

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Проектування та розробка сайту інтернет-магазину декору
з використанням фреймворку Django

Виконав(ла): студент(ка) VI курсу, групи СПмз-61
спеціальності _____

121 інженерія програмного забезпечення

(шифр і назва спеціальності)

	_____	Бойко О.І.
	(підпис)	(прізвище та ініціали)
Керівник	_____	Цуприк Г.Б.
	(підпис)	(прізвище та ініціали)
Нормоконтроль	_____	Стоянов Ю.М.
	(підпис)	(прізвище та ініціали)
Завідувач кафедри	_____	Петрик М.Р.
	(підпис)	(прізвище та ініціали)
Рецензент	_____	Жаровський Р.О.
	(підпис)	(прізвище та ініціали)

АНОТАЦІЯ

Атестаційна робота магістра містить: 71 с., 23 рис., 0 табл., 28 джер.

БАЗА ДАНИХ, ВЕБРОЗРОБКА, ІНТЕРНЕТ-МАГАЗИН, КЛІЄНТСЬКИЙ ІНТЕРФЕЙС, DJANGO, MVC, POSTGRESQL.

Метою кваліфікаційної роботи є проектування та розробка сайту інтернет-магазину декору з використанням фреймворку Django, який забезпечує високу продуктивність, безпеку та зручність використання.

Методи розробки базуються на використанні архітектури MVC, яка забезпечує розподіл функціональності між клієнтською, серверною частинами та адміністративним інтерфейсом.

В результаті роботи було створено функціональний вебдодаток з використанням фреймворку Django на мові програмування Python із динамічним клієнтським інтерфейсом, інтеграцією з базою даних PostgreSQL та системою адміністрування. Для реалізації дизайну використано HTML, CSS і Bootstrap, що забезпечило адаптивність і зручність користувацького інтерфейсу.

Додаток був протестований і відповідає всім вимогам до електронної комерції. Практичне значення роботи полягає у можливості застосування розробленого рішення для створення подібних інтернет-магазинів, адаптованих під різні бізнес-моделі.

ABSTRACT

The master's thesis contains: pages. – 71, pictures. – 23, tables. – 0, bibl.ref. – 28.

CLIENT INTERFACE, DATABASE, DJANGO, MVC, ONLINE STORE, POSTGRESQL, WEB DEVELOPMENT.

The aim of the thesis is the design and development of an online decor store website using the Django framework, which provides high performance, security and ease of use.

The development methods are based on the use of the MVC architecture, which ensures the division of functionality between the client, server parts and the administrative interface.

As a result of the work, a functional web application was created using the Django framework in the Python programming language with a dynamic client interface, integration with the PostgreSQL database and the administration system. HTML, CSS and Bootstrap were used to implement the design, which ensured the adaptability and convenience of the user interface.

The application has been tested and meets all requirements for e-commerce. The practical significance of the work lies in the possibility of applying the developed solution to create similar online stores adapted to different business models.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ СКОРОЧЕНЬ	8
ВСТУП.....	9
1 ОСНОВНІ ОСОБЛИВОСТІ ВЕБ-РЕСУРСІВ ТА ТЕХНОЛОГІЇ РОБОТИ МАРКЕТПЛЕЙСІВ	12
1.1. Основні аспекти веб-розробки	12
1.2. Основні цілі створення вебсайту для електронної торгівлі	13
1.4. Огляд існуючих інструментів веброзробки та обґрунтування вибору фреймворку Django	21
2 АНАЛІЗ ТА ОБҐРУНТУВАННЯ ОБРАННЯ СИСТЕМ ТА МЕТОДІВ РОЗРОБКИ ВЕБСАЙТУ	25
2.1. Розгляд та аналіз технічного завдання	25
2.2. Особливості проектування вебдизайну та структури сайту	26
2.3. Модель MVC	29
2.4. Загальна характеристика та структура роботи з фреймворком Django	30
3 РОЗРОБКА АРХІТЕКТУРИ МЕК ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТНОГО РІШЕННЯ.....	33
3.1. Обрання програмного забезпечення	33
3.2. Опис алгоритму роботи інтернет-магазину та моделювання бази даних ..	35
3.3. Розробка та оптимізація дизайну сайту магазину декору	39
3.4. Реалізація моделі MVC для розробки серверної, клієнтської та адмінчастини сайту	45
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	50
4.1. Охорона праці при роботі з комп'ютерними пристроями	50
4.2. Безпека в надзвичайних ситуаціях.....	53
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	62
ДОДАТКИ.....	65

Додаток А – Публікація у науковому виданні	66
Додаток Б – Алгоритм роботи Корзини.....	68
Додаток В – Алгоритм роботи інтернет-магазину.....	69
Додаток Г – Діаграма прецедентів	70
Додаток Д – Диск із кваліфікаційною роботою магістра.....	71

ПЕРЕЛІК ОСНОВНИХ СКОРОЧЕНЬ

ПЗ – програмне забезпечення;
ТЗ – технічне завдання;
MVC – Модель-Вигляд-Контролер;
ORM – Об’єктно-реляційне відображення;
БД – база даних;
HTML – Мова розмітки гіпертексту;
CSS – Каскадні таблиці стилів;
JS – JavaScript, мова програмування;
Django – фреймворк Python для веброзробки;
PostgreSQL – система управління базами даних;
HTTP – Протокол передачі гіпертексту;
JSON – Формат обміну даними.

ВСТУП

Сучасні тенденції цифровізації бізнесу сприяють швидкому розвитку електронної комерції. Інтернет-магазини стали важливим інструментом для представлення товарів і послуг у цифровому середовищі, що забезпечує їх доступність для широкої аудиторії. Проте розробка якісного інтернет-магазину вимагає застосування сучасних технологій та фреймворків, які гарантують високу продуктивність, безпеку та зручність використання. Фреймворк Django є одним із провідних інструментів у сфері веб-розробки, що дозволяє швидко створювати масштабовані веб-додатки, забезпечуючи їхню ефективність і функціональність.

Тема роботи є актуальною через зростаючі вимоги до якості веб-ресурсів у сфері електронної комерції. Інтернет-магазин декору, створений на базі Django, відповідає сучасним викликам завдяки своїй надійній архітектурі, інтеграції з базами даних, захисту даних користувачів та зручності адміністрування. Використання Django значно спрощує розробку та скорочує час на впровадження, що є критично важливим для бізнесу.

Робота виконана в межах наукових досліджень кафедри програмної інженерії, які спрямовані на впровадження новітніх технологій у розробку веб-додатків. Виконуваний проєкт узгоджується з напрямками досліджень, що стосуються створення програмних рішень для автоматизації бізнес-процесів та розвитку електронної комерції.

Метою роботи є розробка інтернет-магазину декору на основі фреймворку Django з використанням моделі MVC, інтеграцією бази даних та забезпеченням функціонального адміністративного інтерфейсу.

Для реалізації цієї мети були визначені такі завдання:

1. Провести аналіз інструментів веб-розробки та обґрунтувати вибір Django для реалізації проєкту.
2. Розробити архітектуру інтернет-магазину на основі моделі MVC.

3. Реалізувати серверну частину додатка для обробки даних користувачів та управління товарами.
4. Створити клієнтський інтерфейс із динамічним відображенням даних.
5. Розробити адміністративний модуль для управління контентом магазину.
6. Виконати тестування та оптимізацію роботи веб-додатка.

Над питаннями створення та покращення функціональності вебзастосунків працювала велика кількість українських та іноземних спеціалістів, зокрема таких, як Я. Фенвік, Дж. Шах, К. Вертайм, М. Котін, Р. Гавриш Ф. Вірін, О.А. Петрик, та ін.. Таким чином, дана розробка вміщує інформацію про послідовність та основні етапи створення вебсайту для інтернет-магазину, який повинен містити дані про сам маркетплейс, про особливості її товарів, ціни на них, контактну інформацію, новини магазину тощо. Окрім того, каталог товарів матиме як опис самого товару, так і фото всіх продуктів. Також вебсайт повинен швидко відшукуватися пошуковиками та легко і стабільно завантажуватись. Його дизайн повинен бути простий, легкозрозумілий та інтуїтивний.

Процес розробки веб-додатків для електронної комерції.

Предмет дослідження є фреймворк Django як інструмент для проєктування та створення інтернет-магазинів.

Для досягнення мети роботи використані такі методи:

- Аналіз літературних джерел та наявних інструментів веброзробки.
- Проєктування архітектури інтернет-магазину з використанням моделі MVC.
- Тестування веб-додатка для забезпечення його стабільності та відповідності функціональним вимогам.

У роботі реалізовано інтернет-магазин декору у середовищі VS Code з використанням фреймворку Django, що поєднує динамічний клієнтський інтерфейс із зручним адміністративним модулем. Використання PostgreSQL як бази даних дозволило забезпечити високу продуктивність і масштабованість.

Результати роботи можуть бути використані для розробки аналогічних веб-додатків у сфері електронної комерції. Інтернет-магазин декору, створений у межах

цієї роботи, може бути інтегрований у реальне середовище з мінімальними модифікаціями.

Результати досліджень опубліковані в науковій статті та представлені у вигляді тез на XII науково-технічній конференції: Інформаційні моделі, системи та технології

Отже, магістерська робота дозволяє чіткіше охарактеризувати та здійснити аналіз проблеми проєктування та самої розробки сайту інтернет-магазину, акцентуючи увагу на критеріях, які визначають актуалізацію та доцільність вибору технології та інструментів, які будуть використовуватись.

Структура дослідження сформована на основі вищезазначених положень: метою і завданнями та складається із вступу, основної частини, висновків, списку використаної літератури та додатків.

1 ОСНОВНІ ОСОБЛИВОСТІ ВЕБ-РЕСУРСІВ ТА ТЕХНОЛОГІЇ РОБОТИ МАРКЕТПЛЕЙСІВ

1.1. Основні аспекти веб-розробки

Веб-розробка охоплює процес створення веб-сайтів для Інтернету або приватних мереж (інтранет). Її масштаби можуть варіюватися від створення простої статичної сторінки з текстом до розробки складних веб-додатків для електронної комерції чи соціальних мереж. До веб-розробки часто відносять такі завдання, як веб-інжиніринг, дизайн веб-сторінок, створення контенту, роботу з клієнтами, написання клієнтських і серверних сценаріїв, налаштування веб-серверів, забезпечення безпеки мережі та розвиток електронної комерції. У професійному середовищі цей термін зазвичай стосується технічних аспектів, таких як написання коду та створення розмітки. Останнім часом веб-розробка також включає створення систем управління контентом (CMS) [6].

Ці CMS можуть бути створені з нуля, фірмового або з відкритим вихідним кодом. Загалом, CMS діє як проміжне програмне забезпечення між базою даних та користувачем через браузер. Основною перевагою CMS є те, що це дозволяє нетехнічним людям внести зміни до свого веб-сайту без технічних знань.

У великих організаціях і компаніях команди веб-розробників можуть складатися зі сотень фахівців, які працюють за стандартними підходами, такими як методологія Agile, при створенні веб-сайтів. У менших компаніях зазвичай достатньо одного штатного чи контрактного розробника, або ж робота може частково передаватися іншим спеціалістам, наприклад, графічним дизайнерам чи технікам інформаційних систем. Веб-розробка часто є результатом співпраці різних відділів, а не обмежується лише одним підрозділом. Сучасна веб-розробка включає три основні напрями спеціалізації: фронтенд-розробник, бекенд-розробник і розробник повного стека [8].

Фронтенд-розробники працюють над дизайном та візуальним представленням веб-сайту, тоді як бекенд-розробники забезпечують його

функціональність. Вони відповідають за програмування механізмів, які обробляють і зберігають дані.

1.2. Основні цілі створення вебсайту для електронної торгівлі

З моменту комерціалізації Інтернету веб-розробка перетворилася на стрімко зростаючу галузь. Її розвиток стимулюється бізнесами, які прагнуть використовувати свої веб-сайти для продажу товарів і надання послуг клієнтам.

Існує безліч інструментів із відкритим кодом для створення веб-сторінок, таких як BerkeleyDB, GlassFish, LAMP (Linux, Apache, MySQL, PHP) і Perl / Plack, які суттєво знизили витрати на навчання веб-розробці. Додатковим стимулом для розвитку галузі стало зростання популярності програмного забезпечення для веб-розробки типу WYSIWYG, наприклад, Adobe Dreamweaver, BlueGriffon і Microsoft Visual Studio. Хоча для роботи з такими програмами все ще необхідно знати основи мови розмітки HTML або мов програмування, базові навички можна опанувати швидко і почати їх застосовувати.

Постійний розвиток інструментів і технологій дав змогу розробникам створювати більш інтерактивні та динамічні веб-сайти. Крім того, веб-розробники сприяли перетворенню багатьох програм у веб-сервіси, які раніше були доступні лише у вигляді десктопних додатків. Це відкрило нові можливості для децентралізації інформації та поширення контенту. Яскравими прикладами є хмарні сервіси, такі як Adobe Creative Cloud, Dropbox і Google Drive, які дозволяють користувачам працювати з додатками з будь-якої точки, а не залежати від однієї робочої станції для їх використання.

Веб-розробка спричинила значні зміни у сфері комунікації та торгівлі, зокрема через розвиток електронної комерції. Інтернет-аукціони, такі як eBay, докорінно змінили підхід споживачів до пошуку й придбання товарів і послуг. Онлайн-магазини, такі як Amazon.com і Buy.com, революціонізували процес

здійснення покупок, створивши новий досвід для мільйонів користувачів по всьому світу.

Ще одним прикладом значних змін у сфері комунікації, спричинених веб-розробкою, є блоги. Завдяки веб-додаткам, таким як WordPress і Movable Type, з'явилися зручні платформи для створення та ведення блогів. Поширення систем управління контентом з відкритим кодом і корпоративних CMS значно розширило можливості веб-розробки, підвищивши ефективність взаємодії та комунікації в Інтернеті.

Розробка веб-сайтів також значно вплинула на особисті мережі та маркетинг. Тепер веб-сайти не лише інструменти для роботи або бізнесу, а й платформи для спілкування та соціальних взаємодій. Сайти, як-от Facebook і Twitter, надають користувачам можливість комунікувати та організовуватися, створюючи більш особисті й інтерактивні способи залучення аудиторії.

У цифрову епоху люди стали більш залежними від розумних гаджетів, ніж будь-коли раніше. Одночасно галузь електронної комерції стрімко розвивається. За даними Statista, близько 60% людей сьогодні витрачають свої гроші на покупки в інтернет-магазинах, що свідчить про величезну частку інтернет-покупців серед веб-користувачів по всьому світу. Саме тому компанії все частіше інвестують у розробку веб-сайтів для електронної комерції, щоб збільшити свої продажі.

У цьому розділі розглядаються 5 важливих аспектів, які слід врахувати перед розробкою веб-сайту електронної комерції. Першим і одним із найбільш важливих факторів є вибір правильної платформи електронної комерції та гарної теми для сайту. Веб-розробники повинні обрати платформу, яка найкраще відповідає цілям розвитку проекту. Існує кілька популярних платформ для електронної комерції, таких як WooCommerce, Drupal, Magento, Shopify, OSCommerce, Joomla та інші, що дозволяє зробити обґрунтований вибір. Правильна платформа допоможе значно зекономити час і гроші в процесі розробки.

Крім того, вибір правильної теми для веб-сайту має велике значення для розвитку електронної комерції [3]. Це критично важливо для успіху будь-якого онлайн-бізнесу. За даними статистики W3Tech, близько 25% всіх веб-сайтів

електронної комерції не досягають бажаних результатів через невдалий вибір дизайну теми для свого сайту.

Багато веб-сайтів електронної комерції інтегрують функціональність блогу або портфоліо для збільшення взаємодії з користувачами. Це також допомагає компаніям підтримувати довгострокові відносини зі своїми клієнтами. Крім того, Google надає перевагу сайтам, які містять блоги, що може позитивно вплинути на рейтинг вашого веб-сайту електронної комерції. Таким чином, регулярне оновлення контенту через блог може сприяти покращенню видимості сайту в пошукових системах.

SEO (пошукова оптимізація) відіграє ключову роль у розробці веб-сайтів електронної комерції [8]. Для запуску успішного бізнесу в електронній комерції важливо залучити потенційних клієнтів, і саме SEO допомагає збільшити продажі на вашому веб-сайті. З правильними стратегіями SEO ваш сайт зможе зайняти високі позиції на конкурентному ринку. Крім того, якщо користувач шукає ваш сайт в пошукових системах, він з'явиться на перших позиціях завдяки ефективній SEO-оптимізації. Це сприяє підвищенню видимості сайту та збільшенню конверсій.

Таким чином, чим вищу позицію рейтингу займає ваш веб-сайт електронної комерції, тим більше клієнтів автоматично потрапляють на нього. За даними дослідження Statista, близько 35 найбільших веб-сайтів електронної комерції не потрапляють до першої десятки результатів пошукових систем. Це означає, що користувачі рідко потрапляють на ці сайти, оскільки вони не з'являються на видимих позиціях пошукових запитів.

Responsive Design (Адаптивний дизайн) набуває все більшої популярності, оскільки кількість мобільних користувачів зростає, а вебсайти електронної торгівлі отримують великий обсяг трафіку. Це вказує на значну кількість людей, які здійснюють покупки в онлайн-магазинах через мобільні пристрої. За статистикою Mediabistro, 50% продажів вебсайтів електронних продажів здійснюються через мобільні телефони. Тому оптимізація веб-сайту для мобільних пристроїв, а також

для настільних систем, стає необхідною для забезпечення зручного користувацького досвіду на всіх платформах.

Адаптивний веб-дизайн може значно допомогти в досягненні цієї мети, роблячи ваш веб-сайт електронної комерції доступним для різних типів пристроїв. Це спрощує процес розробки для веб-розробників [2]. Крім того, чуйний веб-сайт є важливим інструментом для досягнення успіху вашого веб-сайту електронної комерції, забезпечуючи зручний і ефективний досвід користувачів на всіх платформах.

Хороша маркетингова стратегія є важливою для успішного просування вашого вебсайту електронної торгівлі після його розробки. Для цього слід застосовувати ефективні маркетингові методи, щоб залучити все більше клієнтів. Існує багато способів виходу на ринок для вашого електронного бізнесу, і завдяки надійному маркетинговому агентству ви зможете ідеально просувати свій сайт, забезпечивши його успіх.

Кожен маркетинговий метод має свої переваги та недоліки, але з правильною маркетинговою стратегією та ефективним дизайном ваш бізнес електронної комерції може досягти значного успіху. Комбінація обох факторів допоможе залучити більше клієнтів і забезпечить стабільне зростання вашого онлайн-бізнесу.

Безпека є важливим аспектом для кожного розробника веб-сайтів електронної комерції, оскільки вона допомагає створити довіру серед клієнтів. Існують різні методи захисту, і одним із найважливіших є підтримка SSL (Secure Socket Layer), яка забезпечує шифрування чутливої інформації [7]. Це надзвичайно важливо для користувачів, які вводять платіжні дані, такі як номери дебетових та кредитних карток, адреси, імена, номери телефонів, електронні адреси тощо, оскільки це гарантує їхню безпеку під час обробки на вашому сайті.

Відповідність стандартам PCI (Payment Card Industry) є ще одним важливим методом забезпечення безпеки для підприємств електронної комерції, які приймають цифрові платежі. Стандарти PCI допомагають захищати платіжну інформацію та знижують ризики шахрайства. Згідно з опитуванням компанії Deloitte, близько 40% веб-сайтів електронної комерції відповідають цим

стандартам, що підкреслює важливість їх впровадження для забезпечення безпеки та довіри клієнтів.

1.3. Аналіз аналогічних проєктів та етапи розробки сайту інтернет-магазину

Процес створення сайту інтернет-магазину є складним і вимагає чіткої організації етапів для забезпечення якості. Ось покрокова послідовність для створення такого сайту:

1. Основна концепція сайту: На цьому етапі визначають основну ідею сайту, його цілі та функціональні можливості. Якщо чітко поставлені цілі та враховані потреби замовника, створення стилю, інтерфейсу та розділів буде значно простішим.

2. Технічне завдання (ТЗ): Визначення структури та архітектури сайту, основних компонентів і функцій, таких як окремі сторінки та розділи. Важливо також передбачити CMS для управління контентом, що дозволить замовнику здійснювати зміни без залучення розробників.

3. Технологічна архітектура: Оптимізація технічних аспектів сайту, врахування запитів клієнта та узгодження з бюджетною політикою проєкту.

4. Дизайн веб-сайту: Оформлення сайту, яке повинно бути оригінальним, стильним і привабливим для користувачів. Важливо вибрати дизайнерські рішення, що сприятимуть утриманню відвідувачів на сайті.

5. Програмування: Розробка коду та скриптів, які реалізують функціональність сайту, роблячи його зручним і зрозумілим для користувачів.

6. Наповнення сайту: Введення контенту за допомогою плагінів CMS, що дозволяє редагувати сторінки, додавати медіафайли, створювати гіперпосилання тощо.

7. Тестування: Перевірка сайту на помилки, відповідність вимогам замовника та зручність для кінцевих користувачів. Тестування повинно також включати перевірку на швидкість завантаження і сумісність з різними пристроями.

Для врахування всіх аспектів доцільно також проаналізувати структуру та функціональність інших подібних платформ. Наприклад, маркетплейси, такі як Prom.ua, Crafta.ua, Shafa.ua, Bigl.ua, показали зростання обсягу замовлень на 42% у 2020 році, зокрема, в категорії медичних товарів.

Так, розглянемо один з найбільших сайтів декору в Україні – Decorize [12]. Інтерфейс сайту продемонстровано на рисунку 1.1.

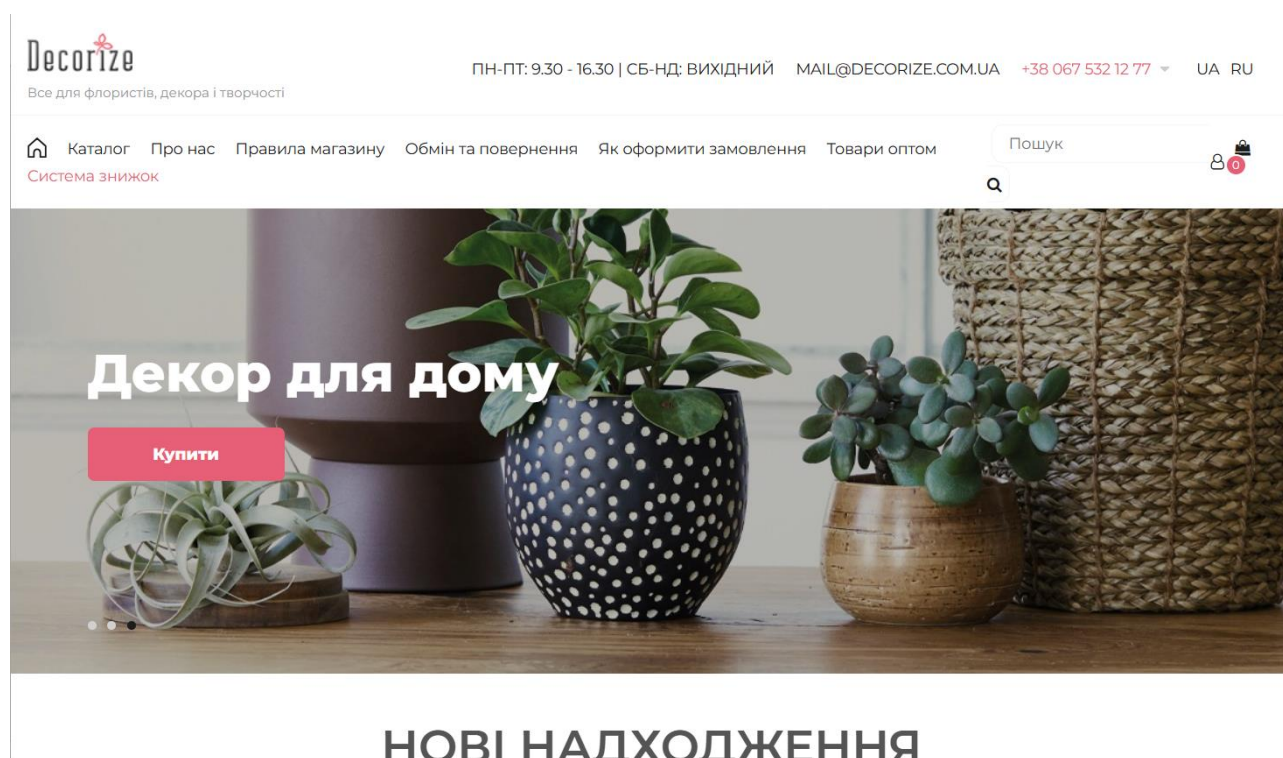


Рис. 1.1 – Інтерфейс та наповненість сайту Decorize [12]

Так, як можемо побачити із скриншота, інтерфейс сайту досить простий і приємний. Особливостями побудови є наступні:

- широка, практично на третину екрану стрічка, на якій позмінно відображаються найосновніші категорії та пропозиції магазину;
- наступним блоком є нові надходження товарів;

- ще 1 блок – акційні пропозиції товарів та гарячі новинки, що дозволяє клієнту одразу сконцентрувати увагу на вигідних пропозиціях;
- у верхній частині сайту розташоване навігаційне меню, у якому розміщено посилання на каталог товарів та посилання на інформацію про магазин, доставку і т.д.;
- у меню каталог досить зручно обирати товари, адже ліворуч з’являється поділ на категорії та панель із зручним фільтром\$
- на верхній панелі розміщена пошукова стрічка, тобто навігація по сайту є досить зручною та зрозумілою;
- у випадяючому вікні праворуч є можливість зареєструватись, зайти в особистий кабінет, розглянути всі особисті дані, здійснити замовлення тощо. Також є всі контактні дані магазину;
- досить швидко завантажується.

Наступний великий онлайн магазин - 4sdecor.com.ua (див. рис. 1.2).

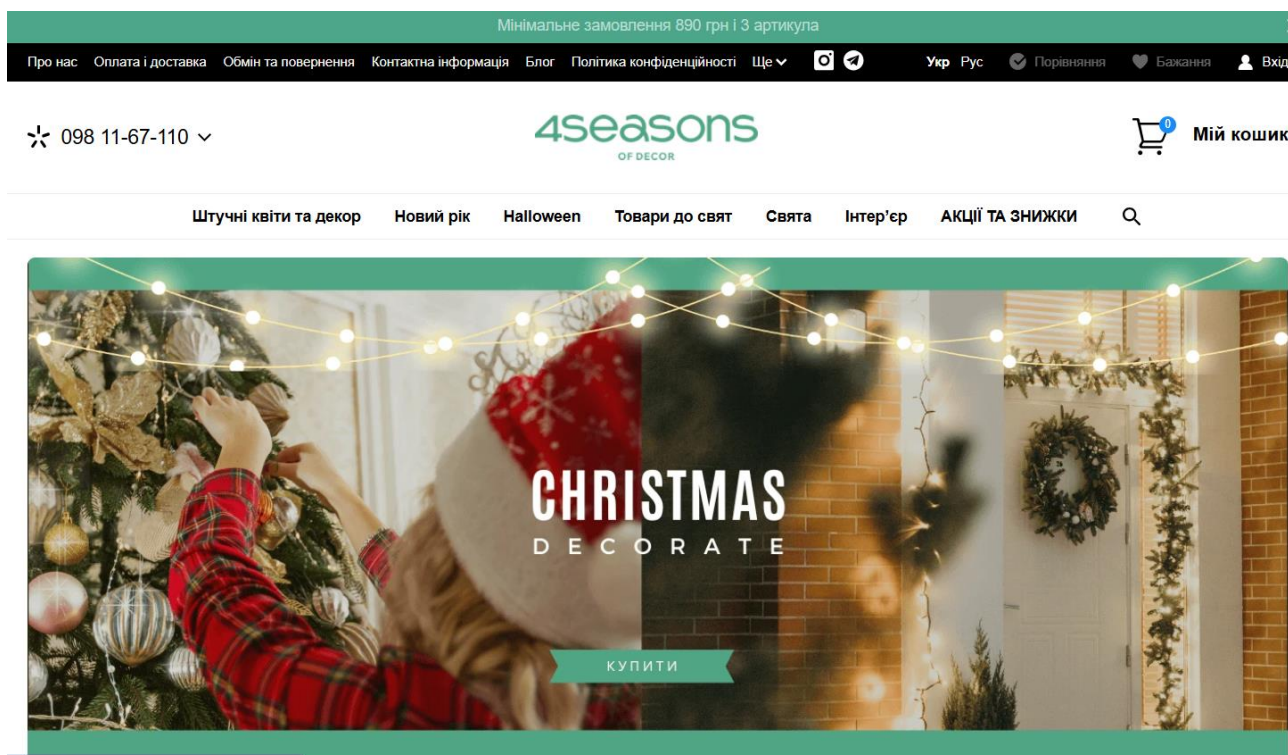


Рис. 1.2 – Інтерфейс та наповненість сайту 4sdecor

Особливостями сайту є:

- як і на попередньому сайті, 3/4 екрану займає стрічка із вигідними пропозиціями та анонсом розіграшів;
- знизу стрічка товарів, котрі користуються найбільшою популярністю та категорії, що знаходяться на знижках;
- навігація досить зручна, вгорі сторінки присутнє меню категорій зі зручною навігацією – при наведенні курсора на категорію розгортається додаткове підменю із детальнішим розбиттям на підкатегорії;
- присутня зручна пошукова панель;
- є особистий кабінет із всією необхідною для клієнта навігацією та зручний «Кошик»;
- наявність значної кількості способу оплати.

Сайт www.fdeco.eu є не таким популярним як два попередні. Інтерфейс простий та зрозумілий (див. рис. 1.3).

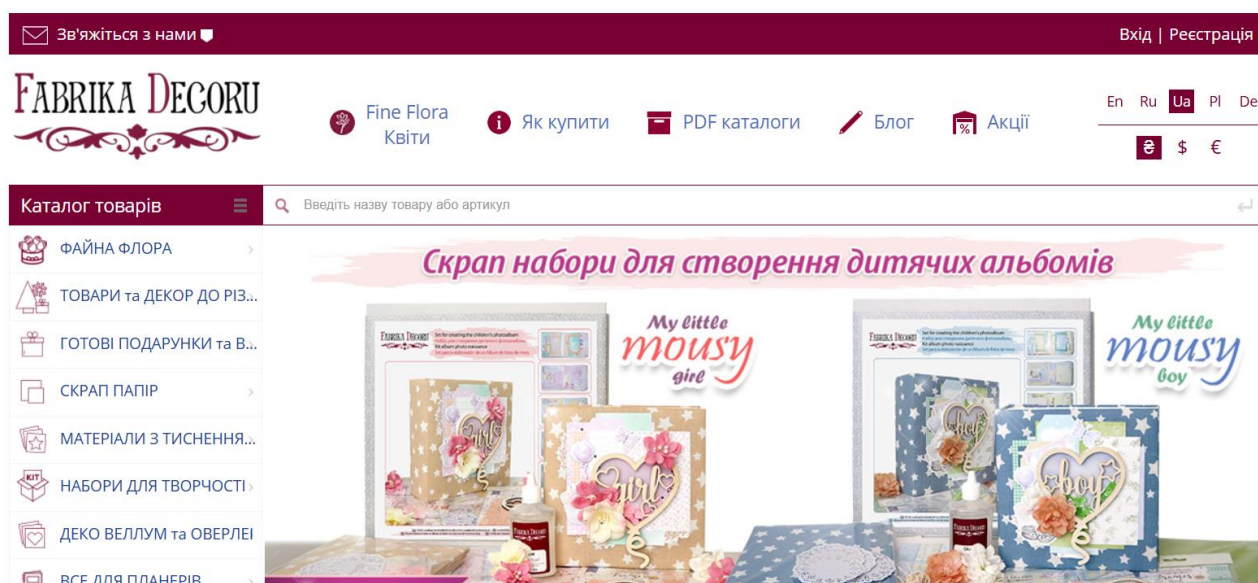


Рис. 1.3 – Інтерфейс та наповненість сайту Фабрика декору fdeco.eu

Слід зазначити, що даний інтернет-магазин є дещо простішим, в порівнянні з попереднім, на основній сторінці відображена практично вся інформація, немає ніяких прихованих випадуючих меню, дизайн дещо застарілий та однотипний. Навігаційні ефекти відволікають від пошуку потрібної інформації. Разом з тим,

присутні всі необхідні для реєстрацій, замовлення та пошуку товарів опції. Третину сторінки займає фото із рекламою, проте різниця в тому, що воно реалізоване звичайним слайдером який самостійно потрібно прогортувати.

Сайт Міс Декору missdecor.com.ua має теж простий і лаконічний дизайн, навігаційне меню, стрічку з найпопулярнішими пропозиціями на третину екрану, панель пошуку, корзину та посилання на особистий кабінет користувача (див. рис. 1.4).

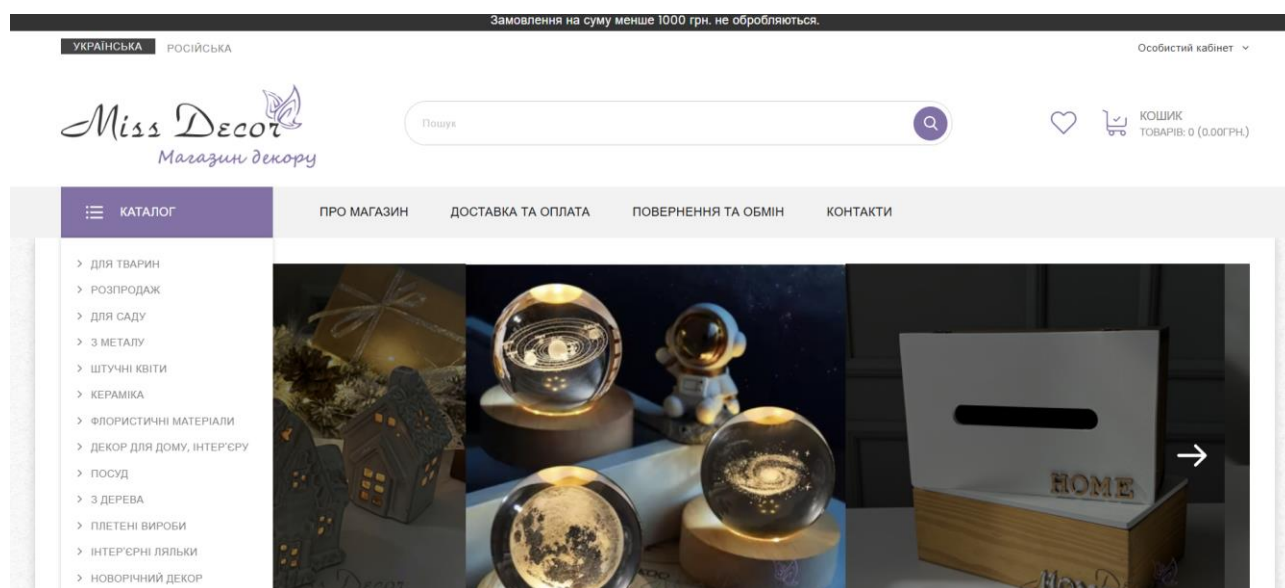


Рис. 1.4 – Інтерфейс та наповненість сайту missdecor.com.ua

1.4. Огляд існуючих інструментів веброзробки та обґрунтування вибору фреймворку Django

Веброзробка охоплює безліч аспектів: від роботи серверної частини та фронтенту до забезпечення безпеки, обробки даних і взаємодії з базами даних. Кожен інструмент або фреймворк має унікальні функціональні можливості, що дозволяє обрати оптимальний варіант для конкретного проекту. У цьому розділі буде проведено аналіз популярних інструментів веброзробки, розглянуто приклади

сайтів, створених за їх допомогою, а також обґрунтовано вибір фреймворку Django для створення сайту декору.

Сучасні інструменти веброзробки поділяються на кілька основних категорій: фреймворки для серверної частини (бекенд), бібліотеки та фреймворки для клієнтської частини (фронтенд), системи управління контентом (CMS), засоби інтеграції з базами даних, інструменти для створення користувацьких інтерфейсів, а також платформи для автоматизації та управління процесом розгортання. Нижче наведено приклади найбільш популярних інструментів у кожній з цих категорій.

1. Фреймворки серверної частини:

- Django (Python): Підходить для швидкого прототипування, забезпечує високу безпеку та підтримує масштабовані вебсайти. Наприклад, сайт Instagram використовує Django для обробки даних користувачів та інтеграції з базою даних.
- Laravel (PHP): Популярний вибір для проектів на PHP, Laravel забезпечує структурований підхід до коду та має широкий набір інструментів для побудови RESTful API. Laravel використовується для сайтів на кшталт Laracasts, платформи навчання з відеоуроками.
- Express.js (Node.js): Легкий фреймворк для створення високонавантажених додатків, особливо для реальних часом проектів. Trello використовує Node.js і Express.js для свого додатку через підтримку асинхронних процесів.

2. Фреймворки клієнтської частини:

- React (JavaScript): Бібліотека для побудови інтерфейсів користувача, особливо популярна для розробки односторінкових додатків. Наприклад, Facebook та Instagram використовують React для швидкого оновлення інтерфейсу без повного перезавантаження сторінки.
- Vue.js (JavaScript): Легка бібліотека для побудови інтерфейсів, яка має схожість із React, але з більшою простотою для новачків. Alibaba використовує Vue для побудови інтерфейсу користувача на сайті електронної комерції.

- Angular (TypeScript): Потужний фреймворк для складних додатків з динамічною логікою. Google використовує Angular для підтримки своїх внутрішніх і зовнішніх інструментів.
- 3. Системи управління контентом (CMS):
 - WordPress (PHP): Найпопулярніша CMS, зокрема для блогів та невеликих вебсайтів, але має обмежені можливості для великих проєктів. The New Yorker використовує WordPress для зручного управління контентом.
 - Drupal (PHP): Підходить для складних структурованих сайтів з високим рівнем налаштування. Використовується, наприклад, Harvard University для управління численними сторінками та контентом.

Сайт декору потребує ефективної обробки даних, високого рівня безпеки та масштабованої архітектури для інтеграції з базами даних, можливості керування великим обсягом контенту (наприклад, каталоги продукції, фотогалереї, форми зворотного зв'язку). Вибір фреймворку Django обумовлений наступними факторами:

1. Швидкість та ефективність розробки: Django слідує принципу «батарейки в комплекті», що означає наявність багатьох стандартних інструментів для веброзробки. Це дозволяє швидко реалізовувати функціональні можливості, що скорочує час розробки та витрати на проєкт.

2. Безпека: Django забезпечує захист від багатьох поширених вразливостей, таких як SQL-ін'єкції, міжсайтове підроблення запитів (CSRF) та міжсайтовий скриптинг (XSS) [15]. Враховуючи, що сайт декору буде обробляти дані користувачів, безпека є ключовою вимогою.

3. Масштабованість та продуктивність: Django дозволяє створювати масштабовані додатки, які можуть обробляти високі навантаження, завдяки використанню баз даних, таких як PostgreSQL. Це важливо для сайту, який передбачає велике завантаження зображень та інші ресурсоємні дані.

4. Підтримка та спільнота: Django має активну спільноту розробників та широкий спектр документації, що робить його зручним для початківців та дозволяє швидко знайти підтримку при вирішенні проблем.

5. Приклади успішних проєктів на Django: Django використовується багатьма великими компаніями, такими як Pinterest та National Geographic (див. рис. 1.5), які мають високі вимоги до обробки даних та стабільності роботи. Це підтверджує його надійність та можливості для роботи з проєктами різного масштабу.

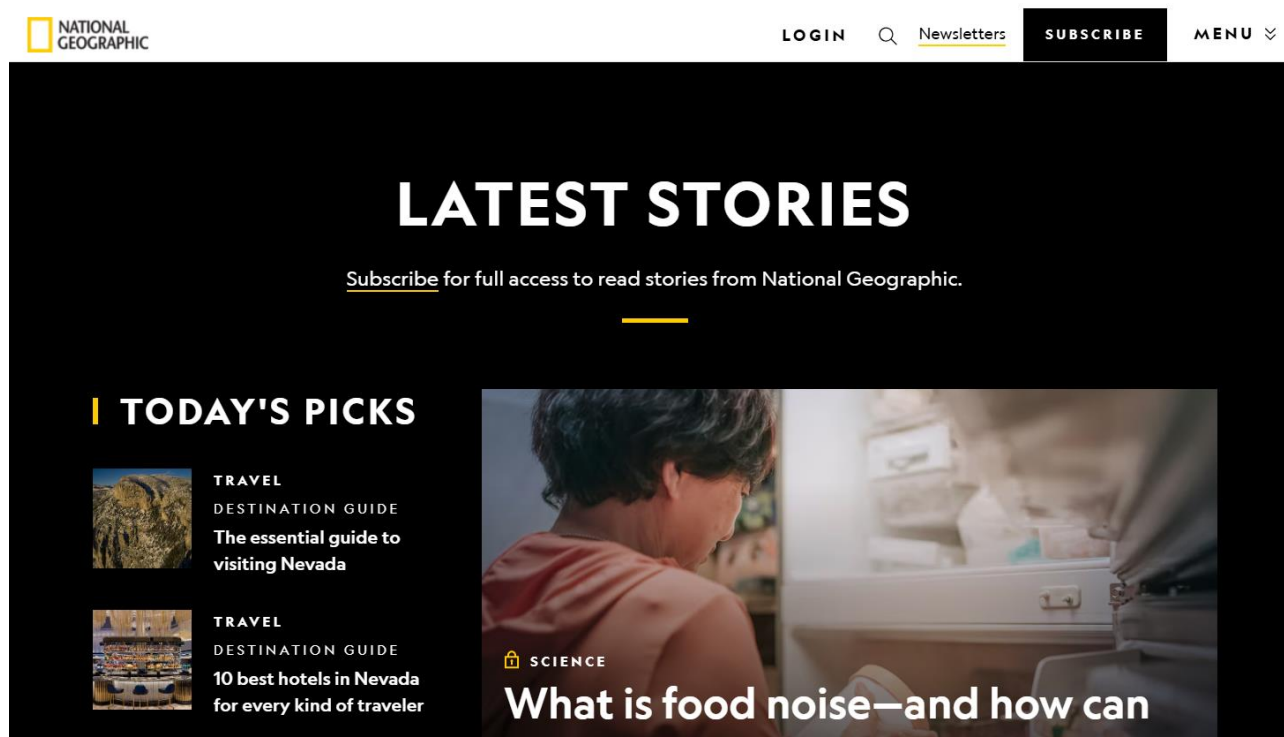


Рис. 1.5. Інтерфейс nationalgeographic.com [16].

Таким чином, враховуючи потреби сайту декору, фреймворк Django є оптимальним вибором для даного проєкту, оскільки він поєднує всі необхідні для проєкту функції: швидкість розробки, безпеку, масштабованість та наявність розвинутої спільноти.

2 АНАЛІЗ ТА ОБГРУНТУВАННЯ ОБРАННЯ СИСТЕМ ТА МЕТОДІВ РОЗРОБКИ ВЕБСАЙТУ

2.1. Розгляд та аналіз технічного завдання

У цьому розділі буде проведено аналіз технічних вимог, які необхідно врахувати під час створення інтернет-магазину та розробки технічного завдання для програмного продукту. Зокрема, потрібно розробити та впровадити модель веб-додатка, а також реалізувати сам сайт.

Таким чином, метою проєкту є розробка сайту для забезпечення можливості торгівлі продукцією декору. У зв'язку з цим визначимо основні вимоги до програмної документації. Варто зазначити, що на кожному етапі розробки програмного забезпечення створюються відповідні документи, що підкреслює важливість належного оформлення програмної документації, яка включає:

1. Специфікацію – перелік та призначення кожного файлу програмного продукту, включно з файлами документації.
2. Програму та текстові файли – коди програмного продукту з відповідними коментарями.
3. Програму та опис методів випробування – перелік і деталізація вимог, які перевіряються під час тестування, із зазначенням способів контролю.
4. Технічне завдання – опис призначення програми, сфер її застосування, рекомендації щодо використання програмного продукту, а також етапи і строки його розробки.
5. Пояснювальну записку – обґрунтування технічних і економічних рішень, прийнятих під час розробки, схеми алгоритмів дій і загальну характеристику роботи програми.

Також до цієї категорії документів належать семантика мови програмування, інструкції з використання, а також діагностично-тестові програми, що застосовуються під час обслуговування технічних засобів.

Процес розробки програмного забезпечення традиційно поділяється на три ключові етапи: технічне завдання – документ, який підлягає узгодженню та затвердженню. робоче проєктування – етап, що охоплює створення програмного продукту, його тестування та підготовку програмної документації. впровадження – фінальний етап, на якому продукт вводиться в експлуатацію [14].

Тестування програмного продукту виконується на етапі його створення за допомогою контрольних тестів, які можуть проводитися самостійно або із залученням тестувальника. Після завершення розробки продукту для його перевірки застосовуються додаткові тестові сценарії.

2.2. Особливості проєктування вебдизайну та структури сайту

У класичному розумінні веб-дизайн охоплює створення та графічне оформлення структури вебсайту, що забезпечує його зручність для користувачів. Основна мета веб-дизайну — візуальне представлення компанії та її діяльності. Унікальні дизайнерські рішення дозволяють виділити сайт серед конкурентів і привернути увагу відвідувачів. Таким чином, ключовими завданнями, які має виконати розробник під час проєктування веб-дизайну, є:

- Оригінальність вирішення — оформлення сайту повинно бути унікальним і виділяти його серед численних інших. Завдяки оригінальному дизайнерському підходу можна значно збільшити прибутковість компанії. Оригінальність проявляється у використанні цікавих графічних рішень в інтерфейсі, а також унікального кольорового оформлення, яке включає авторські творчі підходи.

- Привернення уваги — інтерфейс сайту має бути запам'ятовуваним та здатним привернути увагу широкої аудиторії. Це можна досягти поєднанням унікальних графічних елементів та оригінального, цікавого контенту. Водночас важливо пам'ятати, що сайт повинен бути зручним для сприйняття користувачем, тому варто уникати стандартних шаблонів текстів та надмірної деталізації.

– Інформативність — здатність сайту ефективно передавати необхідну інформацію покупцю. Це досягається завдяки правильно спроектованому та організованому контенту, який доповнюється візуальними елементами, такими як фото, відео тощо.

– Зручність користування досягається завдяки ретельно продуманій та чітко організованій навігаційній системі [5].

Отже, сайт повинен бути цілісним дизайнерським рішенням, яке відображає основну концепцію діяльності продавця, легко сприймається, є гнучким, функціональним, логічно побудованим та простим у використанні. Важливим є ретельне продумування структури сайту. Під структурою мається на увазі логічне розташування сторінок, архітектура сайту, розміщення елементів, зв'язок між сторінками та розділами, об'єднаними спільним стилем і дизайном.

Зазвичай веб-сайти складаються з кількох тематичних розділів, що взаємопов'язані через гіперпосилання. На головній сторінці часто розміщують посилання на ці розділи і описують їх зміст. Перша сторінка зазвичай називається index.htm.

Отже, виникає необхідність у проектуванні зручної та легкозрозумілої структури нашого сайту інтернет-магазину, яка дасть змогу клієнту швидко зорієнтуватись при користуванні вебсайтом. Тому, варто розглянути структуру сайту (див. рис. 2.1).

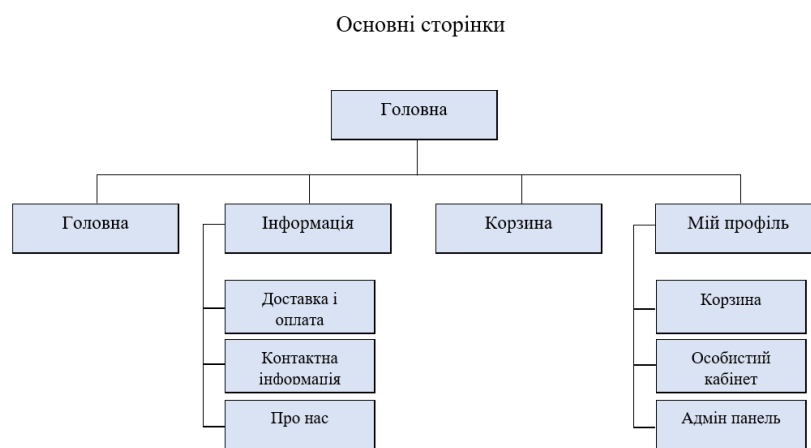


Рис. 2.1 – Візуалізація основних сторінок

Зазначимо, що при проєктуванні структури ми керувалися принципом модульності, що передбачає побудову програми як сукупності окремих невеликих модулів, кожен з яких функціонує за певними правилами. Такий підхід дозволяє значно спростити процес тестування сайту. Функціонал проєкту продемонстровано на рисунку 2.2.

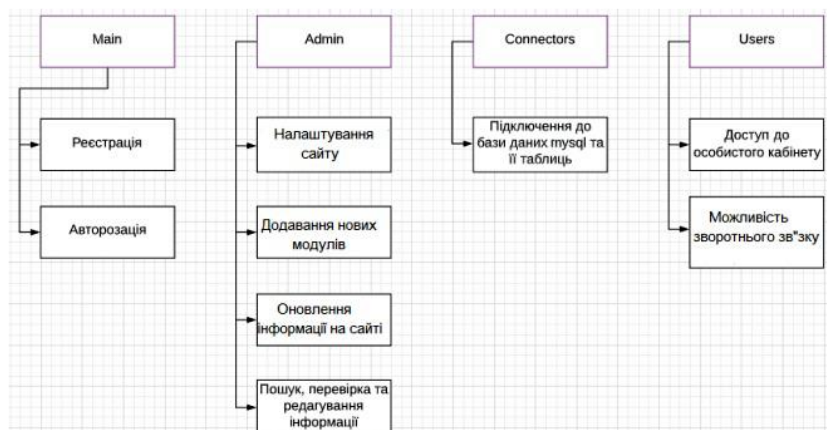


Рис. 2.2 – Структурні модулі сайту

Таким чином, за незареєстрованих користувачів відповідальність несе модуль main який розміщується на початковій сторінці. Після реєстрації на сайті, дані користувача завантажуються в базу даних PostgreSQL, що дає можливість здійснити авторизацію клієнта та надати доступ до особистого кабінету.

Адміністративна частина сайту підпорядковується модулю admin, в якому обробляються вхідні заявки від покупців, перевіряється швидкість з'єднання, редагуються основні компоненти сайту тощо. Підключення до бази даних для збереження інформації здійснюється через connector, що дозволяє покупцю переглядати та редагувати дані, заповнені при реєстрації, переглядати історію замовлень та отримувати зворотний зв'язок.

2.3. Модель MVC

Модель–вигляд–контролер (MVC) — це архітектурний шаблон, який використовується при проектуванні та розробці програмного забезпечення. Він передбачає поділ системи на три взаємопов'язані компоненти: вигляд (інтерфейс користувача), модель (дані) та контролер (модуль керування) [17]. Цей підхід дозволяє відокремити модель даних від вигляду інтерфейсу так, щоб зміни в інтерфейсі мали мінімальний вплив на роботу з даними, а зміни в самих даних могли відбуватися без участі користувацького інтерфейсу. Метою цього шаблону є полегшення модифікацій дизайну програмного забезпечення, що спрощує розширення та внесення змін. Використання MVC також дозволяє повторно застосовувати окремі частини програми. При застосуванні в масштабних проєктах ця модель організовує структуру, знижує складність і полегшує розуміння та супровід коду [9].

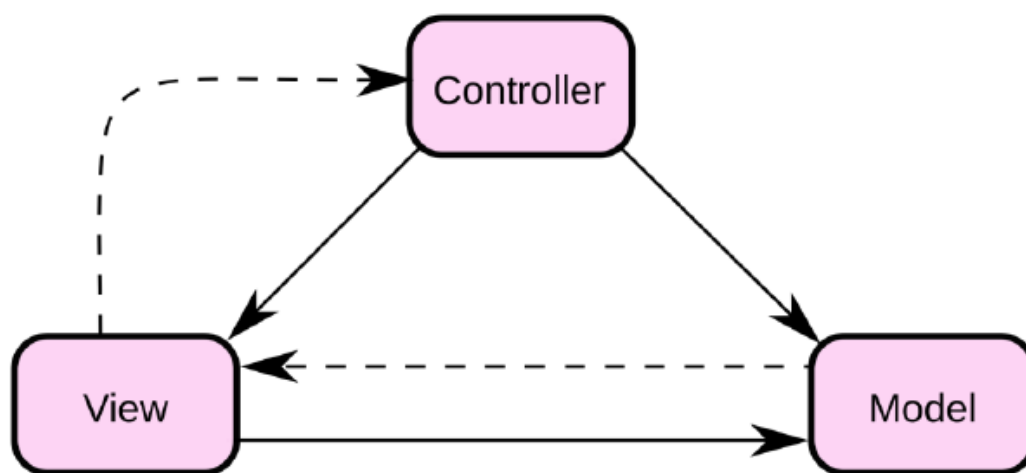


Рисунок 2.3 – Діаграма взаємодії між компонентами шаблону

V — це частина, яка відповідає за те, як матеріал відображається в Django, і за це відповідають відображення (views) та шаблони (templates).

C — частина, яка визначає, як відображати дані, орієнтуючись на вказівки користувача. Це здійснюється через налаштування URLconf фреймворку, який підставляє відповідну функцію Python для конкретного URL.

З огляду на те, що за C відповідає сам фреймворк, а багато процесів відбувається через шаблони, відображення та структури, Django цілком можна вважати MVC фреймворком.

В ідеї MTV:

- М (Model) — модель, яка визначає рівень доступу до основних даних. Вона відповідає за те, як отримати доступ до даних, як їх тестувати та як визначити їх взаємозв'язки між собою.
- Т (Template) — шаблони. Вони відповідають за відображення даних і все, що з цим пов'язано, зокрема за те, як відображення буде організоване на веб-сторінці та в інших документах.
- V (View) — уявлення. Це міст, який з'єднує модель і шаблони. Відповідає за логіку, яка об'єднує необхідні шаблони та модель для правильного відображення даних [19].

2.4. Загальна характеристика та структура роботи з фреймворком Django

Django є високорівневим фреймворком для розробки веб-додатків на мові програмування Python. Він був створений для спрощення процесу розробки завдяки інтегрованим інструментам, які забезпечують швидку розробку, масштабованість і надійність. Django дотримується принципу "Don't Repeat Yourself" (DRY), що мінімізує повторення коду, та підтримує патерн проектування MVC (Model-View-Controller), адаптований як MTV (Model-Template-View).

Фреймворк надає потужний ORM (Object-Relational Mapping), інтегровану систему адміністрування, підтримку авторизації, а також зручні інструменти для роботи з формами, статичними файлами й кешуванням [10].

Для розробки веб-додатків із використанням Django бажано використовувати ізольоване середовище, яке називається віртуальним середовищем. Це дозволяє уникнути конфліктів між залежностями різних проєктів та підтримувати їх

унікальні вимоги. Також, якщо з часом деякі модулі оновляться – створений додаток зможе вільно працювати на використовуваних раніше [19]. Щоб створити віртуальне оточення використовується стандартна команда:

```
python -m venv venv
```

Ця команда створює каталог `venv`, де зберігаються ізольовані залежності Python. Створене середовище перед використанням потрібно активувати командою `venv\Scripts\activate i`, по завершенні роботи – деактивувати. У створене оточення встановлюється Django з допомогою команди:

```
pip install django
```

Працюючи з фреймворком створюють проєкт для конкретного завдання, наприклад для інтернет магазину. Ознак, оскільки такі застосунки є досить об'ємними і мають багато залежностей, зв'язків і сторінок, то їх варто в межах одного проєкту розділяти на додатки, наприклад: головна сторінка, каталог товарів, корзина, користувачі і т.д. [20]. Щоб створити проєкт використовується команда:

```
django-admin startproject myproject,
```

де – `myproject` є ім'ям проєкту.

Django автоматично створює базову структуру файлів і папок:

- `manage.py`: Сценарій для виконання команд Django (запуск сервера, створення додатків, міграцій тощо).
- `myproject/`: Основний каталог, який містить:
 - `__init__.py`: Файл для позначення каталогу як пакету Python.
 - `settings.py`: Головний файл конфігурації проєкту.
 - `urls.py`: Файл для визначення маршрутизації.
 - `wsgi.py`: Сценарій для розгортання на WSGI-сервері.
 - `asgi.py`: Сценарій для розгортання на ASGI-сервері.

Вже на етапі створеного проєкту, його можна запустити командою із `manage.py` – `runserver`. За замовчуванням, сервер працюватиме за адресою: `http://127.0.0.1:8000/` [21].

Однак, щоб розділити проєкт на модулі, потрібно створити окремі додатки. Додатки (apps) у Django — це модульні частини проєкту. Для створення додатку використовується команда:

```
python manage.py startapp appname
```

Кожен додаток має власну структуру:

- models.py: Опис моделей бази даних.
- views.py: Опис логіки відображення сторінок.
- urls.py: (за потреби) маршрути для додатку.
- admin.py: Налаштування адміністративної панелі.
- apps.py: Конфігурація додатку.
- migrations/: Схеми для роботи з базою даних.

В межах виконання проєкту, передбаченою цією роботою, буде створено додатки: goods, users, carts, main. Після створення кожного з них, їх слід додати у файл налаштувань settings.py до списку INSTALLED_APPS, як наведено на рисунку 2.4.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles'  
  
    'main',  
    'goods',  
    'users',  
    'carts',  
]
```

Рис. 2.4 – Список додатків INSTALLED_APPS

Також, у цьому файлі налаштувань прописуються зв'язки з базою даних, шаблонні шляхи, налаштування мови і т.д.

3 РОЗРОБКА АРХІТЕКТУРИ МЕК ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТНОГО РІШЕННЯ

3.1. Обрання програмного забезпечення

Для розробки вебдодатку були вибрані програми, які мають низку унікальних, незамінних властивостей, які полегшують розробку і тестування вебдодатку.

Для оптимізації розробки проекту та зручності були використані наступні додатки:

1. Пакет “ Django ”(серверна платформа).

Django — це високорівневий фреймворк для веб-розробки на мові Python, відомий своєю швидкістю, ефективністю та можливістю швидкого створення безпечних і підтримуваних веб-додатків. Його використання зумовлене наступними факторами:

- швидким процесом розробки;
- зручною архітектурою MVC (Model-View-Controller);
- високим рівнем безпеки завдяки захисту від найпоширеніших веб-загроз, таких як SQL-ін'єкції, XSS-атаки, CSRF-загрози;
- наявністю великої плагінів та бібліотек, що дозволяють швидко інтегрувати додаткові функції [22].

2. СУБД PostgreSQL.

PostgreSQL — це потужна об'єктно-реляційна система управління базами даних, що ідеально підходить для складних веб-додатків. Її вибір обумовлений такими факторами:

- Висока продуктивність. СУБД здатна обробляти великі обсяги даних і підтримує одночасну роботу великої кількості користувачів, що робить її надійним рішенням для масштабних проектів. Її архітектура підтримує ACID-транзакції та повну цілісність даних, що критично важливо для безпеки веб-додатків.

- Розширені можливості роботи з даними. PostgreSQL підтримує роботу з складними типами даних, такими як JSON, JSONB, масиви. Це дозволяє зберігати і обробляти дані без необхідності перетворень.
- Безпека та гнучке управління доступом: PostgreSQL надає багаторівневі моделі безпеки з можливістю налаштування рівнів доступу для кожного користувача або групи користувачів. Вона підтримує аутентифікацію за допомогою SSL-з'єднань, а також інтеграцію з іншими методами захисту даних.
- Підтримка інтеграції з Django: Django має вбудовану підтримку для PostgreSQL, що забезпечує легке налаштування та оптимізацію з'єднання, а також ефективне використання ресурсів бази даних для забезпечення високої продуктивності веб-додатку.

3. Редактор програмного коду VS Code.

Visual Studio Code (VS Code) —безкоштовний та потужний редактор коду з широкими можливостями для розробників, які працюють із різними мовами програмування, включаючи Python і Django. Його вибір обумовлений наступними аспектами:

- Зручний та інтуїтивний інтерфейс. VS Code надає зручний інтерфейс, який можна налаштувати відповідно до потреб розробника, що прискорює процес написання, тестування та відлагодження коду.
- Розширення для Python і Django: VS Code має величезну кількість плагінів і розширень, включаючи специфічні для Python і Django, що забезпечує автозавершення коду, підсвічування синтаксису, налагодження, а також інтеграцію з системами контролю версій (Git) і базами даних.
- Інтеграція з середовищем розробки: VS Code дозволяє запускати Django-сервер безпосередньо з редактора, що значно спрощує процес тестування та налагодження веб-додатку. Також VS Code підтримує Docker і інші технології, що дозволяє легко налаштувати локальне середовище для розробки [11].

- Інструменти налагодження і контролю якості коду: VS Code має вбудовані інструменти для налагодження коду, виявлення помилок та оптимізації. Це дозволяє покращити якість і зменшити час, необхідний для виправлення помилок у програмі.

3.2. Опис алгоритму роботи інтернет-магазину та моделювання бази даних

Розглянемо алгоритми роботи базових програмних складових сайту, зокрема таких: самого веб-сайту, корзини, кабінету користувача, панелі адміністратора магазину декору. Так, алгоритм роботи інтернет-магазину, що відображений у додатку Б.

Користувач має мати можливість обирати, фільтрувати, сортувати товари, переглядати, створювати та змінювати власний кошик товарів у будь-який час, а також виходити з нього задля подальшого пошуку по сайту. Схематичне зображення роботи Корзини продемонстровано у додатку В.

Адмінпанель, в нашому випадку, становитиме доступ до стандартної адмін сторінки, передбаченою фреймворком Django. Продемонструємо схематичне відображення таблиць і зв'язків бази даних (див. рис. 3.1).

Діаграма прецедентів (варіантів використання) – графічна модель, що демонструє сценарії використання системи, взаємодію між користувачами та функціоналом програми. Для побудови діаграми прецедентів інформаційної системи інтернет-магазину «CACTUS store» було розв'язано наступні задачі:

- Визначити акторів: передбачено використання системи такими користувачами як адміністратор вебмагазину «CACTUS store», клієнт та менеджер. Адміністратор відповідає за оновлення, редагування та видалення інформації на сайті. Менеджер взаємодіє зі співробітниками та даними сайту. Клієнт – звичайний користувач сайту, який може переглядати асортимент товарів;

– Визначити перелік прецедентів. Для правильного ведення обліку товарів та замовлень визначені наступні прецеденти: управління користувачами, управління товарами, управління замовленнями, обробка замовлень, перегляд асортименту, оформлення замовлення. Кожен прецедент складається з дій, що відповідають за додавання, редагування, видалення та відображення інформації. Більш детально ознайомитися з прецедентами можна в додатку Г.

Слід зауважити, що загалом алгоритм зв'язку адміністратора із БД продукції повинен бути швидким, простим, легким та інтуїтивно зрозумілим. Всі товари в категоріях повинні бути з'єднані із відповідними підкатегоріями, які повинні зв'язуватись підкатегоріями більш високого рівня, які, в свою чергу міститимуться в батьківській директорії. З метою можливості додавання нової продукції, створюються спеціальні форми введення даних.

Так, адміністратор в адмінпанелі здійснює наповнення контентом сайт – додає категорії товарів, фото, опис товару, формує ціну, знижки, акції і т.д. Покупець же, обравши товар, додає його в корзину, вводить всі необхідні дані щодо особистих даних, способу оплати і доставки, потім ця інформація по новому замовленню повертається до адміністратора.

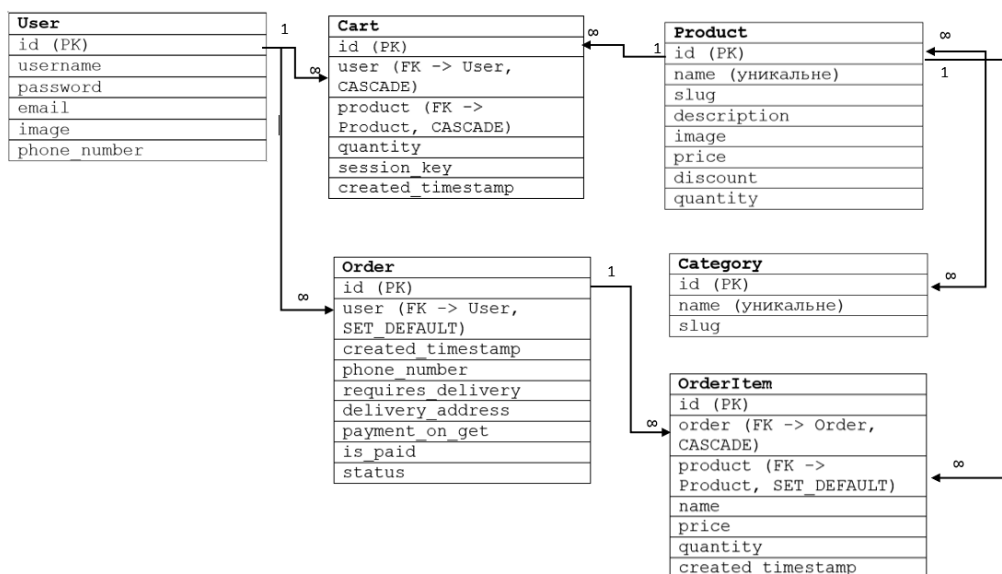


Рис. 3.1 – Схема зв'язків БД

Після підтвердження замовлення, покупець його оплачує відповідно до обраних платіжних систем та отримує обрану продукцію у відділенні фірми-перевізника.

Однією з ключових переваг Django є вбудована ORM (Object-Relational Mapping), яка дозволяє взаємодіяти з базою даних, використовуючи об'єктно-орієнтовану модель замість написання SQL-запитів [23].

Django ORM автоматично трансліює запити Python до SQL. Вона забезпечує наступний функціонал:

- Створення моделей: Усі таблиці бази даних описуються як класи Python, а їхні атрибути відповідають полям таблиць. Цей клас автоматично перетворюється на SQL-запит для створення відповідної таблиці в базі даних.
- CRUD-операції (створення, читання, оновлення, видалення): Django ORM дозволяє виконувати всі основні операції над даними, використовуючи Python-код:
- Зв'язки між моделями: Django ORM підтримує різні типи зв'язків між таблицями: One-to-One, Many-to-One, Many-to-Many.
- Міграції: Django ORM надає механізм міграцій для автоматичного створення, зміни чи видалення таблиць у базі даних. Наприклад:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

З метою управління БД (набір іменованих таблиць, які містять взаємопов'язані інформаційні дані, що зберігаються в ІС через двовимірні таблиці), яка використовується на сайті підключено PostgreSQL, що дає змогу адмініструвати через сервер, завантажувати SQL, а також дивитись наповненість таблиць БД.

Всі таблиці містять масиви, що складається із записів, які містять поля. Зауважимо, що при створенні проєкту Django файл бази даних створюється автоматично. Однак, за замовчуванням використовується модуль sqlite3, який недоцільно використовувати через обмеженість навантаження та функціонал. Саме тому, всі таблиці створені з використанням СУБД PostgreSQL.

Для роботи з БД були створені моделі таблиць через класи у структурі проєкту. Наповнення даних відбувається через адмінпанель (див. рис. 3.2). Або ж це можна робити через СУБД, однак – зручніше через сайт.

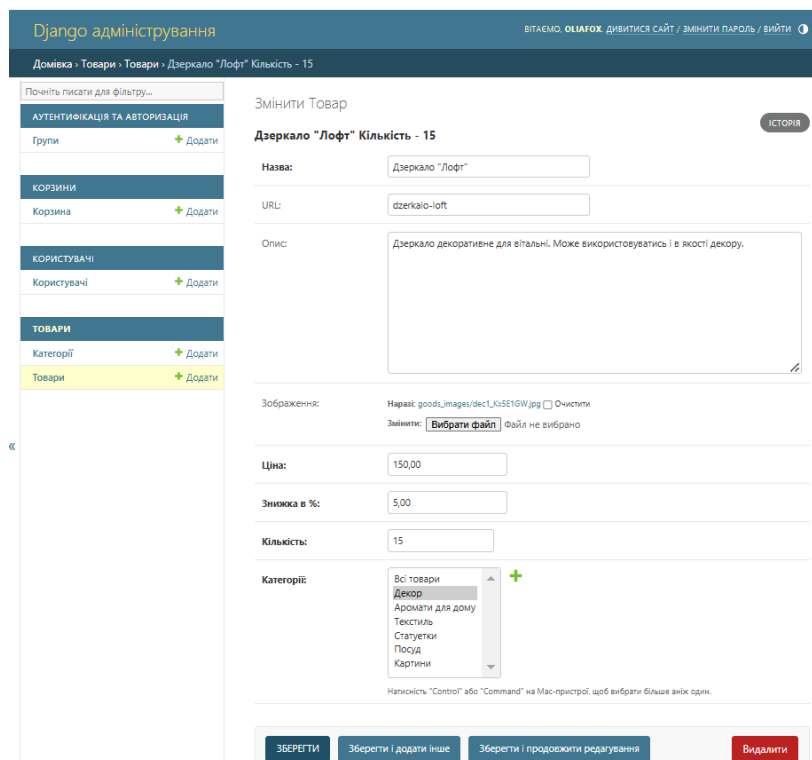


Рис. 3.2 – Наповнення даних через адміністративну панель

Для створення суперкористувача для сайту використовується команда `python manage.py createsuperuser`.

БД нашого сайту інтернет-магазину декору скрадатиметься із наступних 6 базових таблиць: категорії, товари, користувачі, корзина, замовлення і замовлений товар.

Таким чином, інформація що відображена на сайті, формалізується в певних таблицях, яка має певні властивості. Такий набір таблиць для магазину утворюється саме через PostgreSQL. Так, таблиці є наборами певних полів, що відповідають зазначеним атрибутам суті. Поля кожної таблиці, зокрема найменування, ідентифікатор (формується автоматично).

Запити до БД відбуваються шляхом завантаження даних у словник і перебір їх. Для цього не потрібно писати SQL-запити, а слід вказати з якої моделі які поля потрібно вибрати.

3.3. Розробка та оптимізація дизайну сайту магазину декору

Фреймворк Django призначений для розробки бекендової частини сайту. Дизайн сайту розроблявся з використанням HTML та CSS, стилів CSS із bootstrap, декотрих методів javascript і був під'єднаний до сайту [18].

Здебільшого сторінки сайту магазину декору отримали однаковий дизайн, що і на стартовій сторінці (див. рис. 3.3). Всього є дев'ять сторінок, більшість з яких наповнюються динамічно: оформлення замовлення, каталог, товари, головна, про сайт, входу, особистий кабінет, реєстрація і корзина.



Рис. 3.3 – Головна сторінка сайту «CACTUS store»

Для того, щоб у кожній сторінці не прописувати однакові елементи дизайну, створено шаблонний базовий файл base.html, у якому прописано дизайн для хедера, навігаційної панелі, контейнера головної частини сайту футера та модального вікна

корзини. Інші сторінки лиш використовуються базовий дизайн. Для цього на початку кожної сторінки експортується файл із шаблонним дизайном:

```
{% extends "base.html" %}
```

Якщо на сторінці більше нічого не прописати, то завантажуватиметься виключно розмітка базового файлу. Однак, щоб сторінки були унікальними, то у шаблонному файлі base.html замінюються виключно та частина, яка буде відмінною. Тобто, на кожній сторінці, окрім Адмінпанелі, є спільні елементи: навігаційна панель меню із полем пошуку, кнопка каталогу та корзини та футер, який на окремих сторінках буде відсутнім.

Це реалізовується з допомогою блоків або джінджа-кодів. Наприклад, на сторінці Каталогу (див. рис. 3.4) буде змінюватися головна частина – контент, де буде відображатися перелік товарів. Тобто, у базовому файлі є блок пустий блок:

```
{% block content %}
```

```
{% endblock content %}
```

У файлі catalog.html є ідентичний блок, але вміст завантажується як перелік товарів відповідної категорії із БД. Тобто у каталозі прописується лише той код, який буде відмінним від шаблонного. Ідентичною схемою створенні всі інші сторінки сайту. Вміст, що відмінний від шаблонного просто підвантажується у відповідні блоки.

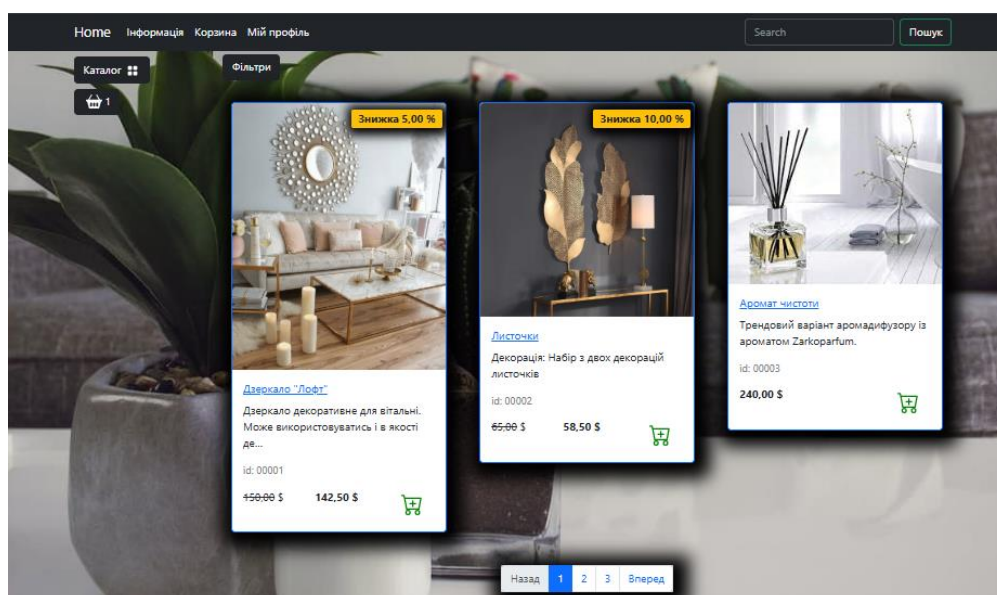


Рис. 3.4 – Сторінка Каталог

Розглянемо детальніше дизайн основних елементів та сторінок сайту. У хедері кожної сторінки розміщене навігаційне меню. Воно відрізняється в залежності від того, чи користувач авторизований чи ні, а також які є права у зареєстрованого юзера. Якщо у користувача є права адміністратора, то йому буде доступне наступне меню:

- Home;
- Інформація (доступно всім):
 - Доставка і оплата;
 - Контактна інформація;
 - Про нас;
- Корзина (доступно всім);
- Мій профіль (доступно тільки авторизованим користувачам):
 - Корзина;
 - Особистий кабінет;
 - Адмінпанель (лише для користувачів із правами суперадміністратора);
 - Вихід;
- Поле пошуку за ключовими словами чи фразами.

Меню Інформація містить випадające меню із посиланням на сторінки, які містять дані про умови доставки та оплати, контактну інформацію про магазин та історію створення і розвитку інтернет-магазину.

Меню Мій профіль доступне лише зареєстрованим користувачам і містить посилання на Корзину, Адмін панель та Особистий кабінет. Сторінка адміністратора доступна лише користувачам рангу суперадмін. Вона перенаправляє користувача за стандартною адресою admin/, можливості і дизайн якої надає структура самого проекту створеного на Django (див. рис. 3.5). Тому для цього пункту меню доробляти дизайн не потрібно.

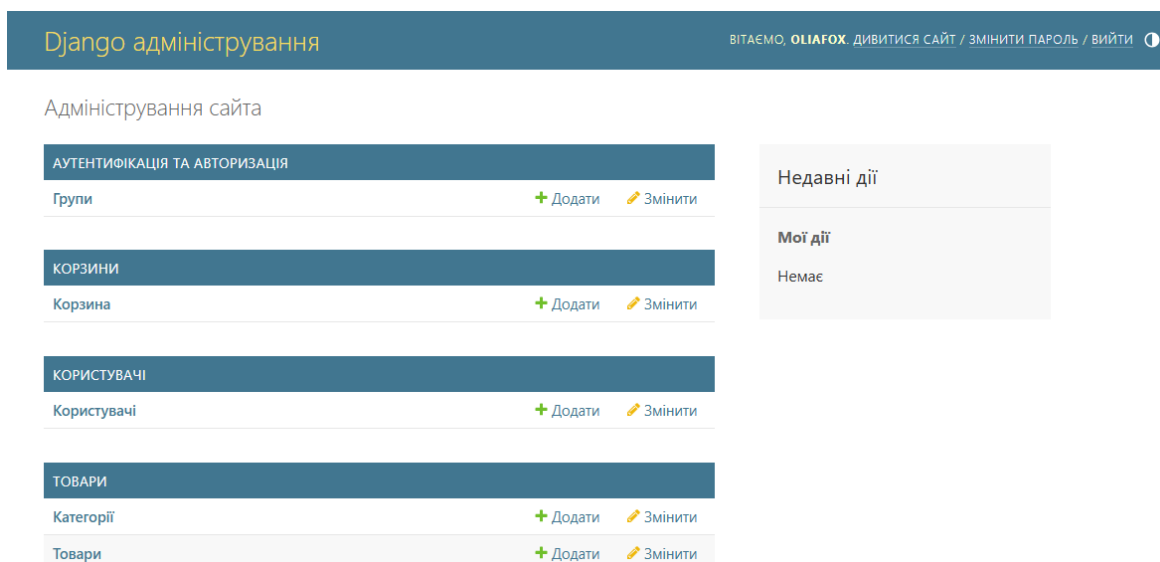


Рис. 3.5 – Сторінка адміністрування

Сторінка Особистий кабінет користувача аналогічно до попередніх завантажує дизайн шаблонного файлу і джінджа-кодами замінює вміст головного контейнера. У ньому розміщено розмітку контейнера з формою редагування особистих даних користувача (див. рис. 3.6, 3.7) та список здійснених раніше замовлень. На даній сторінці відсутній блок підвалу, а всі інші – присутні.

The screenshot shows a form titled "Профіль користувача". At the top, there's a cartoon character of a person with glasses sitting at a desk. Below the character, there are two buttons: "Вибрати файл" and "Файл не вибрано". The form contains several input fields: "Ім'я*" with a placeholder "Введіть ваше ім'я", "Прізвище*" with a placeholder "Введіть ваше прізвище", "Ім'я користувача*" with the value "oliafox", and "Email*" with a placeholder "Введіть ваш email *youremail@example.com". At the bottom, there's a "Зберегти" button.

Рис. 3.6 – Вікно Профіль користувача

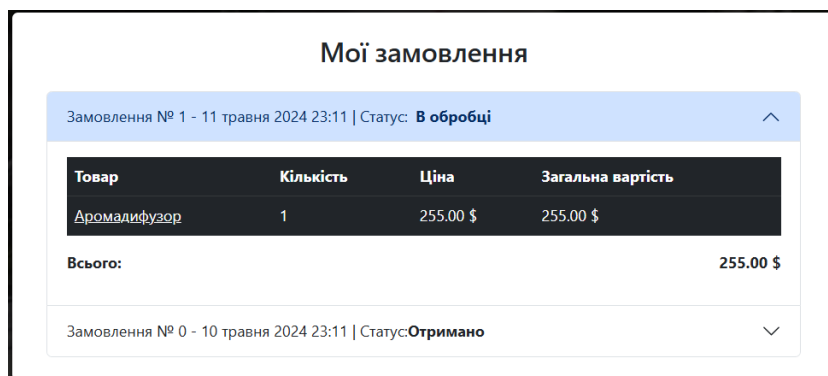


Рис. 3.7 – Вікно Мої замовлення

Значок корзини фіксований і викликає модальне вікно, вигляд головного контейнера якого забезпечує клас modal fade (див рис. 3.8).

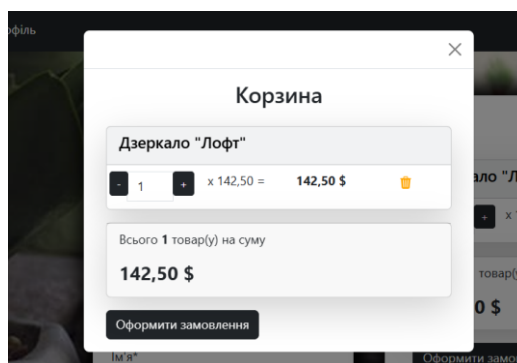


Рис. 3.8 – Вікно Корзина

Меню Каталог теж фіксоване і присутнє на кожній сторінці сайту. Його реалізовано контейнером, що містить кнопки. Розмітка прописана у файлі catalog.html. При натисканні на кнопку завантажується із БД список категорій. Кожна категорія містить товари, які можна переглянути окремо (див. рис. 3.9). Розмітка прописана у файлі product.html. Аналогічно до попередній, у ній замінений блок content, який розміщує в контейнері зображення товару, ціну, опис і кнопку для додавання в корзину.

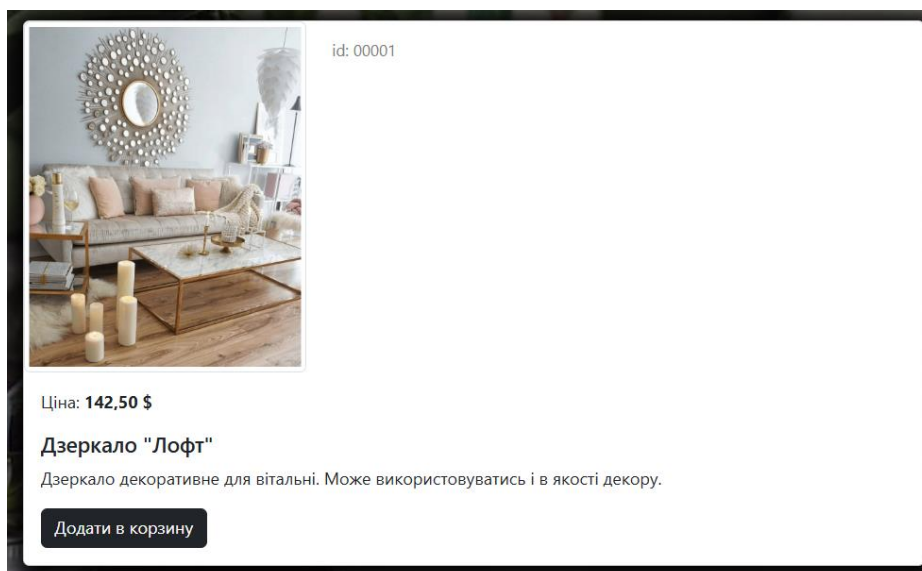


Рис. 3.9 – Вікно Опис товару

На сторінці каталогу розміщено кнопку фільтр (див. рис. 3.10). Також пагінація сторінок, щоб весь перелік товарів не розміщувався на 1 сторінці, а лише по 3 шт.

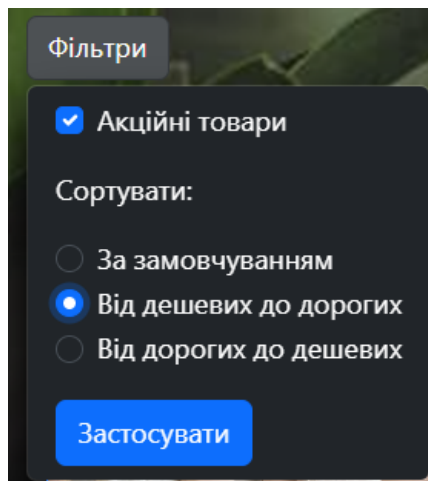


Рис. 3.10 – Фільтр товарів

Для пагінації, фільтру, панелі навігації і поля пошуку використовуються стилі із bootstrap.

Окрім того, сторінка авторизації та авторизації (див. рис. 3.11) організована у файлах login.html та registration.html відповідно, додає до шаблонного файлу контент сторінки із розміткою, що містить форму реєстрації чи реєстраційні поля.

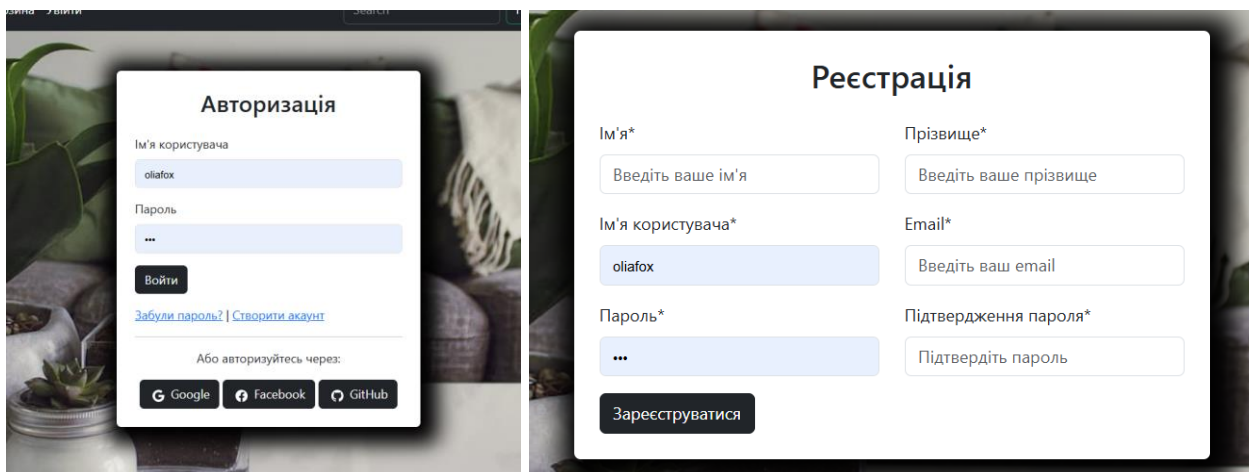


Рис. 3.11 – Вікна Авторизація і Реєстрація

3.4. Реалізація моделі MVC для розробки серверної, клієнтської та адмінчастини сайту

У цій частині роботи представлено практичну реалізацію вебсайту інтернет-магазину декору із застосуванням фреймворку Django, який базується на архітектурі Model-View-Controller (MVC). Розділ охоплює три ключові аспекти розробки: серверна частина, клієнтська частина та адміністративний інтерфейс. Основна мета — продемонструвати, як використання Django забезпечує швидку та ефективну розробку функціонального вебсайту із сучасним дизайном та потужним інструментарієм для адміністрування [23].

Оскільки, розробка сайту інтернет-магазину – це об’ємний проєкт. Тому сайт є одним проєктом, який розділений на додатки: головний (включає головну та інформаційні сторінки), товари (товари і категорії), користувачі (форма реєстрації, авторизації, профіль користувача) і корзини.

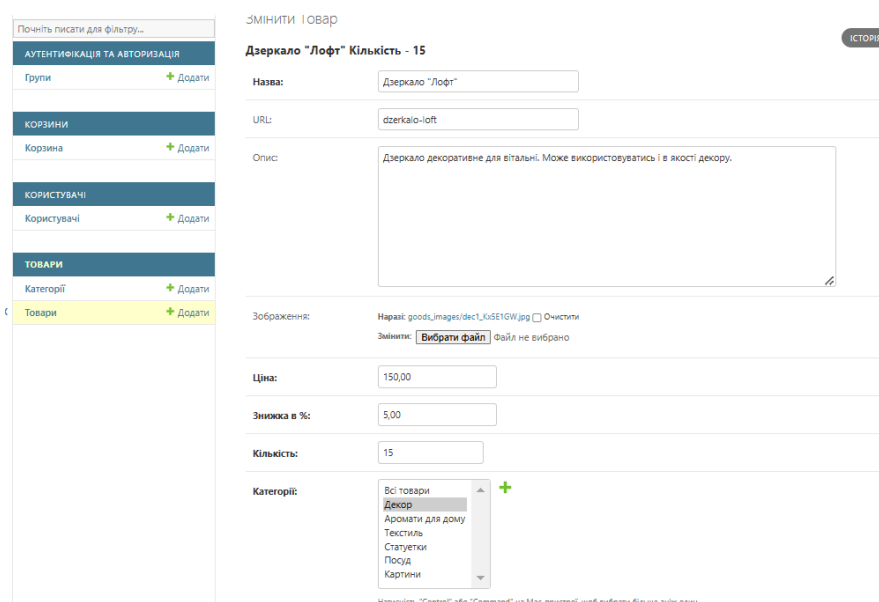
Серверна частина (Model) реалізована на основі моделей Django, які відповідають за роботу з базою даних. Для проєкту було створено кілька ключових моделей, які відображають структуру даних, необхідну для функціонування інтернет-магазину:

- Товари (Products);
- Категорії (Categories);
- Користувач (User);
- Корзина (Cart).

Моделі описані у файлі `models.py` відповідного додатку. Наприклад, модель товару (Product) описана у файлі моделей додатку `goods` і має таку структуру:

```
class Products(models.Model):
    name = models.CharField(max_length=150, unique=True,
verbose_name='Назва')
    slug = models.SlugField(max_length=200, unique=True, blank=True,
null=True, verbose_name='URL')
    description = models.TextField(blank=True,
null=True, verbose_name='Опис')
    image = models.ImageField(upload_to='goods_images', blank=True,
null=True, verbose_name='Зображення')
    price = models.DecimalField(default=0.00, max_digits=7,
decimal_places=2, verbose_name='Ціна')
    discount = models.DecimalField(default=0.00, max_digits=7,
decimal_places=2, verbose_name='Знижка в %')
    quantity = models.PositiveBigIntegerField(default=0,
verbose_name='Кількість')
    category = models.ManyToManyField(Categories, verbose_name='Категорії')
```

Реалізація таблиці наведена на рисунку 3.12.



The screenshot displays the Django Admin interface for the 'goods' app. On the left is a sidebar menu with categories: 'АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ', 'КОРЗИНИ', 'КОРИСТУВАЧІ', 'ТОВАРИ', and 'ТОВари'. The 'ТОВАРИ' section is expanded, showing 'Категорії' and 'Товари'. The main area shows the 'Дзеркало "Лофт" Кількість - 15' form. The form includes fields for 'Назва' (Dзеркало "Лофт"), 'URL' (dзерkalo-loft), 'Опис' (Dзеркало декоративне для віталії...), 'Зображення' (Image upload), 'Ціна' (150,00), 'Знижка в %' (5,00), 'Кількість' (15), and 'Категорії' (Decor). A 'Історія' button is in the top right corner.

Рис. 3.12 – Реалізація таблиці Товари у Адмінпанелі

Ці моделі взаємодіють із базою даних PostgreSQL, забезпечуючи ефективне зберігання та обробку даних.

Клієнтська частина вебсайту реалізована за допомогою представлень (Views) у Django. Для оптимізації розробки використано як функціональні представлення, так і класові. Наприклад, у файлі представлень `views.py` додатку товари:

```
def product(request, product_slug):
    product = Products.objects.get(slug=product_slug)
    context = {
        'product': product
    }
    return render(request, "goods/product.html", context=context)
```

Функція `product` отримує дані про продукт із бази даних за його унікальним `slug`. Далі передає ці дані в шаблон для відображення. Після чого формує сторінку, яка показує детальну інформацію про вибраний продукт. Тобто на сторінку `html` із шаблонів `template`.

Для створення шаблонів використано Django Template Language (DTL), що дозволяє динамічно відображати дані на сторінках [24].

Django Admin надає вбудований адміністративний інтерфейс, який було налаштовано для зручного управління контентом вебсайту. Але кожен таблицю у відповідному додатку потрібно зареєструвати, щоб через панель адміністратора, після проведення міграцій, вона була видимою. Налаштування адміністративного інтерфейсу виконується у файлі `admin.py` кожного додатку, до якого є таблиця БД. Для прикладу, для додатку товарів `goods` у файлі `admin.py` виконано налаштування відображення таблиць наступним чином:

```
@admin.register(Categories)
class CategoriesAdmin(admin.ModelAdmin):
    prepopulated_fields = {
        'slug': ('name',)
    }
@admin.register(Products)
class ProductsAdmin(admin.ModelAdmin):
    prepopulated_fields = {
        'slug': ('name',)
    }
```

Декоратори `@admin.register(Categories)` та `@admin.register(Products)` реєструють моделі `Categories` і `Products` у адміністративній панелі Django. Після

реєстрації, для налаштування їхнього інтерфейсу використовується відповідний клас `CategoriesAdmin` та `ProductsAdmin`.

Класи `CategoriesAdmin` і `ProductsAdmin` успадковують функціонал базового класу `admin.ModelAdmin`. Це дозволяє налаштовувати вигляд і поведінку адміністративного інтерфейсу для відповідних моделей.

Під час створення нового запису `Categories` або `Products`, поле `slug` автоматично генерується на основі значення `name`.

Ці налаштування дозволяють адміністратору зручно працювати з продуктами, категоріями та іншими даними.

У файлах представлення визначається логіка, яка буде використана при зверненні до URL-адрес. А щоб побудувати до них шлях, то у файлі `urls.py` будуються маршрути, тобто визначаються URL-адреси, які викликають функції з файлу `views.py`. В проекті інтернет магазину для додавання товарів до кошику, видалення та зміни кількості використовуються функції `cart_add`:

```
def cart_add(request, product_slug):

    product = Products.objects.get(slug=product_slug)

    if request.user.is_authenticated:
        carts = Cart.objects.filter(user=request.user, product=product)

        if carts.exists():
            cart = carts.first()
            if cart:
                cart.quantity += 1
                cart.save()
        else:
            Cart.objects.create(user=request.user, product=product,
                                quantity=1)

    return redirect(request.META['HTTP_REFERER'])
```

Маршрути, які виникають при виклику функцій із `views.py` кошика скеровуються на відповідні сторінки, які додаються до списку `urlpatterns`:

```
urlpatterns = [
    #path('cart_add/<slug:product_slug>/', views.cart_add, name='cart_add')
    path('cart_add/<slug:product_slug>/', views.cart_add, name='cart_add'),
    path('cart_change/<slug:product_slug>/', views.cart_change,
    name='cart_change'),
    path('cart_remove/<slug:product_slug>/', views.cart_remove,
    name='cart_remove'),
]
```

Використання моделі MVC у Django забезпечує розподіл логіки додатка на три частини:

- Model – відповідає за роботу з базою даних.
- View – формує логіку відображення даних користувачу.
- Template (в Django — частина View) – відповідає за оформлення HTML-сторінок.

Така архітектура спрощує підтримку та розширення коду, дозволяє розробникам паралельно працювати над різними частинами проєкту.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці при роботі з комп'ютерними пристроями

Процес розробки інтернет-магазину зазвичай є тривалим і складним, оскільки потребує роботи з численними технологіями та великими обсягами даних, що підвищує ймовірність виникнення помилок. Відомо, що одним з основних чинників для ефективної роботи з комп'ютером є забезпечення відповідних умов праці. Недостатнє освітлення, обмежений простір чи неякісні дисплеї можуть спричиняти фізичний і згодом психологічний дискомфорт, що, у свою чергу, негативно впливає на результати роботи. Для запобігання цьому необхідно дотримуватися ряду стандартів та правил.

У наказі № 207 від 14.02.2018 НПАОП 0.00-7.15-18 [25] зазначено вимоги, яких слід дотримуватись при роботі з екранними пристроями. Згідно з третьою частиною цього наказу [25], робоче місце працівника повинно відповідати наступним критеріям:

1. Робоче місце повинно мати достатні розміри, щоб працівник міг змінювати робоче положення та рухатися.
2. Випромінювання від екранних пристроїв повинно бути знижене до максимально допустимого рівня з огляду на безпеку та охорону здоров'я працівників.
3. Елементи робочого місця та їх розташування повинні відповідати ергономічним, антропологічним і психофізіологічним вимогам, а також вимогам, що виникають у процесі виконання робіт.
4. Освітлення робочого місця повинно створювати необхідний контраст між екраном і навколишнім середовищем та відповідати вимогам ДСанПІН 3.3.2.007-98.
5. Мікроклімат у виробничих приміщеннях повинен підтримуватися на сталому рівні та відповідати Санітарним нормам мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [26].

6. Робочий стіл чи поверхня повинні бути достатнього розміру та мати низьку відбивну здатність, бути гнучкими для розміщення екрана, клавіатури, документів або устаткування.

7. Робоче крісло повинно бути стійким і дозволяти працівнику легко рухатися та займати зручне положення. Сидіння повинно регулюватися по висоті, спинка — по висоті і мати можливість нахилу. Для зручності має бути передбачена підніжка для тих, кому це необхідно.

Також варто звертати увагу на безпеку, відповідно до четвертої частини наказу [25], де вказані мінімальні вимоги безпеки:

1. Перед початком роботи необхідно очистити екранні пристрої від пилу та інших забруднень.
2. Після завершення роботи пристрої слід вимкнути від живлення.
3. У разі аварійної ситуації потрібно негайно відключити пристрій від електричної мережі.

4. Забороняється:

- Ремонт чи технічне обслуговування, а також налагодження екранних пристроїв на робочому місці працівника під час роботи з ними;
 - Вимикати захисні пристрої або проводити зміни в конструкції та складі екранних пристроїв, а також їх технічне налагодження;
 - Працювати з несправними екранними пристроями, у яких під час роботи виникають нехарактерні сигнали, мигання та інші несправності.
 - Під час виконання робіт з комп'ютером, пов'язаних з нервово-емоційним напруженням мають дотримуватися оптимальні умови мікроклімату відповідно до вимог ДСН 3.3.6.042-99 [26].
5. Під час роботи з комп'ютером, що супроводжується нервово-емоційним напруженням, необхідно забезпечити оптимальні умови мікроклімату відповідно до вимог ДСН 3.3.6.042-99 [26].

Для забезпечення безпеки, згідно з п'ятою частиною наказу «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи

з екранними пристроями» [25], усі екранні пристрої повинні відповідати мінімальним вимогам безпеки.

Згідно з третьою частиною цього наказу [25], робоче місце працівника, який займається розробкою веб-сервісів і підтримує стабільну роботу додатку, повинно відповідати таким вимогам:

1. Екранні пристрої не повинні бути джерелом ризику для працівників.
2. Усе випромінювання повинно бути зведене до мінімального рівня з точки зору безпеки та охорони здоров'я працівників.
3. Символи на дисплеї повинні бути чіткими, відповідного розміру, а також між символами та рядками символів має бути правильна відстань.
4. Зображення на дисплеї повинно бути стабільним, без миготіння чи інших видів несправностей.
5. Яскравість та контрастність символів повинні легко регулюватися і швидко адаптуватися до навколишніх умов.
6. При виборі монітора слід надавати перевагу пристроям, які мають можливість повороту та нахилу екрану.
7. При необхідності монітор може бути закріплений на окремому столі або підставці.
8. Для вибору монітора слід надавати перевагу дисплеям з матовим покриттям, щоб мінімізувати відблиски або відображення світла.
9. При виборі клавіатури слід обирати ту, яка відкидається і є автономною, що дозволяє працівнику обрати зручну робочу позу й уникнути втоми рук.
10. Поверхня клавіатури має бути матовою для уникнення віддзеркалювання.
11. Устаткування, яке входить до робочої станції, не повинно виділяти надлишкове тепло.
12. Під час розробки, вибору, замовлення та модифікації програмного забезпечення, а також при розробці завдань, які передбачають використання екранних пристроїв, роботодавець повинен обирати таке програмне забезпечення, яке відповідає поставленим завданням, є простим у використанні, а де необхідно — адаптованим до рівня знань і досвіду працівника.

Отже, для безпечної та ефективної роботи розробника веб-сервісу створено належні умови праці, які охоплюють все — від організації робочого місця та його оснащення до забезпечення комфортного мікроклімату робочого середовища, відповідно до вимог чинного законодавства.

4.2. Безпека в надзвичайних ситуаціях

Ефективність роботи людини за комп'ютером значною мірою залежить від її функціонального стану. Психофізіологічні та емоційні перенапруження, а також втома оператора можуть призвести до помилок у комп'ютеризованих системах управління, що в свою чергу веде до значних економічних втрат. Статистичні дані свідчать, що від 40% до 75% авіаційних аварій є наслідком людського фактора. Так само відмови комп'ютеризованих систем управління рухом на залізницях та в гірничо-збагачувальних комбінатах, спричинені помилками операторів, складають понад 50% усіх відмов, причому значна частина цих інцидентів пов'язана з невідповідністю між функціональним станом оператора та складністю завдання.

Трудова діяльність користувачів комп'ютерів здійснюється в певному виробничому середовищі, яке безпосередньо впливає на їх функціональний стан. Серед найбільш значущих фізичних факторів виробничого середовища — електромагнітні хвилі різних частот, електростатичні поля, шум, параметри мікроклімату та інші світлотехнічні показники.

Трудовий процес значною мірою впливає на психофізіологічні можливості користувачів комп'ютерів, оскільки їх діяльність пов'язана з великими статичними фізичними навантаженнями, обмеженою руховою активністю, а також напруженнями сенсорного апарату та вищих нервових центрів, що відповідають за увагу, мислення та регуляцію рухів. Крім того, професійна діяльність користувачів комп'ютерів супроводжується високими інформаційними навантаженнями.

Професійні якості, досвід і навички користувачів, що визначають внутрішні засоби їх діяльності, є основою для безпомилкової та надійної роботи, а також для пошуку безпечних методів вирішення виробничих завдань навіть у нестандартних ситуаціях.

Зовнішні фактори діяльності, що здебільшого залежать від ергономічних характеристик організації робочого місця — форми та параметрів його елементів, а також просторового розташування основного і допоміжного устаткування, можуть значно зменшити фізичні та психофізіологічні навантаження, з якими стикаються користувачі комп'ютерів.

Психологічний захист населення націлений на зменшення та нейтралізацію негативних психічних станів і реакцій серед населення під час загрози або виникнення надзвичайних ситуацій (НС) [28]. Основні заходи включають:

- своєчасне застосування ліцензованих та дозволених в Україні інформаційних, психопрофілактичних і психокорекційних методів впливу на особистість;
- виявлення за допомогою психологічних методів чинників, що сприяють виникненню соціально-психологічної напруженості;
- використання сучасних психологічних технологій для нейтралізації негативного впливу чинників НС на населення.

Оскільки робота користувачів комп'ютерів часто передбачає активну взаємодію з іншими людьми, постають питання раціоналізації міжособистісних стосунків. Цей комплекс проблем охоплює як психологічні, так і соціально-психологічні аспекти трудових взаємовідносин, які є факторами «ризиків», що значно впливають на функціональний стан користувачів комп'ютерів.

Визначення та вивчення факторів, які впливають на функціональний стан користувачів комп'ютерів, дозволяє виявити основні причини виникнення станів напруженості, стомлення, стресу, а також здійснити відповідні профілактичні заходи.

Основні фактори, що впливають на функціональний стан користувачів комп'ютера, включають:

1. Середовище – до шкідливих факторів середовища належать:

- 1.1. Фізичні: вплив електромагнітних хвиль різних частот, електростатичних полів, шуму, параметрів мікроклімату, а також світлотехнічні показники;
- 1.2. Хімічні: наявність пилу та шкідливих хімічних речовин, які виділяються при роботі принтерів, копіювальних апаратів та інших пристроїв;
- 1.3. Біологічні: підвищений рівень патогенних мікроорганізмів у повітрі, особливо в приміщеннях з великою кількістю працюючих, поганою вентиляцією або під час епідемій;
- 1.4. Психофізіологічні: напруження зору, уваги, емоційне та інтелектуальне навантаження, тривалі статичні навантаження, монотонність роботи та інші чинники, що впливають на фізіологічний стан користувачів.

2. Трудовий процес – характеризується значними статичними фізичними навантаженнями, недостатньою руховою активністю, напруженнями сенсорного апарату, а також вищих нервових центрів, які забезпечують функції уваги, мислення та регуляції рухів. Крім того, робота користувачів комп'ютерів часто пов'язана з високими інформаційними навантаженнями, що може негативно позначатися на їх функціональному стані.

3. Внутрішні засоби діяльності – це професійні якості та виробничий досвід, що забезпечують надійність і безпомилковість роботи користувачів комп'ютерів, а також допомагають знаходити безпечні методи вирішення виробничих завдань, навіть у нестандартних ситуаціях.

4. Зовнішні засоби діяльності – визначаються ергономічними параметрами організації робочого місця, формою і розмірами його елементів, просторовим розташуванням основного і допоміжного устаткування. Ці фактори можуть значно знизити фізичні та психофізіологічні навантаження на користувачів комп'ютерів.

5. Соціально-психологічні фактори трудових взаємовідносин – важливі для підтримки здоров'я і працездатності операторів комп'ютерів. У професійних користувачів комп'ютерів часто спостерігаються порушення зору, опорно-рухового апарату, а також захворювання центральної нервової, серцево-судинної, імунної систем і шкіри. Велика кількість скарг операторського персоналу на

загальне недомогання, стомлювання, головний біль, порушення зору свідчить про наявність негативного психофізіологічного впливу на самопочуття та працездатність.

Сучасна професія користувача дійсно належить до розумової праці і має ряд характеристик, що суттєво впливають на фізичний та психічний стан працівників. Основні фактори, що визначають цю професію, включають:

- Висока напруженість зорових функцій – тривале використання комп'ютера вимагає постійної концентрації уваги на екрані, що може призвести до втоми очей та розвитку різноманітних захворювань зорового апарату.

- Один із основних аспектів – одноманітна поза – це статичне сидіння, яке може викликати порушення кровообігу, проблеми з хребтом та інші захворювання опорно-рухового апарату.

- Виконання стереотипних рухів – через велике навантаження на м'язи кистей рук при використанні клавіатури та миші підвищується ризик розвитку таких захворювань, як тунельний синдром, проблеми з суглобами рук.

- Нервово-емоційна напруга – в умовах обмеженого часу та високих вимог до швидкості виконання завдань робота з комп'ютером часто супроводжується стресом, що може призвести до зниження ефективності та розвитку психосоматичних розладів.

- Обробка великих обсягів інформації – користувачі ВДТ стикаються з необхідністю обробки великих обсягів даних, що активізує різноманітні психічні функції, зокрема увагу, пам'ять, аналіз та синтез інформації.

- Вплив фізичних факторів – робота з дисплеями на електронно-променевих трубках може створювати додаткові фізичні навантаження, такі як електростатичні поля, радіочастотне випромінювання, які мають потенційно шкідливий вплив на здоров'я користувача.

Всі ці фактори підкреслюють необхідність правильної організації робочого місця та дотримання заходів безпеки для забезпечення здоров'я користувачів.

Текст добре структурує різні типи діяльності професіоналів, які працюють з

комп'ютерами, та вказує на специфічні ризики для їх здоров'я. Ось деякі можливі варіанти для доповнення чи перепису:

1. Класифікація професійної діяльності:

- Перша група діяльності передбачає виконання нескладних, багаторазово повторюваних операцій, таких як робота операторів комп'ютерного набору або працівників довідкових служб. Такі роботи не вимагають високого розумового навантаження, але все одно можуть призвести до втоми через монотонність.
- Друга група включає діяльність, пов'язану з логічними операціями, що повторюються. Це робота інженерів-економістів, проектувальників або операторів автоматизованих виробництв. Така робота вимагає постійної концентрації та інтелектуальних зусиль, що може бути джерелом стресу та фізичного навантаження.
- Третя група стосується діяльності, коли необхідно приймати рішення в умовах нестандартних ситуацій, за відсутності заздалегідь відомого алгоритму. Це робота інженерів-програмістів, диспетчерів на залізничному транспорті чи аеропортах. Цей тип діяльності вимагає високих когнітивних навичок та швидкого реагування, що збільшує рівень стресу та напруги.

2. Психологічні та фізіологічні порушення:

- В умовах інтенсивного розумового навантаження серед користувачів комп'ютерів часто (40–70%) виникають психологічні порушення, такі як нервозність, тривога, роздратованість та замкнутість. Це пов'язано з високим рівнем стресу і тривалою роботою на комп'ютері.
- У США та Європі поширене таке явище, як синдром комп'ютерного стресу (СКС), що включає головний біль, запалення очей, алергії, депресії та роздратованість. Це специфічне захворювання значно впливає на фізичне та психічне здоров'я користувачів ВДТ.

3. Інформаційне перевантаження та його наслідки:

- Інформаційне перевантаження супроводжується низкою специфічних захворювань, відомих як «інформаційні» захворювання. Першим симптомом

таких хвороб є головний біль, який виникає через високий рівень стресу та перевантаження мозку.

4. Професійні захворювання:

- Робота з ВДТ супроводжується підвищеним навантаженням на зір, монотонною працею та впливом різноманітних випромінювань. Це може призвести до професійних захворювань, таких як погіршення зору, головні болі, м'язові болі, а також психічні та нервові розлади.
- Фахівці Всесвітньої організації охорони здоров'я відзначають, що оператори ВДТ більш схильні до таких захворювань, як передчасна стомлюваність, серцево-судинні захворювання, психічні розлади, а також рак.

5. Вплив факторів середовища і трудового процесу:

- Стан здоров'я операторів ВДТ визначається комплексним впливом двох основних факторів: середовища (електромагнітне випромінювання, шум, погана вентиляція) і трудового процесу (монотонність, інтенсивні зорові навантаження, статичні навантаження). Оператори з малим стажем більше страждають від факторів середовища, тоді як досвідченіші працівники з більшим стажем більше зазнають впливу трудового процесу.

Комп'ютерний зоровий синдром (КЗС) є комплексом порушень здоров'я, який часто зустрічається у користувачів персональних комп'ютерів [27]. Це захворювання зазвичай розвивається, коли робоче місце організоване неправильно, наприклад, коли відсутні зручні крісла, підставки для ніг чи рук, а також коли неправильно налаштована висота і відстань монітора. Така організація праці змушує людину перебувати в статичному положенні, що призводить до напруження м'язів, погіршення кровообігу та виникнення головного болю, запаморочення, а також порушень зору, таких як туман чи мерехтіння перед очима.

На розвиток КЗС також впливають фактори навколишнього середовища, такі як поганий мікроклімат, іонізація та мікробне забруднення повітря, а також куріння в робочих приміщеннях. Все це може погіршувати стан здоров'я користувачів ПК. В результаті інтенсивної зорової роботи, користувачі можуть відчувати такі

симптоми, як біль, різь, печіння в очах, а також швидку втомлюваність і проблеми з концентрацією.

Для профілактики КЗС важливо правильно організувати робоче місце, забезпечити комфортні умови для роботи з комп'ютером і регулярно робити перерви. Урахування всіх факторів, що можуть негативно впливати на здоров'я, допомагає розробити ефективні заходи для збереження працездатності та запобігання розвитку таких захворювань.

ВИСНОВКИ

Методика досліджень, застосована у роботі, базувалася на використанні сучасних інструментів розробки веб-додатків. Основою для реалізації проєкту стала архітектура MVC (Model-View-Controller), яка дозволяє ефективно розподілити логіку додатка на три основні частини: Model, View, Template — використовується для створення динамічних HTML-сторінок на основі переданих даних.

Вибір інструментів, таких як Django та PostgreSQL, дозволив оптимізувати процес розробки та забезпечити безпеку, масштабованість і продуктивність системи.

В роботі було охарактеризовано слабкі і сильні сторони інтернет-комерції, здійснено аналіз найбільш популярних маркетплейсів, чиєю специфікою роботи є магазини декору в Україні. Крім того, ми виконали аналіз найбільш затребуваних систем розробки сайтів, розглянули їх переваги та недоліки.

Відтак, аналіз теоретичної блоку, дозволив нам систематизувати практичну частину роботи в контексті розробки сайту інтернет-магазину.

Отже, в якості висновків до кваліфікаційної роботи магістра, хочемо відмітити, що спроектовано та розроблено інтернет-магазин декору, який містить клієнтську, серверну та адміністративну частини. Також реалізовано інтеграцію з базою даних PostgreSQL для надійного зберігання даних про користувачів, замовлення та товари. Було забезпечено зручний адміністративний інтерфейс для управління контентом магазину, включаючи додавання та редагування продуктів, перегляд замовлень та розроблено адаптивний вебдизайн для зручності користувачів на різних пристроях. Проєкт пройшов тестування основних функцій, що підтвердило стабільну роботу додатка та відповідність технічним вимогам.

Сайт магазину декору має високий потенціал для практичного застосування в реальних умовах. Основні переваги:

- Можливість використання як основи для створення аналогічних платформ електронної комерції.
- Простота інтеграції з іншими сервісами, такими як платіжні системи чи служби доставки.
- Гнучкість для адаптації під індивідуальні потреби бізнесу завдяки модульній структурі Django.

Використання безкоштовного програмного забезпечення, такого як Django та PostgreSQL, дозволяє мінімізувати витрати на розробку. Оптимізація роботи магазину через зручний інтерфейс і швидку обробку запитів сприяє залученню клієнтів. Архітектура системи дозволяє легко розширювати функціонал у разі зростання бізнесу.

Інформаційно-пошуковий сайт пройшов локальне тестування, і отримані результати повністю відповідали очікуванням. Було також проведено перевірку на наявність помилок, за підсумками якої підтверджено безперебійну та надійну роботу сайту. Таким чином, усі вимоги, визначені у постановці завдання, були успішно виконані.

У ході дослідження вирішено низку теоретичних і практичних завдань, вдосконалено навички програмування, а також виконано оформлення матеріалів у графічних редакторах.

У результаті досягнуто поставленої мети кваліфікаційної роботи магістра.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – “Інженерія програмного забезпечення” для усіх форм навчання [Текст] – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя – 2020 – 27 с.
2. Pasika DR, Tsupryk HB (2023) Rozrobka odnostorinkovoho veb-zastosunku z vykorystanniam VUE. JS i REACT: porivnialnyi analiz produktyvnostib korystuvatskoho dosvidu ta dotsilnosti [Developing a single-page web application using VUE. JS and REACT: a comparative analysis of user experience performance and expediency]. IMSTT (Tern., 13-14 December 2023), pp. 218-219 [in Ukrainian].
3. Methods of constructing algorithms for comparative test statistical verification of mathematical models of bioobject responses to low-intensity stimuli ЄБ Яворська, БІ Яворський, ГБ Цуприк, РВ Кінаш - Scientific Journal of TNTU, 2023
4. Бойко І.В., Петрик М.Р., Цуприк Г.Б. Дискретні структури (Алгебраїчні та числові системи, комбінаторний аналіз): навчально-методичний посібник для студентів спеціальності 121 «Інженерія програмного забезпечення», аспірантів та викладачів вищих навчальних закладів – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя – 2017 – 64 с.
5. Пасічник В. В. Веб-технології [Текст]: підруч. / В. В. Пасічник, О.В. Пасічник, Д. І. Угрин. – Львів : Магнолія 2013. – 336 с.
6. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Моделювання та видобуток даних (висопродуктивні обчислення у великих алгебраїчних та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: : ТНТУ 2019 – 62 с.
7. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник. Тернопіль: : ТНТУ 2020 – 62 с.

8. Силка М. В., Цуприк Г.Б. Сучасні технології при розробці програмного забезпечення для автоматизованої інформаційної системи обробки замовлень та товарообігу. Збірник тез доповідей VII Міжнародної науково-технічної конференції молодих учених та студентів. Актуальні задачі сучасних технологій – Тернопіль 28-29 листопада 2018 року. — Т. : ТНТУ, 2018. — Том 2. — С. 161–162.
9. О Остапчук, Г Цуприк. Технічні особливості взаємодії між клієнтом та сервером у реальному часі. Збірник тез доповідей X науково-технічної конференції. Інформаційні моделі, системи та технології – Тернопіль 7 грудня 2022 року. — Т. : ТНТУ, 2022— С. 124.
10. Python Developer Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/>.
11. Susan J. Flower. Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization 1st Edition [Текст] / Susan J. Flower. – O'Reilly, 2016. – 124 с.
12. Decorize. Все для флористів, декора і творчості [Електронний ресурс] – Режим доступу до ресурсу: <https://decorize.com.ua/>
13. Andreas Antonopoulos, Gavin Wood Ph.D. Mastering Ethereum: Building Smart Contracts and DApps 1st Edition / Gavin Wood Ph.D. – O'Reilly Media, 2018. – 269с.
14. Docker Developer Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.docker.com/develop/>.
15. Curtin Beau. Django Cookbook: Web Development with Django – Step by Step Guide 2nd edition — CreateSpace Independent Publishing Platform, 2016. — 182 p.
16. National Geographic [Електронний ресурс] <https://www.nationalgeographic.com/>
17. Aratyn Tom. Building Django 2.0 Web Applications . - Packt Publishing, 2018. 406 p.
18. Bootstrap – бібліотека готових рішень URL [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.3/customize/color/>
19. Baumgartner Peter, Malet Yann. High Performance Django. - CreateSpace Independent Publishing Platform, 2015. — 184 p.

20. Bendoraitis Aidas, Kronika Jake. Django 3 Web Development Cookbook: Actionable solutions to common problems in Python web development : 4th Edition. — Packt Publishing, 2020. — 936 p.
21. Dauzon S., Bendoraitis A., Ravindran A. Django: Web Development with Python Packt Publishing, 2016. — 730 p
22. Fabrizio Romano, Gaston C. Hillar, Arun Ravindran. Learn Web Development with Python Packt Publishing, 2018. — 796 p.
23. Jaiswal S., Kumar K. Learning Django Web Development Packt Publishing, 2015. — 405 p.
24. Vincent William S. Django for Beginners: Build websites with Python and Django Independently published, 2019. — 339 p.
25. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508-18>.
26. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text>.
27. Комп'ютерний зоровий синдром [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://linza.com.ua/uk/articles/blog/kompyuternyy-zritelnyy-sindrom-simptomy-i-lechenie/>.
28. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / В.С. Стручок –Тернопіль: ФОП Паляниця В. А., –156 с.

ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА**

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



18-19 грудня 2024 року

**ТЕРНОПІЛЬ
2024**

УДК 004.4'2

Ольга Бойко, магістрантка

Галина Цуприк – к.т.н., доц.

Тернопільський національний технічний університет імені Івана Пулюя

ПРОЄКТУВАННЯ ТА РОЗРОБКА САЙТУ ІНТЕРНЕТ-МАГАЗИНУ ДЕКОРУ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ DJANGO

Olga Boiko, Master Student

Halyna Tsupryk – Ph.D. Assoc. Prof

DESIGN AND DEVELOPMENT OF AN ONLINE DECOR STORE WEBSITE USING THE DJANGO FRAMEWORK

Актуальність теми роботи полягає у зростанні значення електронної комерції, особливо у сфері продажу товарів для дому та декору. У зв'язку з пандемією та поширенням дистанційних моделей споживання, необхідність у зручних та функціональних інтернет-магазинах стає дедалі важливішою. Сучасні умови ринку вимагають створення інноваційних рішень, які забезпечують високу якість обслуговування клієнтів та зручність користування. Django, як інструмент веброзробки, дозволяє швидко та ефективно реалізовувати функціональні та масштабовані інтернет-магазини, що відповідають вимогам часу.

Розроблений інтернет-магазин декору призначений для забезпечення користувачів можливістю зручного перегляду, пошуку та замовлення товарів для прикрашення житлових і комерційних приміщень. Сайт надає широкий вибір категорій товарів, систему фільтрації для швидкого пошуку, а також інтеграцію з платіжними системами для забезпечення безпечних транзакцій.

Об'єктом дослідження є процеси розробки вебдодатків для електронної комерції з використанням сучасних технологій. Зокрема, акцент зроблено на архітектурі вебдодатків, побудованих на фреймворку Django.

Для реалізації проєкту було обрано такі технології:

- Фреймворк Django – для створення серверної частини та управління базами даних.
- СУБД PostgreSQL – для зберігання інформації про товари, користувачів і замовлення.
- HTML, CSS, JavaScript – для розробки інтерфейсу користувача.
- Bootstrap – для адаптивного дизайну.
- Visual Studio Code – як основне середовище розробки.

Розроблений вебсайт ілюструє можливості Django як фреймворку для створення зручних, безпечних і конкурентоздатних платформ. Використання Django сприяє зниженню витрат на розробку, а також забезпечує можливість легкої адаптації до змін ринкових умов. Інтеграція зручного інтерфейсу, функціоналу замовлень і аналітичних інструментів сприяє автоматизації процесів продажу, що в свою чергу підвищує ефективність бізнесу та забезпечує кращий досвід для кінцевого споживача.

Література

1. Michael Dinder. Becoming an Enterprise Django Developer 1st Edition/ Michael Dinder. – Packt Publishing, 2022. – 526 с.
2. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – «Інженерія програмного забезпечення» для усіх форм навчання [Текст] – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя – 2020 – 27 с.

Додаток Б – Алгоритм роботи Корзини

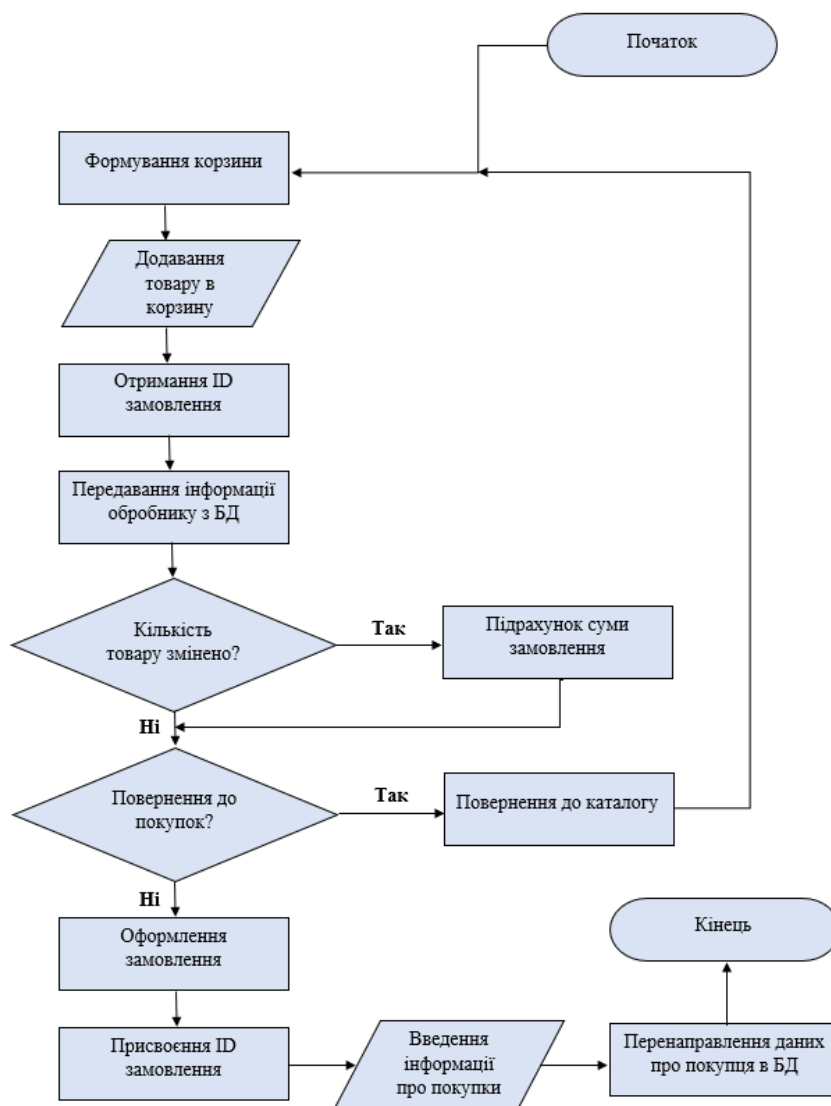


Рис. Б.1 – Алгоритм роботи Корзини

Додаток В – Алгоритм роботи інтернет-магазину

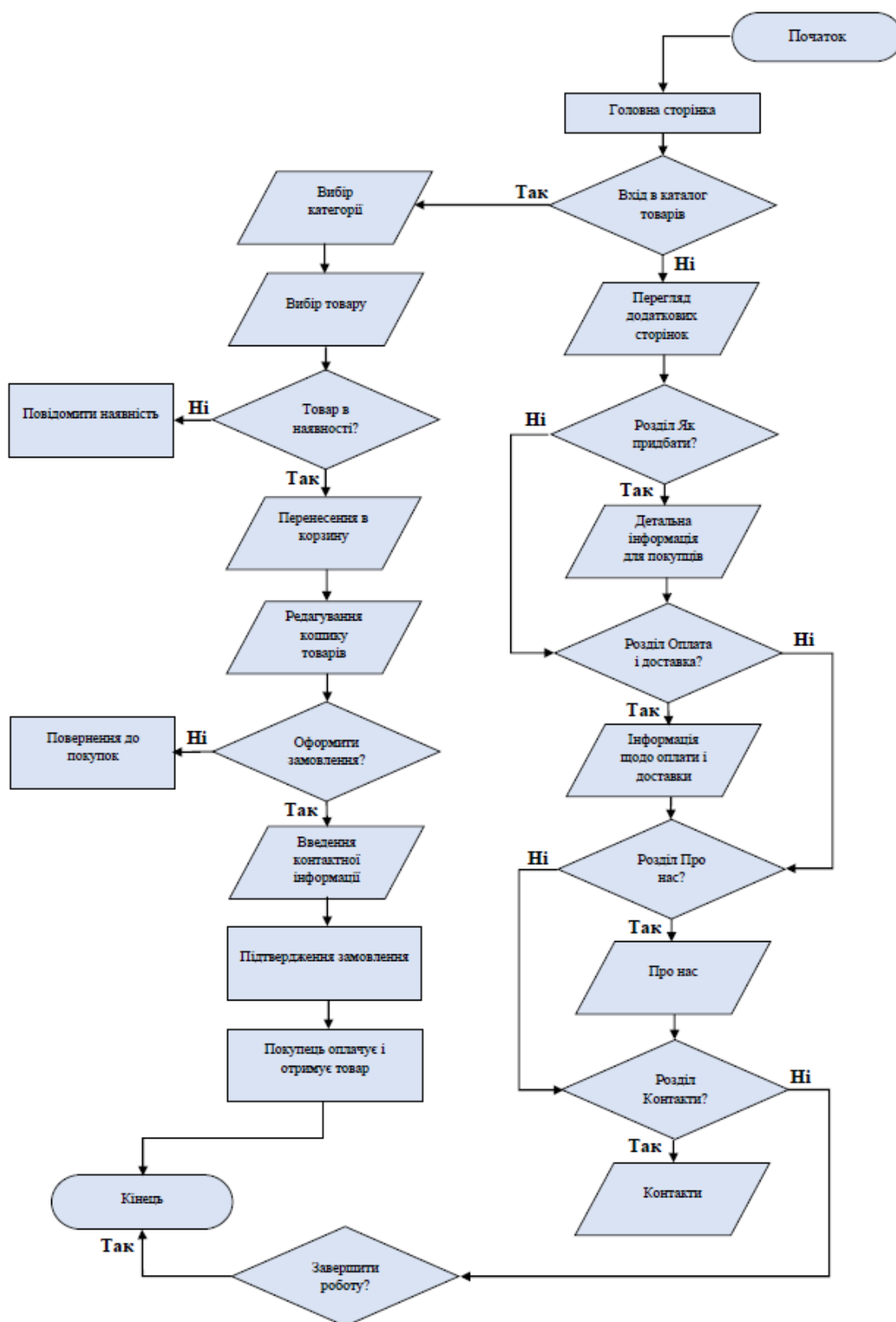


Рис. В.1 – Алгоритм роботи інтернет-магазину

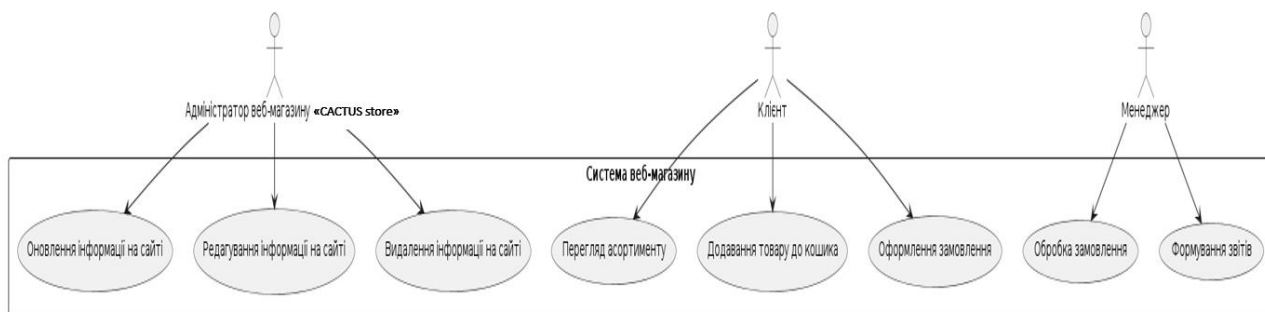


Рис. Г.1 – Діаграма прецедентів

