

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Інтернет-магазин дитячих іграшок»

Виконав студент групи ІПЗ-19-2
КРАВЧЕНКО Олександр Іванович

Керівник роботи:
ЛОКТИКОВА Тамара Миколаївна

Рецензент:
ЛОБАНЧИКОВА Надія Миколаївна

Житомир – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

В.о. зав. кафедри інженерії

програмного забезпечення

Андрій МОРОЗОВ

«14» лютого 2023 р.

ЗАВДАННЯ
на випускню кваліфікаційну роботу

Здобувач вищої освіти: КРАВЧЕНКО Олександр Іванович

Керівник роботи: ЛОКТИКОВА Тамара Миколаївна

Тема роботи: «Інтернет-магазин дитячих іграшок»,

затверджена Наказом закладу вищої освіти від «14» лютого 2023 р., №70/с

Вихідні дані для роботи: Об'єктом дослідження є процес створення та оптимізації інтернет-магазину з метою досягнення успішного функціонування та задоволення потреб клієнтів.

Предметом дослідження є використання фреймворка Yii2 та мови програмування PHP для розробки інтернет-магазину дитячих іграшок.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1	Локтікова Т.М.	14.02.2023р.	14.02.2023р.
2	Локтікова Т.М.	15.03.2023р.	15.03.2023р.
3	Локтікова Т.М.	01.05.2023р.	01.05.2023р.

Керівник _____ Тамара ЛОКТИКОВА

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналізування вимог до програмного продукту	14.02.2023 – 21.02.2023	Виконано
2	Пошук, огляд та аналіз аналогічних програмних продуктів	22.02.2023 – 28.02.2023	Виконано

3	Вивчення матеріалу по конкретній тематиці	01.03.2023 – 14.03.2023	Виконано
4	Планування структури програми	15.03.2023 – 22.03.2023	Виконано
5	Проектування та реалізація архітектури системи інтернет-магазину	23.03.2023 – 31.04.2023	Виконано
6	Написання програмного коду	01.04.2023 – 01.05.2023	Виконано
7	Тестування роботи програми	02.05.2023 – 15.05.2023	Виконано
8	Аналіз одержаних результатів	16.05.2023 – 28.05.2023	Виконано
9	Оформлення пояснювальної записки	29.05.2023 – 14.06.2023	Виконано
10	Попередній захист	05.05.2023	Виконано

РЕФЕРАТ

Випускна робота бакалавра складається з програмного продукту інтернет-магазину дитячих іграшок та пояснювальної записки. Пояснювальна записка до випускної роботи містить текстову частину, викладену на 58 сторінках друкованого тексту, має 38 сторінок додатків. Список використаних джерел містить 20 найменувань і займає 2 сторінки. В роботі наведено 53 ілюстрації та 18 таблиць. Загальний обсяг роботи – 96 сторінок.

Метою роботи є створення інтернет-магазину дитячих іграшок за допомогою HTML, CSS та розробка системи адміністрування за допомогою мови програмування PHP із застосуванням бази даних MySQL.

У роботі визначено головні цілі та завдання для розробки додатку, проаналізовано методи і шляхи вирішення запланованої задачі, описано і обґрунтовано вибір інструментальних засобів для розробки. Наведено алгоритми роботи програмного додатку, а ще архітектуру та взаємодію його основних складових. Також була описана реалізація реєстрації та авторизації користувачів, додавання товарів до кошика за допомогою сесій і додавання відгуків до товарів.

Ключові слова: PHP, WEB-ДОДАТОК, САЙТ, АВТОРИЗАЦІЯ, РЕЄСТРАЦІЯ, УПІД, БАЗА ДАНИХ.

					ІПЗ.КР.Б – 121 – 23 – ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет-магазин дитячих іграшок				Літ.	Арк.	Аркушів
Розроб.		О.І. Кравченко									
Керівник		Т.М. Локтікова								4	96
									Житомирська політехніка, група ІПЗ-19-2		
Рецензент		Н.М. Лобанчикова									
Зав. каф.		А.В. Морозов			Пояснювальна записка						

ABSTRACT

The bachelor's thesis consists of a software product of an online children's toy store and an explanatory note. The explanatory note to the final thesis contains the text part laid out on 58 pages of printed text, has 38 pages of appendices. The list of used sources contains 20 names and occupies 2 pages. The work contains 53 illustrations and 18 tables. The total volume of work is 96 pages.

The purpose of the work is to create an online store of children's toys using HTML, CSS and to develop an administration system using the PHP programming language with the use of a MySQL database.

The work defines the main goals and tasks for the development of the application, analyzes the methods and ways of solving the planned task, describes and substantiates the choice of tools for development. Algorithms of the software application are presented, as well as the architecture and interaction of its main components. The implementation of user registration and authorization, adding products to the cart using sessions and adding reviews to products was also described.

Keywords: PHP, WEB APPLICATION, SITE, AUTHORIZATION, REGISTRATION, YII2, DATABASE.

					III3.KP.Б – 121 – 23 – II3	.
	.					5

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНА ДИТЯЧИХ ІГРАШОК.....	9
1.1 Постановка задачі	9
1.2 Аналіз аналогів програмного продукту	10
1.3 Вибір архітектури програмного продукту	14
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	15
Висновки до першого розділу.....	17
РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНА ДИТЯЧИХ ІГРАШОК	18
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	18
2.2 Розробка бази даних системи.....	23
2.3 Проектування та реалізація алгоритмів роботи системи.....	29
2.4 Реалізація інтернет-магазину дитячих іграшок	31
Висновки до другого розділу	34
РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНТЕРНЕТ-МАГАЗИНОМ ДИТЯЧИХ ІГРАШОК	36
3.1 Порядок встановлення та налаштування параметрів системи.....	36
3.2 Структура інтерфейсу та порядок роботи з інтернет-магазином	37
3.3 Тестування роботи програмного продукту	52
Висновки до третього розділу	54
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТКИ.....	58

ВСТУП

Актуальність теми. Тема інтернет-магазину дитячих іграшок залишається актуальною в наш час. Інтернет-торгівля постійно розвивається і зростає, що створює чудові можливості для бізнесу, зокрема у сфері дитячих іграшок.

Інтернет-магазин дитячих іграшок дозволить батькам швидко та зручно придбати необхідні товари без виходу з дому. Вони можуть переглядати широкий асортимент продуктів, порівнювати ціни, читати відгуки та здійснювати покупки у зручний для себе час.

Ринок дитячих іграшок є великим і розмаїтим. Кожен рік з'являються нові іграшки та тренди, що створює безліч можливостей для підприємців. Інтернет-магазин дозволяє пропонувати широкий асортимент товарів та задовольняти різноманітні потреби клієнтів.

За останні роки спостерігається стабільний ріст онлайн-торгівлі в усьому світі. Споживачі все більше віддають перевагу покупкам в інтернеті, що створює чудові умови для розвитку інтернет-магазинів дитячих іграшок.

Завдяки розвитку технологій та доступності смартфонів та планшетів, діти все частіше використовують електронні пристрої для розваг. Інтернет-магазини можуть відповідати на цей тренд, пропонуючи широкий вибір електронних іграшок та аксесуарів для дітей.

Інтернет-магазин дитячих іграшок є перспективним напрямом бізнесу. Важливо мати якісну продукцію, забезпечувати швидку доставку, добре організований клієнтський сервіс та ефективну маркетингову стратегію для привернення клієнтів.

Метою випускної роботи є розробка інтернет-магазину дитячих іграшок.

Встановлена мета обумовлює наступні завдання:

- визначити та проаналізувати основні вимоги до інтернет-магазину;
- дослідити та визначити існуючі інтернет-магазини;
- розробити загальну архітектуру системи
- проаналізувати та обрати засоби реалізації програмного продукту;

- визначити можливі варіанти використання та розробити об'єктно-орієнтовану структуру системи;
- проектування бази даних;
- розробити алгоритми системи та їх швидкодію;
- забезпечити реалізацію всього необхідного функціоналу для інтернет-магазину;
- описати структури та порядок встановлення системи;
- описати структуру інтерфейсу додатку;
- виконати тестування розробленого додатку.

Об'єктом дослідження є процес створення та оптимізації такого магазину з метою досягнення успішного функціонування та задоволення потреб клієнтів.

Предметом дослідження є використання фреймворка Yii2 та мови програмування PHP для розробки інтернет-магазину дитячих іграшок.

В роботі було використано методи об'єктно-орієнтованого програмування і проектування, а також принципи розробки дизайну інтерфейсів.

Розроблений інтернет-магазин має на меті допомогти користувачеві знайти швидко та зручно необхідні для нього товари, здійснити замовлення та отримати товар хорошої якості і за привабливими цінами.

РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНА ДИТЯЧИХ ІГРАШОК

1.1 Постановка задачі

Розробка повнофункціонального інтернет-магазину для продажу дитячих товарів онлайн. Потрібно створити веб-додаток, який буде надавати користувачам можливість переглядати каталог товарів, здійснювати замовлення, та забезпечувати повну зручність для користувача під час оформлення замовлення.

Опис функціональності:

- реєстрація та авторизація користувачів:
користувачі повинні мати можливість створити обліковий запис або увійти в існуючий обліковий запис для збереження своїх персональних даних;
- перегляд каталогу товарів:
користувачі повинні мати можливість переглядати список доступних товарів у вигляді категорій, а також шукати конкретний товар;
- детальна інформація про товари:
користувачі повинні мати можливість переглядати докладну інформацію про кожен товар, включаючи зображення, опис, ціну, наявність тощо;
- додавання товарів до кошика:
користувачі повинні мати можливість додавати товари до свого кошика покупок, а також видаляти їх з кошика;
- додавання товарів до списку улюбленого:
користувачі повинні мати можливість додавати товари до свого списку улюблених товарів, а також видаляти їх з кошика;
- оформлення замовлення:
користувачі повинні мати можливість оформити замовлення, вказавши необхідні дані, після успішного оформлення замовлення користувач повинен

отримати підтвердження замовлення;

- управління замовленнями:

адміністратори магазину повинні мати можливість переглядати та управляти замовленнями, змінювати їх статуси;

- керування користувачами:

адміністратори магазину повинні мати можливість керувати користувачами, видаляти, корегувати дані за потреби;

- керування категоріями товарів та самими товарами:

адміністратори повинні мати можливість додавати, видаляти, редагувати товари та їх категорії, щоб мати актуальні дані для користувачів.

Технічні вимоги:

- розробка веб-додатку вимагає використання сучасних, популярних та зручних для розробника та користувача веб-технологій;

- зберігання даних про товари, категорії, користувачів та замовлення в базі даних;

- оптимізація для пошукових систем та розробка з урахуванням SEO-правил;

- розробка зручного інтерфейсу для користувача та використання мобільно-адаптивного дизайну;

В цілому задача передбачає розробку програмного коду, дизайну користувацького інтерфейсу, налаштування бази даних та розгортання веб-додатку на сервері. Також необхідно забезпечити тестування та налагодження функціоналу перед запуском.

1.2 Аналіз аналогів програмного продукту

Розробка сучасного та конкурентноспроможного інтернет-магазину потребує чіткого аналізу його конкурентів та аналогічних магазинів.

Під час дослідження багатьох аналогічних програмних продуктів було знайдено схожі за функціоналом та дизайном інтернет-магазини. Тому в процесі

роботи були знайдені та досліджені інтернет-магазини: Goodtoys [1], Jambo [2], Rozum [3].

Rozum – інтернет-магазин з продажу дитячих іграшок, який створений спеціально для задоволення потреб дітей різного віку. Магазин пропонує широкий асортимент якісних, безпечних і цікавих іграшок, що допомагають дитині розвиватися. До того ж, магазин пропонує зручний інтерфейс, який дозволяє з легкістю знайти будь-який товар.



Рисунок 1.1 – Логотип інтернет-магазину Rozum

Основними перевагами Rozum є:

- детальний опис кожного товару;
- зручний пошук та фільтрація товарів;
- чат для зв'язку, під час виникнення питань.

Основними недоліками є:

- перевантаженість інформацією сторінок сайту;
- відсутнє повідомлення про замовлення;
- присутність перекладу на російську мову.

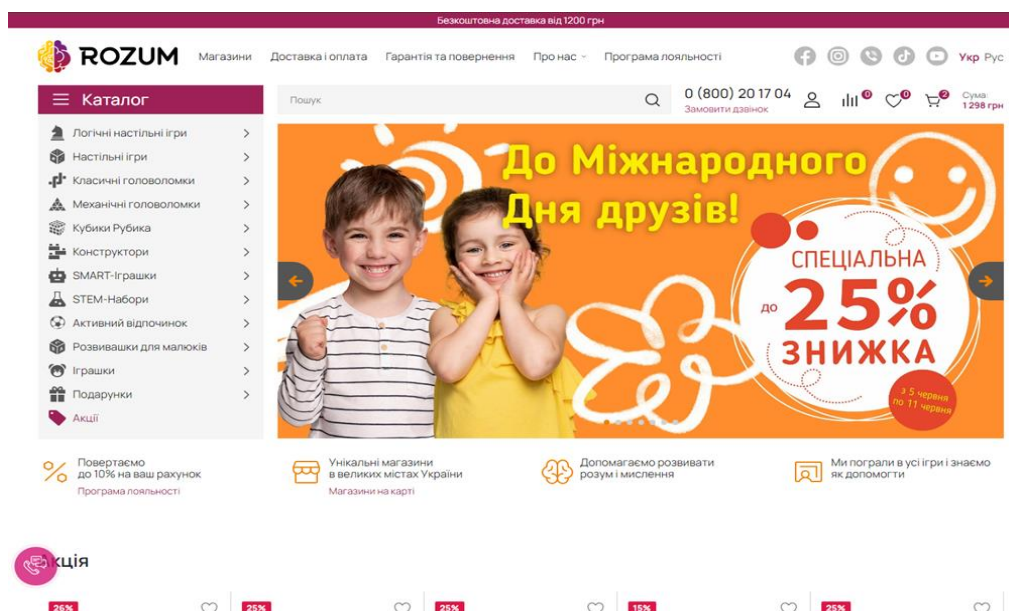


Рисунок 1.2 – Головна сторінка магазину Rozum

Jambo - українська мережа магазинів товарів для дітей. Інтернет-магазин хоче зробити процес покупки товарів швидким, простим і зручним для користувача. Магазин надає широкий асортимент товарів для дітей різного віку і має розумну цінову політику.



Рисунок 1.3 – Логотип інтернет-магазину Jambo

Основними перевагами Jambo є:

- проста і зручна навігація по сайту;
- можливість замовлення дзвінка при виникненні питань;
- безпека та захищеність персональних даних.

Основними недоліками є:

- не зручний інтерфейс користувача;
- обмежена функціональність;
- мінімальна сума замовлення.

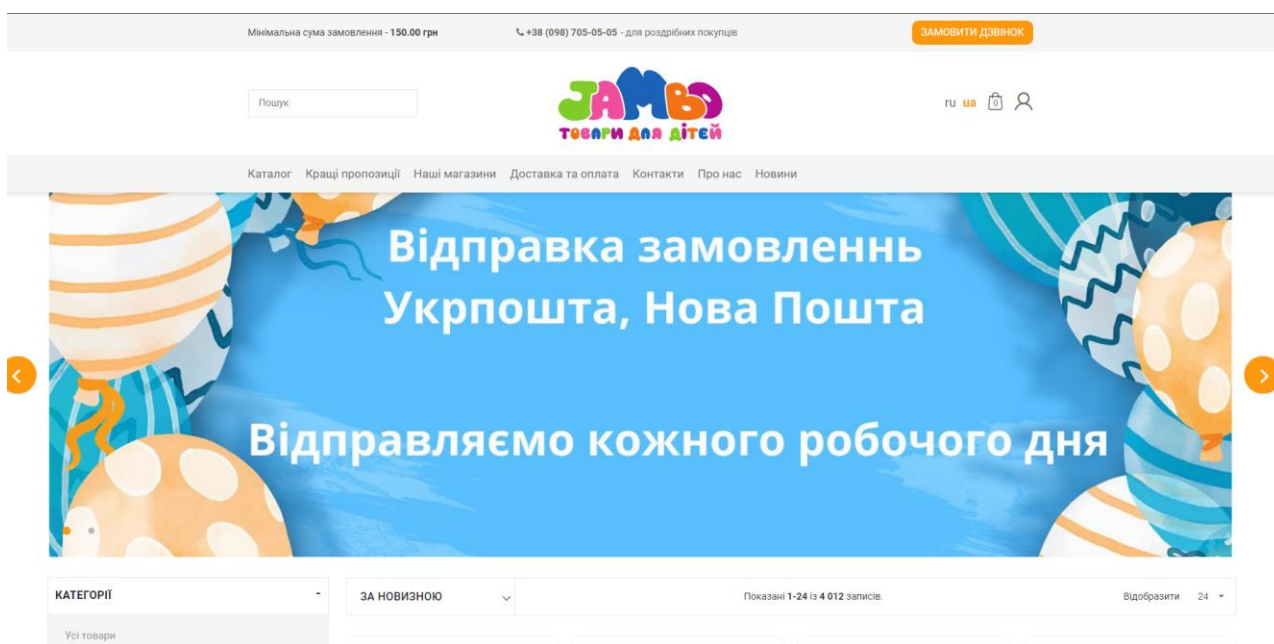


Рисунок 1.4 – Головна сторінка магазину Jambo

Goodtoys – це надійний інтернет-магазин у виборі дитячих іграшок. Він прагне зробити процес купівлі дитячих іграшок приємним для користувача, водночас покладає великий акцент на безпеку дитини. Пропонує зручну систему доставки, що дозволяє отримати замовлення вчасно та без зайвих зусиль.



Рисунок 1.5 – Логотип інтернет-магазину Goodtoys

Основними перевагами Goodtoys є:

- великий асортимент дитячих іграшок;
- знижки, акції;
- безпека.

Основними недоліками є:

- недостатньо параметрів для фільтрації товарів;
- простий дизайн;
- проблеми з версткою;
- проблеми з перекладом на деяких сторінках сайту.

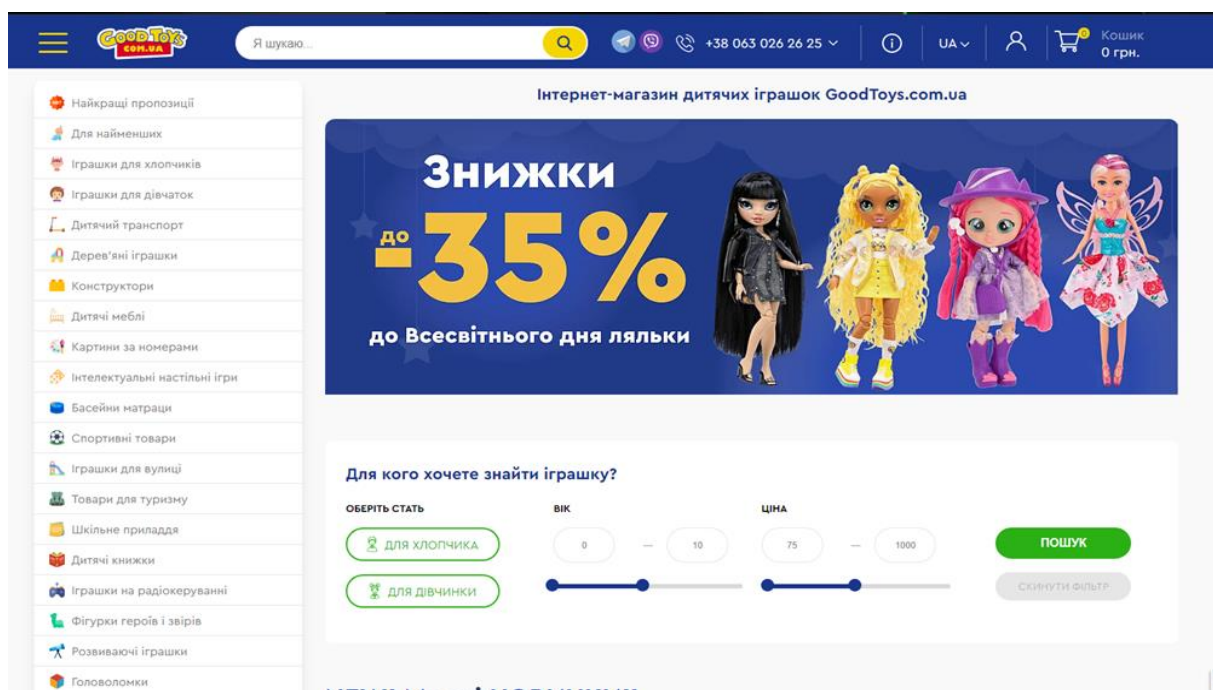


Рисунок 1.6 – Головна сторінка магазину Goodtoys

Підсумовуючи вище наведені переваги та недоліки, складемо порівняльну характеристику згаданих аналогів та власного програмного продукту табл. 1.1.

Таблиця 1.1

Характеристика	Goodtoys	Jambo	Rozum	Власний продукт
Зручність інтерфейсу	+	+	+	+
Дизайн	—+	+	—+	+
Безпека	+	—	—+	+
Можливість онлайн оплати	+	+	+	—
Онлайн підтримка	—	+	-	—
Створення профілю	+	+	+	—

Переглянувши та проаналізувавши аналоги нашого інтернет-магазину, можна зробити висновок, що кожен програмний продукт має свої недоліки та переваги, що створить певну різноманітність та зробить конкуренцію більшою.

Отже, кожен користувач матиме вибір, де йому краще купувати товари, зважаючи на всі плюси та мінуси кожного із інтернет-магазинів.

1.3 Вибір архітектури програмного продукту

Ефективність, надійність, швидкість та масштабованість системи залежить від багатьох факторів, проте одним із головних є вибір архітектури програмного продукту. Також вибір архітектури сильно залежить від конкретних потреб та вимог і від того, що очікується побачити в кінці розробки програмного продукту.

Тому, при виборі архітектури програмного продукту було проаналізовано декілька архітектур: MVVM, MVP та MVC [4].

MVP – це архітектурний шаблон, що використовується для розробки програмних продуктів з графічним інтерфейсом користувача. Вона поділяє компоненти програми на три основні частини: представлення (View), презентер

(Presenter) та модель (Model).

Якщо говорити про MVVM, то ця архітектура допомагає відокремити інтерфейс користувача від самої логіки програмного продукту. Також вона розділяє компоненти програми на три частини: модель (Model), представлення (View) та модель представлення (ViewModel).

При створенні програмного додатку було обрано MVC, яка є однією з найпоширеніших архітектур для розробки веб-додатків і добре підходить для багатьох різноманітних ідей та сценаріїв.

MVC дозволяє поділити логіку додатку на моделі (дані), перегляди (представлення) та контролери (логіка додатку). Як результат, це дозволяє покращити організацію коду, забезпечити легку підтримку та розширення додатку.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Як розробка будь якого програмного продукту вимагає певного набору інструментальних засобів, так і інтернет-магазин потребує своїх засобів.

Тому зважаючи на свій досвід, прочитуючи документацію та аналізуючи засоби і шляхи створення інтернет-магазинів, було обрано таку технологію, як Yii2 [5]. Завдяки цій технології був розроблений інтернет-магазин дитячих іграшок.

Yii2 - є відкритим PHP-фреймворком для швидкої розробки веб-додатків. Він є нащадком фреймворку Yii і пропонує багато нових можливостей та покращень у порівнянні з попередньою версією.

Yii2 надає розробникам потужні інструменти для створення різноманітних веб-додатків, включаючи великі корпоративні системи, соціальні мережі, форуми, електронні магазини. Він побудований з використанням принципів MVC (Model-View-Controller), що дозволяє розділити логіку додатку на компоненти для зручного управління.

Основними перевагами цього фреймворку є:

- висока продуктивність:

фреймворк має ефективну систему кешування, підтримку асинхронних запитів та інші оптимізації, що допомагають забезпечити швидку роботу додатків;

- міцна безпека:

він має вбудовану систему безпеки, яка допомагає захистити додатки від численних загроз. Також фреймворк надає доступ до безпечної обробки форм, авторизації, автентифікації;

- розширюваність:

Yii2 надає гнучкий механізм розширення. Ви можете легко додавати нові компоненти, модулі або розширення до фреймворку, що дозволяє розширити його функціональність без зміни основного додатку.

Для написання коду було обрано середовище програмування PhpStorm [6], де буде виконуватись розробка додатку з використанням PHP [7] версії 7.4, HTML5 [8] та CSS3 [9]. Більш того, PhpStorm надає багато функцій та інструментів для полегшення розробки програмного забезпечення.

Для сервера було обрано OpenServer [10], який є локальним сервером та надає зручне середовище для розробки та тестування веб-додатків. Включає в себе набір серверних технологій, таких як бази даних MySQL [11], PHP.

З метою гарної працездатності системи, забезпечення швидкодії та дотримання ефективного використання ресурсів апаратного забезпечення були сформовані певні вимоги:

1. Вимоги до робочої станції розробника:

- Оперативна пам'ять - 4GB.
- Операційна система – Windows 10.
- Процесор з частотою не менше 3 GHz.
- Браузер.
- Жорсткий диск – 256 ГБ.

2. Вимоги до робочої станції користувача:

Браузер, який відповідає наступним вимогам:

- Chrome 60+.
- Safari 10+ / iOS Safari 10+.
- Edge 12+.
- Firefox ESR+.
- Internet Explorer 11.
- Opera.
- Процесор з частотою від 3 GHz.
- Оперативна пам'ять – 8GB.
- Жорсткий диск - 256 GB.
- Операційна система – Windows 10.

Висновки до першого розділу

Проведений аналіз допоміг визначити основні вимоги та необхідні функціональні можливості для інтернет-магазину дитячих іграшок.

Під час дослідження аналогів програмного додатку були проаналізовані аналогічні інтернет-магазини: Rozum, Goodtoys, Jambo.

На основі поставлених вимог, визначених критеріїв та прийнятих архітектурних рішень було проаналізовано декілька архітектур: MVVM, MVP та MVC і зважаючи на всі переваги та недоліки зроблено рішення про використання архітектури MVC. Засобами реалізації програмного додатку було обрано PHP фреймворк Yii2, HTML5, CSS3, середовищем для розробки коду обрано PhpStorm, а для сервера використано OpenServer. Окрім цього, були визначені апаратні вимоги для кожного елементу системи з метою ефективного використання ресурсів та відповідності програмного забезпечення до всіх визначених потреб.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНА ДИТЯЧИХ ІГРАШОК

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Інтернет-магазин має в собі декілька ролей для користувачів. З метою наочного представлення відносин між елементами системи було створено діаграму варіантів використання та показано на рис. 2.1.

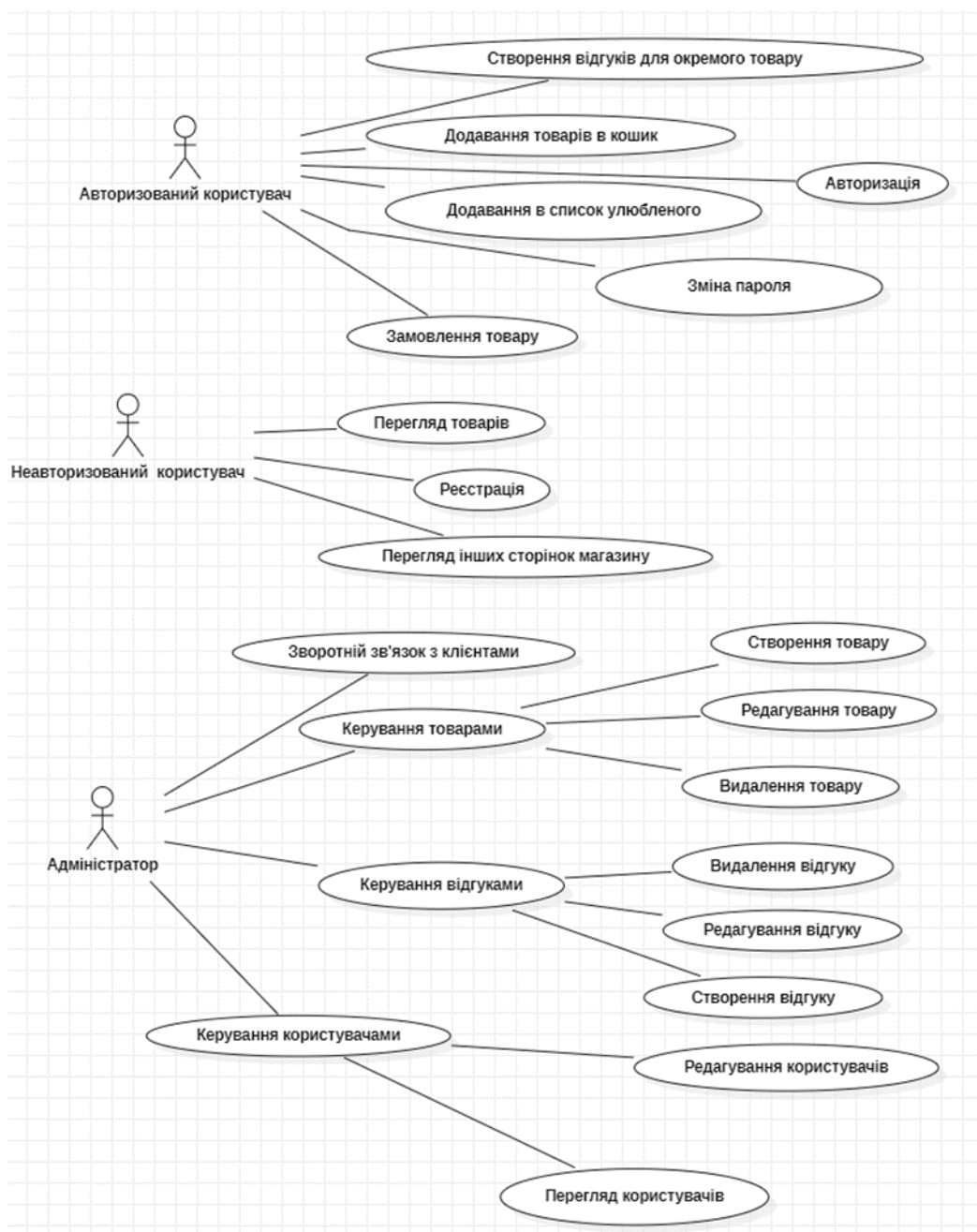


Рисунок 2.1 – Діаграма варіантів використання

Опис кожного преценденту наведено у табл. 2.1–2.7.

Таблиця 2.1

Ім'я варіанту використання: Додавання товарів в список улюбленого та в кошик		Рівень важливості: високий
Основний актор: авторизований користувач		
Короткий опис:		Додавання певних товарів в список улюбленого та в кошик.
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час додавання товарів в список улюбленого або в кошик, потрібно проінформувати користувача про помилку.
Альтернативні/виключні потоки:		Виведена форма для повідомлення помилки

Таблиця 2.2

Ім'я варіанту використання: Перегляд інформації товарів		Рівень важливості: високий
Основний актор: авторизований користувач		
Короткий опис:		Перегляд повної інформації про товар
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час перегляду інформації, то необхідно проінформувати користувача.
Альтернативні/виключні потоки:		Виведена форма для повідомлення помилки

Таблиця 2.3

Ім'я варіанту використання: Проходження реєстрації та авторизації		Рівень важливості: високий
Основний актор: користувач		
Короткий опис:		Проходження реєстрації та авторизації
Передумови:		
Пост-умови:		
Відносини:		

Продовження табл. 2.3

	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час реєстрації або авторизації, то необхідно проінформувати користувача.
Альтернативні/виключні потоки:		Виведена форма для повідомлення помилки

Таблиця 2.4

Ім'я варіанту використання: Керування товарами		Рівень важливості: високий
Основний актор: адміністратор		
Короткий опис:		Додавання вмісту, його редагування та видалення для певного товару.
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час змін, то необхідно проінформувати користувача. Інакше зміни успішно збережено.
Альтернативні/виключні потоки:		

Таблиця 2.5

Ім'я варіанту використання: Керування користувачами		Рівень важливості: високий
Основний актор: адміністратор		
Короткий опис:		Перегляд всіх та певного користувача, видалення, редагування
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час змін, то необхідно проінформувати користувача. Інакше зміни успішно збережено.
Альтернативні/виключні потоки:		

Таблиця 2.6

Ім'я варіанту використання: Керування відгуками		Рівень важливості: високий
Основний актор: адміністратор		
Короткий опис:		Перегляд всіх та певного відгуку видалення, редагування.
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час змін, то необхідно проінформувати користувача. Інакше зміни успішно збережено.
Альтернативні/виключні потоки:		

Таблиця 2.7

Ім'я варіанту використання: замовлення товарів		Рівень важливості: високий
Основний актор: авторизований користувач		
Короткий опис:		Користувач робить замовлення товарів.
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Якщо сталася помилка під час замовлення, то необхідно проінформувати користувача. Інакше користувач отримає успішне повідомлення.
Альтернативні/виключні потоки:		

Бізнес-вимоги:

1. Основні цілі: проект створюється з метою легкого пошуку та купівлі товарів.
2. Можливості: перегляд товарів та замовлення дитячих іграшок.

3. Представлення проекту: проект буде реалізовано у вигляді веб-додатку, що містить зручний інтерфейс та необхідну інформацію за даною тематикою.

Вимоги користувачів

Зовнішні користувачі – авторизований користувач

1. Можливість додавати товар в кошик;
2. Можливість переглядати обраний товар та залишити відгук про нього;
3. Можливість додавати товар в перелік обраного;
4. Можливість перегляду інших сторінок з контентом;
5. Можливість перегляду знижок або акцій;
6. Можливість дати свої рекомендації для покращення роботи інтернет-магазину;
7. Можливість змінити пароль.

Внутрішній користувач – адміністратор

1. Редагування БД;
2. Надання відповідей користувачам.

Згідно архітектури MVC передбачено класи контролерів та моделей, на рис. 2.2 можемо побачити відповідні діаграми класів моделей додатку, які відповідають за зв'язок застосунку з базою даних та формує зручний процес керування додатком.

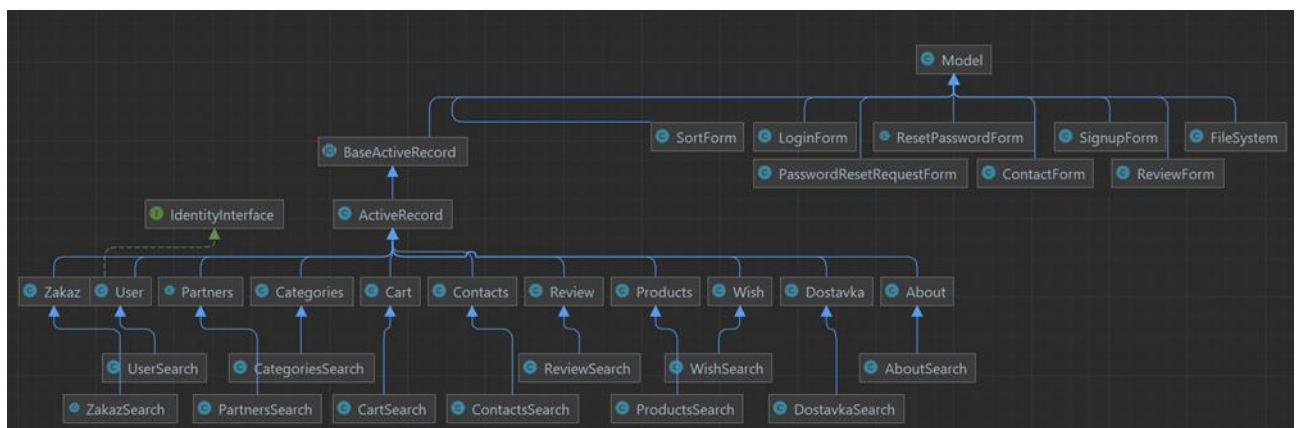


Рисунок 2.2 – Діаграма класів моделей додатку

Також, на рис. 2.3 можемо переглянути відповідні діаграми класів контролерів додатку, що відповідають за формування представлень сторінок додатку.

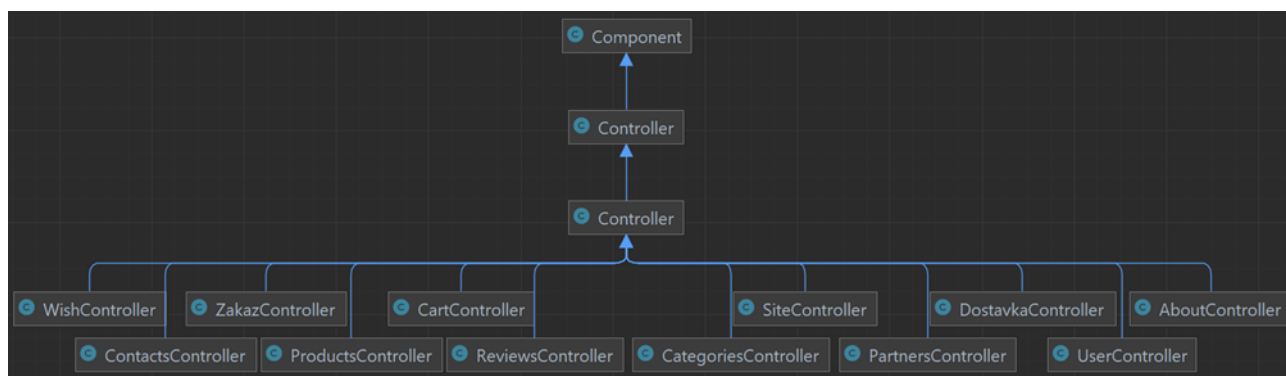


Рисунок 2.3 – Діаграма класів контролерів додатку

2.2 Розробка бази даних системи

Для безпечного та довготривалого збереження даних про користувачів, замовлення, відгуки та інші створено відповідні таблиці. Поля, їх типи та зв'язки показано на рис. 2.4.

Було обрано систему управління базами даних MySQL, яка є потужною та надійною системою.

На основі аналізу вимог для нашого інтернет-магазину було створено схему бази даних, використовуючи діаграми сутність-зв'язок (ER-діаграма [12]), для відображення взаємозв'язків між різними таблицями бази даних.

Для збереження даних про користувачів було створено таблицю User. Поля, їх типи та призначення наведено в табл. 2.8.

Таблиця 2.8

User		
Поле	Тип	Призначення
1	2	3
id	Number	Унікальний ідентифікатор

username	String	Ім'я користувача
1	2	3
auth_key	String	Ключ авторизації
password_hash	String	Захешований пароль
password_reset_token	String	Скидання паролю
email	String	Електронна пошта
status	Number	Статус
created_at	Number	Дата і час створення
updated_at	Number	Дата і час оновлення
role	Number	Роль

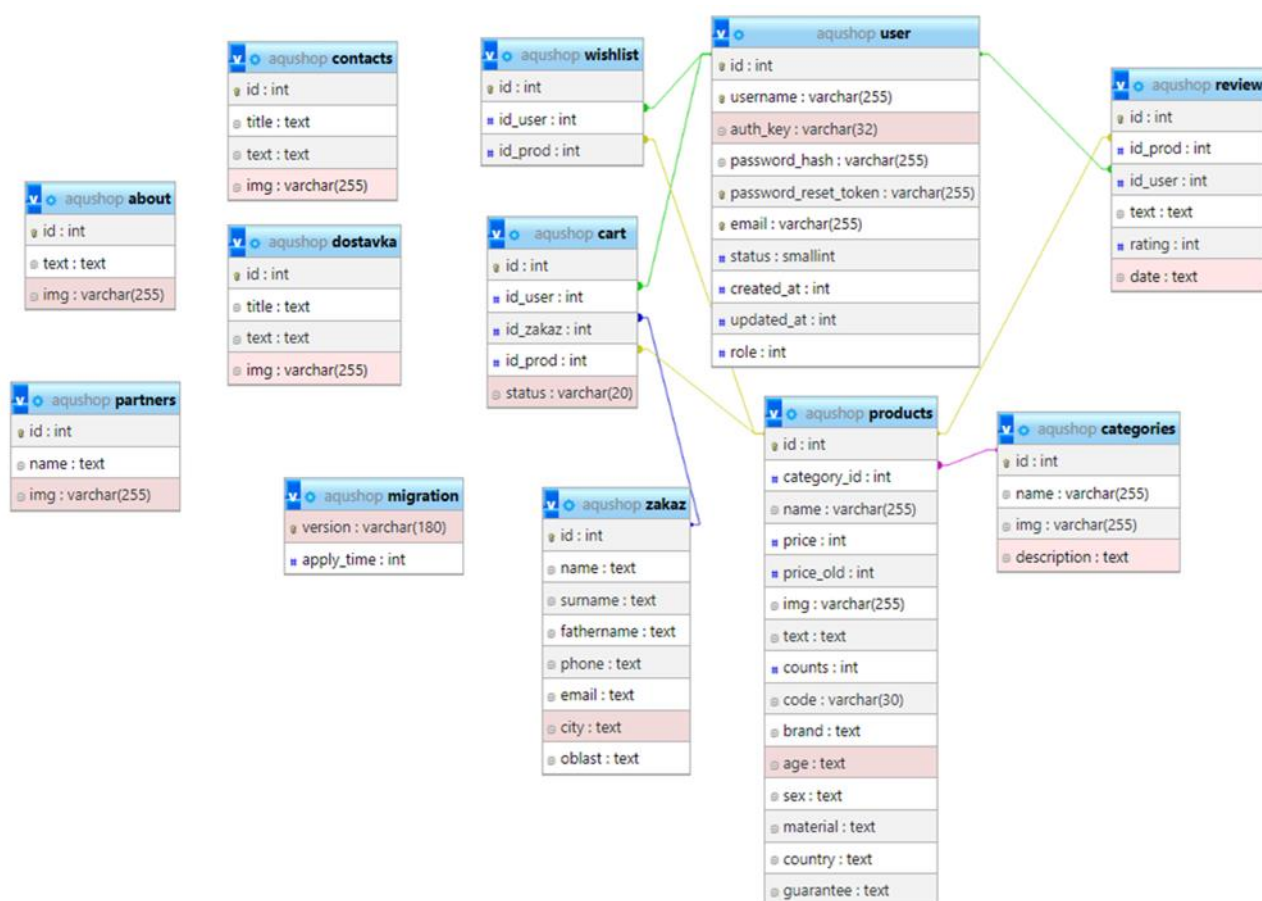


Рисунок 2.4 – ER-діаграма програмного додатку

Для збереження даних про відгуки на товар від користувачів було створено таблицю Review. Поля, їх типи та призначення наведено в табл. 2.9.

Таблиця 2.9

Review		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
id_prod	Number	Унікальний ідентифікатор продукту
id_user	Number	Унікальний ідентифікатор користувача
text	Text	Текст відгуку
rating	Number	Оцінка
date	Text	Дата

Для збереження даних про категорії товарів було створено таблицю Categories. Поля, їх типи та призначення наведено в табл. 2.10.

Таблиця 2.10

Categories		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
name	String	Назва
img	String	Картинка
description	Text	Опис категорії

Для збереження даних про товари було створено таблицю Products. Поля, їх типи та призначення наведено в табл. 2.11.

Таблиця 2.11

Products		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
category_id	Number	Унікальний ідентифікатор категорії
name	String	Назва
price	Number	Ціна
old_price	Number	Стара ціна
img	String	Картинка
text	Text	Опис
counts	Number	Кількість товару
code	String	Артикул
brand	Text	Бренд
age	Text	Вік
sex	Text	Стать
material	Text	Матеріал
country	Text	Країна виробник
guarantee	Text	Гарантія

Для збереження даних про замовлення було створено таблицю Zakaz. Поля, їх типи та призначення наведено в табл. 2.12.

Таблиця 2.12

Zakaz		
Поле	Тип	Призначення
1	2	3
id	Number	Унікальний ідентифікатор

Продовження табл. 2.12

name	Text	Ім'я
1	2	3
surname	Text	Прізвище
fathurname	Text	По-батькові
phone	Text	Телефон
email	Text	Електронна пошта
city	Text	Місто
oblast	Text	Область

Для збереження даних про кошик було створено таблицю Cart. Поля, їх типи та призначення наведено в табл. 2.13.

Таблиця 2.13

Cart		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
id_user	Number	Унікальний ідентифікатор користувача
id_zakaz	Number	Унікальний ідентифікатор замовлення
id_prod	Number	Унікальний ідентифікатор продукту
status	String	Статус

Для збереження даних про список обраного було створено таблицю Wishlist. Поля, їх типи та призначення наведено в табл. 2.14.

Таблиця 2.14

Wishlist		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
id_user	Number	Унікальний ідентифікатор користувача
id_prod	Number	Унікальний ідентифікатор продукту

Для збереження даних про сторінку “Доставка та оплата” було створено таблицю Dostavka. Поля, їх типи та призначення наведено в табл. 2.15.

Таблиця 2.15

Dostavka		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
title	Text	Заголовок
text	Text	Опис
img	String	Картинка

Для збереження даних про сторінку “Контакти” було створено таблицю Contacts. Поля, їх типи та призначення наведено в табл. 2.16.

Таблиця 2.16

Contacts		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
title	Text	Заголовок
text	Text	Опис
img	String	Картинка

Для збереження даних про сторінку “Про нас” було створено таблицю About. Поля, їх типи та призначення наведено в табл. 2.17.

Таблиця 2.17

About		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
text	Text	Опис
img	String	Картинка

Для збереження даних про партнерів нашого інтернет-магазину було створено таблицю Partners. Поля, їх типи та призначення наведено в табл. 2.18.

Таблиця 2.18

Partners		
Поле	Тип	Призначення
id	Number	Унікальний ідентифікатор
name	String	Назва
img	String	Картинка

2.3 Проектування та реалізація алгоритмів роботи системи

Перед тим як користувач матиме можливість додавати товари в кошик і список бажаного, залишати відгуки до товарів, робити замовлення, йому необхідно пройти процес реєстрації в інтернет-магазині та авторизуватися після успішної реєстрації. Тому, алгоритм реєстрації та авторизації нового користувача показаний на рис.2.5.

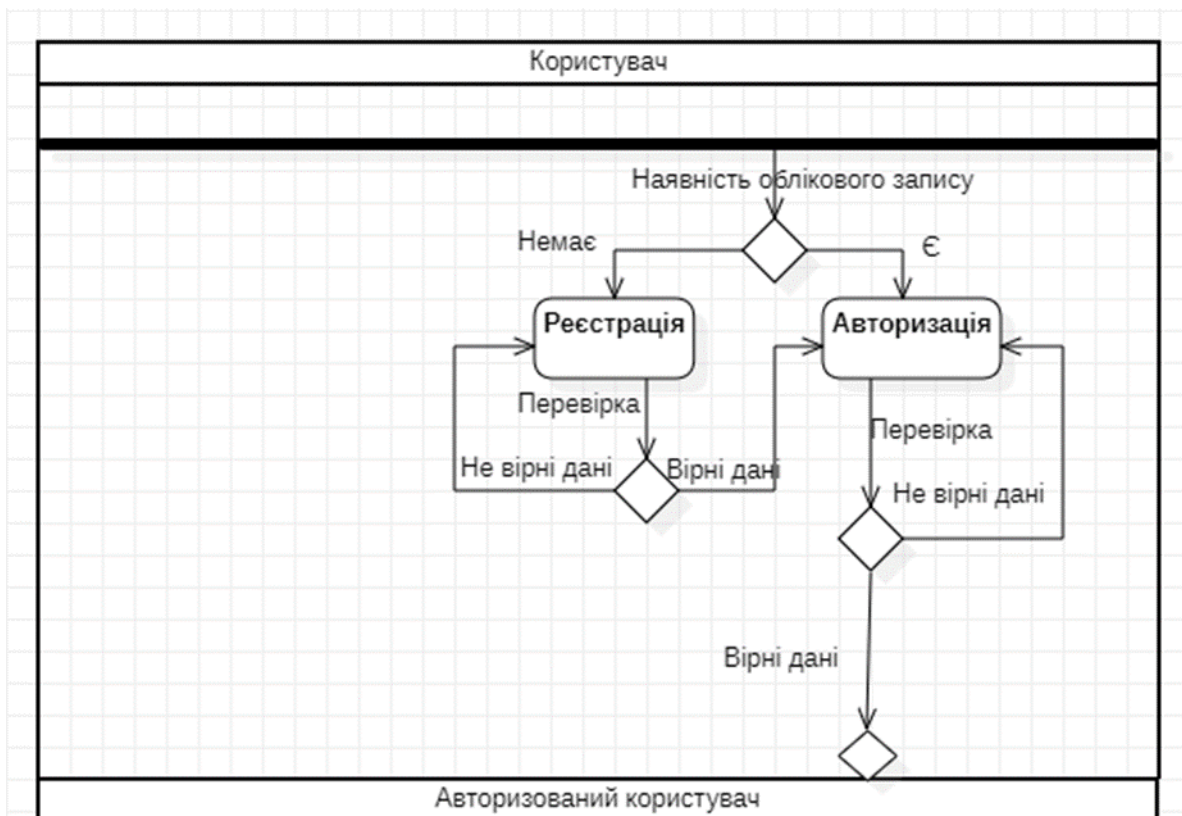


Рисунок 2.5 – Алгоритм реєстрації та авторизації нового користувача

Тому, алгоритм реєстрації та авторизації нового користувача складається з таких кроків:

- йде перевірка на наявність облікового запису, якщо його немає, то користувач повинен пройти процес реєстрації, а якщо обліковий запис наявний то користувач має пройти процес авторизації до системи;
- під час реєстрації перевіряються дані ті, що ввів користувач, якщо вони проходять процес перевірки то користувач успішно проходить реєстрацію, якщо ж не проходить процес перевірки, то йому потрібно перевірити коректність написаних даних;
- так, само і під час авторизації, якщо користувач вводить правильні дані від свого акаунту, то він переходить на головну сторінку інтернет-магазину та стає авторизованим користувачем, якщо ж виникає помилка, то він повинен перевірити введені дані.

Отже, після успішної авторизації, користувач має можливість замовляти товари, додавати їх до кошика, списку обраного та залишати відгуки до товарів.

2.4 Реалізація інтернет-магазину дитячих іграшок

Під час реалізації інтернет-магазину дитячих іграшок було використано архітектурний підхід MVC, який складається з контролерів, моделей, та представлень.

Моделі відповідають за валідацію даних, взаємодіють з базою даних. Представлення відповідають за відображення даних з моделей та взаємодію з користувачами, представлення отримує дані з моделі і відображає їх у відповідному форматі. Контролери виконують посередницьку функцію між моделями і представленнями. Вони обробляють вхідні дані від користувачів, ініціюють виконання дій та забезпечують взаємодію між моделями та представленнями.

На рис. 2.6-2.7 буде наведено логіку процесу реєстрації та авторизації нових користувачів.

```
public function actionSignup()
{
    $model = new SignupForm();

    if ($model->load(Yii::$app->request->post())) {
        if ($user = $model->signup()) {
            if (Yii::$app->getUser()->login($user)) {
                Yii::$app->session->setFlash( key: 'success', value: 'Ви успішно зареєстрували свій новий аккаунт!');
                return $this->goHome();
            }
        }
    }

    return $this->render( view: 'signup', [
        'model' => $model,
    ]);
}
```

Рисунок 2.6 – Процес реєстрації користувача

```

public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {
        Yii::$app->session->setFlash( key: 'success', value: 'Ви успішно увійшли у свій аккаунт!');
        return $this->goBack();
    }

    $model->password = '';
    return $this->render( view: 'login', [
        'model' => $model,
    ]);
}

```

Рисунок 2.7 – Вхід користувача до свого облікового запису

Також, можна подивитися на реалізацію кошика за допомогою сесій, та на процес замовлення товару і відправку повідомлення адміністратору на електронну пошту, про те, що користувач зробив замовлення.

На рис. 2.8-2.9 наведено логіку процесу додавання товарів до кошика з використанням сесій.

```

$productsSession = $session->get( key: 'productsSession');

$arrayID = array();

foreach ($productsSession as $value) {
    array_push( &$array: $arrayID, $value['id']);
}

$products = Products::find()->where(['id' => $arrayID])->asArray()->All();

foreach ($products as $key => $value) {
    $products[$key]['count_cart'] = $productsSession[$key]['count'];
}

return $this->render( view: 'cart', compact( var_names: 'products'));

```

Рисунок 2.8 – Додавання товарів до сесії для реалізації кошика (1 фрагмент)

```

public function actionCart()
{
    $session = Yii::$app->session;
    // $session->destroy();
    $session->open();

    if ($session->has( key: 'productsSession')) {
        $productsSession = $session->get( key: 'productsSession');
    } else {
        $productsSession = array();
    }

    if (isset($_GET['id']) && !empty($_GET['id']) && filter_var($_GET['id'], filter: FILTER_VALIDATE_INT)) {
        $productsArray = Products::find()->where(['id' => $_GET['id']])->asArray()->one();
        Yii::$app->session->setFlash( key: 'success', value: 'Товар успішно доданий до кошика!');
        if (is_array($productsArray) && count($productsArray) > 0) {

            $flag = false;
            for ($i = 0; $i < count($productsSession); $i++) {
                if ($productsSession[$i]['id'] == $_GET['id']) {

                    $flag = true;
                    if ($productsArray['counts'] >= $productsSession[$i]['count'] + 1) {

                        $productsSession[$i]['count']++;
                    }
                    break;
                }
            }

            if (!$flag) {
                array_push( &array: $productsSession, ['id' => $_GET['id'], 'count' => 1]);
            }
        }
    }

    $session->set('productsSession', $productsSession);
}

```

Рисунок 2.9 – Додавання товарів до сесії для реалізації кошика (2 фрагмент)

На рис. 2.10 наведено логіку процесу створення нового замовлення та відправки повідомлення на електронну пошту про успішне замовлення користувачем.

На рис. 2.11 наведено логіку процесу створення нового відгуку від користувача на вибраний ним товар.

```

$model = new Zakaz();

if ($this->request->isPost) {
    if ($model->load($this->request->post()) && $model->save()) {
        $prod = [];

        foreach ($_SESSION['productsSession'] as $item)
        {
            $product = Products::findOne($item['id']);
            $prod[] = ['name' => $product->name, 'price' => $product->price, 'username' => Yii::$app->user->identity->username];
            $cart = new Cart();
            $cart->id_user = Yii::$app->user->getId();
            $cart->id_zakaz = $model->id;
            $cart->id_prod = $item['id'];
            $cart->save();
        }

        //smtp email setting
        $email = Yii::$app->mailer->compose([view: '@app/mail/zakazonemail', ['foots' => $prod]]);
        $email->setFrom(['from@gmail.com' => 'AQUATOYS']);
        $email->setTo(['to' => 'garriq.kravchenko@gmail.com']);
        $email->setSubject(['subject' => 'Замовлення від користувача']);
        $email->setTextBody(['text' => 'Plain text content']);
        $email->send();

        $session = Yii::$app->session;
        $session->destroy();
        Yii::$app->session->setFlash(['key' => 'success', 'value' => 'Ваше замовлення було успішно опрацьовано! Очікуйте на дзвінок від нашого працівника для перевірки та уточнення замовлення']);
        return $this->redirect(['/']);
    }
} else {
    $model->loadDefaultValues();
}

return $this->render([view: 'zakaz', [
    'model' => $model,
]]);

```

Рисунок 2.10 – Створення замовлення та відправка повідомлення на пошту

```

public function actionReview($id)
{
    $model = Review::findOne($id);

    $product = Products::findOne($id);
    $reviews = Review::find()->where(['id_prod' => $id])->all();
    $reviewForm = new ReviewForm();

    return $this->render([view: 'review', [
        'model' => $model,
        'product' => $product,
        'reviews' => $reviews,
        'reviewForm' => $reviewForm,
    ]]);
}

public function actionReviews($id)
{
    $model = new ReviewForm();

    if (Yii::$app->request->isPost)
    {
        $model->load(Yii::$app->request->post());
        Yii::$app->session->setFlash(['key' => 'success', 'value' => 'Ваш відгук було успішно додано!']);
        if ($model->saveReview($id))
        {
            return $this->redirect(['products/one-product', 'id' => $id]);
        }
    }
}

```

Рисунок 2.11 – Створення відгуку про товар

Висновки до другого розділу

Під час виконання проектування інтернет-магазину дитячих іграшок були проаналізовані варіанти використання програмного продукту відповідно до вимог програмного забезпечення. Наприклад, реєстрація, авторизація нових користувачів, додавання відгуків про товар для кожного користувача, додавання

товарів в корзину та список обраного, замовлення товарів, перегляд товарів та категорій.

З метою опису взаємодії компонентів системи було описано алгоритм реєстрації та авторизації нових користувачів.

Також було наведено описи реалізації головних функціональних елементів інтернет-магазину, таких як, реєстрації та авторизація користувачів, створення кошика товарів за допомогою сесій та створення замовлень товарів з відправкою повідомлень на електронну пошту адміністратора про вдале замовлення.

РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНТЕРНЕТ-МАГАЗИНОМ ДИТЯЧИХ ІГРАШОК

3.1 Порядок встановлення та налаштування параметрів системи.

Встановлення та налаштування параметрів системи вимагає встановлення серверного середовища, розгортання серверної частини і клієнтської, а також налаштування зв'язків між компонентами системи.

Тому для показу архітектури програми та розгортання було зроблено діаграму розгортання та показано на рис. 3.1.

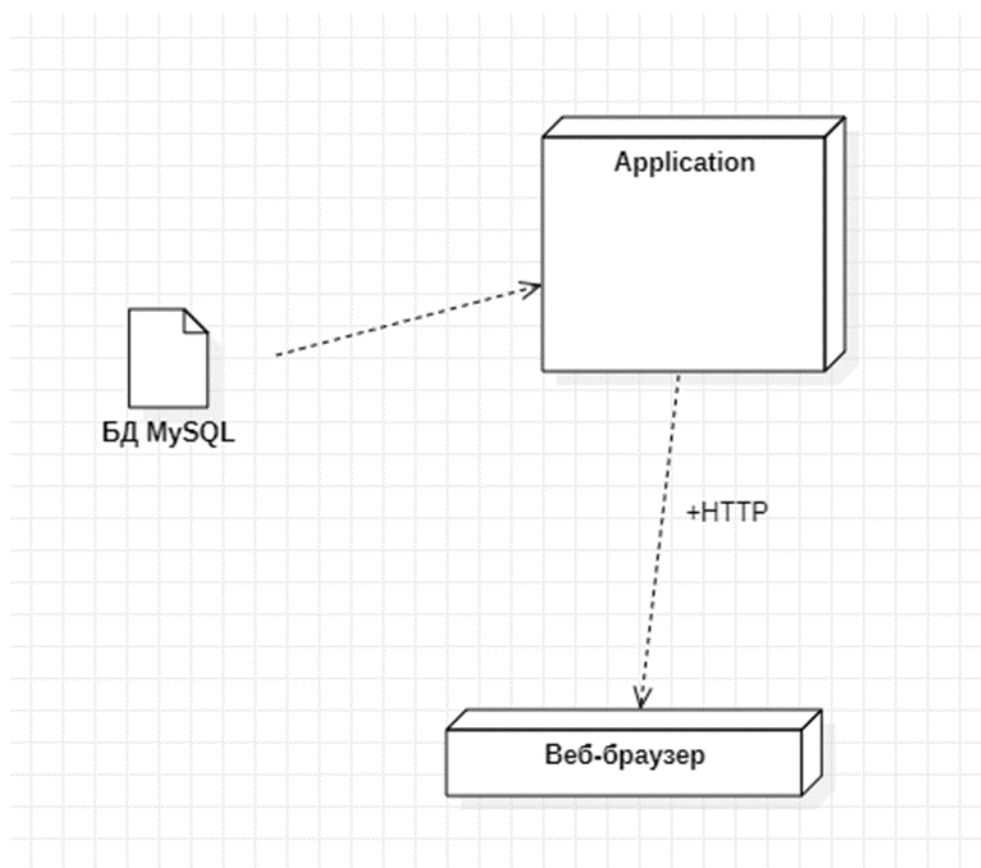


Рисунок 3.1 – Діаграма розгортання

Діаграму розгортання можна описати так, що наш додаток звертається до бази даних, щоб отримати дані, отримує їх із бази та після чого завантажує дані на сторінку веб-браузера з HTML розміткою.

3.2 Структура інтерфейсу та порядок роботи з інтернет-магазином

При вході на сайт людина попадає на головну сторінку, де може переглянути окремі товари з нашого магазину та ознайомитись з переліком наших партнерів, які співпрацюють з нами.

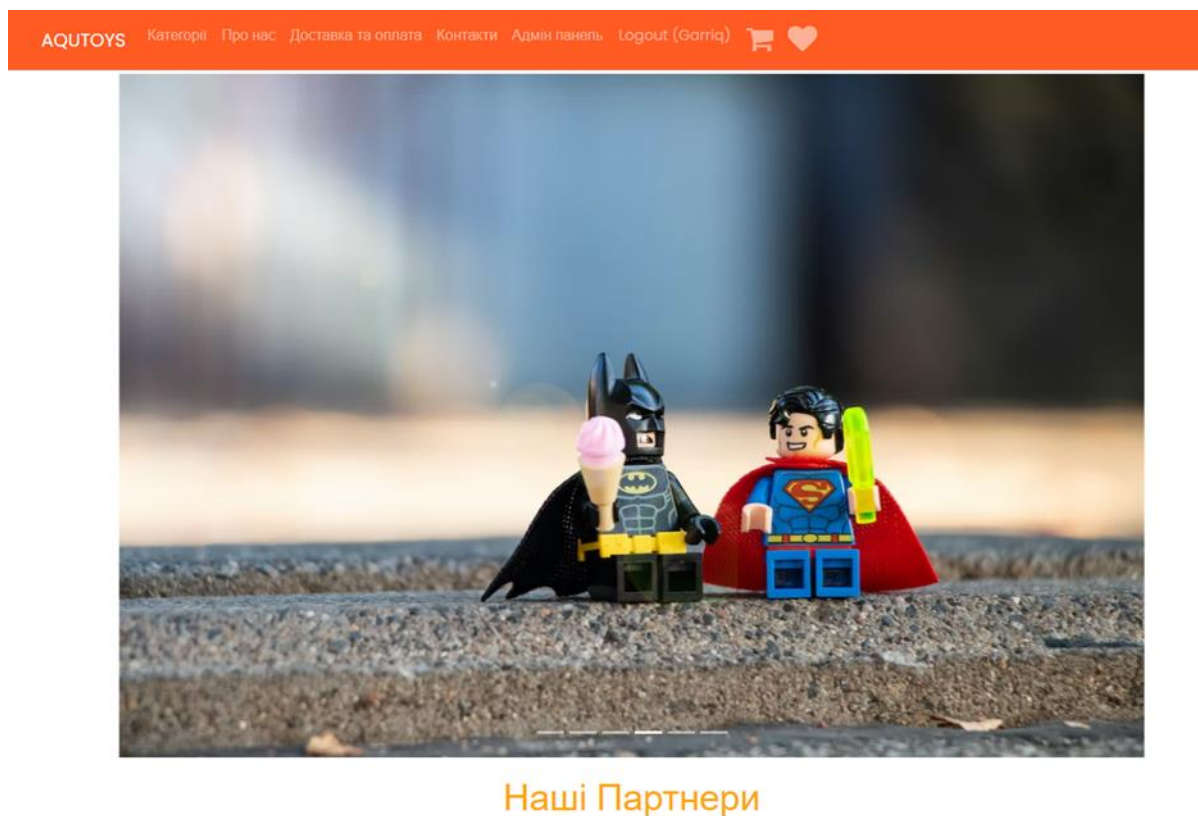


Рисунок 3.2 – Головна сторінка (1 фрагмент)

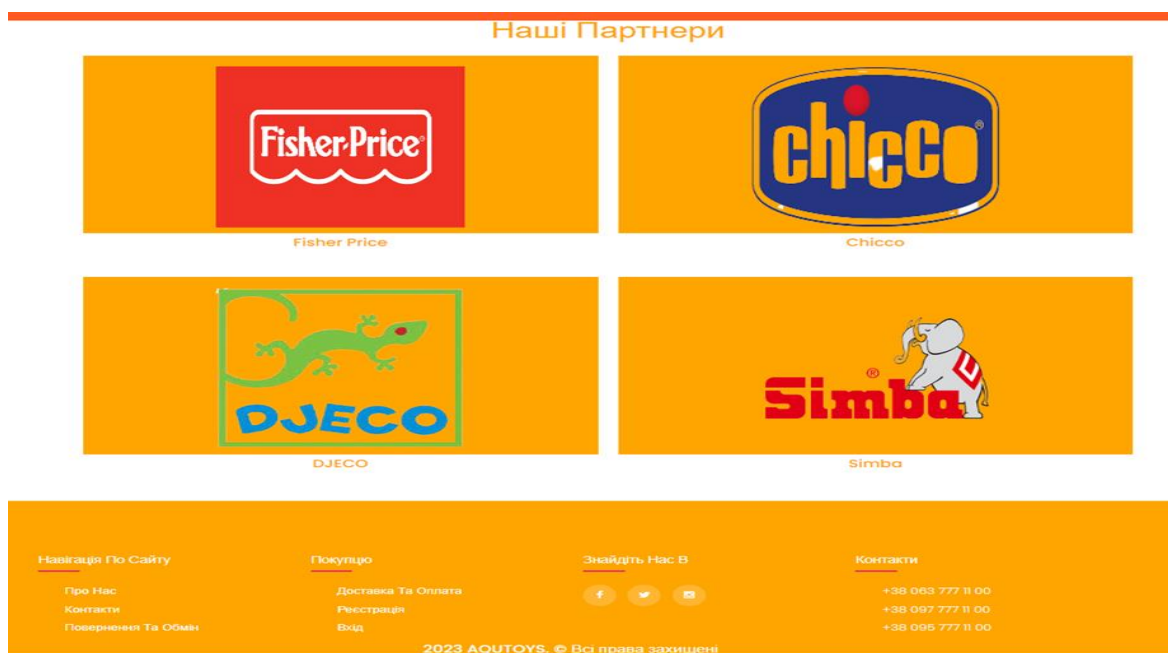


Рисунок 3.3 – Головна сторінка (2 фрагмент)

Після цього, людина може зареєструвати свій новий акаунт, для цього потрібно перейти на сторінку “Реєстрація”.

AQUATOYS Категорії Про нас Доставка та оплата Контакти Регістрація Вхід

[Головна](#) / Регістрація

Регістрація

Будь ласка, заповніть наступні поля для реєстрації:

Ім'я користувача

Електронна пошта

Пароль

Зареєструватися

Рисунок 3.4 – Сторінка “Регістрація”

Для того, щоб пройти реєстрацію потрібно ввести ім'я користувача, електронну пошту та пароль. Пароль повинен бути більше 6 символів, також якщо ім'я користувача та електронна пошта будуть вже зареєстрованими на сайті, то вам потрібно буде ввести інші дані.

[Головна](#) / Регістрація

Регістрація

Будь ласка, заповніть наступні поля для реєстрації:

Ім'я користувача

Це ім'я користувача вже використано.

Електронна пошта

Ця електронна адреса вже зайнята.

Пароль

Зареєструватися

Рисунок 3.5 – Введені дані не пройшли валідацію

Після того як були введені валідні значення, користувача було успішно зареєстровано, і автоматично відбувся вхід користувача на сайт.

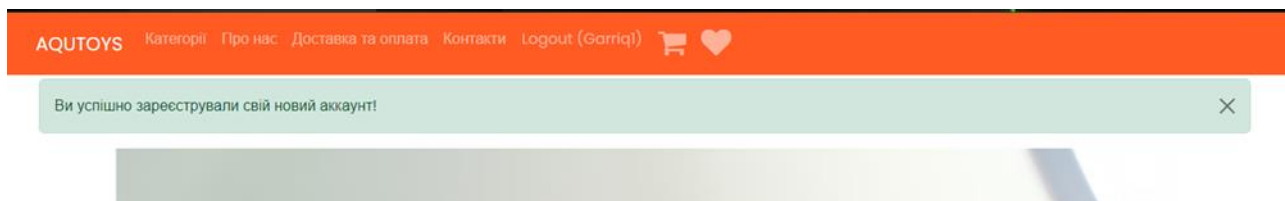


Рисунок 3.6 – Успішна реєстрація користувача

Авторизований користувач на відміну від неавторизованого має можливість додавати товари в кошик та в список бажаного.

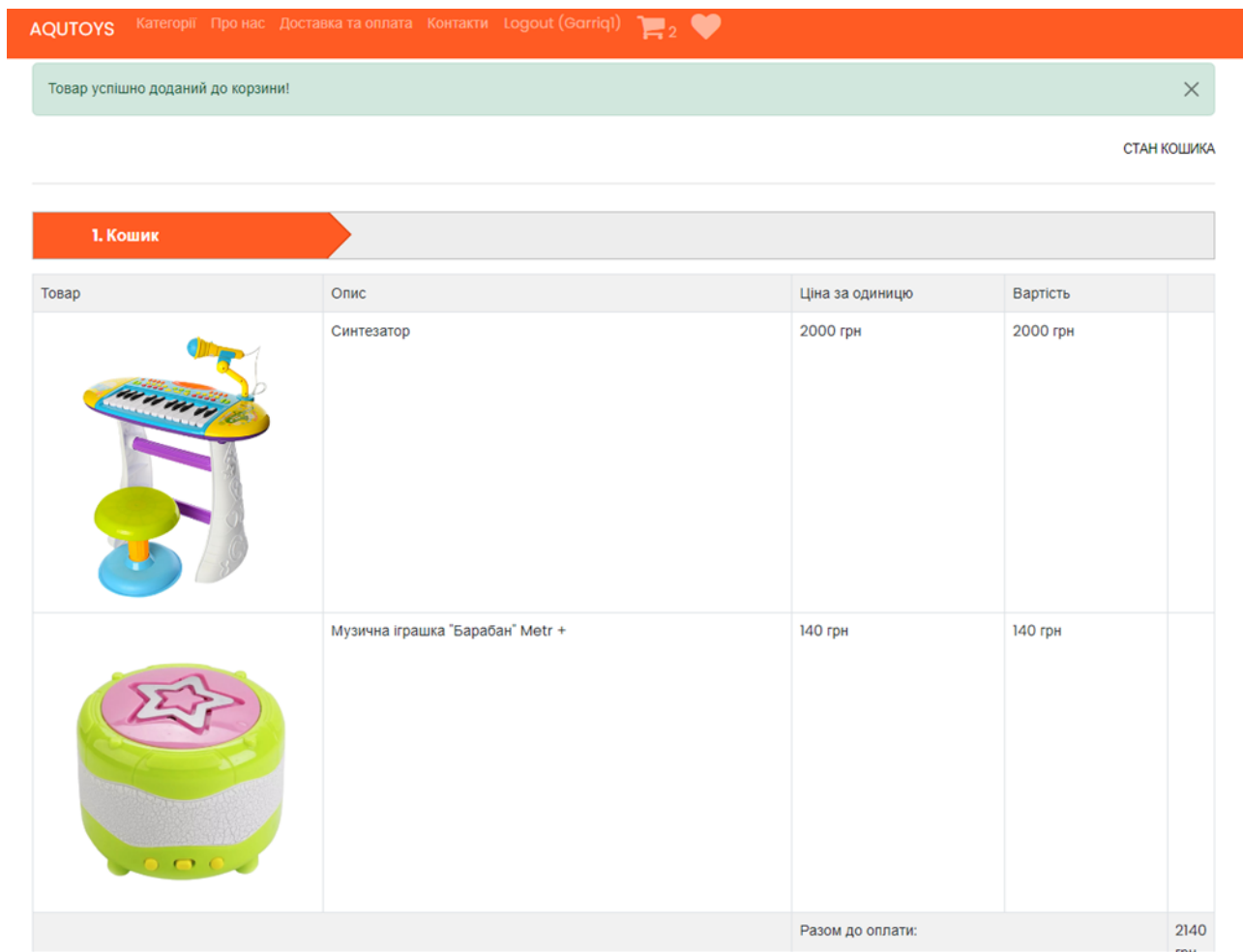


Рисунок 3.7 – Успішне додавання товарів до кошика

Також, є можливість видалити всі товари з кошика.

1. Кошик

В кошику немає товарів

Рисунок 3.8 – Успішне видалення товарів з кошика

1. Улюблене

Товар	Назва	Артикул	Ціна за одиницю	Видалити
	Синтезатор	BB3B3BD	2000 грн	
	Дартс	1201	181 грн	

Рисунок 3.9 – Успішне додавання товарів до списку обраного

Покупець матиме змогу видалити окремий товар або ж всі товари зі списку обраного.

Товар був успішно видалений зі списку обраного!

СТАН ОБРАНОГО

Ваш список улюбленого: 2 товар

1. Улюблене

Товар	Назва	Артикул	Ціна за одиницю	Видалити
	Художній комплект для малювання "Балерина"	DJ09381	799 грн	
	Гітара іграшкова Fan Wingda Toys	819-20	175 грн	

Рисунок 3.10 – Успішне видалення окремого товару зі списку обраного

AQUOTOYS Категорії Про нас Доставка та оплата Контакти Адмін панель Logout (Garriq)  

Список обраного був успішно очищений!

СТАН ОБРАНОГО

Ваш список улюбленого: 0 товар

1. Улюблене

В обраному немає товарів

Рисунок 3.11 – Успішне видалення всіх товарів зі списку обраного

Користувач може переглянути сторінки “Про нас”, “Доставка та оплата”, “Контакти”, “Повернення та обмін”.

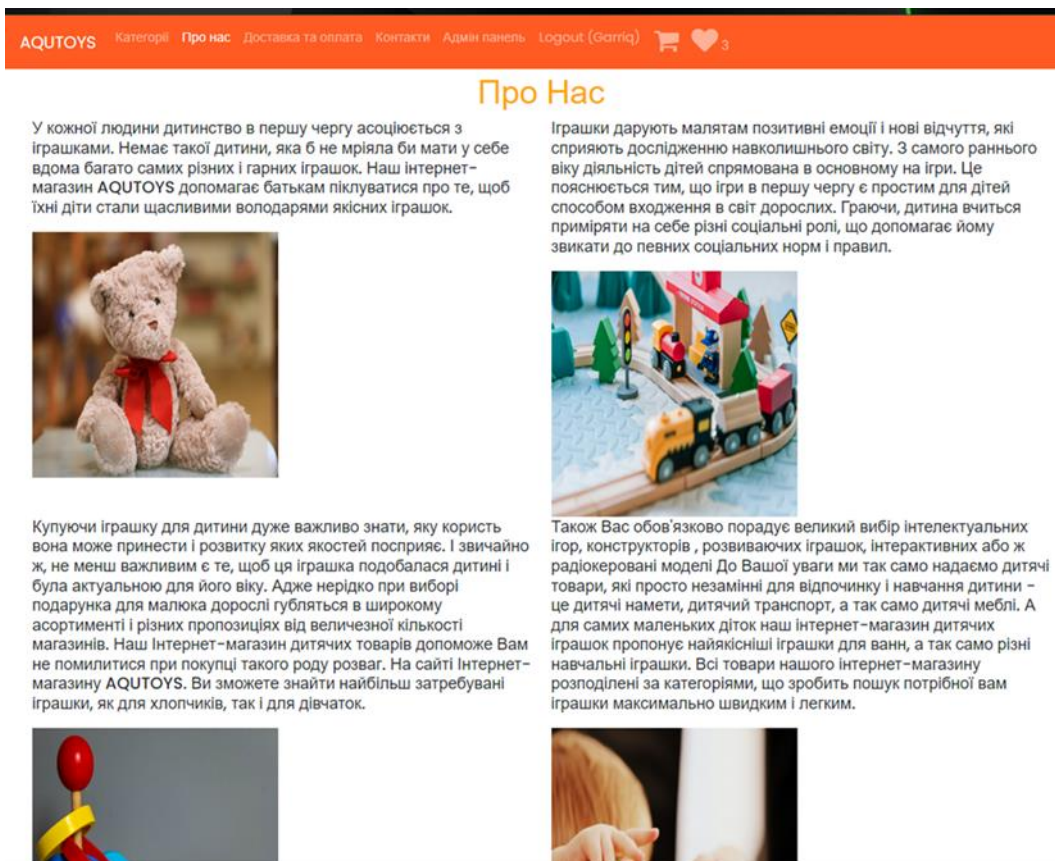


Рисунок 3.12 – Сторінка “Про нас”

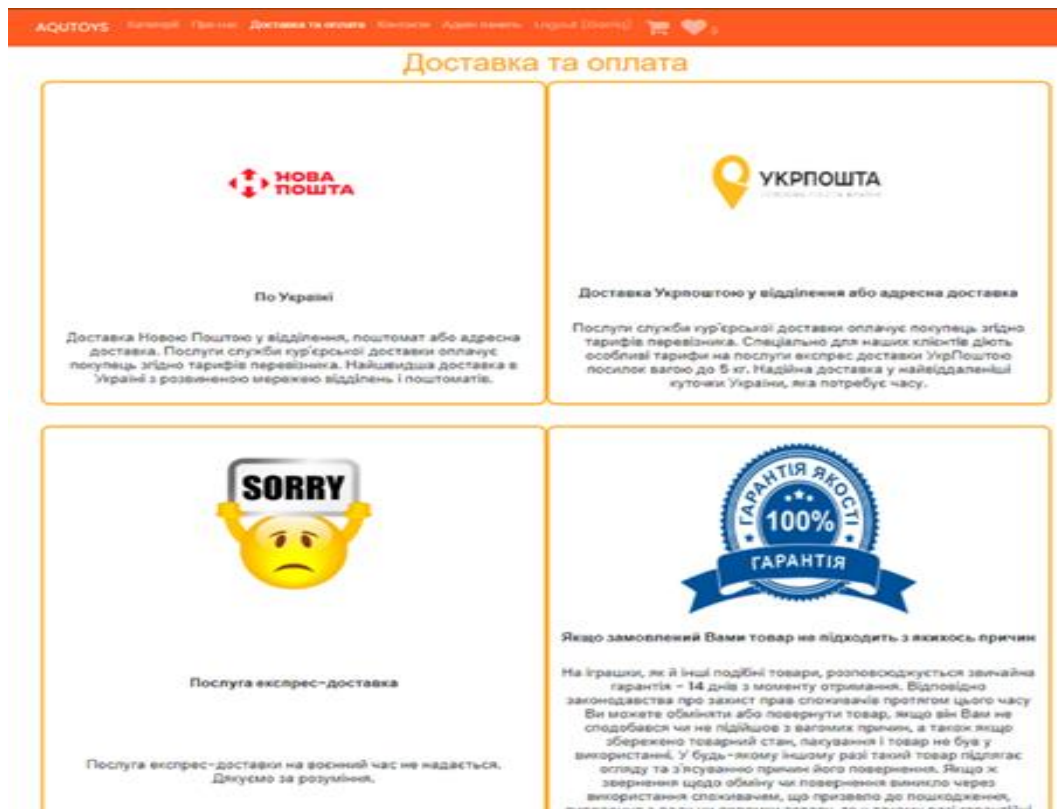


Рисунок 3.13 – Сторінка “Доставка та оплата”

Контакти

Графік роботи

пн-пт: 09:00-19:00

сб-нд: 10:00-17:00



Пошта для зв'язку, пропозицій та відгуків

agutoys@gmail.com



Телефони колл-центру

+38 063 777 11 00

+38 097 777 11 00

+38 095 777 11 00



Наша адреса

Україна, Житомир, вулиця
Київська, 74. 10006



Рисунок 3.14 – Сторінка “Контакти”

Повернення та обмін

Політика повернення та обміну товару

Весь наш асортимент товарів сертифікований в Україні.
Відповідно до закону України, ви можете зробити повернення
або обмін товару з моменту отримання протягом 14 днів.

Головною умовою для цього передбаченим законом, це: товар має бути неушкодженим, та належного вигляду (за винятком м'яких іграшок та надувних дитячих виробів, вони поверненню та обміну не підлягають, відповідно до того ж закону України).

Він не був в експлуатації, зовнішній вигляд іграшки та пакування повинен відповідати тому, який ви її отримали.

У Вас відповідно має бути товарний чек, що підтверджує покупку.
Тоді, ми зі свого боку, відшкодуємо вартість Товару.

(Вартість логістичних компаній та кур'єрів не можна відшкодувати, так як Ви ними скористалися (послуга надана), і якщо прийняли товар, значить, у Вас немає претензій до них.)

Для того, щоб повернути товар, ви можете скористатися двома варіантами.

Для повернення товару Новою Поштою необхідно надіслати товар за реквізитами:

Одержувач: Сидоров Олег +380680968075

Вид доставки: Склад-Склад

Адреса: м. Житомир, 71 відділення

Послуги "Зворотна доставка" та "Накладений платіж" необхідно скасувати

Оплата доставки – за рахунок відправника

Окремо виділимо процес повернення товару Україною шляхом відправки за допомогою сервісу «Нова Пошта».

ШАНОВНІ КЛІЄНТИ!

Завжди перевіряйте той товар, який ви отримуєте
безпосередньо На Складі Нової Пошти, перед тим, як його
забрати!

Адже бувас один випадок на тисячу, коли трапляється заводський брак (ми його 100% заберемо назад, і витрати за логістику також беремо на себе), і також бувас 1 випадок на тисячу, коли при транспортванні товар був пошкоджений (весь товар застрахований, і в у цьому випадку, «Нова Пошта» компенсує збитки). Тому, перевіряйте товар при отриманні, тоді ми зможемо допомогти вирішити проблему.

В іншому випадку, якщо в іграшці виявиться дефект (незалежно від причини), ми не можемо вам допомогти.

Сподіваємося, що дана інформація допоможе вам уникнути негативного досвіду та збільшить довіру до нас.

У будь-якому випадку, телефонуйте до нашого інтернет

Рисунок 3.15 – Сторінка “Повернення та обмін”

Також користувач може переглянути всі категорії товарів, що є в нашому магазині.

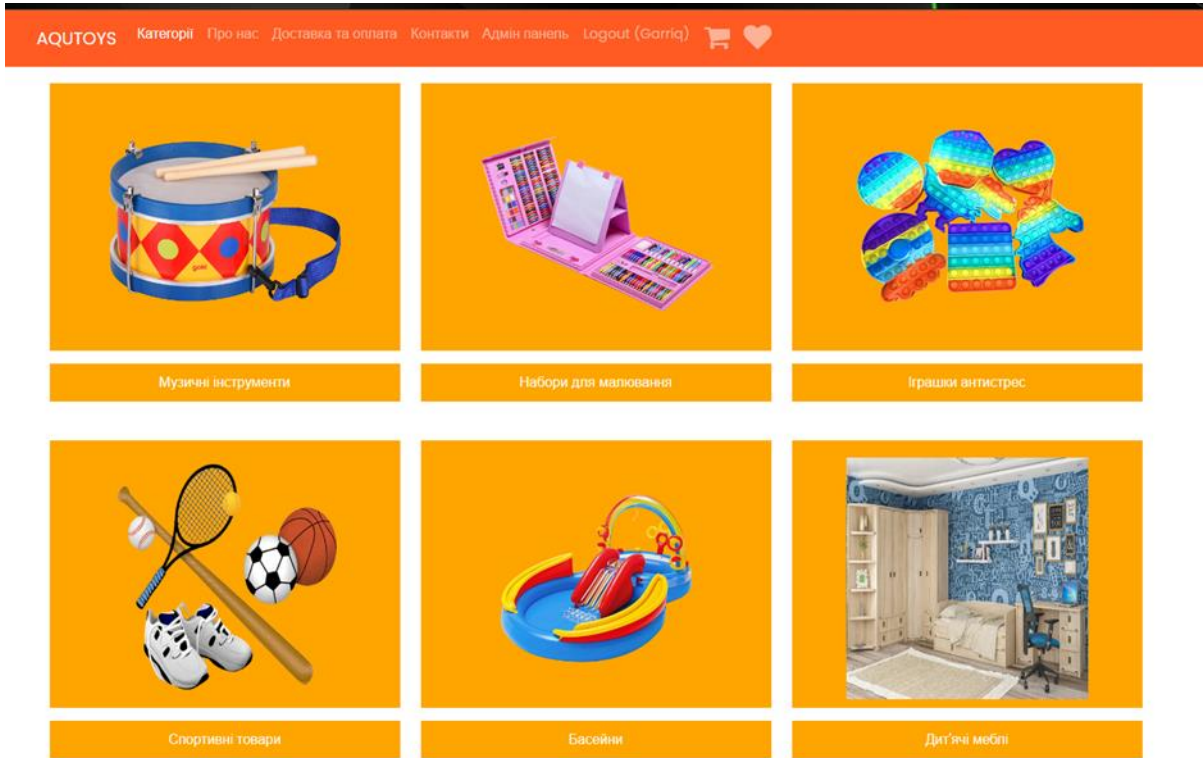


Рисунок 3.16 – Сторінка “Категорії товарів”

Після того, як користувач обрав категорію він переходить на всі товари, які є в обраній категорії товарів.

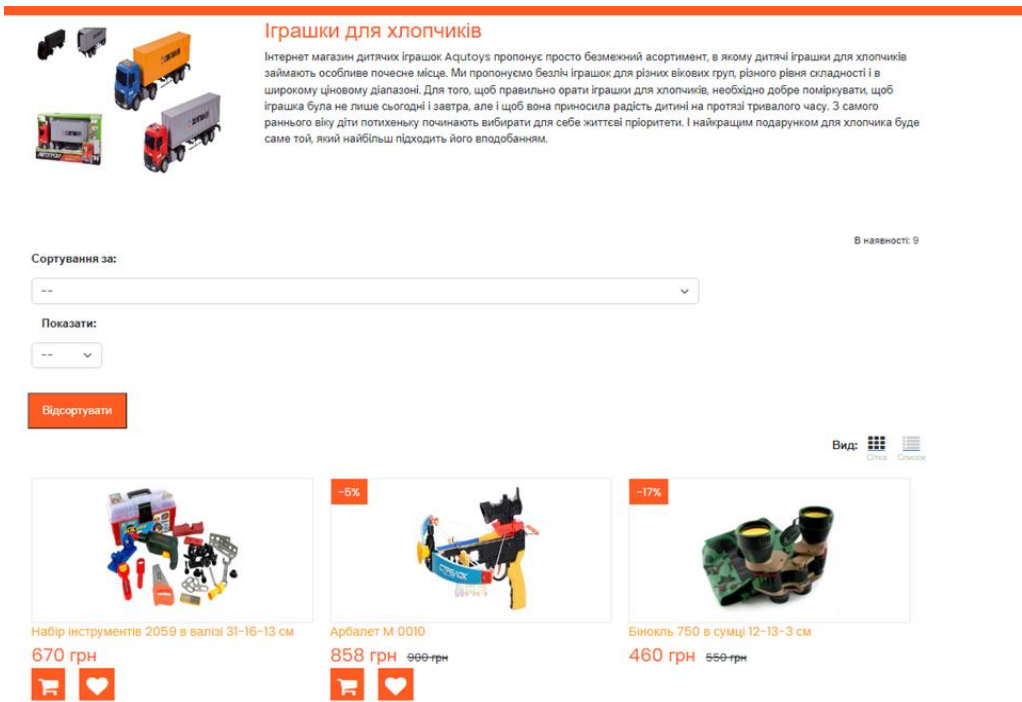


Рисунок 3.17 – Сторінка “Всіх товарів вибраної категорії”(1 фрагмент)



1 2

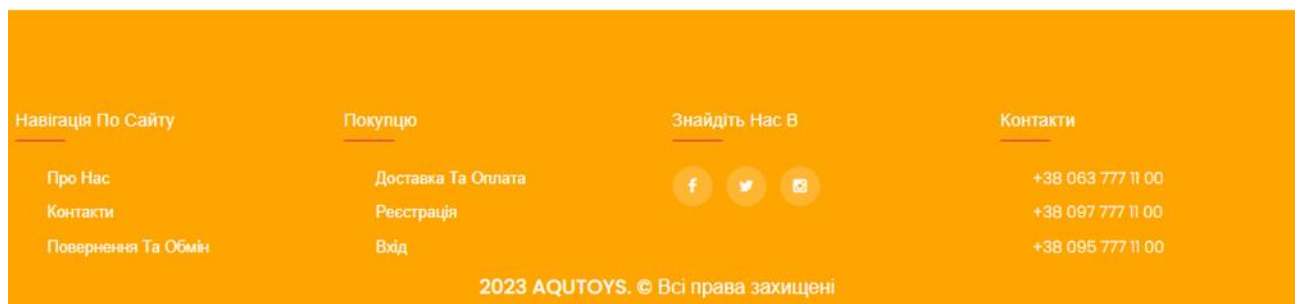


Рисунок 3.18 – Сторінка “Всіх товарів вибраної категорії”(2 фрагмент)

Користувач може змінити вигляд всіх товарів натиснувши кнопку “Список”.

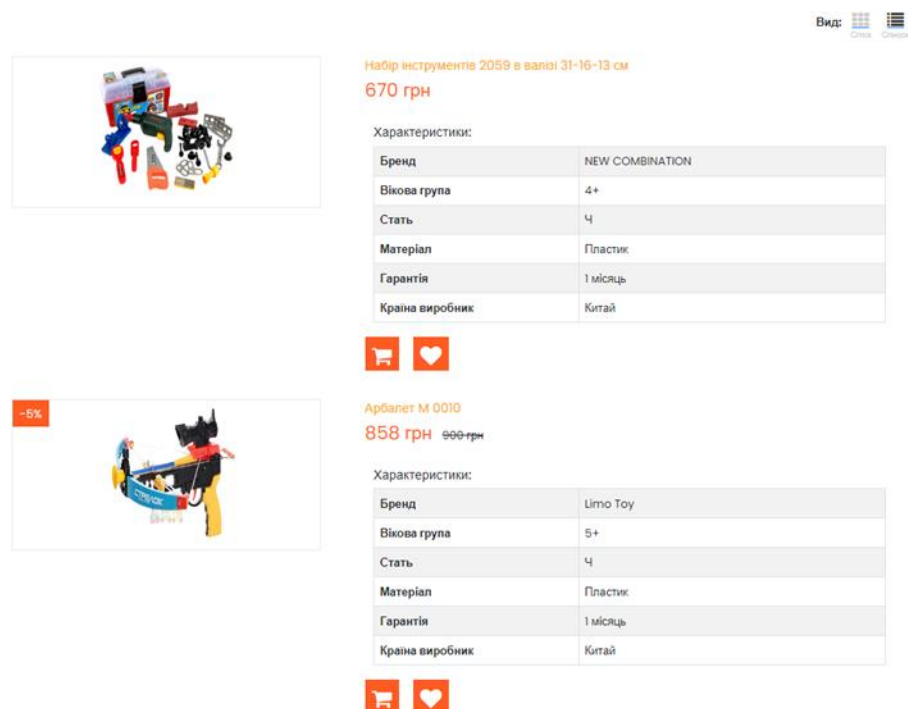


Рисунок 3.19 – Сторінка “Всіх товарів вибраної категорії”(3 фрагмент)

Далі, користувач може відсортувати товари по ціні та за алфавітом, також за замовчуванням на одній сторінці показано 6 товарів, тому користувач може вибрати, яку кількість товарів він хоче бачити на 1 сторінці.

В наявності: 9

Сортування за:

Ціною, по збільшенню ✓ ▾

Показати:

12 ✓ ▾

Відсортувати

Вид:

Grid List

Набір інструментів 2059 в валізі 31-16-13 см

Рисунок 3.20 – Сортування товарів та кількість виведених товарів

Користувач може перейти на сторінку окремого товару, де переглянути більш детальну інформацію про товар.

AQUOTOYS Категорії Про нас Доставка та оплата Контакти Адмін панель Logout (Gorriq)

Набір інструментів 2059 в валізі 31-16-13 см

Артикул: 2059

В наявності: 5

Опис: У Вашій родині підрастає мамин і татів помічник? Тоді такому майстру просто необхідні свої інструменти. Набір інструментів дитячий 2059 зможе втілити мрію малюка, і занурить його в татові турботи! Набір інструментів 2059 – це іграшка для ігор. Всі елементи акуратно розміщені в валізі. Малюк після вдалого ремонту зможе швидко розкласти все по місцях. Всі деталі – це аналог дорослого інструментарію. Це і ключі, і ножи́вка, і шурупверт, і рубанок і багато іншого.

670 грн

Кількість:

В наявності: 5

В корзину В улюблене

Характеристики:

Бренд	NEW COMBINATION
Вікова група	4+
Стать	Ч
Матеріал	Пластик
Гарантія	1 місяць
Країна виробник	Китай

Додати відгук

Відгуки до товару: Набір інструментів 2059 в валізі 31-16-13 см

Рисунок 3.21 – Сторінка окремого товару

Якщо, користувач хоче додати відгук про окремий товар, то він може зробити це натиснувши кнопку “Додати відгук”.

Залишіть свій відгук про товар!



Надіслати

Рисунок 3.22 – Сторінка написання відгуку

Після того, як користувач залишив свій відгук він може його побачити на сторінці товару до якого він залишив відгук.

Додати відгук

Відгуки до товару: Набір інструментів 2059 в валізі 31-16-13 см

Користувач: Garriq

Jun 8, 2023

Відгук: Чудовий товар!



Рисунок 3.23 – Залишений відгук

Далі, якщо користувач хоче щось замовити, він повинен додати товари до кошика, да натиснувши кнопку “Замовити”, він повинен буде заповнити інформацію для замовлення.

AQUTOYS Категорії Про нас Доставка та оплата Контакти Адмін панель Logout (Garriq)

Оформлення замовлення

Ім'я

Олександр

Прізвище

Кравченко

По-батькові

Іванович

+38 (063) 777-77-77

Електронна пошта

timyrshotov@gmail.com

Місто

Житомир

Область

Житомирська

Необхідно заповнити "Область".

Зберегти

Рисунок 3.24 – Заповнена форма оформлення замовлення



Рисунок 3.25 – Повідомлення про успішне замовлення

Після того, як користувач зробив замовлення, адміністратору на електронну пошту приходить повідомлення про те, що користувач зробив замовлення.



AQUOTOYS <garriq.kravchenko@gmail.com>
кому мені ▼

Користувач, Garriq, зробив замовлення!

Назва товару	Ціна за одиницю
Синтезатор	2000 грн
Музична іграшка "Барабан" Metr +	140 грн
Гітара іграшкова Fan Wingda Toys	175 грн

Рисунок 3.26 – Перегляд повідомлення про замовлення

Якщо, користувач при вході на сайт не пам’ятає свій пароль, він може скористатися функцією скидання паролю.

Для цього він повинен ввести свою пошту для отримання подальшої інформації.

[Головна](#) / [Запит на скидання пароля](#)

Запит на скидання пароля

Будь ласка, заповніть свою електронну адресу. Туди буде надіслано посилання для скидання пароля.

Електронна пошта

Надіслати

Рисунок 3.27 – Посилання для зміни паролю на пошті

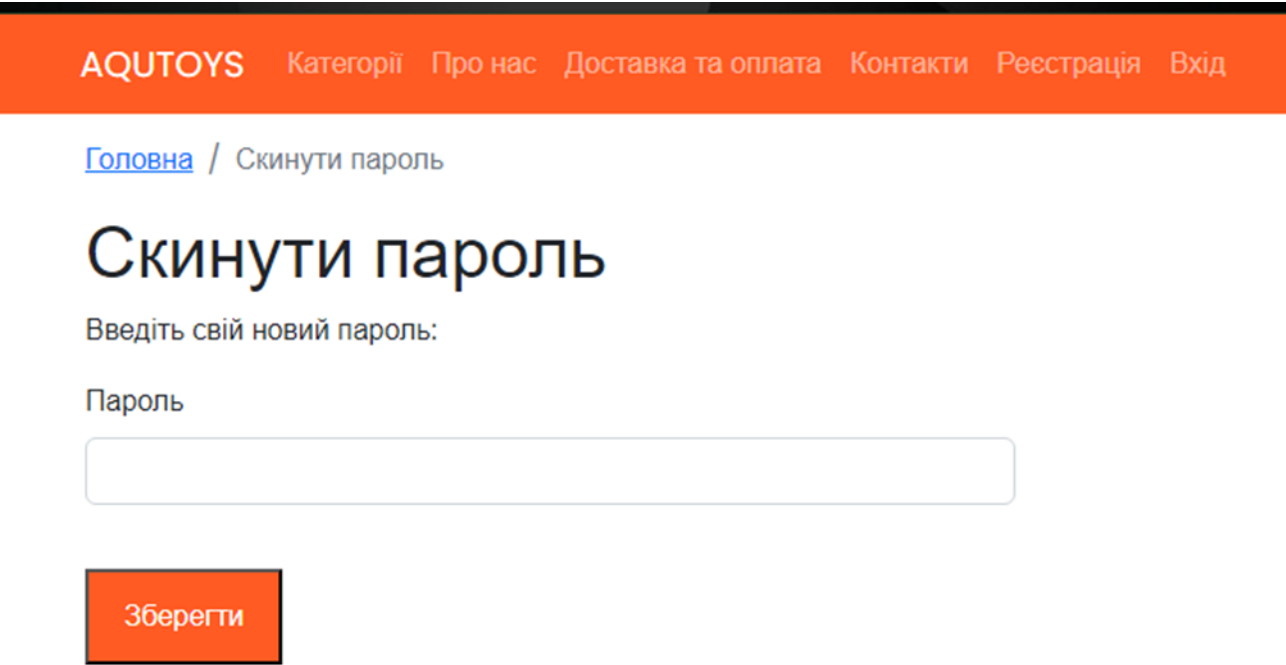


Рисунок 3.28 – Сторінка зі змінною пароля

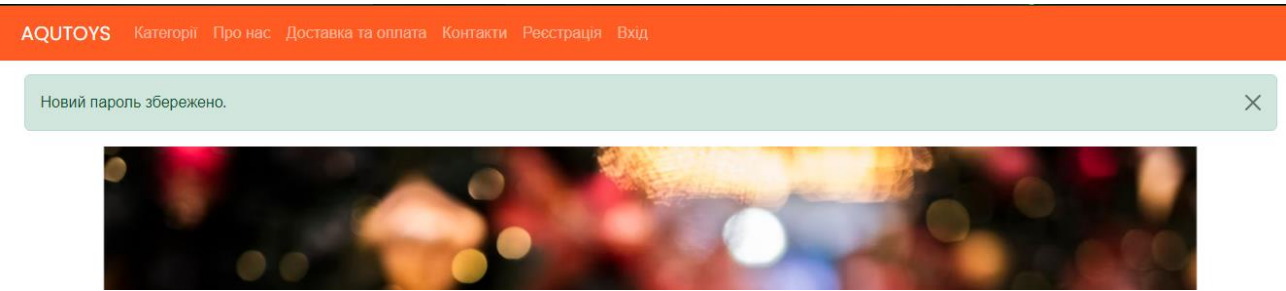


Рисунок 3.29 – Повідомлення про успішну зміну пароля

На сам кінець, залишилася панель адміністратора. Якщо користувач, що зайшов на сайт, має права адміністратора, то він має доступ до панелі адміністратора, щоб забезпечувати працездатність інтернет-магазину.

Адмін-панель

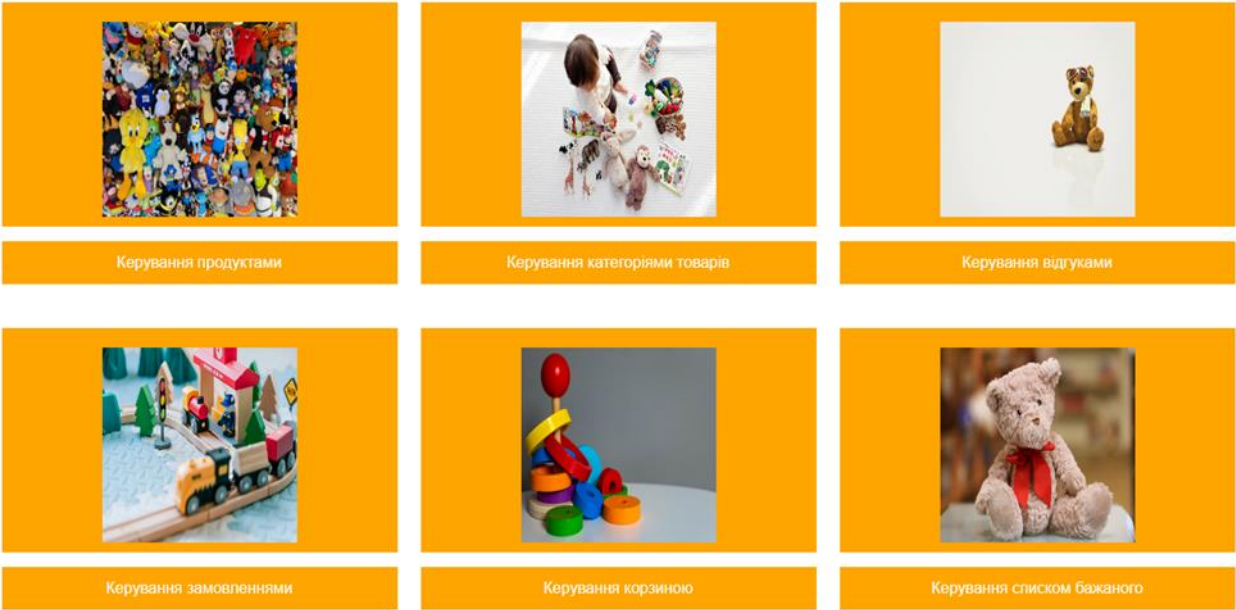


Рисунок 3.30 – Сторінка “Панель адміністратора”

Також на прикладі “Керування категоріями товарів” подивимося функціонал адміністратора.

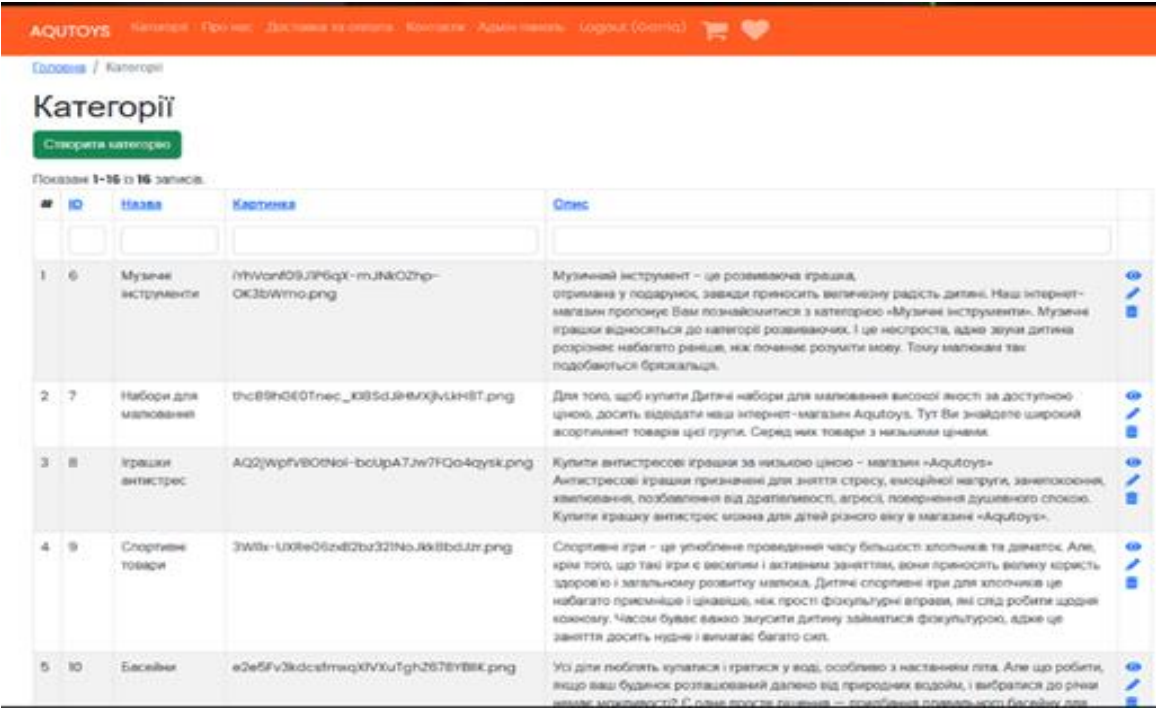


Рисунок 3.31 – Список категорій товарів

Адміністратор має можливість створювати нові категорії, видаляти їх, переглядати та редагувати.

Музичні інструменти

Оновити Видалити

ID	6
Назва	Музичні інструменти
Картинка	
Опис	Музичний інструмент – це розвиваюча іграшка, отримана у подарунок, завжди приносить величезну радість дитині. Наш інтернет-магазин пропонує Вам познайомитися з категорією «Музичні інструменти». Музичні іграшки відносяться до категорії розвиваючих. І це неспроста, адже звуки дитина розрізняє набагато раніше, ніж починає розуміти мову. Тому малюкам так подобаються брязкальця.

Рисунок 3.32 – Перегляд окремої категорії

[Головна](#) / [Категорії](#) / [Музичні інструменти](#) / Оновити

Оновити категорії: Музичні інструменти

Назва

Музичні інструменти



Вибрати файл Файл не вибрано

Опис

Музичний інструмент – це розвиваюча іграшка, отримана у подарунок, завжди приносить величезну радість дитині. Наш інтернет-магазин пропонує Вам познайомитися з категорією «Музичні інструменти». Музичні іграшки відносяться до категорії розвиваючих. І це неспроста, адже звуки дитина розрізняє набагато раніше, ніж починає розуміти мову. Тому малюкам так подобаються брязкальця.

Зберегти

Рисунок 3.33 – Редагування окремої категорії

AQUTOYS

Категорії

Про нас

Доставка та оплата

Контакти

Адмін панель

Logout (Garriq)

Головна

Категорії

Створити категорію

Створити категорію

Назва

Інтерактивні іграшки

Вибрати файл

Інтерактивні-removebg-preview.png

Опис

Завдяки грі малюк пізнає навколишній світ, копіює дорослих, вчиться приймати свої перші рішення. У цьому йому допоможуть інтерактивні іграшки для дітей. Наш магазин іграшок пропонує пупсів, Які плачуть, сміються, п'ють молоко і бруднять пелюшки, ляльок, які вміють розмовляти, співати і навіть танцювати. Різноманітність тварин просто величезна. Вони їдять, бігають за Вашою дитиною, видають звуки. І в окрему категорію можна виділити роботів, які блимають, пересуваються, долають перешкоди і комп'ютери, які можуть навчити Вашу дитину багато чому цікавому і корисному. Для най менших в нашому магазині пропонується аудіовізуальні моделі – співаючі пісеньки, музичні, що говорять. Їх в основному використовують для того, щоб малюка розважити, підняти йому настрій. У нас Ви знайдете велику різноманітність інтерактивних іграшок.

Зберегти

Рисунок 3.34 – Створення нової категорії

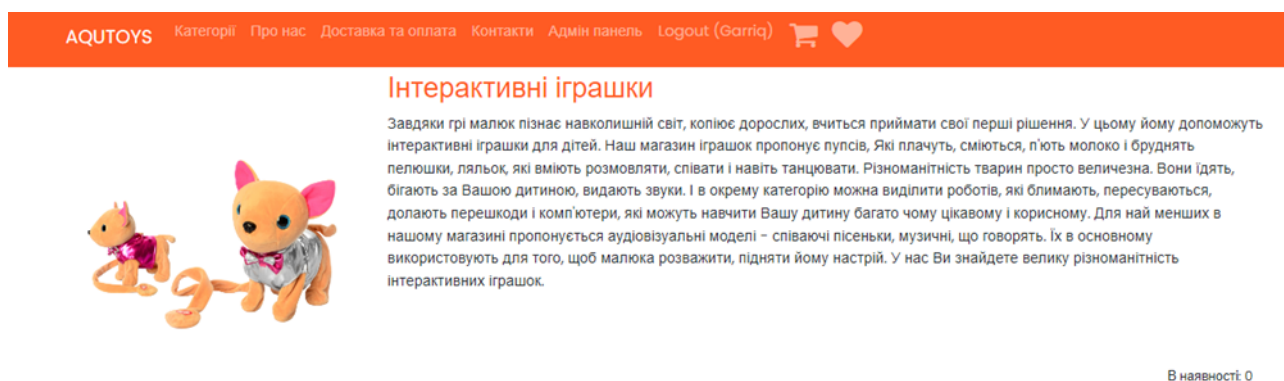


Рисунок 3.35 – Успішне створення нової категорії

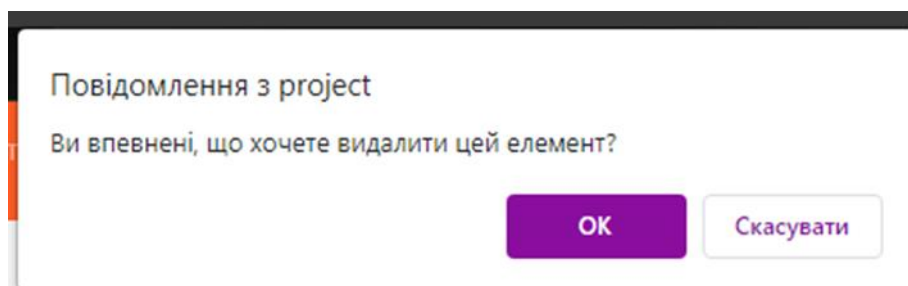


Рисунок 3.36 – Успішне створення нової категорії

3.3 Тестування роботи програмного продукту

Процес тестування програмного забезпечення є дуже важливим для забезпечення якості його роботи. Щоб перевірити коректність роботи програмного комплексу у ситуації, коли користувач замовляє товар, можна виконати наступні дії:

1. Реєстрація користувача на сайті:

передумови: користувач зайшов на головну сторінку сайту та має доступ до сторінки реєстрації;

дії: користувач вводить валідні дані у форму реєстрації, натискає кнопку “Зареєструватися”;

очікуваний результат: користувач отримає повідомлення про успішну реєстрацію на сайті або повідомлення про помилку;

фактичний результат: отримали повідомлення про успішну реєстрацію.

2. Перехід на сторінку категорії товарів:

передумови: користувач аутентифікований в системі;

дії: користувач знаходить сторінку категорії товарів та натискає на обрану категорію;

очікуваний результат: користувач знаходить потрібну йому категорію або ж повертається на головну сторінку;

фактичний результат: користувач знаходить потрібну категорію товарів.

3. Перехід на сторінку товару, який користувач хоче замовити:

передумови: користувач аутентифікований та вибрав потрібну категорію;

дії: користувач знаходить потрібний йому товар та натискає на нього;

очікуваний результат: користувач успішно перейшов на сторінку вибраного товару або ж користувач не вибрав жодного товару і повернеться на сторінку категорії товарів;

фактичний результат: користувач знайшов товар та перейшов на сторінку цього товару.

4. Додавання вибраного товару до кошика:

передумови: аутентифікований користувач та знаходиться на сторінці вибраного товару;

дії: користувач повинен натиснути кнопку “В кошику”;

очікуваний результат: товар успішно доданий до кошика або виникла помилка;

фактичний результат: товар, який користувач додав до кошика успішно доданий.

5. Перехід до оформлення замовлення:

передумови: товари вибрані користувачем повинні лежати в кошику;

дії: користувач перебуваючи в корзині повинен натиснути кнопку “Замовити”;

очікуваний результат: користувач переходить на сторінку оформлення замовлення або ж отримує помилку;

фактичний результат: користувач успішно перейшов на сторінку оформлення замовлення;

6. Оформлення замовлення:

передумови: користувача успішно перенаправлено на сторінку з формою, щоб заповнити дані для замовлення;

дії: користувач повинен заповнити всі поля форми правильною інформацією, для того щоб замовити та успішно отримати товар;

очікуваний результат: користувач отримує повідомлення про успішне замовлення або ж отримує помилку;

фактичний результат: користувач успішно зробив замовлення товарів.

Висновки до третього розділу

В даному розділі було ознайомлено з схемою будови та процесом розгортання кожного компонента інтернет-магазину.

Також, було описано та розглянуто інтерфейс програмного продукту, всі функціональні можливості додатку, дивлячись під якою роллю був користувач.

Більш, того було проведено тестування додатку на прикладі, воно складалось з шагів, для того щоб перевірити і упевнитись в коректності роботи на етапі створення нового замовлення користувачем за допомогою очікуваних та фактичних результатів.

ВИСНОВКИ

В ході виконання випускної роботи бакалавра було спроектовано та розроблено інтернет-магазин дитячих іграшок “Aquatoys”. Для успішного виконання поставленого завдання було зроблено такі дії:

- визначено основні вимоги на необхідні функціональні можливості для програмного продукту;
- було проаналізовано аналогічні інтернет-магазини: Rozum, Goodtoys, Jambo;
- було проаналізовано декілька архітектур таких, як MVVM, MVP, MVC і зважаючи на всі переваги і недоліки зроблено рішення про використання архітектури MVC;
- засобами реалізації програмного продукту було обрано PHP фреймворк Yii2, HTML5, CSS3, середовищем для розробки коду обрано PhpStorm, а для сервера використано OpenServer.
- визначені апаратні вимоги для кожного елементу системи з метою ефективного використання ресурсів та відповідності програмного забезпечення до всіх визначених потреб;
- були проаналізовані варіанти використання програмного продукту відповідного до вимог програмного забезпечення;
- було описано та продемонстровано алгоритм реєстрації та авторизації нових користувачів;
- наведено описи реалізації головних функціональних елементів інтернет-магазину;
- було ознайомлено зі схемою та процесом розгортання кожного елемента інтернет-магазину.

І на сам кінець, було проведено тестування додатку на прикладі, що складалось із кроків, для того, щоб перевірити і упевнитись в коректності роботи на етапі створення нового замовлення користувачем за допомогою очікуваних та фактичних результатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Goodtoys [Електронний ресурс] – Режим доступу до ресурсу:
<https://goodtoys.com.ua/>.
- [2] Jambo [Електронний ресурс] – Режим доступу до ресурсу:
<https://jambo.ua/>.
- [3] Rozum [Електронний ресурс] – Режим доступу до ресурсу:
<https://rozum.com.ua/ua/>.
- [4] MVC [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.yiiframework.com/doc/guide/1.1/basics.mvc>.
- [5] Yii2 Framework [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.yiiframework.com/>.
- [6] PhpStorm [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.jetbrains.com/help/phpstorm/quick-start-guide-phpstorm.html>.
- [7] PHP [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.php.net/docs.php>.
- [8] HTML [Електронний ресурс] – Режим доступу до ресурсу:
<https://developer.mozilla.org/en-US/docs/Web/HTML>.
- [9] CSS [Електронний ресурс] – Режим доступу до ресурсу:
<https://devdocs.io/css/>.
- [10] Open Server Panel [Електронний ресурс] – Режим доступу до ресурсу:
https://biblprog.org.ua/ua/open_server/.
- [11] MySQL [Електронний ресурс] – Режим доступу до ресурсу:
<https://dev.mysql.com/doc/>.
- [12] ER Diagram [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.lucidchart.com/pages/er-diagrams>.
- [13] UML для бізнес-моделювання: для чого потрібні діаграми процесів
[Електронний ресурс] – Режим доступу до ресурсу:
<https://evergreens.com.ua/ua/articles/uml-diagrams.html>.

[14] Use Case Diagram [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>.

[15] Activity Diagram [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>.

[16] Class Diagram [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>.

[17] Unsplash [Электронный ресурс] – Режим доступа до ресурсу:
<https://unsplash.com/>.

[18] Kartik Widget Rating [Электронный ресурс] – Режим доступа до ресурсу:
<https://demos.krajee.com/>.

[19] Fontawesome [Электронный ресурс] – Режим доступа до ресурсу:
<https://fontawesome.com/>.

[20] Gitlab [Электронный ресурс] – Режим доступа до ресурсу:
<https://gitlab.com/>.

ДОДАТКИ

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	.
	.					58

```
CartController.php
class CartController extends Controller
{
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::className(),
                'ruleConfig' => [
                    'class' => AccessRule::className(),
                ],
                'only' => ['view','create', 'update', 'delete'],
                'rules' => [
                    [
                        'actions' => ['view','create', 'update', 'delete'],
                        'allow' => true,
                        'roles' => [
                            User::ROLE_ADMIN
                        ],
                    ],
                ],
            ],
        ];
    }
}
```

```

}

public function actionIndex()
{
    $searchModel = new CartSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionView($id)
{
    return $this->render('view', [
        'model' => $this->findModel($id),
    ]);
}

public function actionCreate()
{
    $model = new Cart();

    if ($this->request->isPost) {
        if ($model->load($this->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    } else {
        $model->loadDefaultValues();
    }

    return $this->render('create', [

```

```

        'model' => $model,
    ]);
}
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($this->request->isPost && $model->load($this->request->post()) &&
        $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render('update', [
        'model' => $model,
    ]);
}
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}
protected function findModel($id)
{
    if (($model = Cart::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('The requested page does not exist.');
}

```

```

}

CategoriesController.php

{
    private $photoSavePlace = '@app/web/images/categories/';
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::className(),
                'ruleConfig' => [
                    'class' => AccessRule::className(),
                ],
                'only' => ['view','create', 'update', 'delete'],

                'rules' => [
                    [
                        'actions' => ['view','create', 'update', 'delete'],
                        'allow' => true,
                        'roles' => [
                            User::ROLE_ADMIN
                        ],
                    ],
                ],
            ],
        ],
    }
}

```

```

];
}
public function actionIndex()
{
    $searchModel = new CategoriesSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionView($id)
{
    return $this->render('view', [
        'model' => $this->findModel($id),
    ]);
}

public function actionCreate()
{
    $model = new Categories();
    $model->scenario = 'create';

    if ($this->request->isPost) {
        if ($model->load($this->request->post())) {
            FileSystem::uploadFile($model, 'img', $this->photoSavePlace);
            if ($model->save()) {
                return $this->redirect(['view', 'id' => $model->id]);
            }
        }
    }
}

```

```

    } else {
        $model->loadDefaultValues();
    }

    return $this->render('create', [
        'model' => $model,
    ]);
}

public function actionUpdate($id)
{
    $model = $this->findModel($id);
    $image = $model->img;

    if ($this->request->isPost && $model->load($this->request->post())) {
        if (!FileSystem::uploadFile($model, 'img', $this->photoSavePlace)) {
            $model->img = $image;
        }
        if ($model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    }

    return $this->render('update', [
        'model' => $model,
    ]);
}

public function actionDelete($id)
{
    $this->findModel($id)->delete();
}

```

```

        return $this->redirect(['index']);
    }

    public function actionCategory()
    {
        $categories = Categories::find()->asArray()->all();

        return $this->render('category', compact('categories'));
    }

    protected function findModel($id)
    {
        if (($model = Categories::findOne(['id' => $id])) !== null) {
            return $model;
        }

        throw new NotFoundHttpException('The requested page does not exist.');
    }
}

```

```

ProductsController.php
class ProductsController extends Controller
{
    private $photoSavePlace = '@app/web/images/products/';

    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [

```

```

        'delete' => ['post'],
    ],
],
'access' => [
    'class' => AccessControl::className(),
    'ruleConfig' => [
        'class' => AccessRule::className(),
    ],
    'only' => ['view','create', 'update', 'delete'],
    'rules' => [
        [
            'actions' => ['view','create', 'update', 'delete'],
            'allow' => true,
            'roles' => [
                User::ROLE_ADMIN
            ],
        ],
    ],
],
];
}

public function actionIndex()
{
    $searchModel = new ProductsSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

```

```

    }

    public function actionView($id)
    {
        return $this->render('view', [
            'model' => $this->findModel($id),
        ]);
    }

    public function actionCreate()
    {
        $model = new Products();

        $model->scenario = 'create';

        if ($model->load($this->request->post())) {
            FileSystem::uploadFile($model, 'img', $this->photoSavePlace);
            if ($model->save()) {
                return $this->redirect(['view', 'id' => $model->id]);
            }
        } else {
            $model->loadDefaultValues();
        }

        return $this->render('create', [
            'model' => $model,
        ]);
    }

    public function actionUpdate($id)
    {
        $model = $this->findModel($id);
        $image = $model->img;
    }

```

```

        if ($this->request->isPost && $model->load($this->request->post())) {
            if(!FileSystem::uploadFile($model, 'img', $this->photoSavePlace)) {
                $model->img = $image;
            }
            if ($model->save()) {
                return $this->redirect(['view', 'id' => $model->id]);
            }
        }

        return $this->render('update', [
            'model' => $model,
        ]);
    }

    public function actionDelete($id)
    {
        $this->findModel($id)->delete();

        return $this->redirect(['index']);
    }

    public function actionProduct()
    {
        if(isset($_GET['id']) && $_GET['id'] != "" && filter_var($_GET['id'],
        FILTER_VALIDATE_INT))
        {
            $id = $_GET['id'];

```

```

$categories = Categories::find()->where(['id' => $id])->asArray()->one();

if(count($categories) > 0)
{
    $model = new SortForm();

    $count_products = count(Products::find()->where(['category_id' => $id])-
>asArray()->all());

    $page = 1;
    $str = null;
    $number = 6;

    if(isset($_GET['page'])    &&    $_GET['page']    !=    ''    &&
filter_var($_GET['page'], FILTER_VALIDATE_INT))
    {
        $page = $_GET['page'];
    }

    if($model->load(Yii::$app->request->post()) && $model->validate())
    {
        if(isset($model->number) && !empty($model->number)){
            $number = $model->number;
        }

        if(isset($model->str)){

            switch($model->str){
                case 0:

```

```

        $products_array = $this->selectListProd($id, ['price'=>
SORT_ASC], $number, $page);
        break;
    case 1:
        $products_array = $this->selectListProd($id, ['price'=>
SORT_DESC], $number, $page);
        break;
    case 2:
        $products_array = $this->selectListProd($id, ['name'=>
SORT_ASC], $number, $page);
        break;
    case 3:
        $products_array = $this->selectListProd($id, ['name'=>
SORT_DESC], $number, $page);
        break;
    default:
        $products_array = $this->selectListProd($id, ['id'=> SORT_ASC],
$number, $page);
        break;
    }
}
else{
    $products_array = $this->selectListProd($id, ['id'=> SORT_ASC],
$number, $page);
}
}
else{
    $products_array = $this->selectListProd($id, ['id'=> SORT_ASC],
$number, $page);
}

```

```

    }

    $count_pages = ceil($count_products / $number);

    if(isset($_GET['view']) && $_GET['view'] == 1)
        $view = 1;
    else
        $view = 0;

    return $this->render('product', compact('categories', 'products_array',
'count_products', 'view', 'model', 'count_pages', 'id'));
    }

}

return $this->redirect(['categories/category']);
}

private function selectListProd($id, $field_sort, $limit, $start)
{
    if($start == 1)
        $start = 0;
    else
        $start = ($start - 1) * $limit;

    return Products::find()->where(['category_id' => $id])->asArray()-
>orderBy($field_sort)->limit($limit)->offset($start)->all();
}

public function actionOneProduct()
{

```

```

        if(isset($_GET['id']) && !empty($_GET['id']) && filter_var($_GET['id'],
FILTER_VALIDATE_INT))
        {
            $id = $_GET['id'];
        }
        else
        {
            throw new NotAcceptableHttpException;
        }

        $product_array = Products::find()->where(['id' => $id])->asArray()->one();
        if(!is_array($product_array) || count($product_array) < 0)
        {
            throw new NotAcceptableHttpException;
        }

        $reviews = Review::find()->where(['id_prod' => $id])->all();

        return $this->render('one-product', compact('product_array','id', 'reviews'));
    }
    protected function findModel($id)
    {
        if (($model = Products::findOne(['id' => $id])) !== null) {
            return $model;
        }

        throw new NotFoundHttpException('The requested page does not exist.');
    }
}

```

ReviewsController.php

```

class ReviewsController extends Controller
{
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::className(),
                'ruleConfig' => [
                    'class' => AccessRule::className(),
                ],
                'only' => ['view','create', 'update', 'delete'],
                'rules' => [
                    [
                        'actions' => ['view','create', 'update', 'delete'],
                        'allow' => true,
                        'roles' => [
                            User::ROLE_ADMIN
                        ],
                    ],
                ],
            ],
        ];
    }

    public function actionIndex()

```

```

{
    $searchModel = new ReviewSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionReview($id)
{
    $model = Review::findOne($id);

    $product = Products::findOne($id);
    $reviews = Review::find()->where(['id_prod' => $id])->all();
    $reviewForm = new ReviewForm();

    return $this->render('review', [
        'model' => $model,
        'product'=>$product,
        'reviews'=>$reviews,
        'reviewForm'=>$reviewForm,
    ]);
}

public function actionReviews($id)
{
    $model = new ReviewForm();

```

```

if(Yii::$app->request->isPost)
{
    $model->load(Yii::$app->request->post());
    Yii::$app->session->setFlash('success', 'Ваш відгук було успішно додано!');
    if($model->saveReview($id))
    {
        return $this->redirect(['products/one-product','id'=>$id]);
    }
}
}

public function actionCreate()
{
    $model = new Review();

    if ($this->request->isPost) {
        if ($model->load($this->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    } else {
        $model->loadDefaultValues();
    }

    return $this->render('create', [
        'model' => $model,
    ]);
}

public function actionView($id)
{
    return $this->render('view', [

```

```

        'model' => $this->findModel($id),
    ]);
}

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($this->request->isPost && $model->load($this->request->post()) &&
        $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render('update', [
        'model' => $model,
    ]);
}

public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

protected function findModel($id)
{
    if (($model = Review::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('The requested page does not exist.');
}

```

```

}

    SiteController.php

<?php

namespace app\controllers;

use app\models\Partners;
use app\models\Products;
use Yii;
use yii\base\InvalidParamException;
use yii\filters\AccessControl;
use yii\web\BadRequestHttpException;
use yii\web\Controller;
use yii\web\Response;
use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\SignupForm;
use app\models>PasswordResetRequestForm;
use app\models\ResetPasswordForm;

class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'only' => ['logout'],
            ],
        ];
    }
}

```

```

        'rules' => [
            [
                'actions' => ['logout'],
                'allow' => true,
                'roles' => ['@'],
            ],
        ],
    ],
    'verbs' => [
        'class' => VerbFilter::class,
        'actions' => [
            'logout' => ['post'],
        ],
    ],
];
}

public function actions()
{
    return [
        'error' => [
            'class' => 'yii\web\ErrorAction',
        ],
        'captcha' => [
            'class' => 'yii\captcha\CaptchaAction',
            'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
        ],
    ];
}

public function actionIndex()
{

```

```

    $partners = Partners::find()->asArray()->all();

    return $this->render('index', compact('partners'));
}

public function actionSignup()
{
    $model = new SignupForm();

    if ($model->load(Yii::$app->request->post())) {
        if ($user = $model->signup()) {
            if (Yii::$app->getUser()->login($user)) {
                Yii::$app->session->setFlash('success', 'Ви успішно зареєстрували свій
новий аккаунт!');
                return $this->goHome();
            }
        }
    }

    return $this->render('signup', [
        'model' => $model,
    ]);
}

public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new LoginForm();

```

```

        if ($model->load(Yii::$app->request->post()) && $model->login()) {
            Yii::$app->session->setFlash('success', 'Ви успішно увійшли у свій
аккаунт!');
            return $this->goBack();
        }

        $model->password = "";
        return $this->render('login', [
            'model' => $model,
        ]);
    }

    public function actionLogout()
    {
        Yii::$app->user->logout();
        Yii::$app->session->setFlash('success', 'Ви успішно вийшли зі свого
аккаунту!');
        return $this->goHome();
    }

    public function actionExchange()
    {
        return $this->render('exchange');
    }

    public function actionAdmin()
    {
        return $this->render('admin');
    }

```

```

public function actionClear()
{
    $session = Yii::$app->session;

    $session->removeAll();
    Yii::$app->session->setFlash('success', 'Товари успішно видалені з кошика!');
    return $this->render('cart', compact('session'));
}

public function actionCart()
{
    $session = Yii::$app->session;
    // $session->destroy();
    $session->open();

    if ($session->has('productsSession')) {
        $productsSession = $session->get('productsSession');
    } else {
        $productsSession = array();
    }

    if (isset($_GET['id']) && !empty($_GET['id']) && filter_var($_GET['id'],
FILTER_VALIDATE_INT)) {
        $productsArray = Products::find()->where(['id' => $_GET['id']])->asArray()-
>one();
        Yii::$app->session->setFlash('success', 'Товар успішно доданий до
кошика!');
        if (is_array($productsArray) && count($productsArray) > 0) {

            $flag = false;

```

```

for ($i = 0; $i < count($productsSession); $i++) {
    if ($productsSession[$i]['id'] == $_GET['id']) {

        $flag = true;
        if ($productsArray['counts'] >= $productsSession[$i]['count'] + 1) {

            $productsSession[$i]['count']++;
        }
        break;
    }
}

if (!$flag) {
    array_push($productsSession, ['id' => $_GET['id'], 'count' => 1]);
}

}

}

$session->set('productsSession', $productsSession);

$productsSession = $session->get('productsSession');

$arrayID = array();

foreach ($productsSession as $value) {
    array_push($arrayID, $value['id']);
}

```

```

$products = Products::find()->where(['id' => $arrayID])->asArray()->All();

foreach ($products as $key => $value) {
    $products[$key]['count_cart'] = $productsSession[$key]['count'];
}

return $this->render('cart', compact('products'));
}

public function actionEmail()
{
    //Smtп email setting
    $content = "<h1>This is html template</h1><div>this is body text</div>";
    $Email = Yii::$app->mailer->compose('@app/mail/passwordResetToken-text',
['content' => $content]);
    $Email->setFrom(['from@gmail.com' => 'AQUTOYS']);
    $Email->setTo('garriq.kravchenko@gmail.com');
    $Email->setSubject('Скидання паролю');
    $Email->setTextBody('Plain text content');
    $Email->send();
    return $this->render('email');
}

public function actionRequestPasswordReset()
{
    $model = new PasswordResetRequestForm();

    if ($model->load(Yii::$app->request->post()) && $model->validate()) {
        if ($model->sendEmail()) {

```

```

        Yii::$app->session->setFlash('success', 'Перевірте свою електронну
пошту, щоб отримати подальші інструкції.');
```

```

        return $this->goHome();
    } else {
        Yii::$app->session->setFlash('error', 'На жаль, ми не можемо скинути
пароль для наданої електронної пошти.');
```

```

    }
}

return $this->render('requestPasswordResetToken', [
    'model' => $model,
]);
}

public function actionResetPassword($token)
{
    try {
        $model = new ResetPasswordForm($token);
    } catch (InvalidParamException $e) {
        throw new BadRequestHttpException($e->getMessage());
    }

    if ($model->load(Yii::$app->request->post()) && $model->validate() &&
$model->resetPassword()) {
        Yii::$app->session->setFlash('success', 'Новий пароль збережено.');
```

```

        return $this->goHome();
    }

    return $this->render('resetPasswordForm', [
        'model' => $model,
    ]);
}

```

```

    }

}

UserController.php
class UserController extends Controller
{
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::className(),
                'ruleConfig' => [
                    'class' => AccessRule::className(),
                ],
                'only' => ['view','create', 'update', 'delete'],
                'rules' => [
                    [
                        'actions' => ['view','create', 'update', 'delete'],
                        'allow' => true,
                        'roles' => [
                            User::ROLE_ADMIN
                        ],
                    ],
                ],
            ],
        ],
    }
}

```

```

        ],
    ];
}

public function actionIndex()
{
    $searchModel = new UserSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionView($id)
{
    return $this->render('view', [
        'model' => $this->findModel($id),
    ]);
}

public function actionCreate()
{
    $model = new User();

    if ($this->request->isPost) {
        if ($model->load($this->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    } else {
        $model->loadDefaultValues();
    }
}

```

```

        return $this->render('create', [
            'model' => $model,
        ]);
    }

    public function actionUpdate($id)
    {
        $model = $this->findModel($id);

        if ($this->request->isPost && $model->load($this->request->post()) &&
            $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }

        return $this->render('update', [
            'model' => $model,
        ]);
    }

    public function actionDelete($id)
    {
        $this->findModel($id)->delete();

        return $this->redirect(['index']);
    }

    protected function findModel($id)
    {
        if (($model = User::findOne(['id' => $id])) !== null) {
            return $model;
        }
    }

```

```

        throw new NotFoundException('The requested page does not exist.');
```

```

    }
}
```

WishController.php

```

class WishController extends Controller
{
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::className(),
                'ruleConfig' => [
                    'class' => AccessRule::className(),
                ],
                'only' => ['view', 'update'],
                'rules' => [
                    [
                        'actions' => ['view', 'update'],
                        'allow' => true,
                        'roles' => [
                            User::ROLE_ADMIN
                        ],
                    ],
                ],
            ],
        ],
    }
}
```

```

        ],
    ];
}

public function actionIndex()
{
    $searchModel = new WishSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionView($id)
{
    return $this->render('view', [
        'model' => $this->findModel($id),
    ]);
}

public function actionWish()
{
    $product = Products::find()->joinWith('wish')->where(['wishlist.id_user' =>
    \Yii::$app->user->getId()])->all();

    return $this->render('wish', compact('product'));
}

public function actionCreate($id)
{
    $model = new Wish();

```

```

$categoryid = Products::findOne(['id' => $id])->category_id;

$model->id_prod = $id;
$model->id_user = \Yii::$app->user->getId();
if (Wish::find()->where(['id_prod' => $id, 'id_user' => \Yii::$app->user-
>getId()])->one()) {
    Yii::$app->session->setFlash('error', 'Товар вже доданий до списку
бажаного!');
    return $this->redirect(['products/product', 'id' => $categoryid]);
} else {
    if ($model->save())
        Yii::$app->session->setFlash('success', 'Товар успішно доданий до списку
бажаного!');
    return $this->redirect(['wish/wish', 'id' => $categoryid]);
}
}

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($this->request->isPost && $model->load($this->request->post()) &&
$model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render('update', [
        'model' => $model,
    ]);
}

```

```

public function actionDeleteall()
{
    $userId = \Yii::$app->user->getId();

    $products = Wish::find()->where(['id_user' => $userId])->all();

    foreach ($products as $product) {
        $product->delete();
    }

    \Yii::$app->session->setFlash('success', 'Список обраного був успішно
очищений!');

    return $this->redirect(['wish']);
}

public function actionDeleteone()
{
    $userId = \Yii::$app->user->getId();

    $product = Wish::findOne(['id_user' => $userId]);

    if ($product !== null) {
        $product->delete();
    }

    \Yii::$app->session->setFlash('success', 'Товар був успішно видалений зі
списку обраного!');

    return $this->redirect(['wish']);

}

public function actionDelete($id)

```

```

{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}
protected function findModel($id)
{
    if (($model = Wish::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundException('The requested page does not exist.');
```

ZakazController.php

```

class ZakazController extends Controller
{
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::className(),
                'ruleConfig' => [
                    'class' => AccessRule::className(),
```

```

        ],
        'only' => ['view','create', 'update', 'delete'],
        'rules' => [
            [
                'actions' => ['view','create', 'update', 'delete'],
                'allow' => true,
                'roles' => [
                    User::ROLE_ADMIN
                ],
            ],
        ],
    ],
];

}

public function actionIndex()
{
    $searchModel = new ZakazSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

public function actionZakaz()
{
    $model = new Zakaz();

    if ($this->request->isPost) {

```

```

if ($model->load($this->request->post()) && $model->save()) {
    $prod = [];

    foreach ($_SESSION['productsSession'] as $item)
    {
        $product = Products::findOne($item['id']);
        $prod[] = ['name' => $product -> name, 'price' => $product -> price,
'username' => Yii::$app->user->identity->username];
        $cart = new Cart();
        $cart->id_user = Yii::$app->user->getId();
        $cart->id_zakaz = $model->id;
        $cart->id_prod = $item['id'];
        $cart->save();
    }

    //Smtп email setting
    $Email = Yii::$app->mailer->compose('@app/mail/zakazonemail', ['foots'
=> $prod]);

    $Email->setFrom(['from@gmail.com' => 'AQUUTOYS']);
    $Email->setTo('garriq.kravchenko@gmail.com');
    $Email->setSubject('Замовлення від користувача');
    $Email->setTextBody('Plain text content');
    $Email->send();

    $session = Yii::$app->session;
    $session->destroy();

    Yii::$app->session->setFlash('success', 'Ваше замовлення було успішно
опрацьовано! Очікуйте на дзвінок від нашого працівника для перевірки та
уточнення замовлення!');

    return $this->redirect(['/']);

```

```

    }
} else {
    $model->loadDefaultValues();
}

return $this->render('zakaz', [
    'model' => $model,
]);
}

public function actionView($id)
{
    return $this->render('view', [
        'model' => $this->findModel($id),
    ]);
}

public function actionCreate()
{
    $model = new Zakaz();

    if ($this->request->isPost) {
        if ($model->load($this->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    } else {
        $model->loadDefaultValues();
    }

    return $this->render('create', [
        'model' => $model,
    ]);
}

```

```

    });
}
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($this->request->isPost && $model->load($this->request->post()) &&
        $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render('update', [
        'model' => $model,
    ]);
}
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}
}

```