

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ
Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

ЗУБОК ДІАНА ВАСИЛІВНА

**ПРОЄКТУВАННЯ ТА РОЗРОБКА САЙТУ ДЛЯ ПРОДАЖУ ДИТЯЧИХ
ТОВАРІВ ЗАСОБАМИ ВЕБТЕХНОЛОГІЙ**

Спеціальність: 122 «Комп'ютерні науки»

Освітньо-професійна програма: Комп'ютерні науки та інформаційні технології
Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»

Науковий керівник:
ГЛИНЧУК ЛЮДМИЛА ЯРОСЛАВІВНА,
кандидат фізико-математичних наук, доцент
кафедри комп'ютерних наук та кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ

Протокол № _____

засідання кафедри комп'ютерних наук
та кібербезпеки

від _____ 2025 р.

Завідувач кафедри

(_____) Гришанович Т. О.

Луцьк-2025

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ВЕБСАЙТІВ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.....	5
1.1. Сучасні тенденції розвитку електронної комерції.....	5
1.2. Огляд вебтехнологій для створення вебсайтів.....	7
1.2.1. Фреймворк Django.....	9
1.2.2. Фреймворк Laravel.....	9
1.2.3. Фреймворк Nest.js.....	11
1.3. Особливості розробки вебсайтів для продажу товарів.....	13
1.4. Огляд та аналіз аналогічних розробок.....	15
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСАЙТУ ДЛЯ ПРОДАЖУ ДИТЯЧИХ ТОВАРІВ.....	22
2.1. Постановка задачі, призначення та вимоги до програмного засобу.....	22
2.2. Вибір моделі розробки програмного засобу.....	23
2.3. Загальний опис проєкту.....	24
2.4. Обґрунтування вибору інструментальних засобів розробки.....	28
2.4.1. Середовище розробки PyCharm.....	28
2.4.2. Клієнтська частина.....	29
2.4.3. Серверна частина.....	31
2.5. Особливості програмної реалізації та основні режими роботи.....	33
2.6. Організація тестування та налагодження програмного засобу.....	42
2.7. Рекомендації по використанню та впровадженню програмного засобу... ..	44
ВИСНОВОК.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТОК А.....	51
ДОДАТОК Б.....	54
ДОДАТОК В.....	58

ВСТУП

Актуальність теми полягає в швидкому розвитку електронної комерції та зростанні попиту на зручні онлайн-платформи для покупки товарів. Батьки все більше шукають можливості придбати дитячі товари через інтернет, оскільки це дозволяє заощадити час, порівняти ціни та зробити покупки в будь-який зручний момент. У наш час вимоги до вебсайтів вимагають використання передових вебтехнологій, що забезпечують адаптивність дизайну, швидкість завантаження сторінок та зручність використання ресурсу на різних пристроях. Крім того, розробка індивідуального вебресурсу дозволяє враховувати специфіку цільової аудиторії, забезпечуючи краще позиціонування на ринку порівняно з універсальними платформами. Створення власного сайту для продажу дитячих товарів сприяє підвищенню довіри клієнтів, оптимізації внутрішніх процесів бізнесу та відкриває можливості для подальшого функціонального розширення.

Таким чином, тема є актуальною, оскільки вона відповідає сучасним вимогам ринку, сприяє розвитку електронної комерції та вдосконаленню користувацького досвіду, що важливо як для потенційних споживачів, так і для бізнесу, який прагне забезпечити стабільний розвиток у конкурентному середовищі.

Метою даної роботи є розробка зручного, адаптивного та функціонального вебсайту для продажу дитячих товарів, що забезпечить просте адміністрування, ефективну взаємодію з користувачем та відповідатиме сучасним вимогам веброзробки й електронної комерції.

Для досягнення визначеної мети потрібно виконати такі **завдання**:

- дослідити сучасні тенденції розвитку електронної комерції;
- проаналізувати сучасні вебтехнології, що використовуються для створення комерційних вебресурсів;

- визначити особливості та вимоги до вебсайтів, орієнтованих на продаж товарів онлайн;
- провести огляд аналогічних вебрішень і проаналізувати їхні переваги та недоліки;
- сформулювати вимоги до функціональності та дизайну вебсайту;
- обґрунтувати вибір моделі процесу розробки та вибраних інструментальних засобів;
- спроектувати структуру сайту та розробити його основні компоненти: каталог товарів, кошик, сторінки товарів, система управління контентом;
- реалізувати програмний засіб відповідно до визначених вимог;
- провести тестування та налагодження вебзастосунку.

Об'єкт дослідження – процес створення вебзастосунків у сфері електронної комерції.

Предмет дослідження – розробка вебсайту для продажу дитячих товарів з урахуванням сучасних вимог до комерційних вебплатформ.

Апробація результату роботи. Результати проведеного дослідження були апробовані у формі тез доповіді на тему «Особливості використання фреймворку Django для створення вебсайту», представленої на II Міжнародній науково-практичній конференції «Проблеми комп'ютерних наук, програмного моделювання та безпеки цифрових систем», яка відбулася 9–11 червня 2025 року на базі практик табору «Гарт». Конференція проводилася кафедрою комп'ютерних наук та кібербезпеки Волинського національного університету імені Лесі Українки.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ВЕБСАЙТІВ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1. Сучасні тенденції розвитку електронної комерції

Електронна комерція продовжує активно розвиватися, змінюючи традиційні підходи до торгівлі та взаємодії з клієнтами. Основні тенденції, що спостерігаються останніми роками, свідчать про глибоку цифрову трансформацію ринку. Поява нових технологій, зміна споживацької поведінки та зростаючі вимоги до зручності і швидкості обслуговування стимулюють бізнеси до постійного оновлення своїх електронних платформ. Водночас конкуренція в онлайн-середовищі змушує компанії впроваджувати інноваційні рішення, щоб утримувати увагу користувачів і забезпечувати якісний клієнтський досвід.

Однією з ключових тенденцій є персоналізація сервісу. Користувачі очікують, що платформа буде розуміти їхні вподобання, попередній досвід покупок і перегляди, пропонуючи саме ті товари, які можуть їх зацікавити. Для цього бізнеси активно використовують аналітику, алгоритми машинного навчання та системи рекомендацій, що формують індивідуальні пропозиції. Компанії, що впроваджують технології штучного інтелекту, скорочують витрати на обслуговування клієнтів до 30%, водночас покращуючи швидкість реагування та рівень задоволеності клієнтів [1]. Персоналізований підхід дозволяє не лише підвищити рівень задоволення клієнтів, а й сприяє зростанню повторних продажів.

Помітним став розвиток мобільної комерції, адже користувачі дедалі частіше здійснюють покупки за допомогою смартфонів або планшетів. Щоб задовольнити потреби мобільних користувачів, компанії впроваджують прогресивні веб-додатки (PWA). PWA – це веб-додатки, які забезпечують

нативний інтерфейс, подібний до додатків, на мобільних пристроях. Вони поєднують гнучкість вебу з функціональністю та продуктивністю нативних додатків, пропонуючи безперебійний та адаптивний користувацький досвід [2].

Варто відзначити, що поширення набула автоматизована взаємодія з клієнтами. Зокрема, широкого поширення набули чат-боти та віртуальні помічники, які можуть відповідати на запити, допомагати з вибором товару, інформувати про акції чи доступність продукції. Автоматизована підтримка дозволяє магазинам обслуговувати значно більшу кількість клієнтів без втрати якості обслуговування. Завдяки сучасним інструментам автоматизації компанії можуть забезпечувати цілодобову підтримку, швидко реагувати на типові запити та оптимізувати навантаження на персонал [3]. Вона також забезпечує постійний доступ до інформації, навіть поза робочими годинами.

Ще однією важливою характеристикою сучасної електронної комерції є омніканальність. Це підхід, за якого взаємодія з клієнтом здійснюється через різні канали – сайт, соціальні мережі, мобільний застосунок, месенджери – з єдиною метою: забезпечити безшовний та зручний досвід. Наприклад, користувач може побачити рекламу товару в Instagram, перейти на сайт магазину, додати товар у кошик, а завершити покупку уже через мобільний застосунок або месенджер. Усі ці канали повинні бути інтегровані між собою, щоб користувач не відчував розриву у процесі покупки. Бізнеси дедалі частіше вибудовують єдину комунікаційну стратегію, що охоплює всі цифрові точки контакту з клієнтом – від соціальних мереж до розсилок на електронну пошту і фізичних магазинів [3].

Крім того, зростає попит на прозорі умови доставки, гнучкі способи оплати, швидке оформлення замовлення та зручні інтерфейси. Споживачі очікують, що весь процес буде максимально простим, швидким і передбачуваним. У зв'язку з цим онлайн-магазини впроваджують покращені логістичні системи, інтегрують платіжні сервіси, надають можливість відстеження замовлення та спрощують повернення товарів. Також важливо,

щоб користувачі мали можливість здійснювати покупки в кілька кроків, без зайвих етапів [3].

Таким чином, електронна комерція сьогодні – це не просто продаж товарів в інтернеті, а ціла екосистема, що постійно вдосконалюється та реагує на виклики часу. Ключовими напрямками її розвитку є персоналізація, автоматизація, омніканальність та клієнтоорієнтованість. Усі ці фактори формують нову якість взаємодії між продавцем і покупцем, підвищують ефективність бізнесу та сприяють формуванню довгострокових відносин із клієнтами.

1.2. Огляд вебтехнологій для створення вебсайтів

У сучасному цифровому середовищі створення вебсайтів стало не лише важливим інструментом представлення бізнесу, інформації чи послуг, а й окремим напрямом у сфері інформаційних технологій. Успішний вебсайт поєднує в собі зручний інтерфейс, привабливий дизайн та ефективну логіку взаємодії з користувачем. Від правильного поєднання технічних рішень і дизайну залежить, чи зможе сайт утримати відвідувача й перетворити його на покупця. Технології веброзробки постійно вдосконалюються, що дозволяє створювати більш функціональні та адаптивні рішення для бізнесу. Умовно всі вебтехнології можна поділити на фронтенд (клієнтська частина) і бекенд (серверна частина).

Фронтенд – це все, що бачить користувач у браузері: текст, кнопки, зображення, меню, а також анімації й інтерактивні елементи. До основних технологій, які застосовуються для створення фронтенду, належать:

1. HTML (HyperText Markup Language) – це мова розмітки гіпертексту, яка використовується для створення структури та макету вебсторінок. Основними компонентами є елементи, теги та атрибути. HTML

використовується для побудови структури та розмітки вебсторінок, відображення тексту зображень та мультимедійного контенту [4].

2. CSS (Cascading Style Sheets) – це мова стилів, призначена для візуального оформлення документів, написаних мовами розмітки, зокрема HTML. CSS використовується для стилізації вебсторінок (кольори, шрифти, відступи тощо) та для адаптивного дизайну, що робить коректним відображення на різних пристроях та екранах [5].
3. JavaScript – це мова програмування, яку зазвичай використовують веброзробники для додавання динамічних взаємодій вебсторінок, програм, серверів і навіть ігор. Ця мова ідеально працює разом з HTML та CSS, доповнюючи CSS у форматуванні елементів HTML [6].

У той час як фронтенд відповідає за взаємодію з користувачем, бекенд виконує обробку даних, реалізацію логіки та зв'язок між різними частинами системи. На графіку (рис. 1.1) зображено найпопулярніші фреймворки станом на січень 2025 року, які найчастіше програмісти використовують для створення бекенду [7, 3:20]. Далі детально розглянемо першу трійку.

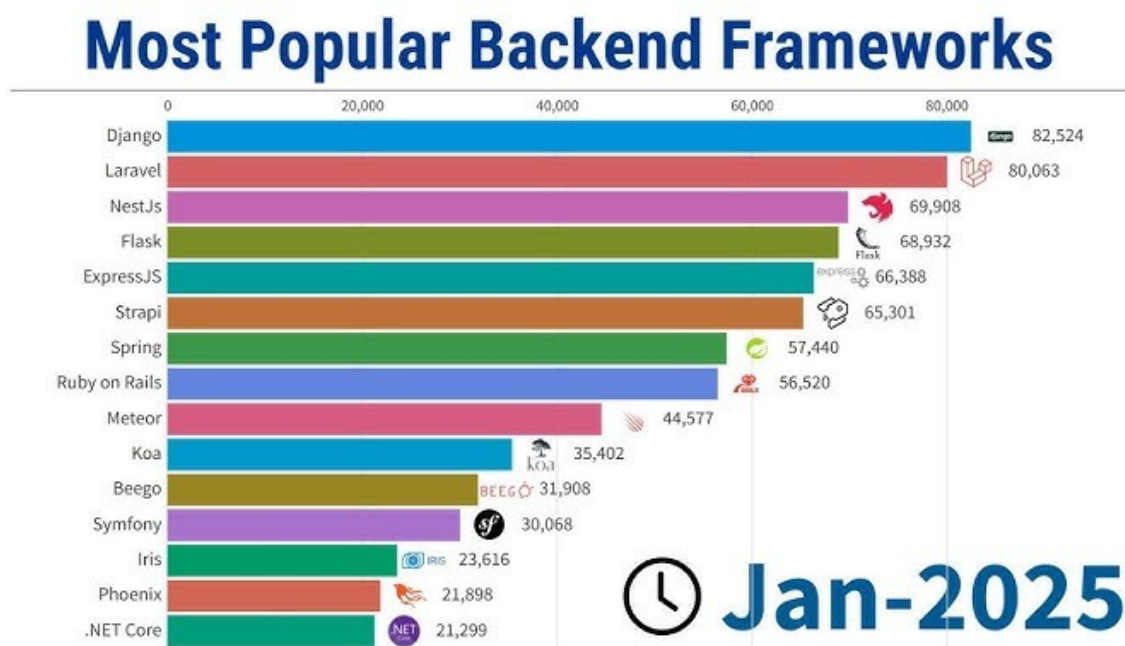


Рис. 1.1 – Графік популярності фреймворків

1.2.1. Фреймворк Django

Django – це високорівневий вебфреймворк для мови програмування Python, створений із фокусом на швидкість розробки, безпеку та масштабованість. Він забезпечує розробника великим набором інструментів, необхідних для створення повноцінного вебзастосунку без потреби у встановленні сторонніх бібліотек. До переваг фреймворку належить наявність вбудованої адміністративної панелі, ORM для роботи з базами даних, система маршрутизації, а також численні механізми захисту від поширених вразливостей. Django активно використовується у великих проєктах, зокрема такими компаніями, як Instagram, Pinterest, Spotify, National Geographic, The Washington Post і навіть Nasa [8]. Попри безліч переваг, Django може здаватися складним для початківців через строгу структуру проєктів та необхідність дотримання визначених правил. Саме цей фреймворк був обраний для створення вебсайту для продажу дитячих товарів. Детальний опис принципів роботи, приклади реалізації функціональності, а також роль Django у створення програмного продукту, що розробляється у межах цієї бакалаврської роботи, буде представлено у 2 розділі.

1.2.2. Фреймворк Laravel

Laravel – це популярний фреймворк для веброзробки, який побудований на мові програмування PHP. Фреймворк був вперше представлений у 2011 році Тейлором Отвеллом і з того часу набув популярності [9]. Фреймворк вирізняється елегантним синтаксисом, який дозволяє створювати чистий, ефективний і легко підтримуваний код. Laravel підходить як для невеликих проєктів, так і для складних корпоративних систем.

Переваги Laravel [9, 10]:

1. Зрозумілий синтаксис завдяки якому розробники швидше адаптуються до роботи з фреймворком.
2. Використовує вбудовану ORM-систему під назвою Eloquent, яка дозволяє працювати з базами даних за допомогою об'єктно-орієнтованого підходу. Це спрощує написання запитів і дає змогу легко керувати зв'язками між таблицями, роблячи роботу з БД більш зрозумілою.
3. Має готові механізми для автентифікації та авторизації, що підвищує рівень безпеки застосунку та зменшує ризик помилок, пов'язаних із реалізацією доступу вручну.
4. Потужний інструмент командного рядка Artisan, який дозволяє автоматизувати типові завдання: створення моделей, контролерів, міграцій тощо.
5. Містить шаблонізатор Blade, який дозволяє створювати гнучкі та багаторазові шаблони HTML-сторінок.
6. Має велику кількість офіційних та сторонніх пакетів (наприклад, Horizon, Passport, Sanctum), які розширюють функціональність проєкту.
7. Активна спільнота розробників надає доступ до документації, форумів, відеоуроків, що полегшує навчання для початківців.
8. Зручність у тестуванні завдяки інтеграції з PHPUnit. Розробник може швидко писати як прості тести, так і комплексні тести застосунку, перевіряючи правильність роботи окремих частин коду.

Недоліки Laravel [9, 10]:

1. Може бути менш ефективним у порівнянні з легшими PHP фреймворками або чистим PHP, особливо при роботі з великими обсягами даних чи під великим навантаженням.
2. Через багатofункціональність новачкам може бути складно одразу зрозуміти всі концепції.

3. Постійні оновлення, що загалом добре, але це може викликати труднощі з підтримкою старих проєктів. Під час оновлення можуть з'являтися несумісності або потреба переписати частину коду.
4. Потребує багато оперативної пам'яті та процесорного часу, що може спричинити потребу у потужнішому сервері або дорожчому хостингу, особливо це актуально для великих або складних застосунків.
5. Важко реалізувати нестандартну функціональність, адже доведеться глибоко занурюватись у внутрішні механізми Laravel.

Загалом, Laravel – це чудовий вибір для тих, хто прагне створювати масштабовані, структуровані та безпечні вебзастосунки, особливо якщо ви плануєте довготривалу підтримку проєкту. Даний фреймворк використовується у багатьох проєктах, зокрема з відомих: BBC, Disney, Twitch, New York Times і Warner Bros [11].

1.2.3. Фреймворк Nest.js

Nest.js – це прогресивний фреймворк для Node.js, його мета – надати розробникам потужний інструмент для створення масштабованих, підтримуваних та ефективних серверних додатків, використовуючи сучасні підходи до архітектури та TypeScript. Натхнений Angular, Nest.js поєднує об'єктно-орієнтоване програмування, функціональне програмування та реактивне програмування для побудови надійних серверних рішень [12].

Переваги Nest.js [12, 13]:

1. Завдяки своїй модульній архітектурі, дозволяє легко організовувати код у модулі, сервіси та контролери. Це сприяє кращій підтримці та масштабуванню додатків, особливо у великих командах.
2. Використовує TypeScript, що забезпечує строгі типи, автоматичне завершення коду, статичну перевірку та покращену інтеграцію з IDE. Це зменшує ймовірність помилок на ранніх етапах розробки, підвищує

продуктивність розробників і полегшує обслуговування коду в майбутньому.

3. Запозичує архітектурні принципи Angular, зокрема декоратори, DI (ін'єкцію залежностей), структуру проєкту. Дана перевага суттєво полегшує входження для фронтенд-розробників або команд, які вже працюють з Angular, адже архітектура знайома та передбачувана.
4. Nest.js має потужну інтеграцію з GraphQL. Використовуючи декоратори та класи, розробники можуть швидко створювати схеми без необхідності писати багато повторюваного коду.
5. Швидко розвивається завдяки відкритому коду, сильній спільноті та регулярним оновленням. Існує велика кількість пакетів, генераторів, шаблонів, що спрощують інтеграцію з популярними технологіями.
6. Ідеально інтегрується з `nx.dev` – інструментом для роботи з монорепозіторіями. Дозволяє масштабовано керувати проєктами, мати єдиний репозіторій для клієнтської і серверної частин, повторно використовувати модулі, перевіряти залежності та авторизовувати CI/CD.

Недоліки Nest.js [12,13]:

1. Хоча архітектура є продуманою та структурованою, новачки можуть зіткнутися з труднощами у розумінні таких понять, як декоратори, DI-контейнери, життєвий цикл компонентів.
2. Для маленьких проєктів може виявитися занадто «важким». Його модульна структура, численні файли й класи можуть ускладнити розробку, не даючи суттєвих переваг порівняно з легшими фреймворками.
3. Оскільки Nest.js спроектовано з фокусом на сучасний стек (TypeScript, модульність), його інтеграція в застарілі проєкти з JavaScript може бути складною. Потрібне повне переписування архітектури, що не завжди можливо.

Отже, Nest.js – це потужний фреймворк, який пропонує сучасний підхід до створення серверних застосунків з використанням Node.js. Він поєднує

переваги об'єктно-орієнтованого програмування, реактивності та строгого типування. Його архітектура чудово підходить для масштабних проєктів, мікросервісів, монорепозіторіїв та корпоративного програмного забезпечення. Однак використання Nest.js має сенс тоді, коли команда знайома з TypeScript, Angular або має потребу у добре структурованому, масштабованому проєкті. У випадку простих сервісів або обмежених ресурсів доцільніше розглянути легші альтернативи. Nest.js активно використовуються такими відомими компаніями: Adidas, BMW, IBM, JetBrains, GitLab, Mercedes-Benz [14].

1.3. Особливості розробки вебсайтів для продажу товарів

Розробка вебсайтів для продажу товарів має низку специфічних вимог і особливостей, що зумовлені необхідністю забезпечення зручного користувацького досвіду, високої надійності, безпеки та ефективної взаємодії з базою даних. Такий сайт повинен виконувати не лише інформаційну, а й комерційну роль, спрямовану на стимулювання користувачів до здійснення покупки. Розглянемо детально функції, які повинні бути у кожного інтернет-магазину [15, 16].

Каталог товарів є ключовим елементом будь-якого вебсайту для продажу, адже саме через нього здійснюється основна навігація користувачів по асортименту магазину. Якісний каталог не має обмежень у кількості категорій чи товарних позицій – він має масштабуватись як для малих, так і для великих торговельних майданчиків, залежно від ресурсів сервера. Однією з важливих складових є наявність стандартних і користувацьких характеристик товарів, що дозволяє налаштовувати гнучкі фільтри за матеріалом, брендом, розміром, ціною тощо. Завдяки цьому користувачі можуть швидко знайти потрібний товар без зайвих кроків. Система каталогу має підтримувати динамічне відображення варіацій одного й того ж товару (наприклад, зміну кольору або розміру без переходу на іншу сторінку). Інтерфейс каталогу повинен підтримувати функції

попереднього перегляду товару, порівняння позицій, швидке застосування фільтрів без перезавантаження сторінки, а також відображення актуальних цій та наявних товарів на складі.

Важливою складовою сайту для продажу товарів є кошик, який виступає проміжним етапом між вибором товару та його придбанням. Сучасні вебсайти повинні реалізовувати інтерактивний кошик, що дозволяє користувачу змінювати кількість обраних товарів, видаляти позиції, переглядати підсумкову вартість покупки, а також переходити до етапу оформлення замовлення. Кошик має бути доступним із будь-якої сторінки сайту, наприклад, у вигляді іконки у верхньому куті, що оновлюється в реальному часі. Процес оформлення замовлення повинен бути максимально спрощеним. Сучасні фахівці радять використовувати модель «односторінкового оформлення», де користувач вводить усі необхідні дані (ім'я, номер телефону, адресу доставки) на одній сторінці без додаткових переходів. Вебсайти для продажу товарів часто потребують інтеграції з платіжними системами. Незважаючи на те, що під час початкової розробки можливо обійтись без підключення сторонніх API, програмна архітектура має передбачати можливість такої інтеграції у майбутньому.

Окрему увагу слід приділити адміністративній панелі сайту, яка забезпечує керування товарами, замовленнями та користувачами. Через цю панель адміністратор може додавати нові позиції до каталогу, змінювати опис товарів, оновлювати ціни, керувати залишками товарів на складі, встановлювати знижки, а також переглядати та обробляти замовлення. Система адміністрування має бути зручною, інтуїтивно зрозумілою та безпечною, з обмеженням доступу для сторонніх користувачів. Безпека – це ще один ключовий аспект. Оскільки сайт обробляє персональні дані користувачів, необхідно реалізувати захищене з'єднання через HTTPS, використання надійних механізмів авторизації й автентифікації, захист від поширених атак.

Ще однією важливою особливістю є адаптивний дизайн. Зважаючи на те, що значна частина користувачів заходить на сайти з мобільних пристроїв, інтерфейс має коректно відображатися на смартфонах і планшетах. Елементи повинні автоматично підлаштовуватись під розмір екрана. Оптимізація швидкодії сайту є критичною, адже повільне завантаження сторінок негативно впливає на досвід користувачів і позиції у пошукових системах. Щоб залучати нових клієнтів та аналізувати ефективність роботи сайту, доцільно впровадити аналітичні інструменти (Google Analytics, Facebook Pixel тощо). Вони дозволяють відстежувати поведінку користувачів, джерела трафіку, найпопулярніші сторінки та інші метрики.

Отже, розробка сайту для продажу товарів передбачає не лише реалізацію базового функціоналу, а й створення надійної, масштабованої, захищеної системи з урахуванням сучасних вимог до дизайну, продуктивності та аналітики. Успішний сайт для електронної комерції повинен забезпечувати зручну взаємодію з користувачем, ефективну роботу з даними і бути готовим до подальшого зростання і розвитку.

1.4. Огляд та аналіз аналогічних розробок

Для розробки якісного та функціонального вебсайту з продажу дитячих товарів важливо врахувати досвід існуючих програмних рішень аналогічного призначення. З цією метою було здійснено аналіз трьох інтернет-магазинів. Розгляд кожного з них охоплює основні характеристики, функціональні можливості, а також сильні й слабкі сторони. Такий підхід дозволяє виявити вдалі рішення, яких варто дотримуватись, і недоліки, яких слід уникати під час створення власного сайту.

«Будинок іграшок» – це інтернет-магазин, що пропонує широкий асортимент товарів [17]. Для створення цього сайту використані такі технології,

як мови програмування PHP та TypeScript для бекенду, для бази даних Firebase та MySQL.

На головній сторінці сайту можна переглянути різні пропозиції (акційні товари, іграшки певних брендів, іграшки за певною класифікацією). Сайт дозволяє додавати товари до кошика, створювати списки бажаного, оформлювати замовлення онлайн з вибором типу доставки. Крім того, реалізовано особистий кабінет користувача з історією замовлень і збереженими адресами доставки. Також представлено вітрину знижок, бонусну програму та блок із рекомендаціями товарів (рис. 1.2).

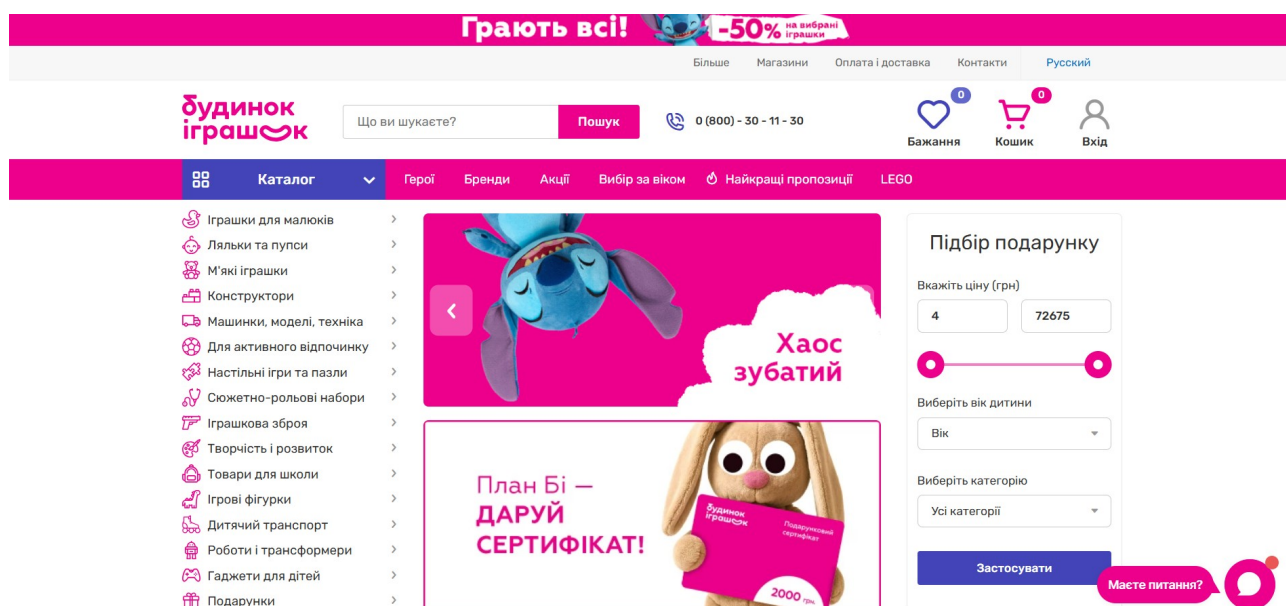


Рис. 1.2 – Головна сторінка сайту «Будинок іграшок»

Для прикладу оберемо з каталогу категорію «Ігрові фігурки», після чого відкриється нова сторінка з відповідними товарами. З лівого боку є фільтри, можна обрати фільтрацію за брендом, ціною, віком, рейтингом, країною, видом іграшки тощо (рис. 1.3). Варто відзначити, що сайт має функцію за допомогою якої можна зв'язатись з онлайн-консультантом через месенджери або безпосередньо в чаті на сторінці сайту.

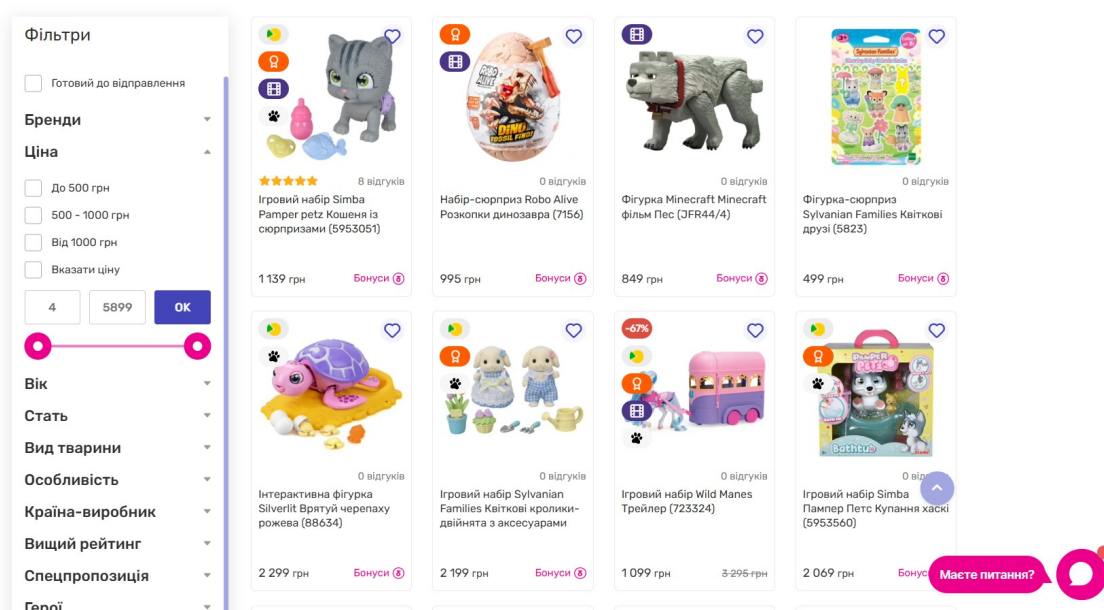


Рис. 1.3 – Каталог товарів «Будинок іграшок»

Переваги:

- зручна навігація та якісний інтерфейс;
- великий вибір категорій та брендів;
- можливість зв'язатись з онлайн-консультантом;
- функціональний особистий кабінет;
- програма лояльності з бонусами.

Недоліки:

- перевантажене головне меню, що може ускладнити пошук інформації;
- вподобані можна додавати лише зареєстрованим користувачам;
- перегляд товарів можливий лише за категоріями, не можна переглянути всі товари одразу.

«Чудо острів» – магазин іграшок вебсайт для якого розроблений компанією BRANDER [18]. Використано такі технології, як мова програмування PHP для бекенду та база даних MySQL.

Головна сторінка має подібний інтерфейс до попереднього прикладу. Тут теж є реалізований особистий кабінет, пошук, вподобані та кошик. Також є каталог товарів, де іграшки поділені за категоріями (рис. 1.4).

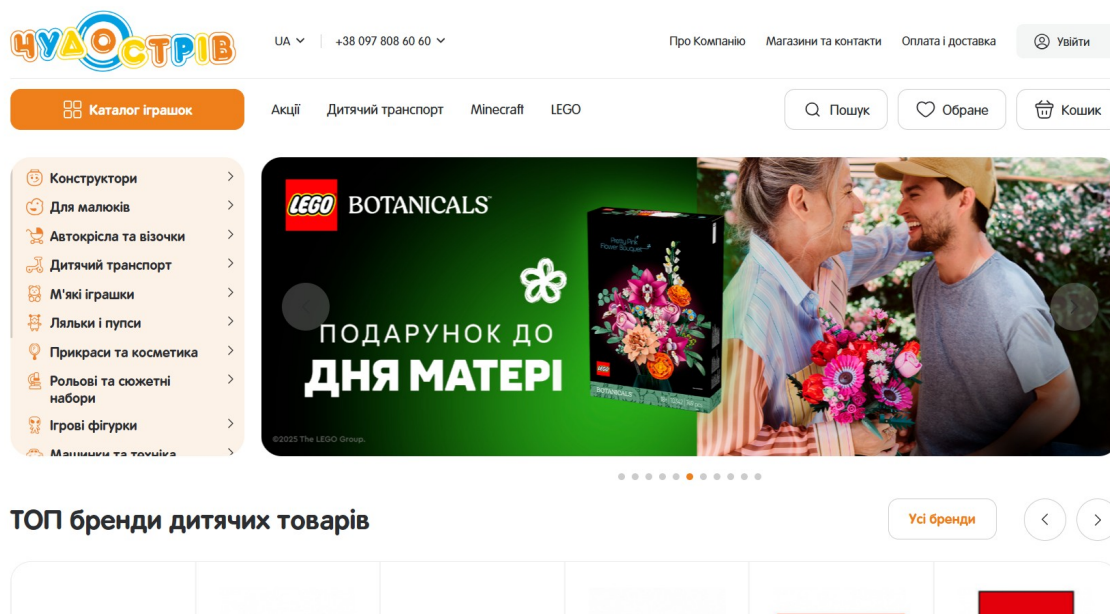


Рис. 1.4 – Головна сторінка сайту «Чудо острів»

Для перегляду каталогу оберемо категорію «М'які іграшки». З лівого боку розташовані фільтри (за категорією, віком, брендом тощо). Товари можна переглядати, додавати до вподобаних та до кошика (рис. 1.5). Також на сайті можна зв'язатись з онлайн-консультантом, який допоможе вирішити питання, якщо такі виникнуть.

Переваги:

- яскравий та приємний інтерфейс;
- програма лояльності;
- можливість перевірити наявність товару у різних магазинах;
- можливість зв'язатись з онлайн-консультантом;
- реалізований особистий кабінет.

Недоліки:

- повільне завантаження окремих сторінок;
- до вподобаних можна додавати товари лише зареєстрованим користувачам;
- перегляд каталогу можливий лише за категоріями, не можна переглянути всі товари одразу.

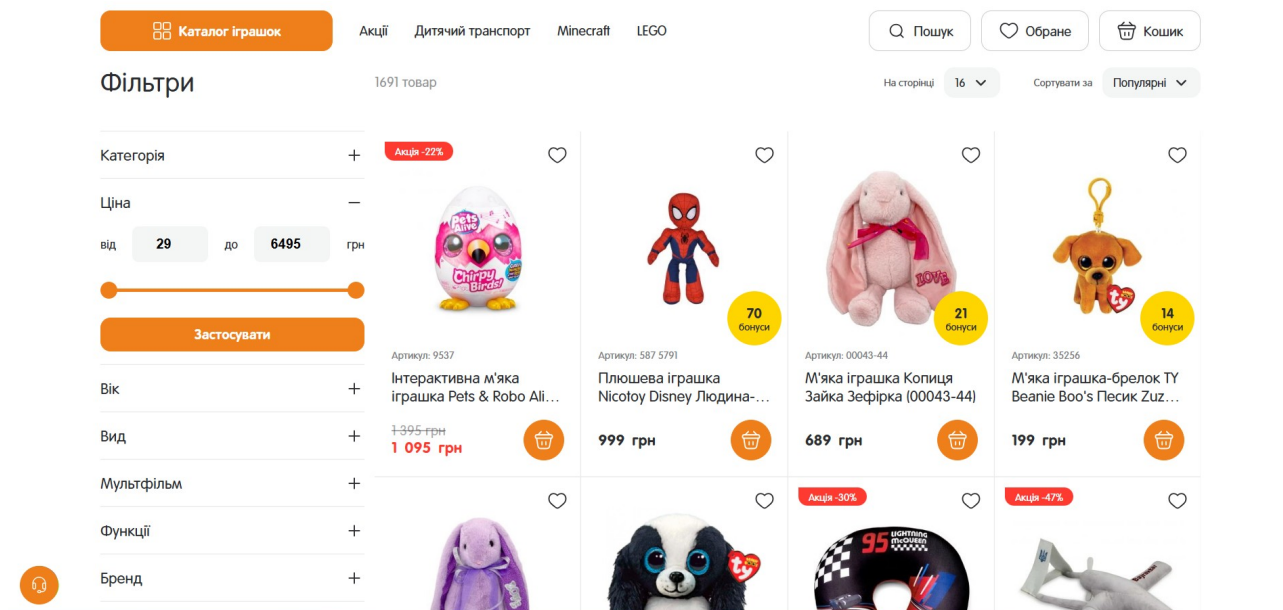


Рис. 1.5 – Каталог товарів «Чудо острів»

«Goodtoys» – магазин іграшок з великим асортиментом товарів [19]. Інформацію про технології, які використано при розробці не вдалось знайти.

Головна сторінка цього вебзастосунку виглядає простіше за дві попередні розглянуті, однак така мінімалістичність сприяє зручнішій навігації. У верхній частині розміщено пошукову стрічку з підказками, що дозволяє швидко знаходити потрібні товари, а також іконки для входу до особистого кабінету та перегляду кошика. Незважаючи на зручне розташування елементів, на сайті відсутня функція додавання товарів до списку вподобаних, що ускладнює повернення до зацікавлених позицій (рис. 1.6).

Зліва розташовано зручне меню каталогу товарів — покупець може обрати потрібну категорію з представленого списку. Наприклад, при виборі категорії «Набори для творчості», користувач отримує список відповідних товарів із можливістю базової фільтрації за ціною, брендом та віковою категорією (рис. 1.7). У той же час відсутня можливість впорядкувати результати за рейтингом, популярністю чи новизною.



Рис. 1.6 – Головна сторінка сайту «Goodtoys»

Переваги:

- привабливий інтерфейс;
- проста й інтуїтивна навігація;
- великий вибір товарів;
- функціональний особистий кабінет;
- є бонусна система з накопичувальним рахунком;
- наявність зручної системи пошуку з підказками;
- можна залишити відгук та переглянути відгуки інших покупців.

Недоліки:

- немає функції онлайн-консультант;
- немає можливості додати до вподобаних;
- перегляд каталогу можливий лише за категоріями, не можна переглянути всі товари одразу;
- обмежені можливості фільтрації.

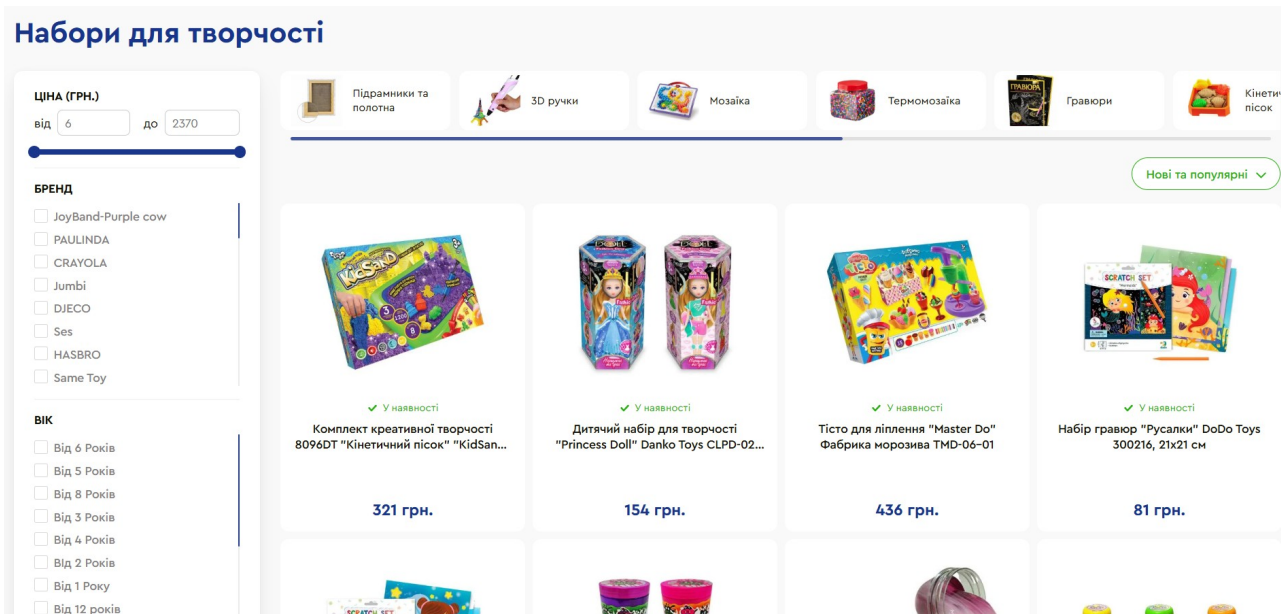


Рис. 1.7 – Каталог товарів «Goodtoys»

Аналіз трьох аналогічних програмних розробок дозволяє зробити висновок, що більшість сучасних інтернет-магазинів дитячих товарів мають схожу структуру: каталог, картки товарів, кошик, оформлення замовлення та акційні пропозиції. Найбільш функціонально насиченими є сайти «Будинок іграшок» та «Чудо острів», які мають особисті кабінети, системи лояльності та зручну фільтрацію товарів. Натомість «Goodtoys» є прикладом спрощеної реалізації з мінімальним функціоналом, що може бути корисним як базовий варіант для старту. Для створення власного сайту доцільно врахувати такі аспекти: забезпечити просту та інтуїтивну навігацію, реалізувати ефективну систему фільтрації, додати розділ акцій та бонусів та передбачити зручне оформлення замовлення з можливістю вибору міста та відділення.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСАЙТУ ДЛЯ ПРОДАЖУ ДИТЯЧИХ ТОВАРІВ

2.1. Постановка задачі, призначення та вимоги до програмного засобу

Головною задачею даного проєкту є розробка вебсайту для продажу дитячих товарів, який забезпечить зручний та безпечний процес вибору, замовлення та оплати товарів для батьків та інших покупців. Потрібно створити ефективний інтерфейс для взаємодії з користувачем, що дозволить швидко знаходити потрібний товар, додавати його в кошик та оформлювати замовлення. Важливими задачами є також забезпечення можливості адміністрування товарів та замовлень для адміністратора магазину, що дозволить здійснювати оперативне оновлення інформації, управління наявністю товарів та моніторинг замовлень.

Призначення даної розробки полягає у створенні вебсайту, який стане зручним інструментом для батьків та інших покупців дитячих товарів. Сайт має забезпечити комфортний процес ознайомлення з асортиментом, вибору та замовлення товарів без необхідності відвідування фізичного магазину.

Функціональні вимоги до програмного засобу:

- каталог товарів з категоріями та реалізованою фільтрацією;
- система пошуку товарів за ключовими словами;
- можливість додавати товари до вподобаних;
- кошик для зберігання вибраних товарів з підрахунком загальної суми та можливістю редагувати кількість або видаляти товари;
- оформлення замовлення;
- адміністративна панель для додавання, редагування та видалення товарів, управління замовленнями.

2.2. Вибір моделі розробки програмного засобу

У межах цього проєкту для розробки вебзастосунку було обрано інкрементну модель життєвого циклу програмного забезпечення. Інкрементальна модель – це методологія, яка передбачає побудову системи невеликими, керованими частинами або інкрементами. Кожен інкремент представляє частину функціональності системи, і після кожної фази надається робоча версія продукту [20]. У розробці вебсайту для продажу дитячих товарів було реалізовано кілька ключових ітерацій, кожна з яких охоплює окрему частину функціоналу.

Першим кроком став етап проєктування структури сайту. На цьому етапі було створено базову архітектуру вебресурсу, включно з головною сторінкою, навігаційним меню, шаблонами для сторінок категорій і товарів. Також було розроблено файлову структуру проєкту, обрано стек технологій (мови програмування, бібліотеки, середовище розробки), а ще налаштовано серверну частину й середовище для тестування застосунку. Основна мета цього етапу – це створити основу, на яку в подальшому будуть нашаровуватись функціональні компоненти.

У межах другої ітерації було реалізовано каталог товарів. Користувач отримує можливість переглядати іграшки. У базу даних було додано приклади товарів із такими атрибутами, як назва, опис, категорія, ціна, вік та фотографія. В інтерфейсі реалізовано фільтрацію товарів за певними параметрами (наприклад, ціна, вік), що спрощує користувачу пошук необхідної продукції. Додатково реалізовано можливість переходу на сторінку окремого товару з детальним описом і зображенням.

На третій ітерації зосереджено увагу на реалізації кошика покупця. Було створено можливість додавати товари до кошика з будь-якої сторінки сайту, переглядати та змінювати їхню кількість та видаляти. Усі дані про вибрані товари зберігаються у сесії. Також реалізується можливість додати товари до

вподобаних, де можна їх переглянути та видалити. Такі можливості дозволяють зручно керувати замовленням, не створюючи додаткових бар'єрів для користувача.

Четверта ітерація була присвячена реалізації оформлення замовлення. Створюється зручна форма, яку користувач заповнює для підтвердження покупки. Форма включає необхідні поля: ПІБ, контактний номер телефону, місто і відділення пошти та коментар до замовлення. Після заповнення форми замовлення зберігається в базі даних, а кількість відповідного товару автоматично зменшується.

Завершальним етапом стала ітерація, пов'язана з візуальним вдосконаленням сайту та тестуванням функціональності. Було перевірено коректність роботи усіх функцій. Виправлено знайдені помилки. Особливу увагу приділено зручності інтерфейсу, читабельності текстів, логіці розміщення елементів, а також загальній швидкодії сайту.

Таким чином, використання інкрементної моделі дозволило розробити повноцінний вебзастосунок, поступово вводячи основні можливості, забезпечити гнучкість процесу розробки та вчасно виявляти й усувати недоліки. Кожен етап був логічно завершеним і пов'язаний із загальним баченням функціоналу сайту, що дозволяє в майбутньому легко розширити його можливості.

2.3. Загальний опис проєкту

Розроблене програмне забезпечення є вебзастосунком для інтернет-магазину дитячих товарів. Архітектура вебсайту побудована за класичною клієнт-серверною моделлю. Взаємодія між клієнтською частиною, сервером і базою даних забезпечує ефективне опрацювання запитів користувачів, підтримку актуальної інформації про товари та замовлення, а також стабільну

роботу системи загалом. Такий підхід дозволяє реалізувати масштабований, надійний та зручний у використанні сервіс для продажу дитячих товарів.

У системі передбачено два основні типи користувачів: звичайний користувач, який виступає покупцем, та адміністратор, який здійснює контроль за вмістом сайту. Діаграма використання сайту, наведена на рисунку 2.1, відображає ключові функціональні можливості, доступні кожному з користувачів.

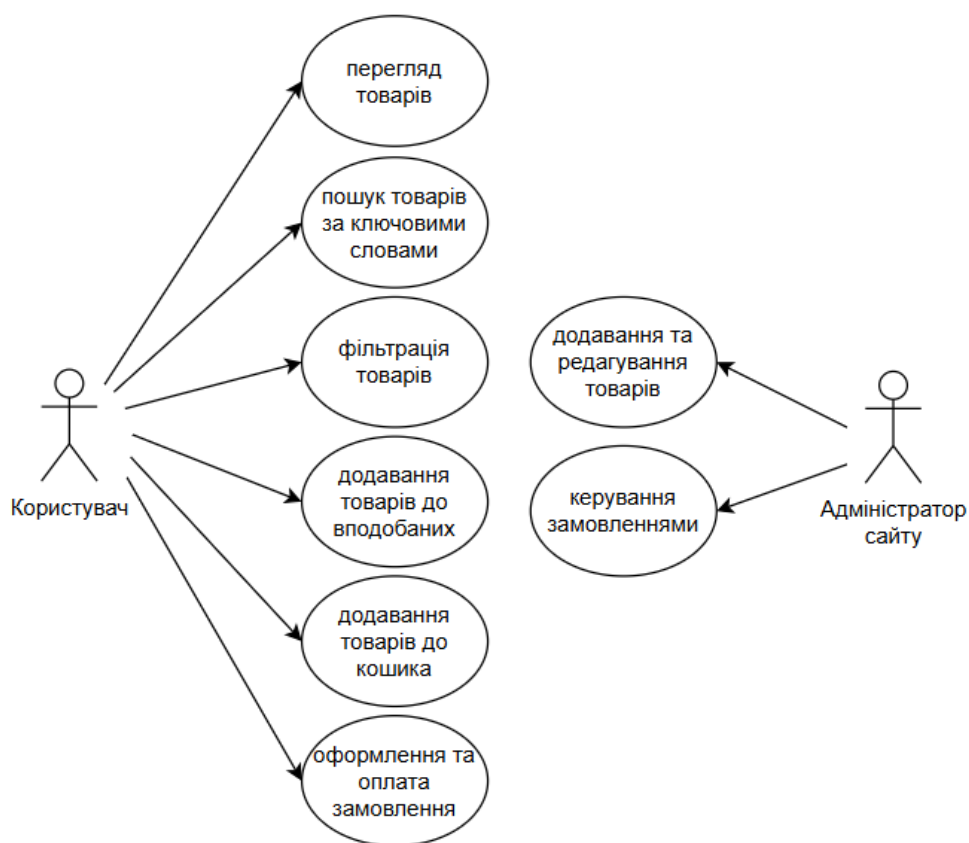


Рис. 2.1 – Діаграма використання сайту

З боку користувача система підтримує перегляд асортименту товарів, пошук за ключовими словами, перегляд розділу зі знижками, додавання товарів до вподобаних та до кошика, перегляд самого кошика та подальше оформлення замовлення. У разі потреби користувач також може очистити кошик або

скасувати замовлення. Адміністратор, у свою чергу, має доступ до адміністративної панелі, де реалізовані функції додавання, редагування й видалення товарів, керування інформацією про знижки, а також перегляд та оновлення статусу замовлень. Загалом, діаграма використання дає змогу побачити логіку взаємодії користувача та адміністратора з сайтом на рівні основних функціональних сценаріїв.

Для більш глибокого розуміння процесу обробки запитів у системі доцільно розглянути діаграму послідовності, наведену на рисунку 2.2. На цій діаграмі послідовно показано, як саме здійснюється обмін даними між користувачем, інтерфейсом, сервером і базою даних у процесі виконання конкретної задачі.

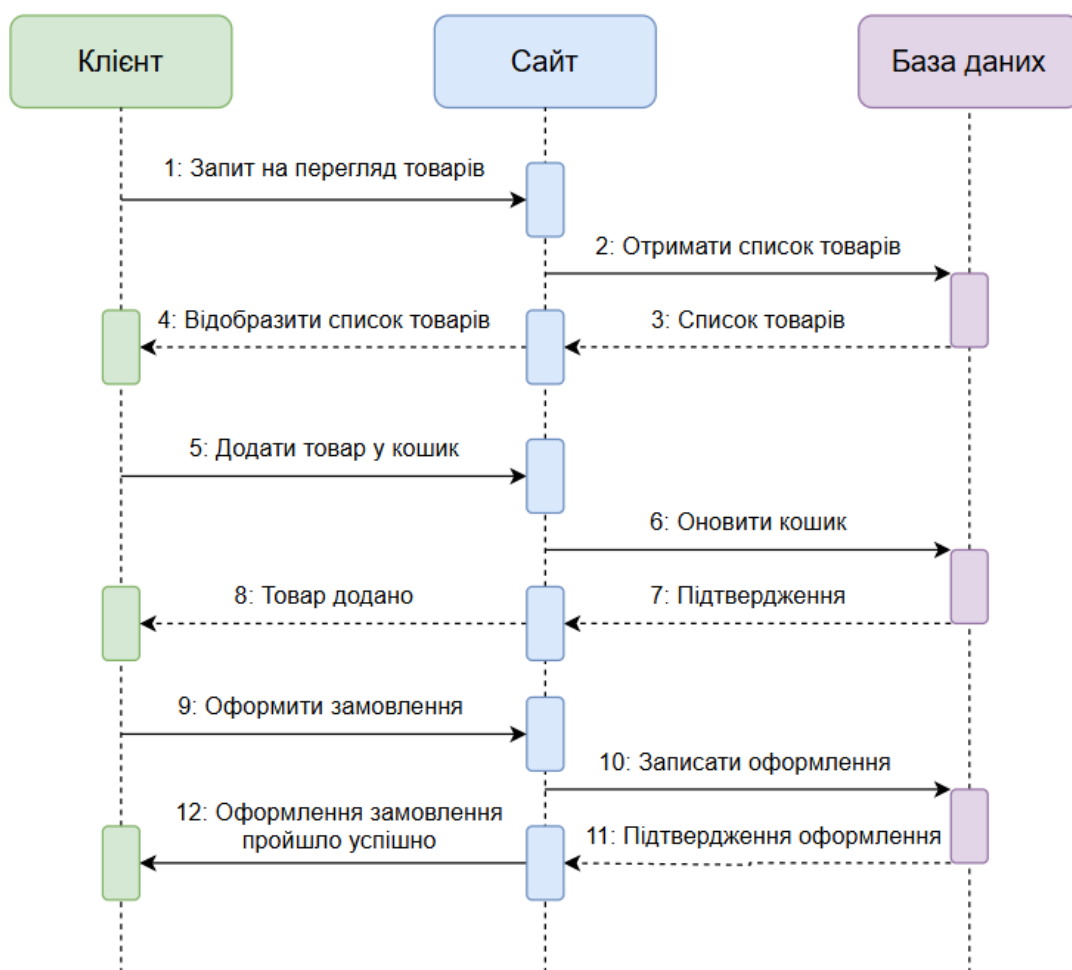


Рис 2.2 – Діаграма послідовності

Взаємодія починається з вибору користувачем відповідного розділу на сайті, після чого клієнтська частина надсилає запит до серверної на отримання списку категорій. Сервер обробляє цей запит, формує відповідний SQL-запит до бази даних SQLite, отримує список категорій і передає його назад клієнту. Після цього користувач обирає одну з категорій, що спричиняє повторний запит до серверної частини, яка звертається до бази даних з фільтром за обраною категорією. Отримані товари виводяться на інтерфейс із відповідною інформацією, включаючи назву, опис, зображення та ціну.

У разі переходу до оформлення замовлення користувач заповнює форму із контактною інформацією, після чого дані надсилаються на сервер для обробки. Сервер формує SQL-запит на додавання нового запису в таблицю замовлень, а також виконує оновлення даних про наявність товарів у таблиці продуктів, зменшуючи відповідну кількість. У разі успішного завершення операції система повертає підтвердження, яке відображається користувачу у вигляді повідомлення про успішне замовлення.

У даному розділі наведено опис архітектури вебзастосунку інтернет-магазину дитячих товарів, який побудовано за класичною клієнт-серверною моделлю. Користувачі взаємодіють із системою через вебінтерфейс, надсилаючи запити до серверної частини, яка обробляє отримані дані, реалізує основну функціональність та управляє бізнес-логікою проєкту. Для збереження та обробки інформації застосовується вбудована база даних SQLite, яка забезпечує надійне і ефективне зберігання даних про товари, користувачів, а також замовлення, що є ключовими елементами системи. Використані діаграми варіантів використання та послідовності дають змогу зрозуміти основні сценарії взаємодії користувача з системою, що сприяє кращому уявленню структури та функціональних можливостей розробленого вебзастосунку. Загальна архітектура забезпечує надійну та зручну роботу як для кінцевих користувачів, так і для адміністраторів системи.

2.4. Обґрунтування вибору інструментальних засобів розробки

2.4.1. Середовище розробки PyCharm

PyCharm – це інтегроване середовище розробки (IDE), яке використовується для програмування на Python [21]. Такий вибір середовища розробки зумовлений широкими можливостями у роботі з Python та Django, зручним інтерфейсом, високою стабільністю та широкою підтримкою сучасних інструментів веброботки.

Основні функції PyCharm [22, 23]:

1. PyCharm аналізує контекст і пропонує інтелектуальне автодоповнення для змінних, функцій, класів, ключових слів, а також імпортів.
2. Середовище автоматично виявляє помилки ще під час написання коду: синтаксичні помилки, помилки типів, неініціалізовані змінні та інші потенційні проблеми.
3. PyCharm має потужний вбудований налагоджувач із можливістю встановлення точок зупину, перегляду значень змінних у режимі реального часу та крокового виконання коду.
4. Підтримує популярні фреймворки тестування (unittest, pytest), дозволяючи запускати, переглядати та аналізувати тести безпосередньо з середовища розробки.
5. Підтримує безпечне автоматизоване перейменування, витягування функцій, інлайнінг та інші види рефакторингу, що допомагає покращити структуру коду без ризику несправності логіки.
6. Дозволяє підключатися до баз даних (зокрема SQLite), переглядати структуру таблиць, виконувати SQL-запити та керувати даними без сторонніх інструментів.

7. Вбудована підтримка HTML, CSS, JavaScript, шаблонів Django тощо дозволяє розробляти як клієнтську, так і серверну частини проєкту в одному середовищі.

2.4.2. Клієнтська частина

Клієнтська частина (фронтенд) вебзастосунку відповідає за відображення інтерфейсу користувача та забезпечення зручної взаємодії з функціональністю сайту. Саме цю частину бачить користувач, коли заходить на сайт. У межах проєкту клієнтська частина реалізована з використанням HTML, CSS та JavaScript.

HTML – це мова гіпертекстової розмітки, яка використовується для створення та структурування контенту в Інтернеті. Вона є основою кожної веб-сторінки, визначаючи, як текст, зображення, відео та інші мультимедійні елементи відображаються в браузері [24].

Основні функції HTML:

- дозволяє визначити заголовки, абзаци, списки, таблиці, зображення, посилання та інші елементи;
- за допомогою тегів контент впорядковується в логічні блоки;
- забезпечує створення форм з полями введення, кнопками, прапорцями тощо, що дозволяє отримувати дані від користувача;
- підтримка вставки зображень, відео, аудіо та інших мультимедійних елементів;
- дозволяє зв'язувати сторінки між собою через гіперпосилання, що є основою навігації у веброботі.

CSS – мова каскадних таблиць стилів (CASS) і використовується для стилізації елементів, написаних мовою розмітки, такою як HTML. Вона відокремлює контент від візуального представлення сайту. HTML є основою контенту сайту, а CSS формує весь візуальний дизайн та естетику [25].

Основні функції CSS:

- дозволяє змінювати вигляд тексту, фону, меж, відступів, розмірів та інших властивостей елементів;
- за допомогою медіазапитів можна змінювати стилі залежно від розміру екрана, пристрою тощо;
- дозволяє задавати точне розміщення за допомогою таких властивостей, як position, flexbox, grid;
- можна додавати динамічні ефекти (анімацію, плавні зміни кольору, розміру, положення тощо);
- стилі можна задавати для класів або ідентифікаторів і використовувати їх багаторазово в різних елементах.

JavaScript – це мова сценаріїв або програмування, яка дозволяє реалізовувати складні функції на веб-сторінках. Щоразу, коли веб-сторінка не просто відображає статичну інформацію, а своєчасно відображає оновлення контенту, інтерактивні карти, анімовану 2D/3D графіку, відео-музикальні автомати з прокруткою тощо, можна сміливо стверджувати, що тут використовується JavaScript [26].

Основні функції JavaScript [27]:

- дозволяє створювати динамічний контент, обробляти події користувача, реагувати на натискання кнопок, рух миші тощо;
- виконується у всіх популярних браузерях без потреби в додаткових інструментах;
- можливість створювати плавні переходи, зміни стилів, рух об'єктів тощо;
- реалізація меню, що розкривається, слайдерів, вкладок, модальних вікон тощо;
- підтримує об'єктно-орієнтовані концепції, включаючи прототипне наслідування;
- Забезпечує асинхронну взаємодію з сервером без перезавантаження сторінки за допомогою технології AJAX.

2.4.3. Серверна частина

Серверна частина (бекенд) відіграє ключову роль у функціонуванні сучасних вебзастосунків. Саме вона відповідає за обробку запитів, а також за логіку роботи застосунку, яка не доступна користувачу напряду через браузер. На відміну від клієнтської частини, яка виконується безпосередньо у веббраузері користувача, серверна частина працює на віддаленому сервері – комп'ютері, який цілодобово обробляє запити та повертає результати. Завдяки серверу можна реалізувати динамічні функції: формування списків товарів з бази даних та надсилання замовлень.

У рамках цього проєкту серверна частина реалізована за допомогою мови програмування Python з використанням фреймворку Django. Цей технологічний стек забезпечує надійну обробку запитів, взаємодію з базою даних, управління логікою роботи сайту, а також підтримку авторизації, форм і сесій. У якості бази даних використано SQLite, яка є вбудованою в Django за замовчуванням і забезпечує простоту використання без потреби окремого серверного програмного забезпечення. Розглянемо детальніше усі технології, які було використано для створення бекенду.

Python – це мова програмування, яка часто використовується для створення вебсайтів і програмного забезпечення, автоматизації завдань і проведення аналізу даних. Зазвичай використовується для розробки вебсайтів та програмного забезпечення, автоматизації завдань, аналізу та візуалізації даних [28].

До основних функціональних переваг Python належить [29]:

- проста та зрозуміла структура коду, що спрощує розробку, тестування й подальший супровід програмного забезпечення;
- можливість застосування різних стилів програмування для зручного вирішення завдань будь-якої складності;

- велика кількість готових бібліотек та модулів, що значно пришвидшує розробку функціоналу;
- кросплатформеність, що дає змогу запускати програми на різних операційних системах без необхідності їх доопрацювання;
- легка інтеграція з іншими мовами програмування та технологіями;
- активна спільнота користувачів і розробників, яка забезпечує стабільний розвиток, доступ до документації та швидке вирішення технічних питань.

Django – це широко відомий веб-фреймворк в екосистемі Python. Він значно спрощує та прискорює розробку веб-сайтів та програм, надаючи програмісту вже готові інструменти. Використовуючи Джанго, ви зможете без зайвих зусиль створювати програми з багаторівневою структурою. На відміну від традиційних HTML редакторів, з Django можна не турбуватися про проблему дублювання коду. Інструмент працює за принципом DRY (Don't Repeat Yourself), що мінімізує повторне використання коду [30].

Функціональні особливості Django [31, 32]:

- чітке розділення логіки програми, структури даних та користувацького інтерфейсу;
- вбудована система маршрутизації запитів;
- готовий інтерфейс адміністративної панелі з підтримкою багатьох мов;
- зручні засоби для роботи з формами, автентифікацією користувачів та обробкою запитів;
- інструменти для тестування та налагодження коду;
- підтримка масштабування;
- вбудована документація;
- зручний шаблонізатор для створення динамічних HTML-сторінок.

SQLite – це компактна вбудована реляційна база даних з відкритим кодом. SQLite не є окремим сервером баз даних, як, наприклад, MySQL або PostgreSQL [33].

У рамках цієї роботи використано SQLite як основну систему управління базами даних, оскільки вона інтегрована у фреймворк Django за замовчуванням. Це дозволяє зосередитися на розробці логіки вебзастосунку без потреби в додатковій конфігурації серверної частини бази даних.

Основні переваги використання SQLite [34, 35]:

- підтримка стандарту SQL, що забезпечує зручну роботу з запитами та сумісність із іншими СУБД;
- вбудована в Django, що дозволяє одразу почати роботу без встановлення додаткових компонентів;
- не потребує запуску окремого серверу, вся база працює локально в межах застосунку;
- база зберігається у вигляді одного файлу, що спрощує перенесення, копіювання та резервне збереження;
- швидка робота при обробці невеликих обсягів даних;
- низькі системні вимоги, підходить для проєктів з обмеженими ресурсами;
- забезпечує надійність і стабільність зберігання даних;
- дозволяє працювати із транзакціями, що важливо для забезпечення цілісності даних;
- не потребує конфігурації доступів, портів, авторизації чи мережових з'єднань.

2.5. Особливості програмної реалізації та основні режими роботи

На початковому етапі було створено проєкт в Django з назвою `website`, який виконує роль головного контейнера для всієї логіки застосунку. У межах цього проєкту також було створено додаток `main`, що містить основний функціонал магазину. Django чітко структурує проєкт, поділяючи його на окремі компоненти, кожен із яких виконує визначену роль (рис. 2.3).

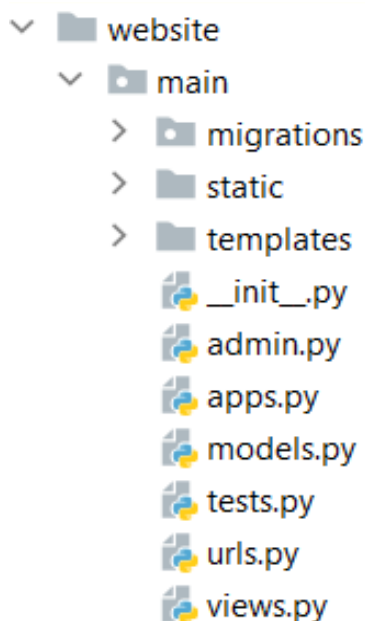


Рис. 2.3 – Структура проєкту

Короткий опис основних файлів і директорій та їхнє призначення:

- `models.py` – файл, у якому описуються модулі даних, у рамках проєкту тут створено класи, що відповідають таблицям у базі даних;
- `admin.py` – файл у якому реєструються модулі для керування в адміністративній панелі;
- `urls.py` – файл, що відповідає з маршрутизацію, тут задаються шляхи, за якими користувач може переходити на різні сторінки сайту;
- `views.py` – файл, у якому реалізовано логіку обробки запитів користувачів, у ньому визначено функції та класи, які відповідають за відображення сторінок сайту, обробку форм, роботу з кошиком тощо;
- `apps.py` – містить конфігурацію додатка, включно з його ім'ям та іншими параметрами, які використовує Django при запуску;
- `tests.py` – призначений для написання модульних тестів, що дозволяють перевірити коректність роботи окремих частин застосунку;
- `templates/` – директорія з HTML-шаблонами, що визначають вигляд інтерфейсу користувача;

- static/ – містить статичні файли (стили CSS, скрипти JavaScript, зображення тощо).

У рамках додатку main було створено кілька моделей, які реалізують основну логіку сайту з продажу дитячих товарів. Усі моделі зосереджені у файлі models.py. Кожна з них відповідає певному об'єкту предметної області. Product – модель, що зберігає інформацію про товари, представлені на сайті. Містить такі поля: name (назва товару), description (опис), price (ціна), quantity (доступна кількість), image (головне зображення товару), is_discounted (булеве поле, що вказує на наявність знижки. На рисунку 2.4 зображено, який вигляд має таблиця товарів на адміністративній панелі.

Виберіть toy щоб змінити

Дія: Вперед 0 з 48 обрано

НАЗВА	КАТЕГОРІЯ	ВІК ДИТИНИ	ЦІНА	НАЯВНІСТЬ
☐ Багаторазові водні розмальовки від Avenir Морські тваринки	Розмальовки	3+	450,00	14
☐ Багаторазові водні розмальовки від Avenir Динозаври	Розмальовки	3+	450,00	12
☐ Водний пістолет з рюкзаком Восьминіг	Ігри для активного відпочинку	3+	990,00	12
☐ Водний пістолет з рюкзаком Краб	Ігри для активного відпочинку	3+	990,00	10
☐ Стікорова мозаїка «Тваринки»	Стікербуки	2+	370,00	20
☐ Художній набір від Tookyland з рулонною розмальовкою	Набори для малювання	2+	1100,00	23
☐ Набір вітражного розпису «Світ принцес»	Набори для малювання	3+	710,00	19
☐ Набір для малювання на воді	Набори для малювання	3+	850,00	17
☐ Творчий набір з помпонами	Творчі набори	3+	590,00	26
☐ Творчий набір з помпонами	Творчі набори	3+	440,00	28
☐ Розмальовки із сухою палітрою Динозаври	Розмальовки	3+	450,00	18
☐ Розмальовки із сухою палітрою Сад	Розмальовки	3+	450,00	25
☐ Пальчикові фарби	Фарби Крайони	1+	460,00	6
☐ Наліпки на нігті від Tookyland	Наліпки	2+	370,00	20

ВІДФІЛЬТРУВАТИ

За Категорія

Всі

Творчі набори

Набори для малювання

Розмальовки

Фарби | Крайони

Наліпки

Стікербуки

Ігри для активного відпочинку

За Вік дитини

Всі

1+

2+

3+

4+

Рис. 2.4 – Таблиця товарів (Toys)

ToyPhoto – модель для збереження додаткових фотографій товарів. Вона пов'язана з моделлю Product за допомогою зовнішнього ключа (ForeignKey) і дозволяє прикріплювати кілька фото до одного товару. Містить такі поля: product (зв'язок з товаром) та photo (файл зображення). Вигляд таблиці на

рисунку 2.5. Завдяки цій моделі можна відображати галерею товарів, що підвищує візуальну привабливість сайту.

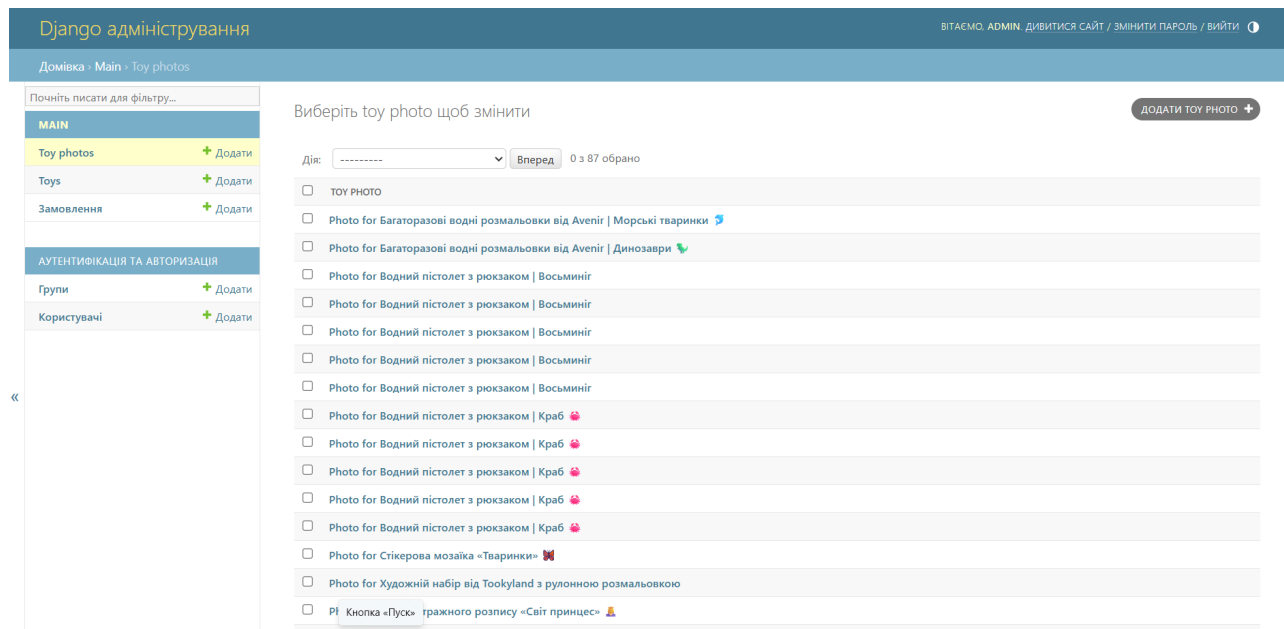


Рис. 2.5 – Таблиця додаткових фото товарів (Toy photos)

Одним із ключових представлень (views), реалізованих у додатку, є функція `toy_catalog`, яка відповідає за виведення списку іграшок із можливістю фільтрації, пошуку, сортування та пагінації. Це представлення використовується для формування каталогу товарів на сайті, враховуючи параметри, які надсилає користувач через GET-запит.

Розглянемо реалізацію пошуку товарів за ключовими словами (рис. 2.6). На початку у функції перевіряється наявність пошукового запиту (`search`) у параметрах GET-запиту. Якщо запит є, система виконує попередню обробку тексту: видаляє зайві символи, пробіли та приводить текст до нижнього регістру. Далі запит розбивається на окремі слова, і для кожного з них формується умова пошуку в полі назви (`name`) або опису (`description`) товару. Пошук нечутливий до регістру, що робить його зручним для користувача. Усі товари, що відповідають запиту відображаються у результатах.

```

# Пошук за назвою або описом
if search_query:
    # Видаляємо спеціальні символи (наприклад, "|") і зайві пробіли
    search_query = re.sub(r'[\W\s]', '',
                        search_query)
    search_query = ' '.join(search_query.split())
    # Перетворюємо рядок у нижній регістр для нечутливого до регістру порівняння
    search_query_lower = search_query.lower()

    # Розбиваємо пошуковий запит на окремі слова
    query_words = search_query_lower.split()
    query = Q() # Створюємо порожній об'єкт
    for word in query_words:
        # Створюємо умови пошуку для кожного слова в назві та описі товару
        query |= Q(name__icontains=word) | Q(
            # Використовуємо 'icontains' для нечутливого до регістру пошуку
            description__icontains=word)
    # Фільтруємо список іграшок за складеним запитом
    toys_list = toys_list.filter(query)

```

Рис. 2.6 – Реалізація пошуку за ключовими словами

Для прикладу введемо у пошукову стрічку слово «фарби» (рис. 2.7). Система виконує пошук і виведе всі наявні товари, у назві або описі яких згадується цей термін. Це дає змогу швидко знайти потрібний продукт без необхідності переглядати увесь каталог вручну.

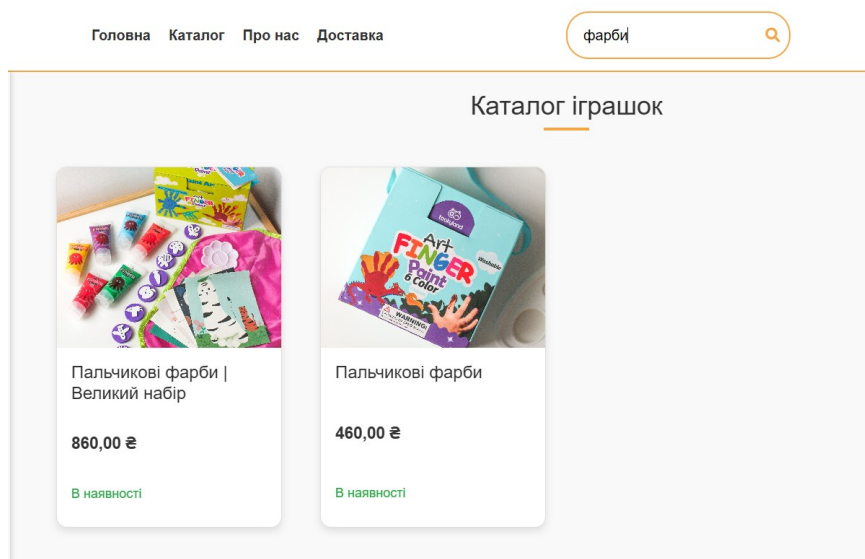


Рис. 2.7 – Приклад пошуку товарів за ключовими словами

Далі у функції `toy_catalog` реалізується фільтрація за категоріями та акційними товарами (рис. 2.8). Якщо користувач обрав одну або кілька категорій, товари відфільтровуються згідно з ними (через поле `category`). Окрім того, якщо у запиті присутній параметр `on_sale`, то відображаються лише ті товари, що мають встановлену знижку, за допомогою фільтрації по булевому полю `is_on_sale`.

```
# Фільтрація по категоріях
if categories:
    toys_list = toys_list.filter(category__in=categories)

# Фільтрація акційних товарів
if on_sale:
    toys_list = toys_list.filter(is_on_sale=True)
```

Рис. 2.8 – Реалізація фільтрації товарів за категоріями та акціями

Наступним етапом є фільтрація за віком (рис. 2.9). Користувач може вибрати одну або кілька вікових груп (наприклад, «2+», «3+»).

```
# Фільтрація по вікових групах
if age_groups:
    # Створюємо список для всіх вікових груп, які потрібно включити
    selected_ages = []
    for age in age_groups:
        if age == '4+':
            selected_ages.extend(['1+', '2+', '3+', '4+'])
        elif age == '3+':
            selected_ages.extend(['1+', '2+', '3+'])
        elif age == '2+':
            selected_ages.extend(['1+', '2+'])
        elif age == '1+':
            selected_ages.append('1+')

    # Видаляємо дублікати (якщо такі є)
    selected_ages = list(set(selected_ages))

    # Фільтруємо товари за віковими групами
    toys_list = toys_list.filter(age_group__in=selected_ages)
```

Рис. 2.9 – Реалізація фільтрації товарів за віком

Для кожної вибраної групи формуються відповідні варіанти фільтрації. Наприклад, якщо обране «3+», то враховуються також товари рекомендовані для віку «1+» і «2+», оскільки вони підходять для дітей віком від одного року і старше. Це реалізовано шляхом динамічного формування списку відповідних груп і фільтрації по полю `age_group`.

Для більш точного підбору товарів реалізовано також фільтрацію за ціною (рис. 2.10). З запиту зчитуються значення мінімальної та максимальної ціни, після чого товари фільтруються по діапазону `price_gte` та `price_lte`. Якщо користувач не вказав значення, використовуються значення за замовчуванням. Крім того, реалізовано сортування товарів за ціною. Якщо обрано відповідний параметр (`sort_by_price`), товари можуть бути впорядковані у зростаючому (`asc`) або спадному (`desc`) порядку за вартістю.

```
# Фільтрація по ціні
min_price = request.GET.get('min_price', 0)
max_price = request.GET.get('max_price', 10000)

try:
    min_price = float(min_price)
    max_price = float(max_price)
except ValueError:
    min_price = 0
    max_price = 10000

toys_list = toys_list.filter(price__gte=min_price, price__lte=max_price)

# Сортування за ціною
if sort_by_price:
    if sort_by_price == 'asc':
        toys_list = toys_list.order_by('price') # Від найменшої до найбільшої
    elif sort_by_price == 'desc':
        toys_list = toys_list.order_by('-price') # Від найбільшої до найменшої
```

Рис. 2.10 – Реалізація фільтрації та сортування за ціною

Візьмемо для прикладу наступні параметри фільтрації: категорія – «Творчі набори», вікова група – «3+», сортування за ціною – від 0 до 600. У результаті система відобразила лише ті товари, які відповідають заданим умовам, що значно спрощує процес вибору та дозволяє користувачу швидко знайти саме те, що його цікавить (рис. 2.11).

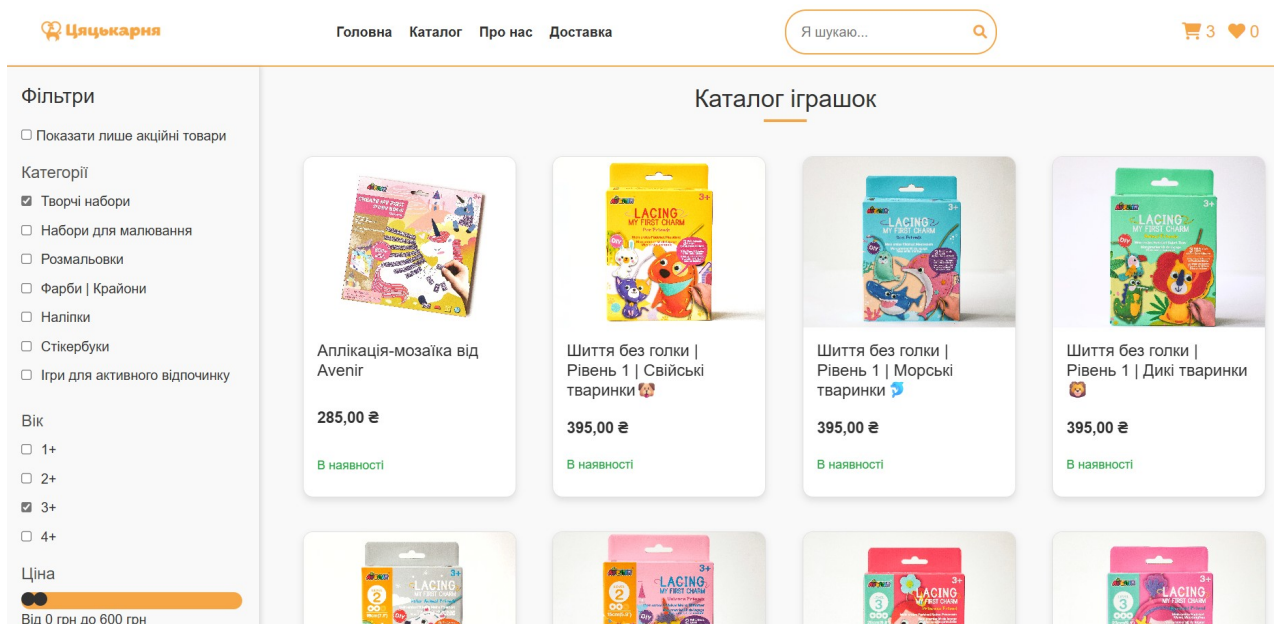


Рис. 2.11 – Приклад фільтрації товарів

Для відображення детальної інформації про окремий товар реалізовано функцію `toy_detail`, яка обробляє відповідний запит користувача (рис. 2.12).

```
def toy_detail(request, toy_id):
    toy = get_object_or_404(Toy, id=toy_id) # Отримуємо товар за ID
    context = {
        'toy': toy,
    }
    return render(request, 'toys/card.html', context)
```

Рис. 2.12 – Реалізація детальної інформації про певний товар

Вона викликається коли користувач натискає на певну картку товару зі сторінки каталогу або головної сторінки. За допомогою `get_object_or_404()` з бази даних отримується об'єкт моделі `Toy` за його унікальним ідентифікатором `toy_id`. Якщо такого товару не існує, користувачу покаже сторінку з помилкою.

Знайдений об'єкт передається в шаблон `card.html` через словник `context`, щоб відобразити всю необхідну інформацію (назву, фото, опис, вік, ціну, наявність, знижки тощо). Цей підхід дозволяє ефективно реалізувати сторінку з докладною інформацією про товар та забезпечує зручний і логічний перехід від загального списку товарів до конкретного продукту (рис. 2.12).

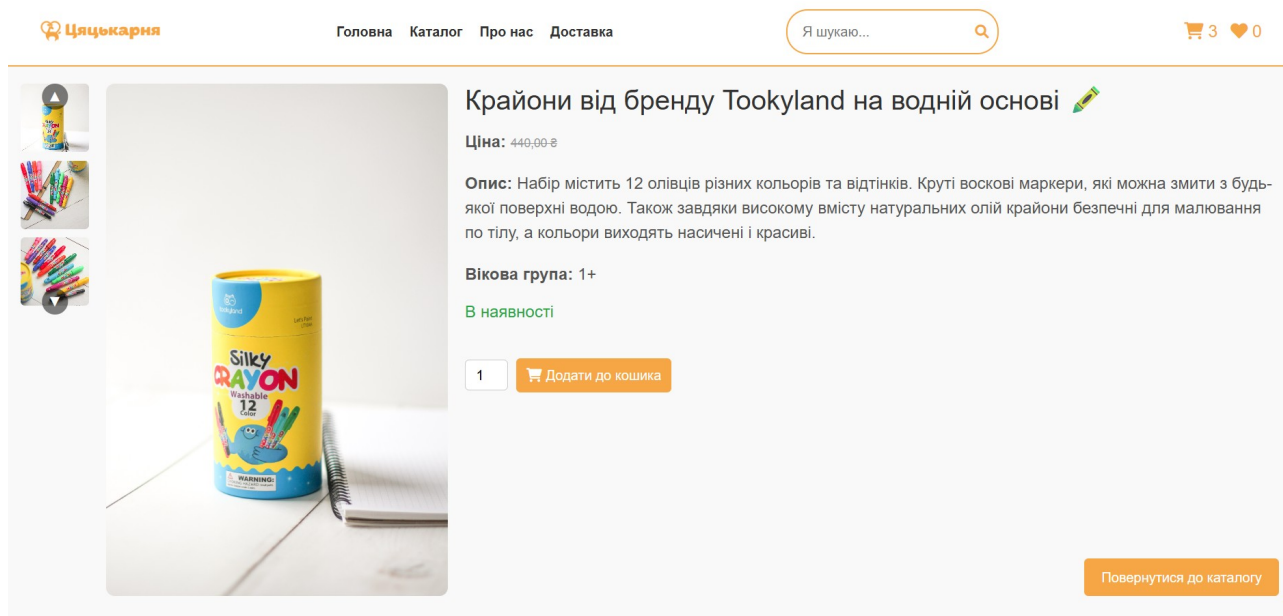


Рис. 2.12 – Відображення конкретного товару на сторінці

Оформлення замовлення реалізовано у вигляді зручної вебформи, що складається з двох частин: зліва — блок для введення контактних даних і параметрів доставки, справа — детальна інформація про вміст замовлення (рис. 2.13). Користувач заповнює обов'язкові поля, зокрема ім'я, прізвище, номер телефону, електронну адресу, місто та вибирає відділення Нової Пошти. За бажанням можна залишити коментар до замовлення. Праворуч відображаються

товари, які були додані до кошика, із зазначенням назви, кількості, ціни (з урахуванням знижок) та суми за кожну позицію. Нижче автоматично підраховується загальна кількість товарів, сума замовлення та виводиться інформація про доставку. Після натискання кнопки «Підтвердити замовлення» усі введені дані надсилаються на сервер, де обробляються та зберігаються у відповідних таблицях бази даних. Зокрема, зберігаються контактні дані клієнта, інформація про обраний спосіб доставки, а також всі товари, що були додані у кошик на момент оформлення. Таким чином, створюється повний запис замовлення, який надалі може бути переглянутий, опрацьований або змінений адміністратором через внутрішню адмін-панель.

Оформлення замовлення

Контактні дані

Ім'я *

Прізвище *

Телефон *

Email *

Доставка

Місто *

Відділення Нової Пошти *

Коментар до замовлення

Підтвердити замовлення

Ваше замовлення



Ігровий набір «Чарівний пісок»
~~4750,00 ₴~~ **1575,00 ₴**
 Кількість: 1
 Сума: 1575.00 ₴



Аплікація-мозаїка від Avenir
 285,00 ₴
 Кількість: 1
 Сума: 285.00 ₴



Шиття без голки | Рівень 1 | Морські тваринки
 395,00 ₴
 Кількість: 1
 Сума: 395.00 ₴

Всього товарів:	3
Сума:	2255.00 ₴
Доставка:	За тарифами перевізника
До сплати:	2255.00 ₴

Рис. 2.13 – Оформлення замовлення

2.6. Організація тестування та налагодження програмного засобу

У процесі розробки сайту для продажу дитячих товарів було проведено комплексне тестування функціональних можливостей моделей товарів і замовлень, а також інтерфейсу каталогу. Це забезпечило коректну роботу основних користувацьких сценаріїв, включаючи створення товарів, фільтрацію, оформлення замовлень та обробку знижок. Для реалізації автоматизованого тестування було використано вбудований в Django фреймворк unittest за допомогою класу `django.test.TestCase`, який дозволяє створювати тести для перевірки логіки та взаємодії з базою даних.

У рамках тестування моделі товару перевірялася можливість створення об'єкта з усіма необхідними полями, включаючи назву, опис, категорію, ціну, кількість на складі, вікову категорію, а також інформацію про знижки. Результат цього тесту підтверджує, що об'єкт товару створюється коректно, і його властивості зберігаються відповідно до заданих значень (рис. В.1). Окрему увагу приділено перевірці відображення остаточної ціни (`final_price`), яка залежить від наявності акційної ціни. Якщо товар є акційним і має вказану знижку, система правильно використовує значення `sale_price`. Якщо ж товар не має знижки або акційна ціна відсутня, застосовується звичайна ціна (рис. В.2). Також, протестовано модель фотографій товару. Було перевірено, що зображення правильно асоціюється з відповідним товаром, і зберігається файл із вказаною назвою (рис. В.3).

Інтерфейс каталогу товарів проходив тестування на коректну роботу фільтрації. Перевірено функціональність відбору за категорією, де результати відповідають вибраній категорії і не включають товари інших типів (рис. В.4). Аналогічно перевірено фільтрацію за віковими групами, що дозволяє користувачу бачити лише ті товари, які підходять для обраної вікової категорії (рис. В.5). Тестування фільтрації акційних позицій підтвердило, що користувач може відобразити лише товари, які мають знижки (рис. В.6). Було також

перевірено коректну роботу фільтра за діапазоном цін (рис. В.7), а також сортування за ціною у зростаючому порядку.

Окремим етапом стало тестування моделі замовлення та її взаємозв'язку з моделлю товарних позицій у замовленні. Перевірено створення об'єкта замовлення з усіма основними полями, включаючи контактну інформацію, місто, відділення доставки, коментар, статус і загальну суму (рис. В.8). Далі протестовано можливість додавання позицій до замовлення, включаючи кількість одиниць та ціну кожного товару (рис. В.9). У результаті тесту підтверджено правильну роботу зв'язку між замовленням та його позиціями. Також було перевірено, що статус замовлення відповідає одному з дозволених варіантів, зазначених у моделі (рис. В.10).

Отже, проведене тестування підтвердило стабільність логіки роботи з товарами, каталогом та оформленням замовлень. Усі ключові аспекти, включаючи обробку знижок, фільтрацію й взаємодію з базою даних, функціонують відповідно до вимог, що свідчить про надійність реалізованої системи.

2.7. Рекомендації по використанню та впровадженню програмного засобу

Розроблений вебзастосунок призначений для автоматизації процесів інтернет-магазину дитячих товарів, забезпечуючи зручний інтерфейс для користувачів. Практичне використання програмного продукту дозволяє підвищити ефективність управління асортиментом, оптимізувати процес оформлення замовлень та покращити якість обслуговування клієнтів.

Для коректної роботи системи необхідно забезпечити стабільне підключення до мережі Інтернет. Серверне обладнання має відповідати мінімальним технічним характеристикам: мати не менше 4 ГБ оперативної пам'яті, процесор із середнім рівнем продуктивності, а також достатній обсяг

вільного дискового простору для зберігання бази даних і файлів застосунку. Клієнтська частина системи сумісна з сучасними веб-браузерами, що підтримують стандарти HTML5, CSS3 та JavaScript, що забезпечує коректне відображення інтерфейсу на різних пристроях, включно з персональними комп'ютерами, планшетами та смартфонами. Для зручної роботи рекомендується використовувати пристрої з роздільною здатністю екрану не нижче 1024×768 пікселів.

Впровадження програмного засобу можливо як на локальному сервері підприємства, так і на хмарних платформах, що підтримують розгортання вебзастосунків і управління базами даних. Архітектура системи передбачає можливість масштабування та інтеграції з іншими інформаційними системами підприємства, що сприяє її адаптації до потреб бізнесу.

ВИСНОВОК

У результаті виконання бакалаврської роботи було спроектовано та реалізовано вебсайт для продажу дитячих товарів, що забезпечує базову функціональність для взаємодії з користувачем. Сайт дозволяє переглядати каталог товарів, здійснювати фільтрацію за категорією, віковою групою, ціною та акційними пропозиціями, а також створювати замовлення з вибором кількості товарів і введенням контактної інформації. Реалізовано адміністративну частину, що дає змогу додавати та редагувати товари, прикріплювати фото, керувати наявністю та знижками. Проведене тестування підтвердило коректну роботу основних функцій сайту, зокрема моделі товару, замовлення, логіки формування остаточної ціни, фільтрація та збереження інформації про замовлення. Попри позитивні результати, проєкт має певні обмеження. Зокрема, на поточному етапі реалізації відсутня можливість онлайн-оплати замовлень. У подальшому доцільно реалізувати інтеграцію з платіжними сервісами, що значно покращить зручність користувачів і дозволить автоматизувати процес обробки замовлень. Крім того, в майбутньому варто впровадити особистий кабінет покупця з можливістю перегляду історії замовлень, редагування даних і відстеження статусу замовлення. Поставлені завдання було виконано в повному обсязі, а розроблений вебсайт є якісною основою для подальшого розвитку функціоналу й вдосконалення користувацького досвіду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 5 Key Retail Tech Trends Shaping Future E-commerce Growth. *Number Analytics - AI Statistical Software*. URL: <https://www.numberanalytics.com/blog/key-retail-tech-trends-shaping-future-ecommerce-growth> (дата звернення: 06.05.2025).
2. eCommerce Technology: Top 10 Trends Shaping 2025's Retail Landscape. *Digital Silk*. URL: <https://www.digitalsilk.com/digital-trends/ecommerce-technology/> (дата звернення: 06.05.2025).
3. The Future Of E-Commerce Technology. *Forbes*. URL: <https://www.forbes.com/councils/forbesagencycouncil/2023/08/25/the-future-of-e-commerce-technology/> (дата звернення: 06.05.2025).
4. HTML - Hypertext Markup Language. *Freelancer*. URL: <https://www.freelancer.com.ua/what-is-html> (дата звернення: 08.05.2025).
5. W3Schools.com. *W3Schools Online Web Tutorials*. URL: https://www.w3schools.com/css/css_intro.asp (дата звернення: 08.05.2025).
6. What Is JavaScript: A Beginner's Guide to the Basics of JS. *Hostinger Tutorials*. URL: <https://www.hostinger.com/tutorials/what-is-javascript> (дата звернення: 08.05.2025).
7. Statistics and data. Most Popular Backend Frameworks 2011/2025, 2025. *YouTube*. URL: <https://www.youtube.com/watch?v=-WrRlgj9IUg> (дата звернення: 08.05.2025).
8. The 15 Most Popular Websites Using Django | SECL Group. *SECL Group*. URL: <https://seclgroup.com/15-websites-and-web-apps-built-with-django/> (дата звернення: 08.05.2025).
9. Top 6 Advantages and Disadvantages of Laravel for web Development. *Digital Applications - Software Company - Software House in Pakistan*. URL: <https://dgaps.com/advantages-disadvantages-of-laravel-519> (дата звернення: 09.05.2025).

10. Laravel framework - Advantages and Disadvantages explained. *phpspider*. URL: <https://www.phpspiderblog.com/the-advantages-and-disadvantages-of-laravel-a-comprehensive-analysis/> (дата звернення: 09.05.2025).

11. Top Laravel Companies 2025 | Built In. *Built In | Tech Jobs Personalized*. URL: <https://builtin.com/companies/tech/laravel-companies> (дата звернення: 09.05.2025).

12. Advantages and disadvantages of NestJS. *JavaScript Custom Software Development Company*. URL: <https://themobilereality.com/blog/advantages-and-disadvantages-of-nestjs> (дата звернення: 10.05.2025).

13. Beighton B. NestJS the Pros and Cons. *Medium*. URL: <https://bradbeighton.medium.com/nestjs-the-pros-and-cons-aff714607b07> (дата звернення: 10.05.2025).

14. Documentation | NestJS - A progressive Node.js framework. *Documentation | NestJS - A progressive Node.js framework*. URL: <https://docs.nestjs.com/discover/companies> (дата звернення: 10.05.2025).

15. Best E-commerce Website Features in 2025. Part 1: Client Side. *SECL Group*. URL: <https://seclgroup.com/e-commerce-website-features-part-1/> (дата звернення: 10.05.2025).

16. Must-Have Features for a Successful Online Shopping Website - Building Your Website — Strikingly. *How To Make a Website – Free Website Builder | Strikingly*. URL: <https://www.strikingly.com/content/blog/must-have-features-for-online-shopping-website.amp/> (дата звернення: 10.05.2025).

17. Дитячі іграшки Будинок Іграшок - купити іграшку для дитини в Києві, Україні на Ві.юа. *Дитячі іграшки Будинок Іграшок - купити іграшку для дитини в Києві, Україні на Ві.юа*. URL: <https://bi.ua/ukr/> (дата звернення: 11.05.2025).

18. Інтернет-магазин іграшок Чудо Острів: широкий асортимент товарів для дітей від 0 до 16 років. *Інтернет-магазин іграшок Чудо Острів: широкий асортимент товарів для дітей від 0 до 16 років*. URL: <https://chudo-ostriv.com.ua/>

srsltid=AfmBOopb_U0obaQkg8avdZl5O_vbEHa6oia3o8buXQRpMlcT572k9PBP
(дата звернення: 11.05.2025).

19. Goodtoys | Інтернет-магазин дитячих іграшок. *Goodtoys | Інтернет-магазин дитячих іграшок.* URL: <https://goodtoys.com.ua/?srsltid=AfmBOooyVzjicsLSTkEJCYdbWfW4He1NlfSjUtQXoJfAWbBnIXQ7F8bW>
(дата звернення: 11.05.2025).

20. Інкрементальна модель життєвого циклу розробки ПЗ: ключові переваги та етапи. *Custom Web & Mobile Development Company - New Line Technologies.* URL: <https://newline.tech/inkrementalna-model-zhittyevogo-ciklu-rozrobki-programnogo-zabezpechennya-uk/> (дата звернення: 12.05.2025).

21. GeeksforGeeks. What is PyCharm? - GeeksforGeeks. *GeeksforGeeks.* URL: <https://www.geeksforgeeks.org/what-is-pycharm/> (дата звернення: 13.05.2025).

22. JetBrains. PyCharm Features - JetBrains Python IDE. *JetBrains.* URL: <https://www.jetbrains.com/pycharm/features/> (дата звернення: 13.05.2025).

23. Pycharm це IDE від JetBrains для програмування на Python. *FoxmindEd.* URL: <https://foxminded.ua/pycharm-tse/> (дата звернення: 13.05.2025).

24. Gaur B. What is HTML: Features, Importance, and Key Elements. URL: <https://www.digitalregenesys.com/blog/what-is-html> (дата звернення: 13.05.2025).

25. What is CSS and how does it work?. *Hostinger Tutorials.* URL: <https://www.hostinger.com/tutorials/what-is-css> (дата звернення: 13.05.2025).

26. What is JavaScript? - Learn web development | MDN. *MDN Web Docs.* URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript (дата звернення: 13.05.2025).

27. Javascript Features — Studytonight. *Studytonight - Best place to Learn Coding Online.* URL: <https://www.studytonight.com/javascript/javascript-features>
(дата звернення: 13.05.2025).

28. What Is Python Used For? A Beginner Guide. *Coursera*. URL: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> (дата звернення: 18.05.2025).

29. Що таке Python і для чого він потрібен. *GoIT*. URL: <https://goit.global/ua/articles/shcho-take-python-i-dlia-choho-vin-potriben/> (дата звернення: 18.05.2025).

30. Що таке Django і як його застосовувати в Python. *Highload.tech - media для розробників*. URL: <https://highload.tech/uk/shho-take-django-i-yak-z-nym-pratsyuvaty-v-python/> (дата звернення: 18.05.2025).

31. Створення сайту на Python Django | Wezom. *IT-компанія повного циклу розробки програмних продуктів WEZOM - Київ, Україна*. URL: <https://wezom.com.ua/ua/blog/razrabotka-sajtov-na-python-django> (дата звернення: 18.05.2025).

32. 11 Top Django Python Features to Accelerate your Web App Development Workflow. *GoodFirms*. URL: <https://www.goodfirms.co/blog/django-python-advantages> (дата звернення: 18.05.2025).

33. FREEhost.UA. *Хостинг в Україні – FreeHost*. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-sqlite/> (дата звернення: 18.05.2025).

34. SQLite. *Aspose.GIS*. URL: <https://products.aspose.com/gis/uk/net/gis-formats/sqlite/> (дата звернення: 18.05.2025).

35. Features Of SQLite. *SQLite Home Page*. URL: <https://www.sqlite.org/features.html> (дата звернення: 18.05.2025).

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

ВСТУП

Даний проєкт передбачає розробку вебсайту для продажу дитячих товарів. Програмний продукт дозволить користувачам зручно переглядати каталог товарів та оформлювати замовлення. Метою розробки є створення сучасної платформи, що забезпечить ефективний продаж дитячих товарів та зручний досвід для користувачів.

ПІДСТАВИ ДЛЯ РОЗРОБКИ

Робота з розробки вебсайту для продажу дитячих товарів здійснюється на підставі написання бакалаврської роботи.

ПРИЗНАЧЕННЯ РОЗРОБКИ

Програмний продукт призначений для забезпечення зручного та ефективного процесу онлайн-покупок. Основні функції включають:

- перегляд каталогу;
- пошук товарів;
- фільтрація товарів;
- додавання товарів до кошика;
- оформлення замовлення.

Вебсайт розробляється для використання в середовищі електронної комерції.

ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Програмний продукт повинен забезпечувати виконання наступних функцій:

- можливість перегляду товарів та їх фільтрація за категоріями, ціною та віком;
- швидкий пошук товарів за назвою або ключовими словами;
- додавання товарів до вподобаних;

- додавання, редагування та видалення товарів з кошика;
- оформлення замовлення;
- адміністративна панель з можливістю управління товарами та замовленнями.

Вимоги до надійності:

- стабільна робота у різних браузерах;
- коректне завантаження та відображення контенту;
- мінімізація часу відновлення після збоїв;
- можливість швидкого внесення виправлень у код.

Для забезпечення стабільної роботи програмного продукту необхідно мати стабільне інтернет-з'єднання та сучасні браузери, такі як Chrome, Firefox, Safari та Edge.

Серверне обладнання повинно включати процесор з 4 ядрами та частотою 2.5 ГГц або вище, оперативну пам'ять 8 ГБ або більше, а також дисковий простір не менше 100 ГБ.

Програмний продукт повинен забезпечувати сумісність із сучасними стандартами та технологіями. Для розробки використовуються такі інструменти: HTML і CSS для створення інтерфейсу, JavaScript для динамічної взаємодії з користувачем, Django для бекенду (обробка запитів) та SQLite для зберігання даних товарів. Вхідні дані включають інформацію про товари та дані користувачів. Вихідні дані – це HTML-сторінки для відображення інтерфейсу.

ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документів входить текст бакалаврської роботи та технічне завдання.

ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ

Розробка вебсайту дозволить скоротити витрати на обслуговування клієнтів за рахунок автоматизації процесів, таких як перегляд товарів та оформлення замовлень. Оперативна обробка запитів через зручний інтерфейс зменшить навантаження на персонал, що дозволить зменшити витрати на його

утримання. Крім того, завдяки швидкому доступу до інформації про товари та можливості самостійного оформлення замовлень, знижується ризик втрати клієнтів. Це робить вебсайт економічно вигідним для бізнесу, з можливістю швидкої окупності та масштабування.

СТАДІЇ І ЕТАПИ РОЗРОБКИ

1. Аналіз вимог та визначення цілей проєкту.
2. Проєктування архітектури системи.
3. Розробка фронтенду та бекенду.
4. Реалізація функціональності, а саме каталог товарів з можливістю фільтрації, кошик для покупок, вподобані та пошук за ключовими словами.
5. Тестування, виявлення та усунення помилок.
6. Введення сайту в експлуатацію.

ПОРЯДОК КОНТРОЛЮ Й ПРИЙМАННЯ

Приймання результатів роботи здійснюється в процесі здачі бакалаврської роботи.

ДОДАТОК Б

КЕРІВНИЦТВО КОРИСТУВАЧУ

1. Загальні відомості

Вебсайт для магазину дитячих товарів створено з метою забезпечення зручної онлайн-платформи для покупців. Сайт дозволяє користувачам переглядати доступний асортимент, обирати необхідні товари та оформляти замовлення.

2. Функціональне призначення

Даний програмний продукт призначений для організації онлайн-продажу дитячих товарів. За допомогою вебсайту користувачі можуть ознайомитися з товарами, скористатися пошуком і фільтрами для зручної навігації по каталогу, переглядати акційні пропозиції, додавати товари до кошика та оформляти замовлення. Адміністратор сайту має можливість додавати нові товари, редагувати наявні позиції, змінювати опис, ціну та зображення продукції.

3. Умови застосування програми

Для роботи з вебсайтом необхідний пристрій із доступом до мережі Інтернет (комп'ютер, планшет, смартфон) та сучасний браузер (наприклад, Google Chrome, Microsoft Edge, Safari та інші).

4. Опис роботи програми

Вебсайт для продажу дитячих товарів розроблений з метою полегшити процес покупок і взаємодії з клієнтами. Програма має наступні функціональні можливості:

- користувач може оглянути головну сторінку та її вміст (рис. Б.1);
- користувач може ознайомитися з різними категоріями товарів та переглянути якусь із них (рис. Б.2);
- користувач може здійснювати пошук товарів за ключовими словами, отримуючи результати, що відповідають запиту (рис. Б.3);
- користувач може переглянути актуальні акційні пропозиції (рис. Б.4);

- користувач може додавати товари до кошика (рис. Б.5);
- після вибору товарів користувач може оформити замовлення та ввести дані для доставки (рис. Б.6).

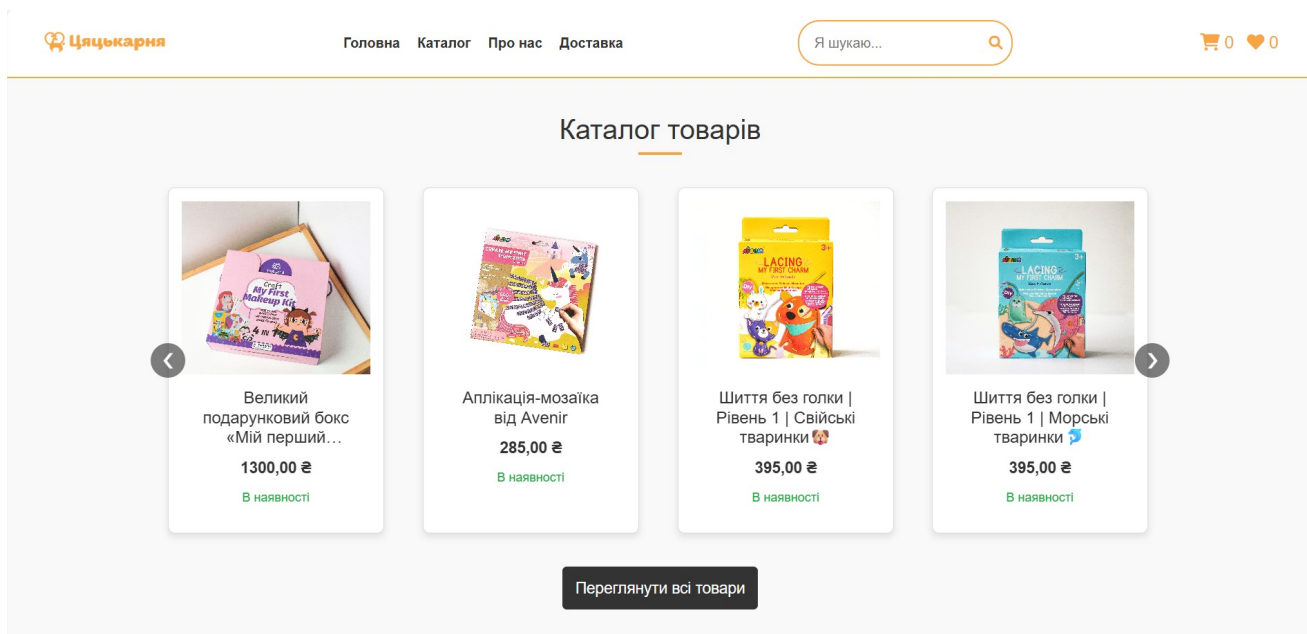


Рис. Б.1 – Головна сторінка вебсайту

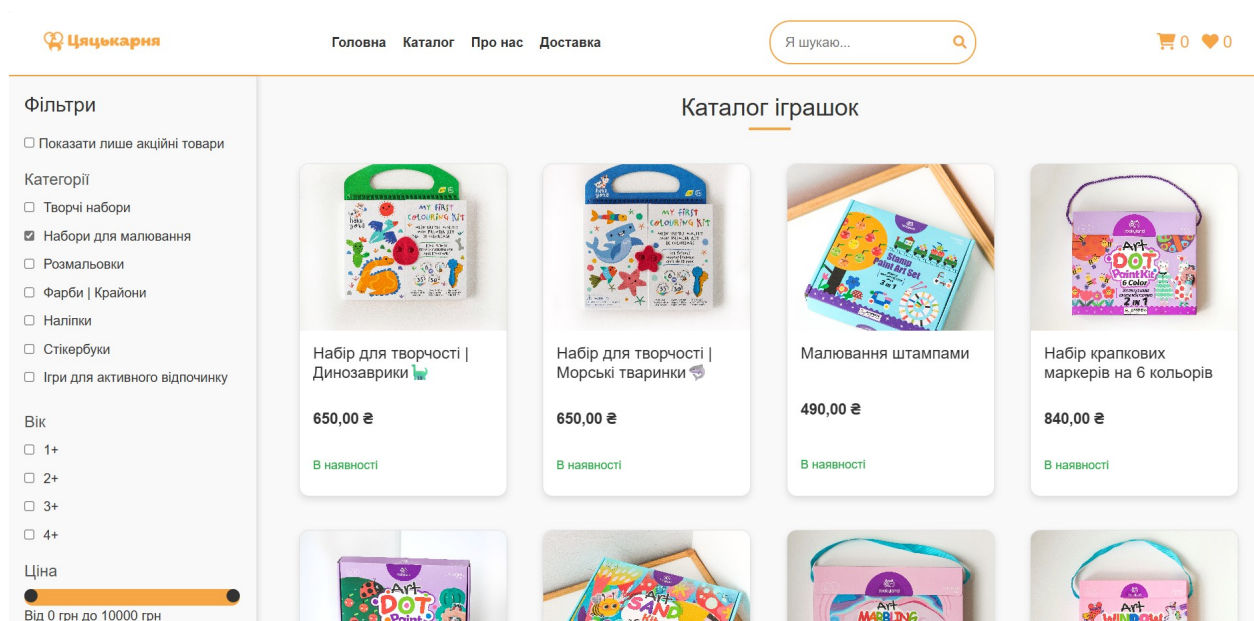


Рис. Б.2 – Перегляд категорії «Набори для малювання»

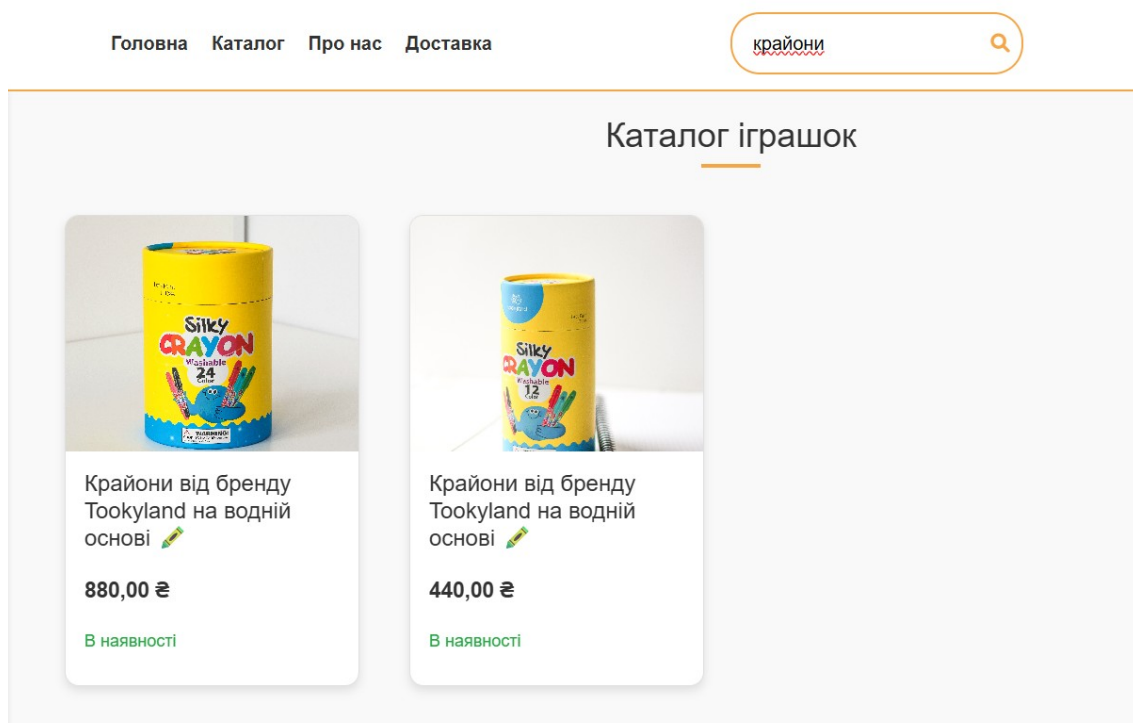


Рис. Б.3 – Пошук за ключовими словами

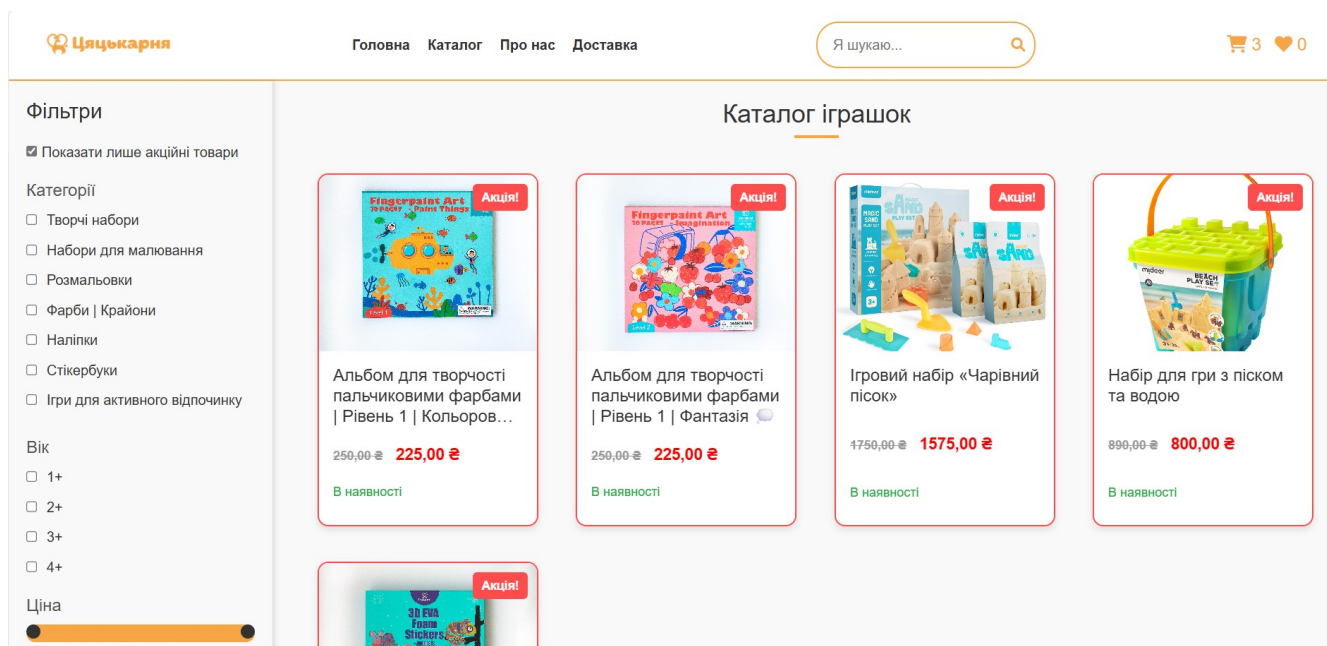


Рис. Б.4 – Перегляд акційних товарів

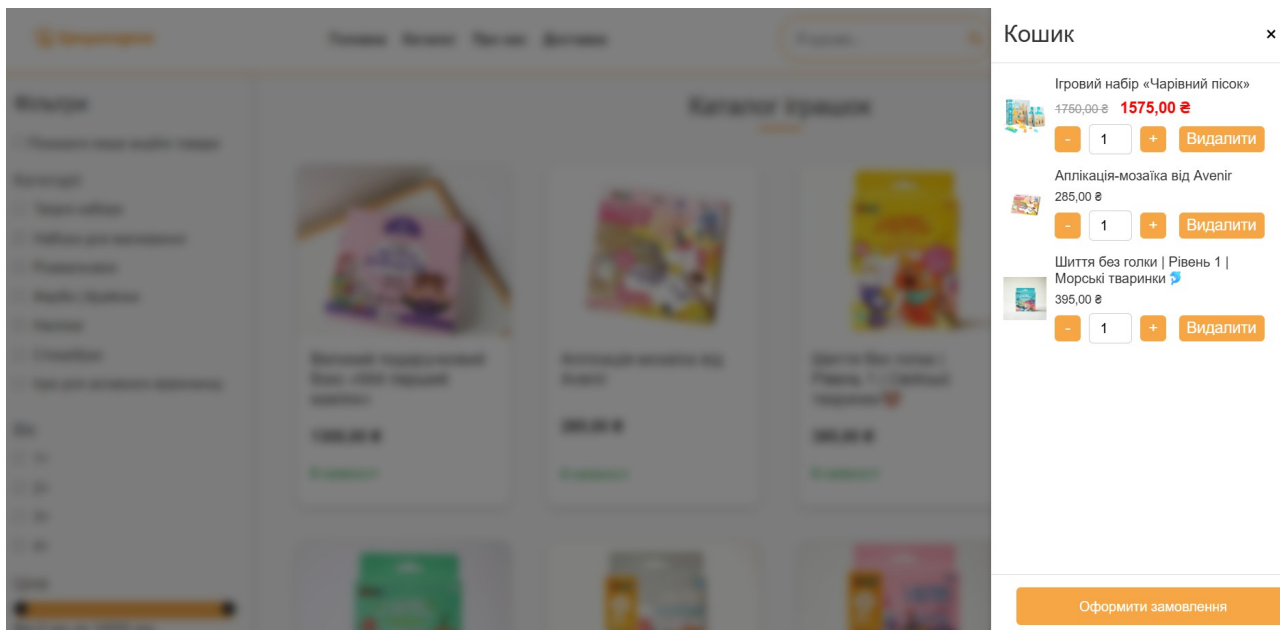


Рис. Б.5 – Перегляд кошика

Контактні дані

Ім'я *

Прізвище *

Телефон *

Email *

Доставка

Місто *

Відділення Нової Пошти *

Коментар до замовлення

Ваше замовлення

Ігровий набір «Чарівний пісок»
1750,00 ₴ **1575,00 ₴**
Кількість: 1
Сума: 1575.00 ₴

Аплікація-мозаїка від Avenir
285,00 ₴
Кількість: 1
Сума: 285.00 ₴

Шиття без голки | Рівень 1 | Морські тваринки
395,00 ₴
Кількість: 1
Сума: 395.00 ₴

Всього товарів: 3

Сума: 2255.00 ₴

Доставка: За тарифами перевізника

До сплати: 2255.00 ₴

Рис. Б.6 – Оформлення замовлення

ДОДАТОК В

МОДУЛЬНІ ТЕСТИ

```
def test_create_toy(self):
    toy = Toy.objects.create(
        name="Набір для творчості",
        description="Чудовий набір для розвитку уяви",
        category="creative_sets",
        price=1500.00,
        stock=10,
        age_group='3+',
        is_on_sale=True,
        sale_price=1200.00
    )
    self.assertEqual(toy.name, "Набір для творчості")
    self.assertEqual(toy.final_price, 1200.00) # перевірка final_price
    self.assertTrue(toy.is_on_sale)
```

Рис. В.1 – Тестування створення товару

```
def test_final_price_without_sale(self):
    toy = Toy.objects.create(
        name="Фарби",
        description="Набір фарб для дітей",
        category="paints",
        price=300.00,
        stock=5,
        age_group='2+',
        is_on_sale=False,
        sale_price=None
    )
    self.assertEqual(toy.final_price, 300.00) # очікуємо звичайну ціну

def test_final_price_with_sale_but_no_sale_price(self):
    toy = Toy.objects.create(
        name="Розмальовка",
        description="Книжка-розмальовка",
        category="coloring",
        price=500.00,
        stock=3,
        age_group='4+',
        is_on_sale=True,
        sale_price=None
    )
    self.assertEqual(toy.final_price, 500.00) # очікуємо fallback до price
```

Рис. В.2 – Тестування відображення ціни

```
def test_toy_photo(self):
    toy = Toy.objects.create(
        name="Наліпки",
        description="Веселі наліпки",
        category="stickers",
        price=300.00,
        stock=20,
        age_group='1+'
    )

    image = SimpleUploadedFile("test.jpg", b"file_content", content_type="image/jpeg")
    photo = ToyPhoto.objects.create(toy=toy, photo=image)

    self.assertEqual(photo.toy.name, "Наліпки")
    self.assertIn("test.jpg", photo.photo.name)
```

Рис. В.3 – Тестування додавання додаткових фото до товару

```
def setUp(self):
    Toy.objects.create(name="Наліпки з тваринами",
                       category="stickers", age_group="3+",
                       price=500, stock=10, is_on_sale=True)
    Toy.objects.create(name="Багаторазові водні розмальовки",
                       category="coloring", age_group="2+",
                       price=300, stock=5, is_on_sale=False)
    Toy.objects.create(name="Набір для малювання",
                       category="art_sets", age_group="1+",
                       price=1200, stock=7, is_on_sale=True)

def test_filter_by_category(self):
    response = self.client.get(reverse('toy_catalog') + '?category=stickers')
    self.assertContains(response, "Наліпки з тваринами")
    self.assertNotContains(response, "Розмальовка динозаври")
```

Рис. В.4 – Тестування фільтрації за категорією

```
def test_filter_by_age_group(self):
    response = self.client.get(reverse('toy_catalog') + '?age=3+')
    self.assertContains(response, "Наліпки з тваринами")
    self.assertContains(response, "Набір для малювання")
    self.assertNotContains(response, "Розмальовка динозаври") # бо 2+
```

Рис. В.5 – Тестування фільтрації за віком

```
def test_filter_on_sale(self):
    response = self.client.get(reverse('toy_catalog') + '?on_sale=1')
    self.assertContains(response, "Наліпки з тваринами")
    self.assertContains(response, "Набір для малювання")
    self.assertNotContains(response, "Розмальовка динозаври")
```

Рис. В.6 – Тестування фільтрації акційних позицій

```
def test_filter_by_price_range(self):
    response = self.client.get(reverse('toy_catalog') + '?min_price=40&max_price=60')
    self.assertContains(response, "Наліпки з тваринами")
    self.assertNotContains(response, "Розмальовка динозаври")
    self.assertNotContains(response, "Набір для малювання")

def test_sort_by_price_asc(self):
    response = self.client.get(reverse('toy_catalog') + '?sort_by_price=asc')
    toys = list(response.context['page_obj'].object_list)
    self.assertTrue(toys[0].price <= toys[1].price <= toys[-1].price)
```

Рис. В.7 – Тестування фільтрації за ціною

```
def test_create_order_user(self):
    order = OrderUser.objects.create(
        first_name="Олена",
        last_name="Петренко",
        phone="380961112233",
        email="olena@example.com",
        city="Київ",
        warehouse="Відділення №5",
        comments="Будь ласка, подзвоніть перед доставкою",
        status="new",
        total_price=850.00
    )

    self.assertEqual(order.first_name, "Олена")
    self.assertEqual(order.status, "new")
    self.assertEqual(order.total_price, 850.00)
    self.assertIsNotNone(order.created_at)
    self.assertIsNotNone(order.updated_at)
```

Рис. В.8 – Тестування створення замовлення

```

def test_create_order_item(self):
    order = OrderUser.objects.create(
        first_name="Іван",
        last_name="Іваненко",
        phone="380931234567",
        email="ivan@example.com",
        city="Львів",
        warehouse="Відділення №12",
        comments="",
        status="processing",
        total_price=300.00
    )

    item = OrderItem.objects.create(
        order=order,
        product_name="Набір для малювання",
        product_price=300.00,
        quantity=1
    )

    self.assertEqual(item.order, order)
    self.assertEqual(item.product_name, "Набір для малювання")
    self.assertEqual(item.quantity, 1)
    self.assertEqual(item.product_price, 300.00)

```

Рис. В.9 – Додавання товарів до замовлення

```

def test_order_status_choices(self):
    order = OrderUser.objects.create(
        first_name="Тарас",
        last_name="Гончар",
        phone="380991234567",
        email="taras@example.com",
        city="Харків",
        warehouse="Відділення №2",
        comments="",
        status="delivered",
        total_price=200.00
    )

    self.assertIn(order.status, dict(OrderUser.STATUS_CHOICES).keys())

```

Рис. В.10 – Тестування статусу замовлення

АНОТАЦІЯ

Зубок Д. В. Проектування та розробка сайту для продажу дитячих товарів засобами вебтехнологій. Рукопис.

Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 122 Комп'ютерні науки. Волинський національний університет імені Лесі Українки, Луцьк, 2025 р.

Актуальність теми зумовлена розвитком електронної комерції, зокрема в сфері продажу дитячих товарів, що потребує зручних онлайн-інструментів із простим інтерфейсом і надійною функціональністю.

Мета роботи – створення вебзастосунку для продажу дитячих товарів із функціями перегляду каталогу, пошуку, фільтрації, додавання товарів до кошика та оформлення замовлення.

Програмний продукт складається з клієнтської частини, реалізованої з використанням HTML, CSS та JavaScript, а також серверної частини, створеної на основі фреймворку Django мовою програмування Python. Для збереження інформації про товари та замовлення використано вбудовану базу даних SQLite. Реалізовано функції пошуку, фільтрації за категоріями, відображення товарів із знижками, формування кошика та оформлення замовлення з автоматичним додаванням інформації до бази даних.

У роботі наведено опис структури бази даних, ключових представлень (view-функцій), шаблонів, а також алгоритмів, які забезпечують основну логіку програми. Також подано пропозиції щодо розширення функціональності у майбутньому, зокрема можливість впровадження системи авторизації та підключення платіжного модуля.

Ключові слова: вебзастосунок, дитячі товари, інтернет-магазин, електронна комерція, Django, Python, база даних, автоматизація замовлень.