистою тканиною [28].

Основним обладнанням робочого місця користувача є монітор, системний блок та клавіатура. Робочі місця повинні розташовуватися на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1 м, між собою на відстані не менше 1,5 м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

Робочі місця слід розташовувати так, щоб уникнути попадання в очі прямого світла. Джерела освітлення рекомендується розташовувати з обох боків екрану паралельно напрямку погляду. Для уникнення світлових відблисків екрану, клавіатури в напрямку очей користувача, від світильників загального освітлення або сонячних променів, необхідно використовувати антиполюсківі сітки, спеціальні фільтри для екранів, захисні козирки, на вікнах – жалюзі.

При роботі з текстовою інформацією (у режимі введення даних та редагування тексту, читання з екрану) найбільш фізіологічним правильним є зображення чорних знаків на світлому (білому) фоні.

Монітор повинен бути розташований на робочому місці так, щоб поверхня екрана знаходилася в центрі поля зору на відстані 400-700 мм від очей користувача. Рекомендується розміщувати елементи робочого місця так, щоб витримувалася однакова відстань очей від екрана, клавіатури, тексту.

Зручна робоча поза при роботі з комп'ютером забезпечується регулюванням висоти робочого столу, крісла та підставки для ніг. Раціональною робочою позою може вважатися таке положення, при якому ступні працівника розташовані

горизонтально на підлозі або підставці для ніг, стегна зорієнтовані у горизонтальній площині, верхні частини рук – вертикальні.

Узагальнюючи вимоги до режиму праці та відпочинку для дорослих користувачів, слід сказати, що найбільш важкою вважається творча робота в режимі діалогу з ПЕОМ – її тривалість не повинна перевищувати шести годин на день. Для будь-якого дорослого користувача тривалість безперервної роботи за монітором не повинна перевищувати двох годин, після чого необхідна перерва 15-20 хвилин.

ВИСНОВКИ

Розробка та створення сайту для меблевого магазину дозволить фабриці зайняти нову нішу на ринку продаж, зокрема, у сфері інтернет-торгівлі та збільшити свій прибуток. Окрім того, структура бази даних дозволить легко створювати нові категорії, а значить – можлива реалізація і супутніх товарів, що підвищує конкурентоспроможність підприємства.

У ході виконання дипломного проекту проведено аналіз предметної області, досліджено схожі програмні продукти, що дозволило виділити основні функції інтернет магазину та вимоги до його написання та супроводу.

У проектній частині сформовано технічне завдання на розробку сайту. Розроблено концептуальну, логічну та фізичну моделі даних. За допомогою СУБД MySQL та мови SQL спроектовано базу даних «db_shop», яка містить дані, необхідні для роботи з сайтом.

У практичній частині проведено аналіз інтерфейсу користувача, вікон додатку, описано основні модулі та методи їх розробки, визначено та описано алгоритми їх реалізації. Проведено тестування додатку, описано декілька тесткейсів та виправлено виявлені помилки.

Проведено аналіз основних правил техніки безпеки та охорони праці при роботі з комп'ютером.

У результаті виконання дипломного проекту створено веб-застосування «Інтернет магазин продажу меблевої продукції», що дозволяє фабрикам організовувати віддалений режим торгівлі з покупцями та пришвидшувати процес замовлення товарів, тим самим збільшуючи свій прибуток.

ПЕРЕЛІК ПОСИЛАНЬ

1. Інтернет магазин «Rozetka». URL: <https://rozetka.com.ua/ua/> (дата звернення 25.11.2022).
2. Інтернет магазин «COMFY». URL: <https://comfy.ua/ua/> (дата звернення 26.11.2022).
3. Інтернет магазин «FOTOS». URL: <https://f.ua/> (дата звернення 27.11.2022).
4. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань : навч. посіб. Львів : Магнолія-2006, 2012. 584 с.
5. Шаров С. В., Осадчий В. В. Бази даних та інформаційні системи : навч. посіб. Мелітополь : МДПУ ім. Б. Хмельницького, 2014. 352 с.
6. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom Database Systems: The Complete Boo.
7. Anthony Molinaro SQL Cookbook: Query Solutions and Techniques for Database Developers, 2011. 1096 с.
8. Методичні вказівки до виконання курсової роботи з дисципліни «Бази даних» для спеціальності 5.05010301 «Розробка програмного забезпечення» / упоряд. Н. В. Оляніна. Гусятин : Вид-во ГК ТНТУ, 2015. 49 с.
9. Методичні вказівки для побудови uml діаграм до дипломного проекту для студентів спеціальності 121 «Інженерія програмного забезпечення» / упоряд. А. Л. Біленький. Гусятин : Вид-во ГК ТНТУ, 2019. 13 с.
10. Розробка uml діаграми варіантів використання. URL: <https://studfiles.net/preview/5200239/page:6/> (дата звернення 08.04.2023).
11. Діаграма варіантів використання. URL: <http://5fan.ru/wievjob.php?id=21296> (дата звернення 10.04.2023).
12. Jon Duckett. HTML and CSS: Design and Build Websites.
13. Douglas Crockford. JavaScript: The Good Parts.

14. Youtube канал Glo Academy Курс Веб-розробник 10.0. URL: <https://www.youtube.com/watch?v=n9kx5Vk0jgc&list=PL3LQJkGQtzc5G7wIQfVqBMEprnTKZlAXf> (дата звернення 10.12.2022).
15. Youtube канал Glo Academy CSS Фічі | CSS-features. URL: <https://www.youtube.com/watch?v=Kdal-3AfeRc&list=PL3LQJkGQtzc56HquxrkwPdQt9Q1wHm21P> (дата звернення 10.12.2022).
16. Youtube канал Magisters Навчання сучасним професіям PHP для початківців. URL: <https://www.youtube.com/user/WebMagistersRu> (дата звернення 10.12.2022).
17. VIDEEasy as hell jQuery plugin for video backgrounds. URL: <https://vodkabears.github.io/vide/> (дата звернення 15.12.2022).
18. Mobile-Friendly Test. URL: <https://search.google.com/test/mobile-friendly> (дата звернення 10.05.2023).
19. Glenford J. Myers. The Art of Software Testing, 2010. 464 с.
20. Srinivasan Desikan, Gopalaswamy Ramesh. Software Testing: Principles and Practices, 2011. 544 с.
21. Arbon Jason, Carollo Jeff, Whittaker James A. How Google Tests Software. Boston : Addison-Wesley Professional, 2012. 281 с.
22. Copeland Lee. A Practitioner's Guide to Software Test Design. Norwood : Artech House, 2014. 486 с.
23. Lisa Crispin, Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams, 2012. 374 с.
24. Методичні вказівки до проведення тестування програмного забезпечення для студентів спеціальності 121 «Інженерія програмного забезпечення» / упоряд. Р. І. Чаплінський. Гусятин : Вид-во ГК ТНТУ, 2022. 13 с.
25. Розрахунок та обґрунтування собівартості програмного продукту. Методичні вказівки до виконання економічного розділу дипломної роботи для студентів спеціальності 5.005010301 «Розробка програмного забезпечення» / упоряд. : К. І. Барціховська. Гусятин : Вид-во ГК ТНТУ, 2017. 50 с.

26. Про охорону праці : Закон України від 14.10.1992 р. № 2694-ХІІ. Дата оновлення 27.12.2019. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 14.04.2023).

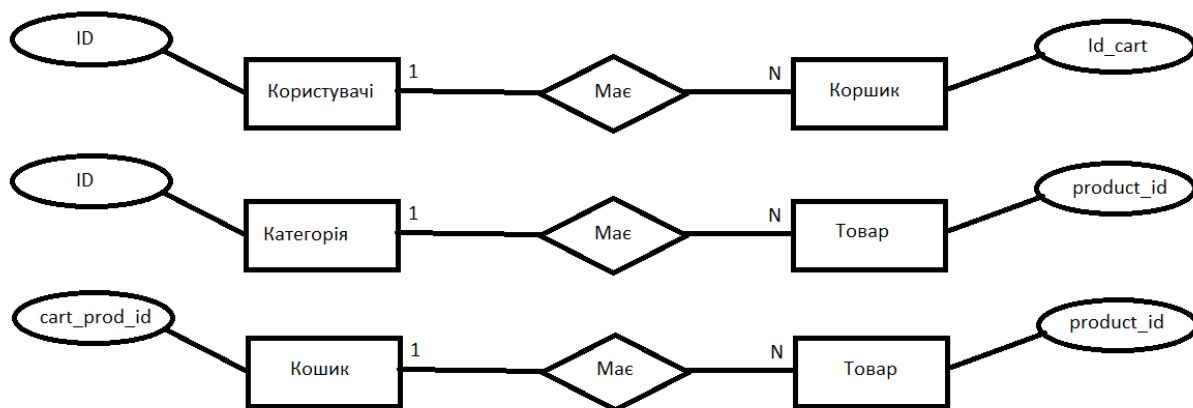
27. Гандзюк М. П., Желібо Є. П., Халімовський М. О. Основи охорони праці : підручник / за ред. М. П. Гандзюка. К. : Каравела, 2008 . 384 с.

28. ДСТУ 8604 : 2015 р. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги (дата звернення 10.06.2023).

ДОДАТКИ

Додаток А

ER-діаграма бази даних «shop_db»



Додаток Б

Лістинг Index.php

```

<?php
require_once("db_connection.php");
require_once("db_connectionAuth.php");

if(isset($_GET["sort"]))
{
    $sorting = $_GET["sort"];
}
else{
    $sorting = NULL;
}
switch($sorting)
{
    case'price-asc':
        $sorting = 'price ASC';
        $sort_name = 'Дешево-Дорого';
        break;
    case'price-desc':
        $sorting = 'price DESC';
        $sort_name = 'Дорого-Дешево';
        break;
    case'popular':
        $sorting = 'count DESC';
        $sort_name = 'Популярне';
        break;
    case'news':
        $sorting = 'datetime DESC';
        $sort_name = 'Новинки';
        break;
    case'brand':
        $sorting = 'title ASC';
        $sort_name = 'Від А до Я';
        break;
    default:
        $sorting = 'products_id DESC';
        $sort_name = 'Без сортування';
}

?>
<?php if(isset($_SESSION['logged_user']))?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang = "en" lang="en">
<head>
    <meta charset="UTF-8">
    <link href="https://fonts.googleapis.com/css2?family=Gotu&display=swap"
rel="stylesheet">
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <link href="css/reset.css" rel="stylesheet" type="text/css">

```


Продовження Додатку Б

```

<link href="css/style.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="js/jquery-1.8.2.min.js"></script>
<script type="text/javascript" src="js/jcarousellite_1.0.1.js"></script>
<script type="text/javascript" src="js/shop-script.js"></script>
<title>Інтернет магазин меблів</title>
</head>
<body>
  <div id="block-body">
    <?php
      require_once("include/block-header.php");
    ?>
    <div id="block-content">
      <div id="block-sorting">
        <p id="nav-breadcrumbs"><a href="index.php">Головна сторінка</a> \ <span>Всі
товари</span></p>
          <ul id="options-list">
            <li>Вид:</li>
            <li></li>
            <li></li>
            <li>Сортування:</li>
            <li><a id="select-sort"><?php echo $sort_name;?></a>
              <ul id="sorting-list">
                <li><a href="index.php?sort=price-asc">Дешево-
Дорого</a></li>
                <li><a href="index.php?sort=price-desc">Дорого-
Дешево</a></li>
                <li><a href="index.php?sort=popular">Популярне</a></li>
                <li><a href="index.php?sort=news">Новинки</a></li>
                <li><a href="index.php?sort=brand">Від А до
Я</a></li>
              </ul>
            </li>
          </ul>
        </div>
      <ul id="block-tovar-grid">
        <?php
          $num = 12;
          $page = isset($_GET['page']) ? $_GET['page'] : 1;
          $offset = $num * ($page - 1);
          /*$count = mysqli_query("SELECT COUNT(*) FROM table_products WHERE visible = '1'",
$link);
          $temp = mysqli_fetch_array($count);
          if ($temp[0] > 0)
          {
            $tempcount = $temp[0];

            $total = (($tempcount - 1) / $num) + 1;
            $total = intval($total);

            $page = intval($page);
            if(empty($page) or $page < 0) $page = 1;
            if($page > $total) $page = $total;
            $start = $page * $num - $num;
            $qry_start_num = "LIMIT $start, $num";

```

Продовження Додатку Б

```

}*/

$result = mysqli_query($link, "SELECT * FROM table_products WHERE visible='1' ORDER
BY $sorting LIMIT $num OFFSET $offset");
if (mysqli_num_rows($result) > 0)
{
    $row = mysqli_fetch_Array($result);
    do
    {
        if ($row["image"] != "" /*&&
file_exists("../uploads_images/".$row["image"]*/)
        {
            $img_path = $row["image"];
            $max_width = 200;
            $max_height = 200;
            list($width, $height) = getimagesize($img_path);
            $ratioh = $max_height/$height;
            $ratiow = $max_width/$width;
            $ratio = min($ratioh, $ratiow);
            $width = intval($ratio*$width);
            $height = intval($ratio*$height);
        }else
        {
            $img_path = "../img/no-image.png";
            $width = 110;
            $height = 200;
        }
        echo'

        <li>
        <div id="image" class="block-images-grid">
        <a href="#"></a>
        </div>
        <p class="style-title-grid"><a
href="post.php?post_id='.$row["products_id"].'">'.$row["title"].'</a></p>
        <ul class="reviews-and-counts-grid">
        <li><p>'.$row["view"].'</p></li>

Продовження Додатку Б

        <li><p>'.$row["comment"].'</p></li>
        </ul>

        <a href="#" class="btn btn-outline-success"> В корзину</a>
        <p class="style-price-
grid"><strong>'.$row["price"].'</strong>грн.</p>
        <div class="mini-features">
        </div>
        </li>
        ';
    }
    while ($row = mysqli_fetch_Array($result));
}
else
{

```

Продовження Додатку Б

```

    }
?>
</ul>
<ul id="block-tovar-list">
<?php

$result = mysqli_query($link, "SELECT * FROM table_products");
if (mysqli_num_rows($result) > 0)
{
    $row = mysqli_fetch_Array($result);

    do
    {
        if ($row["image"] != "" /*&&
file_exists("../uploads_images/".$row["image"]*/)
        {
            $img_path = $row["image"];
            $max_width = 200;
            $max_height = 200;
            list($width, $height) = getimagesize($img_path);
            $ratioh = $max_height/$height;
            $ratiow = $max_width/$width;
            $ratio = min($ratioh, $ratiow);
            $width = intval($ratio*$width);
            $height = intval($ratio*$height);
        }else
        {
            $img_path = "../img/no-image.png";
            $width = 110;
            $height = 200;
        }
        echo'
        <li>
        <div class="block-images-list">
        
        </div>
            <ul class="reviews-and-counts-list">
            <li><p>'. $row["view"].'</p></li>
            <li><p>'. $row["comment"].'</p></li>
            </ul>
            <p class="style-title-list"><a
href="post.php">'. $row["title"].'</a></p>
            <a class="btn btn-dark" value="Знайти"></a>
            <p class="style-price-
list"><strong>'. $row["price"].'</strong>грн.</p>
            <div class="style-text-list">
            </div>
        </li>
        ';
    }
    while ($row = mysqli_fetch_Array($result));
}

?>
</ul>

```

Продовження Додатку Б

```
</div>
<div id="block-right">
<?php
    require_once("include/block-category.php");
    require_once("include/block-parameter.php");
    require_once("include/block-news.php");
?>
</div>
<?php
    require_once("include/block-footer.php");
?>
</div>

</body>
</html>
```

Додаток В

Лістинг Login.php

```

<?php
    require_once("db_connectionAuth.php");

    $data = $_POST;
    if(isset($data['do_login']))
    {
        $errors = array();
        $user = R::findOne('user', 'login = ?', array($data['lg_username']));
        if ($user)
        {
            if(password_verify($data['lg_password'] = "ADMIN", $user->password)){
                $_SESSION['logged_user'] = $user;
                header('Location: adminMenu.php');
            }
            if(password_verify($data['lg_password'], $user->password)){
                $_SESSION['logged_user'] = $user;
                header('Location: index.php');
            }
            else
            {
                $errors[] = 'От халепа... Такий пароль відсутній!!!';
            }
        } else
        {
            $errors[] = 'Ой, якась помилка... Це точно ваш Логін?';
        }
    }
    if ( ! empty($errors))
    {
        echo '<div style="color: red;">'.array_shift($errors).'

```

Продовження Додатку В

```

    <div class="login-form-1">
        <form action="login.php" method="post" id="login-form" class="text-left">
            <div class="login-form-main-message"></div>
Продовження Додатку В
                <div class="main-login-form">
                    <div class="login-group">
                        <div class="form-group">
                            <label for="lg_username" class="sr-only">Ім'я
користувача</label>
                                <input type="text" class="form-control"
id="lg_username" name="lg_username" placeholder="Логін">
                            </div>
                        <div class="form-group">
                            <label for="lg_password" class="sr-
only">Пароль</label>
                                <input type="password" class="form-control"
id="lg_password" name="lg_password" placeholder="Пароль">
                            </div>
                        <div class="form-group login-group-checkbox">
                            <input type="checkbox" id="lg_remember"
name="lg_remember">
                                <label for="lg_remember">Запам'ятати</label>
                            </div>
                    </div>
                    <button name="do_login" type="submit" class="login-button"><i
class="fa fa-chevron-right"></i></button>
                </div>
                <div class="etc-login-form">
                    <p>Забули пароль? <a href="#forgot-password-form">Натисніть
тут</a></p>
                    <p>Новенький? <a href="registration.php">створити новий
акаунт</a></p>
                </div>
            </form>
        </div>
        <!-- end:Main Form -->
Продовження Додатку В
    </div>

<!-- FORGOT PASSWORD FORM -->
<div class="text-center" style="padding:50px 0">
    <div class="logo">Забули пароль?</div>
    <!-- Main Form -->
    <div class="login-form-1">
        <form method="post" id="forgot-password-form" class="text-left">
            <div class="etc-login-form">
                <p></p>
            </div>
            <div class="login-form-main-message"></div>
            <div class="main-login-form">
                <div class="login-group">
                    <div class="form-group">
                        <label for="fp_email" class="sr-only">Email
адреса</label>

```

Продовження Додатку В

```

                                <input      type="text"      class="form-control"
id="fp_email" name="fp_email" placeholder="Введіть свій E-mail">
                                </div>
                                </div>
                                <button  type="submit"  class="login-button"><i  class="fa  fa-
chevron-right"></i></button>
                                </div>
                                </form>
                                </div>
                                <!-- end:Main Form -->
</div>
```

Додаток Г
Диск з програмою