

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

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

на тему:

«Розробка інтерактивного каталогу коміксів»

Виконав студент групи ЗПЗ-20-1
ОСАДЧУК Євгеній Русланович

Керівник роботи:
ЧЕПЮК Ларіна Олексіївна

Консультант:
ЛОКТИКОВА Тамара Миколаївна

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2024

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

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ОСАДЧУК Євгеній Русланович**

Керівник роботи: **ЧЕПЮК Ларіна Олексіївна**

Тема роботи: **«Розробка інтерактивного каталогу коміксів»,**

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р. № 74/с**

Вихідні дані для роботи: **об'єктом дослідження є інформаційні технології для розробки інтерактивного каталогу коміксів, а предметом дослідження є засоби, алгоритми та методи, що забезпечують ефективну роботу інтерактивного каталогу коміксів.**

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Локтікова Т.М.	26.02.2024р.	26.02.2024р.
2	Локтікова Т.М.	11.03.2024р.	11.03.2024р.
3	Локтікова Т.М.	24.04.2024р.	24.04.2024р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебсайту та пояснювальної записки. Пояснювальна записка до кваліфікаційної роботи містить текстову частину, викладену на 54 сторінках друкованого тексту. Список використаних джерел містить 12 найменувань і займає 1 сторінку. У роботі наведено 30 рисунків та 1 таблиця. Також у роботі є 28 сторінок додатків. Загальний обсяг роботи становить 82 сторінки.

Метою роботи є розробка інтерактивного каталогу, який забезпечить ефективні вибір та замовлення коміксів.

Після проведених огляду й аналізу можливих архітектурних рішень для системи було обрано архітектуру MVC.

Для реалізації інтерактивного каталогу були застосовані сучасні інформаційні технології – PHP, JavaScript, jQuery та MySQL.

Розроблена система відрізняється зручним та адаптивним інтерфейсом, який забезпечує інтерактивний процес вибору та замовлення коміксів.

Інтерактивний каталог пройшов необхідні види тестування, що допомогли усунути можливі помилки системи.

КЛЮЧОВІ СЛОВА: ВЕБСАЙТ, JAVASCRIPT, MVC, MYSQL, PHP.

					ІПЗ.КР.Б-121-24-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Осадчук Є.Р.			Розробка інтерактивного каталогу коміксів. Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник		Чепюк Л.О.					3	82
						Житомирська політехніка, група ЗІПЗ-20-1		
Н. контр.		Тичина О.П						
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

The bachelor's thesis consists of a website and an explanatory note. The explanatory note to the qualification paper contains a text part laid out on 54 pages of printed text. The list of used sources contains 12 names and occupies 1 page. The work contains 30 figures and 1 table. There are also 28 pages of appendices in the work. The total volume of work is 82 pages.

The goal of the work is to develop an interactive catalog that will ensure effective selection and ordering of comics.

After reviewing and analyzing possible architectural solutions, the MVC architecture was chosen for the system.

Modern information technologies - PHP, JavaScript, jQuery and MySQL - were used to implement the interactive catalog.

The developed system is distinguished by a convenient and adaptive interface that provides an interactive process of selecting and ordering comics.

The interactive catalog passed the necessary types of testing, which helped eliminate possible system errors.

KEYWORDS: JAVASCRIPT, MVC, MYSQL, PHP, WEBSITE.

		Осадчук Є.Р.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРАКТИВНОГО КАТАЛОГУ.....	8
1.1 Постановка задачі	8
1.2 Аналіз аналогів програмного продукту	9
1.3 Вибір архітектури інтерактивного каталогу	13
1.4 Обґрунтування вибору інструментальних засобів для реалізації системи	18
Висновки до першого розділу.....	21
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ	22
2.1 Визначення варіантів використання системи	22
2.2 Розробка бази даних	24
2.3 Розробка функціональних алгоритмів роботи програми	28
2.4 Реалізація системи	31
Висновки до другого розділу	41
РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНТЕРАКТИВНИМ КАТАЛОГОМ	43
3.1 Структура інтерфейсу системи	43
3.2 Тестування системи	47
Висновки до третього розділу	50
ВИСНОВКИ	51
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ.....	54

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних

ООП – об’єктно-орієнтоване програмування

ПЗ – програмне забезпечення

CMS – система управління контентом

CRUD – основні операції, які використовуються для управління даними (створення, читання, оновлення, видалення)

MVC – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення, що дозволяє розділити додаток на три компоненти: Model, View, Controller

PHP – мова програмування (гіпертекстовий препроцесор)

		Осадчук Є.Р.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У даній кваліфікаційній роботі представлений процес створення програмного забезпечення (ПЗ) інтерактивного каталогу, який стане цікавим та корисним для людей-фанатів коміксів чи просто зацікавлених у цій тематиці.

Актуальність роботи полягає в застосуванні сучасних технологій об'єктно-орієнтованого підходу в програмуванні. Дана кваліфікаційна робота відображає на прикладі роботу ООП. Проект зберігає дані в БД та може надавати аналіз даних, що зберігаються.

Мета роботи полягає в розробці інтерактивного каталогу коміксів з використанням об'єктно-орієнтованого підходу і користувацьких класів.

Для досягнення поставленої мети необхідними є виконання таких завдань:

1. Здійснити огляд й аналіз аналогів системи, дослідити їхні переваги та недоліки, сформулювати вимоги до розроблюваної системи.
2. Обрати із обґрунтуванням методи і засоби побудови та проектування системи.
3. Створити систему із застосуванням обраного інструментарію, побудувати архітектуру та розробити основні компоненти системи.
4. Здійснити тестування системи.

Для вирішення поставлених завдань були використані такі методи дослідження: аналіз, синтез, систематизація, зіставлення.

Об'єктом дослідження є інформаційні технології для розробки інтерактивного каталогу коміксів.

Предметом дослідження є засоби, алгоритми та методи, що забезпечують ефективну роботу інтерактивного каталогу коміксів.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ НАПРЯМІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРАКТИВНОГО КАТАЛОГУ

1.1 Постановка задачі

Інтерактивний каталог коміксів створюється для задоволення потреб шанувальників коміксів, які шукають зручний спосіб вибору та замовлення своїх улюблених видань через Інтернет. Цей каталог пропонуватиме широкий асортимент коміксів, від класичних серій до новинок і рідкісних знахідок, завдяки зручному та привабливому вебінтерфейсу. Основною метою проекту є створення середовища, де користувачі зможуть легко знаходити, переглядати і замовляти комікси, забезпечуючи при цьому максимально комфортний та інтуїтивно зрозумілий досвід користування.

На головній сторінці сайту будуть виставлені акційні пропозиції, новинки та найпопулярніші комікси, що дозволить користувачам одразу ж побачити найцікавіші та актуальні пропозиції. Для більш детального вибору доступний каталог з розширеними можливостями фільтрації за жанрами, авторами, ціною, та іншими важливими параметрами. Кожен комікс на сайті матиме свою окрему сторінку з детальним описом, зображеннями обкладинок, відгуками від інших користувачів і рекомендаціями схожих товарів, які можуть зацікавити замовника.

Пошукова система сайту буде оснащена функцією пошуку, що дозволить користувачам ефективно знаходити комікси, використовуючи ключові слова або назви. Кошик для покупок дозволить користувачам зберігати обрані товари до моменту оформлення замовлення, яке буде максимально спрощено для забезпечення легкості та зручності замовлень.

Для адміністраторів сайту будуть доступні інструменти для управління контентом, що включають додавання нових товарів, оновлення інформації про існуючі комікси та видалення неактуальних позицій. Система аналітики забезпечить звіти про продажі, популярність товарів та поведінку користувачів на сайті, допомагаючи керувати запасами та планувати маркетингові заходи.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

Управління профілями користувачів включатиме можливість налаштування ролей та доступів, а також управління індивідуальними налаштуваннями користувачів.

1.2 Аналіз аналогів програмного продукту

На сьогодні в Україні існує досить мала кількість сайтів для тих, хто хоче знайти та замовити комікси та дізнатись щось нове про них.

Першим ми розглянемо сайт <https://cosmic.com.ua> [1].

Сайт Cosmic Shop є інтернет-магазином, який спеціалізується на продажу коміксів, артбуків, колекційних фігурок, футболок та іншого мерчу (рис. 1.1). Він вирізняється широким вибором продукції, зокрема, представлено офіційні видання відомих брендів, таких як Marvel і Dark Horse Comics, а також ексклюзивні колекційні видання. Магазин надає можливість покупцям придбати товари як українською, так і англійською мовами, що робить його доступним для широкого кола користувачів.

Дизайн сайту орієнтований на зручність користувачів: він включає чітке розділення на категорії, можливість додавання товарів у список бажань та прості опції для покупки. Також на сайті регулярно проводяться акції та розпродажі, що акцентовано спеціальними маркуваннями товарів із зазначенням знижок.

Сайт також забезпечує детальний опис кожного товару, включаючи інформацію про видавця, автора, кількість сторінок, мову та рік випуску, що допомагає покупцям ухвалити обґрунтоване рішення перед купівлею. Зокрема, наявність різних артбуків, таких як "Мистецтво «Любові, смерті + роботів»", подається з глибоким зануренням у контент серіалу, що відображає високий рівень деталізації та оригінального підходу до кожного видання.

Аналіз сайту Cosmic Shop виявив кілька потенційних недоліків, які можуть впливати на досвід користувачів:

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

1. Навігація та організація контенту: хоча сайт і добре структурований, деякі користувачі можуть знайти навігацію трохи перевантаженою або неінтуїтивною. Велика кількість категорій та підкатегорій може ускладнити швидке знаходження конкретних товарів.

2. Інформація про товари: іноді описи товарів можуть бути недостатньо деталізованими. Користувачам може бракувати інформації про розміри, матеріали або інші специфікації, що важливі для прийняття рішення про покупку.

3. Мобільна оптимізація: хоча сайт функціональний на мобільних пристроях, деякі елементи дизайну або функціональності можуть не оптимально відображатися або працювати на смартфонах та планшетах (рис. 1.2).

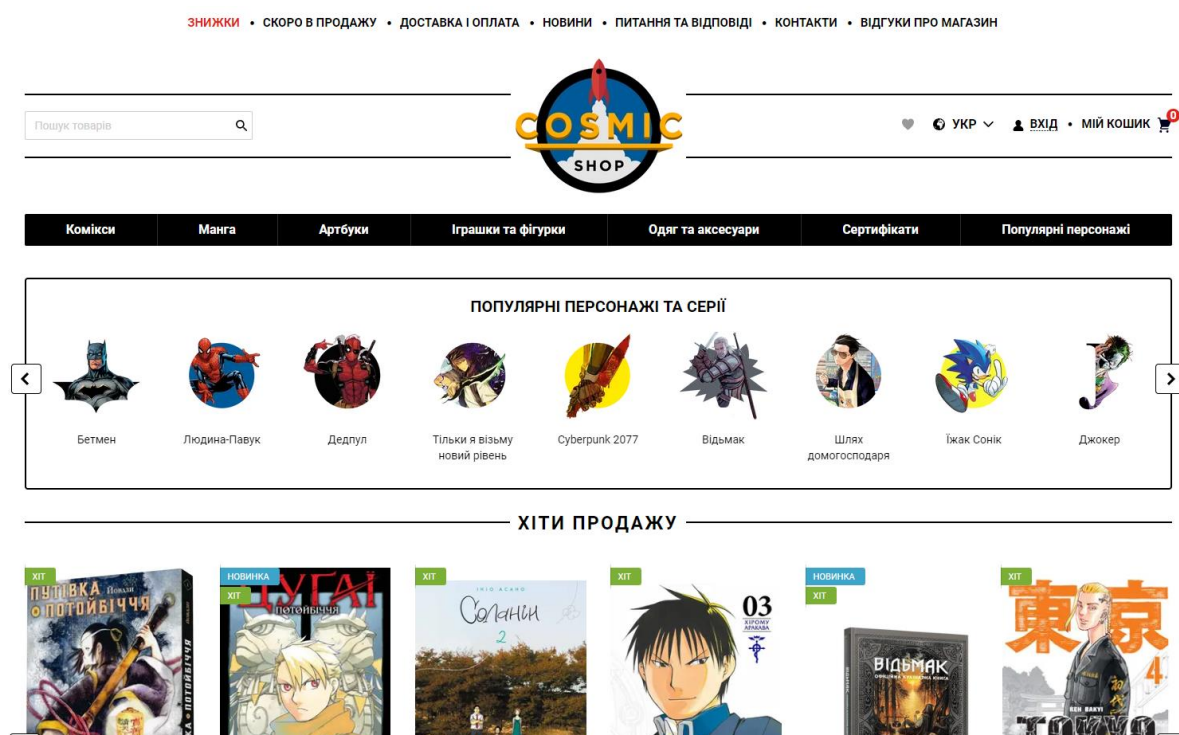


Рисунок 1.1 - Інтерфейс сайту Cosmic Shop

Наступним розглянемо сайт <https://worldofcomics.ua> [2].

World of Comics — це інтернет-магазин в Україні, який спеціалізується на продажу коміксів, манги та іншого гік-мерчу (рис. 1.3). Магазин пропонує широкий асортимент оригінальних та ліцензійних продуктів, що робить його популярним серед любителів коміксів. World of Comics надає широкий вибір коміксів, включаючи популярні серії від Marvel та DC, такі як Deadpool, Iron Man та Batman, доступні в різних випусках та спеціальних виданнях. Магазин

		Осадчук Є.Р.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

також пропонує ексклюзивні товари, що може привернути увагу колекціонерів та відданих шанувальників, які шукають унікальні товари. Сайт має адаптивний дизайн (рис. 1.4).

Сайт підтримує функції фільтрації за популярністю або ціною і пропонує опції для сортування товарів, що покращує досвід покупок.

Серед недоліків можна виділити такі:

1. Складність навігації: незважаючи на широкий асортимент продукції, нові користувачі можуть зіткнутися з певними складнощами в навігації через велику кількість товарів та категорій.

2. Інформація про товари: деякі описи товарів могли б бути детальнішими, особливо щодо специфікацій, які можуть цікавити завзятих колекціонерів.

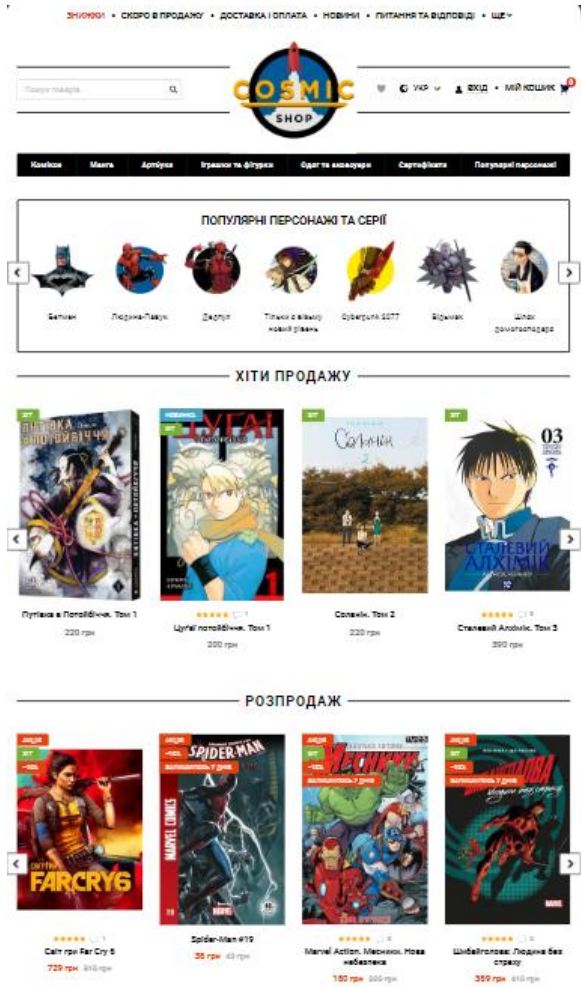


Рисунок 1.2 - Адаптивність Cosmic Shop для телефону

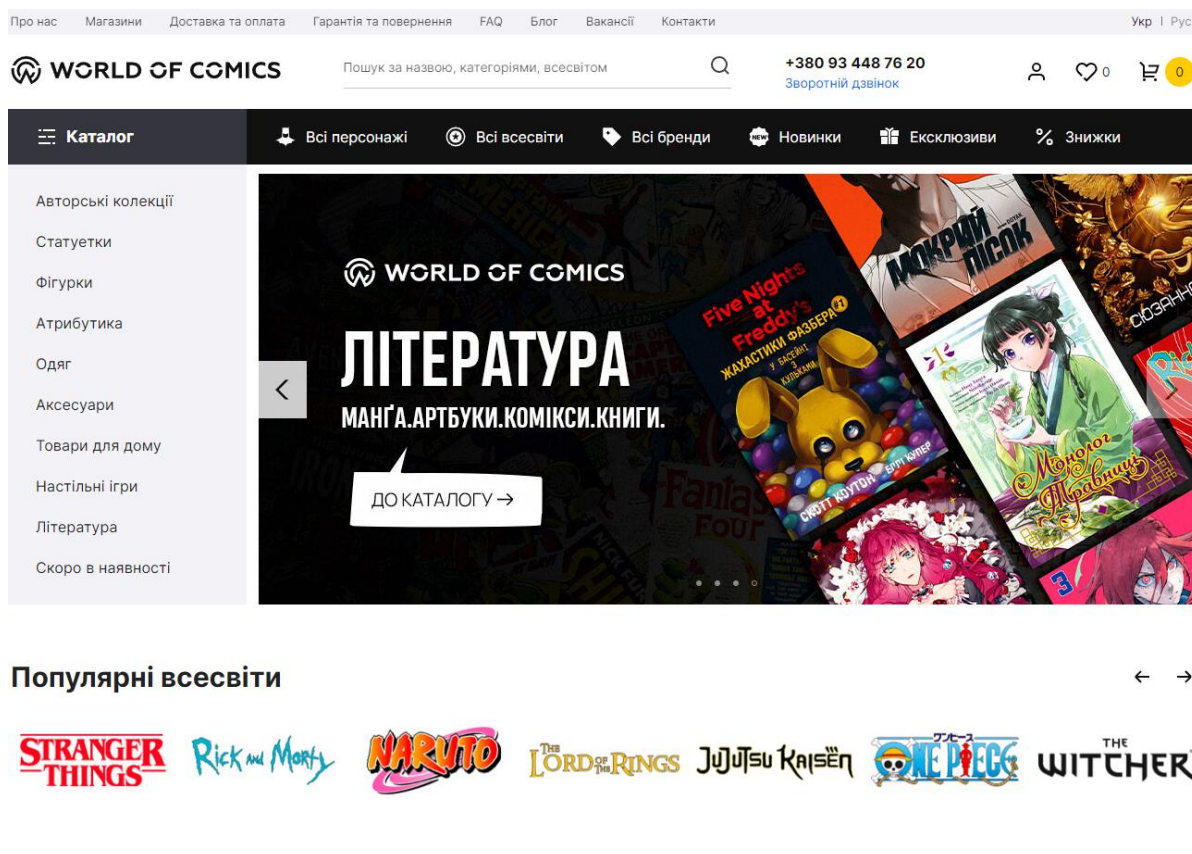


Рисунок 1.3 - Інтерфейс сайту World of Comics

На основі переглянутих сайтів можна скласти таблицю порівняльної характеристики за основними критеріями (табл. 1.1).

Таблиця 1.1

Порівняльна характеристика аналогів програмного продукту

Назва сайту/Критерій	Адаптивність	Можливість збільшення зображення	Швидкий доступ до верхньої панелі	Читабельність при адаптуванні	Вдалість кольорової гамми	Зрозумілість інтерфейсу	Інтернет-магазин
cosmic.com.ua	+	+	-	+	+	+	+
worldofcomics.ua	+	-	+	+	-	-	+

Головним завданням буде створити такий сайт, який буде мати наступні переваги:

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

1. зручність у використанні;
2. приємний інтерфейс;
3. адаптивність;
4. швидке завантаження.

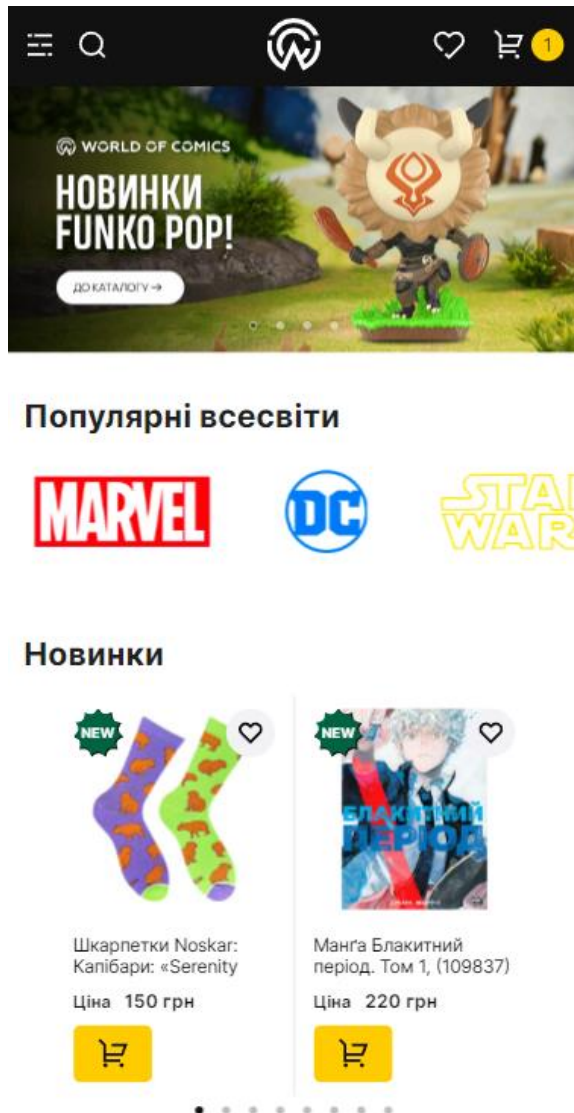


Рисунок 1.4 - Адаптивність сайту World of Comics

1.3 Вибір архітектури інтерактивного каталогу

У роботі потрібно розробити вебсайт, який надаватиме можливості реєстрації, авторизації, пошуку товару, отримання детальної інформації про товар, замовлення товару.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

При розробці даного сайту буде використовуватися об'єктно-орієнтований підхід до проектування ПЗ, адже він має безліч переваг над іншими методами програмування.

Об'єктно-орієнтоване програмування (ООП) — це метод програмування, заснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування. Таким чином, програма являє собою набір об'єктів, що мають стан та поведінку. Об'єкти взаємодіють використовуючи повідомлення. Будеться ієрархія об'єктів: програма в цілому — це об'єкт, для виконання своїх функцій вона звертається до об'єктів що містяться у ньому, які у свою чергу виконують запит шляхом звернення до інших об'єктів програми [3].

Об'єктно-орієнтоване програмування дозволяє розробникам більш ефективно управляти складністю програмного забезпечення за рахунок інкапсуляції, наслідування та поліморфізму. Інкапсуляція дозволяє обмежити доступ до деяких компонентів об'єкта та управляти тим, як зовнішній світ взаємодіє з об'єктом. Це полегшує модифікацію та розширення програм, оскільки зміни в одній частині системи не впливають на інші. Наслідування дозволяє класам успадковувати характеристики та поведінку від інших класів, що сприяє повторному використанню коду та зменшенню дублювання. Поліморфізм дає змогу об'єктам різних класів, які виконують однаковий інтерфейс, бути використаними взаємозамінно, що забезпечує гнучкість та масштабованість системи [4].

Великою перевагою ООП є його здатність моделювати реальний світ, що робить дизайн програмного забезпечення інтуїтивно зрозумілим. Об'єкти в програмі можуть представляти реальні сутності, такі як користувачі, замовлення, продукти тощо, а їхні взаємодії моделюють реальні процеси та відносини. Це сприяє як кращому розумінню програми, так і спрощенню її розвитку та підтримки [5].

Незважаючи на численні переваги, ООП також може виявитися складним для використання у великих системах, де неправильне застосування

		Осадчук Є.Р.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

наслідування може призвести до надмірно складних ієрархій, що ускладнюють розуміння та підтримку коду. Однак, за допомогою ретельного планування архітектури та дизайну, можливо мінімізувати такі проблеми та максимально використовувати переваги ООП для створення ефективних, масштабованих та легко підтримуваних програмних рішень.

Важливою частиною даного проекту є зберігання та обробка інформації.

Зберігання інформації – комплекс заходів, спрямований на забезпечення збереження повноти і цілісності сформованих даних про певну інформацію, створення і підтримання належних умов для їх використання, а також запобігання несанкціонованому доступу, поширенню і використанню.

Найзручніший спосіб збереження інформації є база даних.

База даних – це інструмент, використовуючи який, можна збирати й упорядковувати інформацію. У базах даних можна зберігати відомості про людей, продукти, замовлення тощо.

Комп'ютеризована БД – це контейнер об'єктів. Одна БД може містити кілька таблиць.

Тому, в даній програмі буде використано базу даних MySQL Server.

Для роботи з даним програмним продуктом потрібно знати та використовувати мову SQL.

Мова SQL – мова програмування, призначена для управління даними в системах керування базами даних. Дана мова містить ключові слова, які слугують для визначення дій над даними. Наприклад, слово «select» використовується для вибірки даних з таблиць, «insert» - для додавання даних до таблиць, «update» - оновлення, «delete» видалення [6].

Модель даних (Data model) — абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними. Використовується механізм представлення і організації просторової моделі даних, або растрової моделі даних. Вона не залежить від комп'ютерної системи і пов'язана тільки зі структурою даних.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Безпека даних в базах даних є важливим аспектом у сучасних інформаційних системах. Механізми авторизації та аутентифікації дозволяють контролювати доступ до даних, запобігаючи несанкціонованому доступу. За допомогою ролей та політик доступу адміністратори можуть визначати, хто має право читати, редагувати або видаляти певні дані. Шифрування даних, як на стадії зберігання, так і при передачі, забезпечує додатковий рівень захисту, захищаючи чутливу інформацію від витоку та зловживань.

Оптимізація запитів є ще одним критичним аспектом управління базами даних, що впливає на продуктивність системи. Використання індексів допомагає підвищити швидкість виконання запитів, зменшуючи час відгуку бази даних на запити користувачів. Ефективне проектування схеми бази даних, яке включає нормалізацію даних, допомагає уникнути надмірної редундантності та спрощує управління даними. Аналітика запитів і профілювання баз даних дозволяють ідентифікувати та оптимізувати «вузькі місця» у продуктивності [7].

Ще одним важливим аспектом є розробка бізнес-логіки на рівні бази даних за допомогою збережених процедур та тригерів. Це дозволяє централізувати та оптимізувати обробку даних, зменшуючи навантаження на додатки клієнтської сторони та забезпечуючи більш високу надійність і безпеку обробки даних. Збережені процедури виконують предопрацьовані запити до даних, що значно підвищує ефективність виконання складних операцій, які потребують множинних звернень до бази даних.

Для зручного, гнучкого та інтерактивного написання модулів до сайту використовується MVC патерн [8].

Модель–вигляд–контролер (або Модель–представлення–контролер, Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

(вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

Використання MVC-шаблону забезпечує відмінну підтримку тестування, оскільки кожен компонент (модель, вигляд, контролер) може бути тестований окремо. Це особливо корисно у великих проектах, де потрібно забезпечити високий рівень надійності та якості програмного забезпечення. Модель може бути протестована без інтерфейсу користувача, вигляд може бути перевірений на відповідність дизайну без реальної бізнес-логіки, а контролери можуть бути перевірені на правильність передачі даних між виглядом та моделлю. Це розділення також дозволяє командам працювати більш незалежно, ефективно координуючи свої зусилля.

Крім того, MVC сприяє більшій гнучкості при інтеграції з іншими архітектурними шаблонами та дизайн-патернами. Наприклад, можна інтегрувати шаблони репозиторій або стратегія для моделі, щоб покращити управління даними або надати адаптивність у поведінку програми. Для вигляду можна використовувати декоратор або компоновщик, щоб збільшити можливості відображення без зміни основних механізмів взаємодії. Ця гнучкість важлива для розвитку системи, яка з часом може потребувати нових функцій чи інтеграції з додатковими сервісами.

Завдяки чіткому відокремленню компонентів, MVC також дозволяє краще масштабувати програмні проекти. Розподілення завдань між моделлю, виглядом і контролером допомагає управляти розростанням коду та спрощує його рефакторинг та оптимізацію. Можливість незалежного оновлення

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

компонентів програми без впливу на інші частини системи є великою перевагою, яка робить великі та складні програми більш стійкими до помилок та легшими у підтримці.

1.4 Обґрунтування вибору інструментальних засобів для реалізації системи

У сучасному світі веброзробки існує безліч мов програмування, які можна використовувати для створення інтерактивного каталогу, таких як PHP, JavaScript, Python, Ruby та інші.

PHP — це одна з найпопулярніших мов веброзробки у світі. Ця популярність не є випадковою: PHP спеціально розроблена для створення вебсторінок і вебдодатків, що робить її ідеальним вибором для проектів, які вимагають стабільної та ефективної взаємодії з сервером. Це особливо важливо для інтернет-магазинів, де потрібно обробляти велику кількість транзакцій та запитів користувачів [9].

JavaScript є дуже популярною мовою, особливо в контексті розробки клієнтської частини з використанням бібліотеки React або фреймворку Angular. JavaScript також може використовуватися на сервері (Node.js), що дозволяє створювати повністю ізоморфні додатки. Проте, в порівнянні з PHP, Node.js може вимагати більш складного управління асинхронним кодом та може не мати такої широкої підтримки готових рішень для електронної комерції, як Magento чи WooCommerce на PHP.

Python є ще одним чудовим кандидатом з потужними фреймворками, такими як Django та Flask. Django особливо добре підходить для швидкої розробки за рахунок свого "батарейки включено" підходу, пропонуючи вбудовану панель адміністратора і розширені інструменти для роботи з базами даних. Тим не менш, PHP все ще виграє за рахунок спеціалізації на веброзробці і кращої інтеграції з вебсерверами, що може сприяти кращій продуктивності в розробці інтернет-магазинів.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Ruby зі своїм фреймворком Ruby on Rails також заслуговує уваги як потужне рішення для швидкої розробки вебдодатків з "конвенцією над конфігурацією". Rails пропонує ряд інструментів, які можуть спростити створення комплексних вебдодатків. Однак, масштабування Ruby on Rails може бути викликом порівняно з PHP, особливо у великих, високонавантажених проектах, де PHP видається більш оптимальним завдяки своїй швидкодії та взаємодії з сервером.

В кінцевому підсумку, хоча існує багато мов, які можуть бути використані для розробки інтернет-магазину коміксів, PHP залишається привабливим вибором через його цільову орієнтацію на веброзробку, велику спільноту, обширну підтримку і зрілість управління вебдодатками. Ці характеристики роблять PHP ідеальним для проектів, де потрібно швидко запустити надійний та масштабований інтернет-магазин.

Розглянемо переваги PHP детальніше.

PHP має велику кількість доступних ресурсів і спільноту. Завдяки своїй популярності, PHP має одну з найбільших спільнот розробників, які надають безліч відкритих джерел та інструментів. Це означає, що розробники можуть швидко знайти відповіді на свої запитання або готові рішення для типових проблем, що зустрічаються під час розробки інтернет-магазинів.

Безпека в PHP постійно покращується завдяки регулярним оновленням та вдосконаленням від спільноти та розробників. Для інтернет-магазину важливо забезпечити захист даних користувачів і фінансових транзакцій, і PHP надає всі необхідні інструменти для реалізації високого рівня безпеки.

На завершення, PHP є мовою з високою продуктивністю за рахунок широкої підтримки на серверах і оптимізації для роботи з вебзастосунками. Це означає, що інтернет-магазини на PHP можуть швидко обробляти велику кількість запитів, що є критично важливим для комерційних платформ, особливо в періоди високого навантаження, наприклад, під час акцій або розпродажів.

Середовище розробки обиралось між PHPStorm, Visual Studio Code (VS Code), Eclipse PDT, NetBeans.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

PHPStorm є одним з найпопулярніших інтегрованих середовищ розробки (IDE) для PHP і є частиною портфеля продуктів компанії JetBrains, відомої своїми потужними інструментами для розробників. Вибір PHPStorm для розробки інтернет-магазину коміксів на PHP можна пояснити кількома ключовими аспектами, які виділяють це IDE серед конкурентів.

Visual Studio Code (VS Code) - це легкий, але потужний редактор від Microsoft, який став дуже популярним серед розробників завдяки своїй гнучкості та великому вибору розширень. Хоча VS Code може бути налаштований для роботи з PHP, він не пропонує такої глибокої інтеграції та спеціалізованих функцій для PHP, як PHPStorm.

Eclipse PDT - інше популярне IDE для PHP, яке пропонує багато інструментів для розробки. Однак, воно може бути досить важким для системи і менш інтуїтивним у порівнянні з PHPStorm.

NetBeans - ще один відомий вибір для PHP-розробників, який підтримує широкий спектр мов програмування. Подібно до Eclipse, NetBeans може виявитися трохи важким для системи і не має такої зручної підтримки плагінів, як PHPStorm.

Враховуючи ці аспекти, PHPStorm є відмінним вибором для професійної PHP-розробки завдяки своїм комплексним функціям, спеціалізації на PHP та високому рівню підтримки розробників, що робить розробку ефективною і комфортною.

PHPStorm спеціально розроблений для мови PHP. Це означає, що IDE має розширену підтримку PHP коду, включаючи автоматичне завершення коду, рефакторинг, точне виявлення помилок і підказки, що в значній мірі підвищує продуктивність розробника. PHPStorm також підтримує всі сучасні версії PHP і відстежує оновлення мови, забезпечуючи користувачам доступ до останніх функцій мови.

На відміну від деяких інших PHP-орієнтованих IDE, PHPStorm також пропонує великий спектр інструментів для розробки фронтенду. Він підтримує JavaScript, HTML, CSS, Sass, LESS, TypeScript і багато інших технологій,

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

що робить його універсальним інструментом для розробки повноцінних веб-додатків.

RHPStorm має вбудовані інструменти для роботи з базами даних, що дозволяє розробникам з легкістю управляти базами даних, виконувати SQL-запити та візуалізувати схеми баз даних без потреби в сторонніх програмах. Крім того, RHPStorm пропонує велику кількість плагінів і інтеграцію з такими системами контролю версій, як Git, SVN, Mercurial.

Висновки до першого розділу

В ході виконання першого розділу було праналізовано програмні продукти схожого функціоналу та сформовано вимоги до сайту, що розробляється. Отримано теоретичні засади для подальшого написання програмного коду, використовуючи ООП, MVC паттерн, базу даних.

Також, було визначено необхідні інструменти для реалізації інтерактивного каталогу, а також інтерфейсу та функціоналу сайту, що розробляється – PHP, JavaScript, СУБД MySQL, jQuery [10].

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Визначення варіантів використання системи

На підставі отриманих у попередньому розділі даних можна розробити алгоритм роботи програми, який допоможе реалізувати функціонал та методи виконання поставленої задачі. На рис. 2.1 представлено загальний алгоритм роботи програми.

Оскільки застосовується патерн MVC, загальний алгоритм вебдодатку простий. Він буде використовуватися при кожній дії користувача. Схема алгоритму представлена нижче.

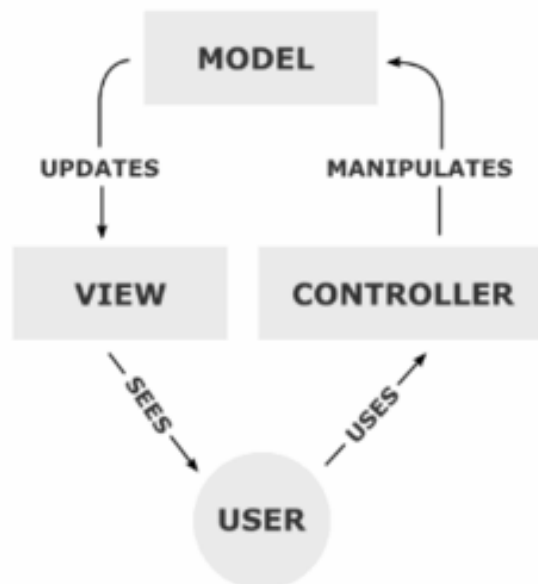


Рисунок 2.1 - Загальний алгоритм роботи вебдодатку

Для кращого розуміння загального алгоритму роботи вебдодатку, розробленого на основі патерна MVC (Model-View-Controller), розглянемо кожен етап взаємодії користувача з додатком більш детально. Це дозволить уявити процес обробки запитів і відповідей у вебдодатку.

1. Введення адреси або натискання на посилання користувачем

Коли користувач вводить URL адресу в браузері або натискає на посилання, він ініціює запит до сервера. За це відповідає клієнтська сторона, яка формує HTTP запит та відправляє його на сервер за вказаною адресою. У

цьому контексті важливо, що URL адреса асоціюється з конкретними маршрутами в коді сервера, які визначають, до якого контролера потрібно звернутися.

2. Реакція браузера на дії користувача і звернення до контролера

Сервер отримує запит і, використовуючи вбудовану систему маршрутизації, визначає, який контролер повинен обробити запит. Маршрутизація — це процес визначення реакції сервера на різні запити до компонентів вебдодатку.

3. Виклик відповідного методу контролера

Після того, як контролер ідентифіковано, він викликає специфічний метод, що відповідає за обробку цього запиту. Метод контролера має за мету координувати процеси, які мають відбутися для забезпечення відповідної реакції на запит.

4. Звернення методу контролера до моделі

Метод контролера взаємодіє з моделлю, яка відповідає за доступ до даних. Залежно від потреби, метод може замовляти в моделі вилучення, внесення або оновлення даних у базі даних.

5. Доступ до бази даних через модель

Модель виконує запити до бази даних, щоб отримати необхідну інформацію або внести зміни. Завдяки архітектурі MVC, контролери не взаємодіють безпосередньо з базою даних, а роблять це через модель, що дозволяє краще структурувати код та забезпечити його гнучкість.

6. Повернення даних у модель

Після виконання запитів до бази даних, модель обробляє отримані дані і передає їх назад у контролер. Це можуть бути дані, отримані з бази, або результати обробки даних.

7. Передача даних з моделі в метод контролера

Контролер приймає дані від моделі і готує їх до передачі в представлення. Цей крок часто включає додаткову логіку обробки, необхідну для правильного відображення даних користувачеві.

8. Передача даних контролером в представлення

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

Контролер передає оброблені дані в представлення, яке відповідає за їх візуалізацію. Представлення — це шар, який формує кінцевий HTML-код або інший формат відповіді, який буде відправлено користувачу.

9. Відправлення відповіді користувачеві

Останній крок — це відправлення сформованої відповіді з сервера назад у браузер користувача. Відповідь може містити HTML-сторінку, JSON-об'єкти або будь-який інший формат даних, який використовується для забезпечення функціоналу вебдодатку.

Цей загальний алгоритм роботи демонструє, як MVC патерн дозволяє ефективно розділити логіку обробки даних, взаємодію з користувачем і представлення даних, забезпечуючи гнучкість і масштабованість вебдодатків.

2.2 Розробка бази даних

Етап створення бази даних є критично важливим у процесі розробки системи, оскільки база даних відіграє ключову роль у забезпеченні збереження, організації та доступу до великого обсягу інформації, необхідної для функціонування каталогу. Ось основні аспекти важливості цього етапу:

Організація даних: база даних дозволяє структуровано та ефективно зберігати дані про продукти, користувачів, замовлення та інші важливі аспекти бізнесу. Це спрощує управління даними та доступ до них.

Масштабованість: чітко структурована база даних дозволяє легко масштабувати каталог, додавати нові продукти, категорії та функціональні можливості без зниження продуктивності.

Ефективність: оптимізована база даних забезпечує швидкий доступ до даних, що критично важливо для забезпечення хорошого досвіду користувача, особливо при високому навантаженні на систему.

Безпека: надійна база даних допомагає забезпечити безпеку інформації про клієнтів, що є критично важливим для збереження довіри клієнтів.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтеграція: база даних служить основою для інтеграції з іншими системами, такими як CRM, ERP, а також зовнішніми API.

Процес створення бази даних для каталогу є таким:

Визначення вимог: розробка детального переліку вимог до бази даних, з урахуванням специфіки товарів, користувачів та бізнес-процесів системи.

Проектування схеми: створення схеми бази даних, яка включає всі необхідні таблиці, поля, типи даних, індекси, зовнішні та внутрішні ключі. Це може включати таблиці для продуктів, користувачів, замовлень, категорій продуктів, відгуків тощо.

Розробка бази даних: використання SQL для створення бази даних та таблиць згідно з проектом.

Заповнення даними: введення первинних даних у базу даних, можливо, через імпорт з існуючих джерел або вручну.

Тестування та оптимізація: перевірка бази даних на предмет помилок, повільних запитів та інших потенційних проблем. Оптимізація за допомогою індексування, рефакторингу запитів або зміни структури.

Захист та резервне копіювання: реалізація заходів безпеки для захисту даних та створення стратегії регулярного резервного копіювання.

Деплоймент: розгортання бази даних на сервері та її інтеграція з інтерфейсом каталогу.

Цей процес забезпечує створення надійної, ефективної та безпечної бази даних, готової до інтеграції з системою.

На рис. 2.2 зображено схему бази даних розроблюваного інтерактивного каталогу.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

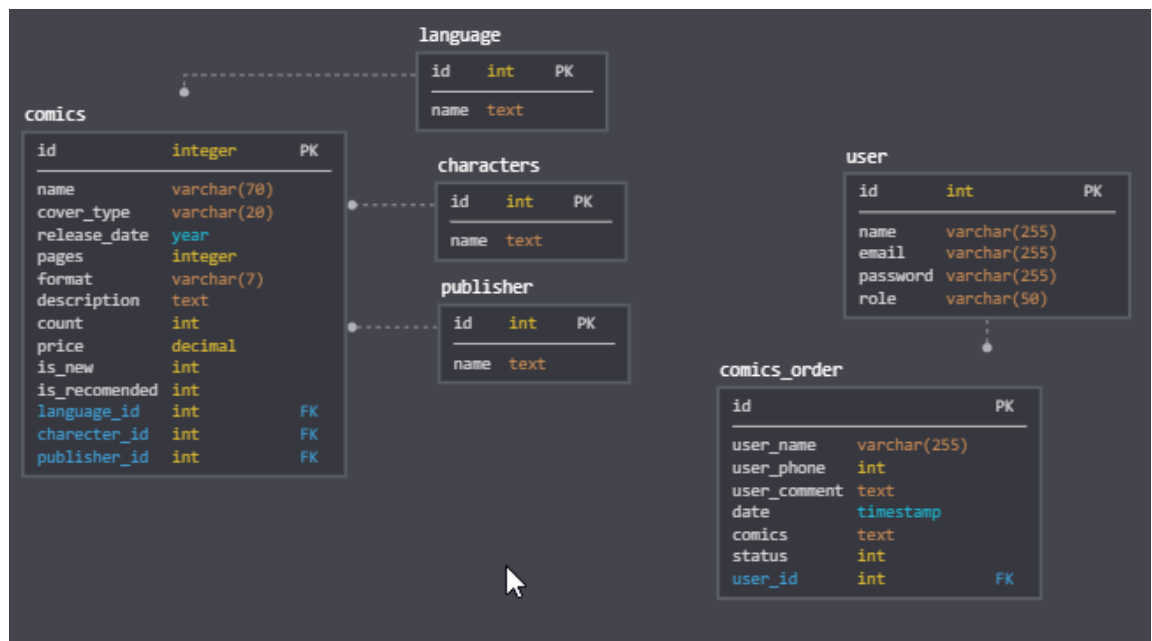


Рисунок 2.2 - Діаграма бази даних

Ця база даних містить кілька пов'язаних таблиць, що здаються частиною системи для управління коміксами. Ось їхній опис:

comics – таблиця для зберігання інформації про комікси. Включає поля для назви, типу обкладинки, дати випуску, кількості сторінок, формату, опису, кількості екземплярів, ціни, позначки про новизну, рекомендації, а також зовнішні ключі для мови, персонажа та видавця.

language – таблиця для мов, на які перекладено комікси. Містить ідентифікатор та назву мови.

characters – таблиця для персонажів, що з'являються в коміксах. Включає поля для ідентифікатора та назви персонажа.

publisher – таблиця для видавців коміксів. Містить ідентифікатор та назву видавця.

user – таблиця для користувачів системи. Містить поля для ідентифікатора, імені, електронної пошти, пароля та ролі користувача.

comics_order – таблиця для замовлень коміксів. Включає поля для ідентифікатора, імені користувача, телефону користувача, коментаря, дати замовлення, коміксів, статусу замовлення та зовнішнього ключа до таблиці користувачів.

Процес створення цієї бази даних за допомогою MySQL [11, 12]:

1. Встановлення MySQL.

2. Створення бази даних:

```
CREATE DATABASE comics_db;
```

```
USE comics_db;
```

3. Створення таблиць: для кожної таблиці використовується окрема команда CREATE TABLE. Наприклад, для таблиці comics:

```
CREATE TABLE comics (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(70),  
    cover_type VARCHAR(20),  
    release_date YEAR,  
    pages INT,  
    format VARCHAR(7),  
    description TEXT,  
    count INT,  
    price DECIMAL,  
    is_new INT,  
    is_recomended INT,  
    language_id INT,  
    character_id INT,  
    publisher_id INT,  
    FOREIGN KEY (language_id) REFERENCES language(id),  
    FOREIGN KEY (character_id) REFERENCES characters(id),  
    FOREIGN KEY (publisher_id) REFERENCES publisher(id)  
);
```

3. Аналогічно створюються інші таблиці.

4. Створення взаємозв'язків: зазвичай використовують зовнішні ключі для створення взаємозв'язків між таблицями, що дозволяє зберігати цілісність даних.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей процес дозволяє систематизовано побудувати базу даних, що підтримує управління коміксами, замовленнями та користувачами.

2.3 Розробка функціональних алгоритмів роботи програми

Маючи загальний алгоритм роботи вебдодатку, що отримано попередньо, можемо почати безпосередню роботу над проектом.

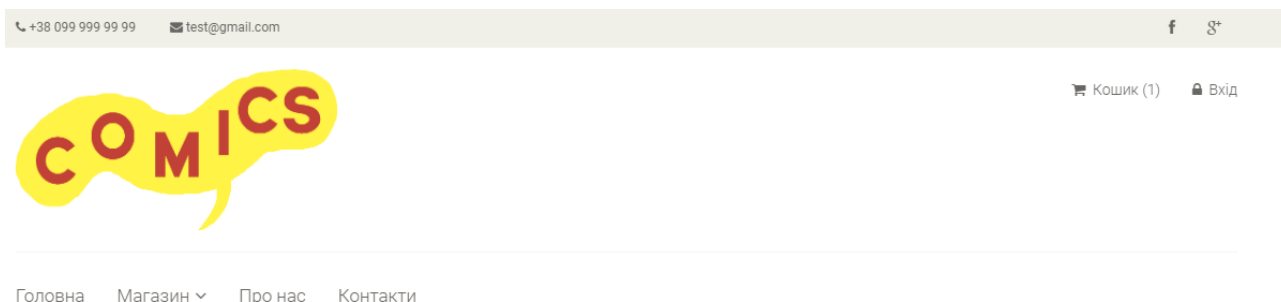
Одним з найважливіших компонентів розроблюваної системи є авторизація користувачів. Для авторизації та роботи з акаунтами було створено клас User, в якому написані основні функції роботи з користувачами: реєстрація, авторизація, редагування профіля та перевірка коректності введених даних.

Приклад реалізації реєстрації:

```
public static function register($name, $email, $password)
{
    $db = Db::getConnection();
    $sql = 'INSERT INTO user (name, email, password) '
        . 'VALUES (:name, :email, :password)';
    $result = $db->prepare($sql);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
    $result->bindParam(':email', $email, PDO::PARAM_STR);
    $result->bindParam(':password', $password, PDO::PARAM_STR);
    return $result->execute();
}
```

На рис. 2.3 представлено інтерфейс форми реєстрації.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				28
Змн.	Арк.	№ докум.	Підпис	Дата		



Реєстрація на сайті

Ім'я
E-mail
Пароль
Реєстрація

Рисунок 2.3 - Реєстрація користувача

Надалі заходячи у особистий кабінет він має можливість змінювати власні дані. Користувач може бути адміністратором та просто користувачем. Адміністратор має змогу додавати, редагувати та видаляти інформацію, звичайні користувачі можуть тільки переглядати товари та робити замовлення, а також редагувати свою інформацію.

Всі дані про користувачів зберігаються у спеціально створеній таблиці (рис. 2.4).

		<u>id</u>	<u>name</u>	<u>email</u>	<u>password</u>	<u>role</u>
<input type="checkbox"/>		9	admin	admin@gmail.com	123456	admin
<input type="checkbox"/>		10	test	test@gmail.com	123456	

Рисунок 2.4 - Таблиця з користувачами

Інші дані також зберігаються в спеціальних таблицях. Таблиця персонажів була створена для зручності пошуку коміксів за персонажами (рис. 2.5).

←T→ ▼ id name				
<input type="checkbox"/>				1 Бетмен
<input type="checkbox"/>				2 Месники
<input type="checkbox"/>				4 Шазам
<input type="checkbox"/>				5 Капітан Марвел
<input type="checkbox"/>				6 Аквамен
<input type="checkbox"/>				7 Капітан Америка

Рисунок 2.5 - Таблиця з персонажами

Також створені такі таблиці для мови перекладу(чи оригіналу) коміксу та видавництва, які їх публікують (рис. 2.6 – 2.7).

←T→ ▼ id name				
<input type="checkbox"/>				1 English
<input type="checkbox"/>				3 Українська
<input type="checkbox"/>				4 Японська

Рисунок 2.6 - Таблиця з мовами

←T→ ▼ id name				
<input type="checkbox"/>				1 MARVEL
<input type="checkbox"/>				2 DC
<input type="checkbox"/>				3 Vertigo

Рисунок 2.7 - Таблиця з видавництвами

Основною є таблиця comics, в якій зберігаються дані про комікси, а також ідентифікатори, які пов'язують її з іншими таблицями (рис. 2.8).

id	name	publisher_id	cover_type	release_date	pages	format	language_id	character_id	description	count	price	is_new	is_recommended
1	Нові Месники. Том 1. Все помирає	1	Твердий	2019	156	16x24		2	2 Найсильніша та найрозумніша команда Всесвіту Марве...	10	200.00	0	0
2	Doomsday Clock #1	2	Твердий	2019	38	17x26		1	3 Всесвіт DC стикається з найбільшою загрозою - проф...	10	140.00	1	1
3	Бетмен. Повернення темного лицаря	2	Твердий	2016	264	21x28		2	1 Шедевр сучасного графічного роману, на сторінках я...	5	440.00	0	1
4	Вартові	3	Твердий	2018	523	16,5x25		3	8 Один із головних бестселерів світу коміксів. Єдини...	5	495.00	0	1
5	Шазам!	2	Твердий	2019	200	17x26		2	4 Біллі Бетсон, простий підліток із Філадельфії, ніч...	10	405.00	0	0
6	Життя Капітана Марвел	1	М'який	2019	128	16x27		3	5 Вона одна з найсильніших героїнь не лише на Землі,...	5	200.00	0	0
7	Аквамен. Книга 2. Інші	2	Твердий	2019	160	16,5x25		3	6 На думку MTV Geek, «історія Аквамена, як і раніше,...	5	250.00	0	1
8	Капітан Америка. Смерть Капітана Америка	1	Твердий	2019	572	17x26		2	7 Він був кумиром мільйонів, прикладом для наслідува...	5	695.00	0	0
9	Batman Secret Files #1	2	М'який	2018	39	17x25		1	1 Пориньте в раніше невідомі розслідування Бетмена в...	5	145.00	0	1

Рисунок 2.8 - Таблиця з коміксами

Також створена таблиця, в яку записуються дані про замовлення користувачів (рис. 2.9).

id	user_name	user_phone	user_comment	user_id	date	comics	status
1	Тест	990999999	ТЕСТОВЕ ПОВІДОМЛЕННЯ	1	2024-05-18 14:09:18	{"2":1}	1
2	Тест	990999999	test	1	2024-05-18 15:23:06	{"2":1}	1
3	Юра	665656565	test	1	2024-05-18 14:03:06	{"1":1}	1
4	ТЕСТ	737373737	ТЕСТУЄМО ВІДПРАВЛЕННЯ ЗАМОВЛЕННЯ	1	2024-05-19 10:32:26	{"8":1,"7":1}	1

Рисунок 2.9 - Таблиця з замовленнями

Таким чином відбувається контроль і аналіз даних, що надає користувач.

2.4 Реалізація системи

На даному етапі розробки відбувається перехід до безпосередньої розробки ПЗ. На даний момент вирішені всі організаційні та технічні питання з приводу розроблюваного додатку. Чітко сформовано задачі, які повинний виконувати вебдодаток, вирішені питання по їх реалізації. Тому, притримуючись завдання та вище оговорених положень можна переходити до розробки інтерфейсу додатку.

Для забезпечення безпечної роботи вебресурсу налаштовуємо переадресацію сайту до файлу index.php. Переадресація дозволить заблокувати непотрібні файли на сервері і чітко контролювати дані, що будуть відображатися користувачам. Так як ми використовуємо Apache-сервер, то переадресацію

реалізуємо за допомогою конфігураційного файлу «.htaccess». Розглянемо конфігурації файлу «.htaccess»:

```
AddDefaultCharset utf-8
```

```
RewriteEngine on
```

```
RewriteBase /
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule ^(.*)$ index.php
```

Надалі ми притримуємось MVC патерну для моделювання схеми нашого проекту.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

У рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами. Модель (Model) відповідає за зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле) і передає дані у модель.

Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Вигляд може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації.

Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Модель інкапсулює ядро даних і основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу даних.

Вигляд може мати декілька взаємопов'язаних областей, наприклад різні таблиці і поля форм, в яких відображаються дані.

У функції контролера входить відстеження визначених подій, що виникають в результаті дій користувача. Контролер дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас.

Таким чином ми створюємо шаблон для зовнішнього вигляду сайту, представляючи з середину частину, що стосується кожної сторінки окремо.

Наприклад наведемо фрагмент основного коду для головної сторінки:

```
<?php include ROOT . '/views/layouts/header.php'; ?>
```

```
<section>
```

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

```
<div class="row">
```

```
<div class="col-sm-3">
```

```
<div class="left-sidebar">
```

ВИВЕДЕННЯ КАТЕГОРІЙ ДЛЯ СОРТУВАННЯ

```
<h2>Каталог</h2>
```

```
<div class="panel-group category-products">
```

```
<p>Видавці</p>
```

```
<?php foreach ($publishers as $publisherItem): ?>
```

```
<div class="panel panel-default">
```

```
<div class="panel-heading">
```

```
<h4 class="panel-title">
```

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<a href="/publisher/<?php echo
$publisherItem['id'];?>">
    <?php echo $publisherItem['name'];?>
</a>
</h4>
</div>
</div>
<?php endforeach; ?>
<br>
<p>Персонажі</p>
<?php foreach ($character as $characterItem): ?>
    <div class="panel panel-default">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a href="/character/<?php echo
$characterItem['id'];?>">
                    <?php echo $characterItem['name'];?>
                </a>
            </h4>
        </div>
    </div>
<?php endforeach; ?>
<br>
<p>Мови</p>
<?php foreach ($language as $languageItem): ?>
    <div class="panel panel-default">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a href="/language/<?php echo
$languageItem['id'];?>">

```

		Осадчук Є.Р.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Чепюк Л.О.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

                <?php echo $languageItem['name'];?>
            </a>
        </h4>
    </div>
</div>
</div>
<?php endforeach; ?>
</div>
</div>
</div>

<div class="col-sm-9 padding-right">
    <div class="features_items"><!--features_items-->
        <h2 class="title text-center">Останні товари</h2>

        <?php foreach ($latestComics as $comics): ?>
            <div class="col-sm-4">
                <div class="product-image-wrapper">
                    <div class="single-products">
                        <div class="productinfo text-center">
                            
                            <h2><?php echo $comics['price']; ?> грн</h2>
                            <p>
                                <a href="/comics/<?php echo $comics['id']; ?>">
                                    <?php echo $comics['name']; ?>
                                </a>
                            </p>
                            <a href="#" class="btn btn-default add-to-cart" data-
id="<?php echo $comics['id']; ?>"><i class="fa fa-shopping-cart"></i>В ко-
шик</a>

```

		Осадчук Є.Р.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Чепюк Л.О.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        </div>
        <?php if ($comics['is_new']): ?>
            
        <?php endif; ?>
    </div>
</div>
</div>
</div>
<?php endforeach; ?>
СЛАЙДЕР ДЛЯ ВИВЕДЕННЯ РЕКОМЕНДОВАНИХ ТОВАРІВ
</div><!--features_items-->

```

```

    <div class="recommended_items"><!--recommended_items-->
        <h2 class="title text-center">Рекомендовані товари</h2>

        <div class="cycle-slideshow"
            data-cycle-fx=carousel
            data-cycle-timeout=5000
            data-cycle-carousel-visible=3
            data-cycle-carousel-fluid=true
            data-cycle-slides="div.item"
            data-cycle-prev="#prev"
            data-cycle-next="#next"
        >
            <?php if($sliderComics)foreach ($sliderComics as
$sliderItem): ?>
                <div class="item">
                    <div class="product-image-wrapper">
                        <div class="single-products">
                            <div class="productinfo text-center">

```

```


<h2><?php echo $sliderItem['price']; ?>
грн</h2>
<a href="/comics/<?php echo $sliderItem['id'];
?>">
<?php echo $comics['name']; ?>
</a>
<br/><br/>
<a href="#" class="btn btn-default add-to-cart"
data-id="<?php echo $sliderItem['id']; ?>"><i class="fa fa-shopping-cart"></i>В
КОШИК</a>
</div>
<?php if ($sliderItem['is_new']): ?>

<?php endif; ?>
</div>
</div>
</div>
</div>
<?php endforeach; ?>
</div>

<a class="left recommended-item-control" id="prev"
href="#recommended-item-carousel" data-slide="prev">
<i class="fa fa-angle-left"></i>
</a>
<a class="right recommended-item-control" id="next"
href="#recommended-item-carousel" data-slide="next">
<i class="fa fa-angle-right"></i>

```

		Осадчук Є.Р.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Чепюк Л.О.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```
</a>
```

```
</div>
```

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

```
</div>
```

```
</div>
```

```
</section>
```

```
<?php include ROOT . '/views/layouts/footer.php'; ?>
```

Відповідний код для контролера:

В цьому контролері прописано сценарій для головної сторінки та підключено дані з інших файлів, які будуть там відтворюватись. Вище наведено код головної сторінки, на якій є цикли з виведенням критеріїв пошуку, останніх коміксів та слайдеру.

```
public function actionIndex()
{
    $publishers = Publisher::getPublishersList();
    $character = Character::getCharactersList();
    $language = Language::getLanguagesList();
    $latestComics = Comics::getLatestComics(6);
    $sliderComics = Comics::getRecommendedComics();
    require_once(ROOT . '/views/site/index.php');
    return true;
}
```

Розглянемо детальніше функцію `Publisher::getPublishersList()`;

```
public static function getPublishersList()
{
```

```

$db = Db::getConnection();

$result = $db->query('SELECT id, name FROM publisher ORDER BY
id, name ASC');
$i = 0;
$publishersList = array();
while ($row = $result->fetch()) {
    $publishersList[$i]['id'] = $row['id'];
    $publishersList[$i]['name'] = $row['name'];
    $i++;
}
return $publishersList;
}

```

Ця функція призначена для отримання списку видавців з бази даних. Вона є статичною, тому може бути викликана без створення екземпляру класу, до якого вона належить. Всередині функції спочатку встановлюється з'єднання з базою даних за допомогою методу `Db::getConnection()`. Це з'єднання потім використовується для виконання SQL-запиту, який вибирає поля `id` і `name` з таблиці `publisher` та сортує результати за `id` та `name` в порядку зростання.

Після виконання запиту, результати обробляються в циклі `while`. В кожній ітерації циклу дані одного видавця (вибрані з результату запиту) додаються до масиву `$publishersList`. Кожен видавець представлений як асоціативний масив з ключами `id` та `name`, що відповідають ідентифікатору та назві видавця відповідно. Індексація масиву `$publishersList` ведеться за допомогою змінної `$i`, яка інкрементується з кожною ітерацією циклу. Наприкінці функції масив `$publishersList`, що містить всі витягнуті дані, повертається як результат роботи функції.

Такий підхід дозволяє ефективно зібрати всю необхідну інформацію про видавців для подальшого використання в програмі, наприклад, для відображення списку видавців у вебінтерфейсі або для інших операцій з даними.

		Осадчук Є.Р.			ПІЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Також розглянемо функцію Comics::getLatestComics().

```
public static function getLatestComics($count =
self::SHOW_BY_DEFAULT)
{

    $db = Db::getConnection();

    $sql = 'SELECT id, name, price FROM comics '
        . 'ORDER BY id DESC '
        . 'LIMIT :count';

    $result = $db->prepare($sql);
    $result->bindParam(':count', $count, PDO::PARAM_INT);
    $result->setFetchMode(PDO::FETCH_ASSOC);

    $result->execute();
    $i = 0;
    $comicsList = array();
    while ($row = $result->fetch()) {
        $comicsList[$i]['id'] = $row['id'];
        $comicsList[$i]['name'] = $row['name'];
        $comicsList[$i]['price'] = $row['price'];
        $i++;
    }
    return $comicsList;
}
```

Функція getLatestComics() призначена для отримання списку останніх коміксів з бази даних. Вона приймає параметр \$count, який визначає кількість коміксів, що буде вибрано; за замовчуванням це значення визначено константою SHOW_BY_DEFAULT класу.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

Спочатку в функції встановлюється з'єднання з базою даних за допомогою методу `Db::getConnection()`. Далі формується SQL-запит, який вибирає поля `id`, `name` і `price` з таблиці `comics`, сортує результати в порядку спадання за `id` (щоб отримати найновіші записи) і обмежує вибірку кількістю, зазначеною в параметрі `$count`.

Цей SQL-запит підготовлюється до виконання за допомогою методу `prepare()` і передачі параметра `$count` з використанням методу `bindParam()`, який забезпечує безпечне введення параметра у запит, прив'язуючи його до плейсхолдера `:count` в SQL-запиті з вказівкою його типу як цілочисельного (`PDO::PARAM_INT`).

Після підготовки, запит виконується методом `execute()`. Результати запиту обробляються в циклі `while`, де кожен рядок результату витягується і конвертується в асоціативний масив. Ці дані (ідентифікатор, назва та ціна коміксу) збираються у масив `$comicsList`. Індексція в масиві `$comicsList` ведеться за допомогою змінної `$i`, яка інкрементується з кожною ітерацією циклу.

Функція повертає масив `$comicsList`, який містить інформацію про останні комікси, що може бути використано в інших частинах програми для відображення списку новинок або для інших бізнес-логік.

Таким чином, використовуючи усі ці функції, ми можемо кожен раз лише заповнювати необхідний контент, збираючи як пазл кінцевий вигляд кожної сторінки.

Висновки до другого розділу

Було проведено розробку загального та функціональних алгоритмів системи, описано принципи виконання основних методів роботи. Розроблено базу даних для інтерактивного каталогу, яка містить шість таблиць. Також представлено деталізовану розробку сторінок.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Розроблено загальний алгоритм функціонування роботи програми. Була розроблена інтерфейсна частина, змодельовані вкладки сайту, та спроектовано програмний код сайту.

В результаті було отримано готовий програмний продукт з функціональністю, який виконує всі поставлені задачі та вимоги.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНТЕРАКТИВНИМ КАТАЛОГОМ

3.1 Структура інтерфейсу системи

На даному етапі практично реалізовано сайт, тому необхідно пояснити його роботу. Розглянемо головну сторінку сайту, куди користувач потрапляє, щойно перейшовши за посиланням (рис. 3.1).

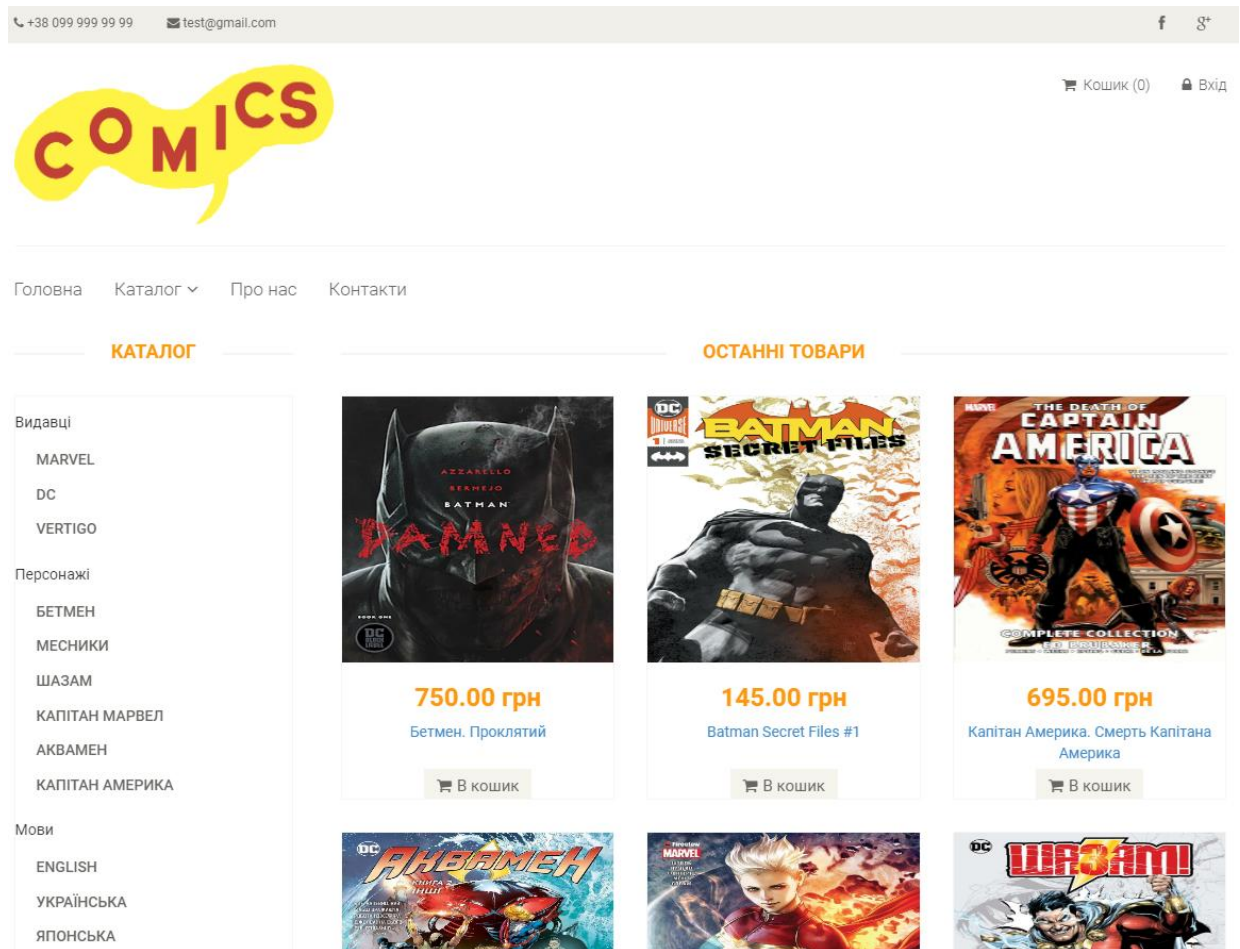


Рисунок 3.1 - Головна сторінка

Користувач бачить наступні пункти меню:

- «Головна» (безпосередньо стартова сторінка);
- «Каталог»;
- «Про нас»;
- «Контакти»;
- «Вхід/Реєстрація»;

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

- «Кошик».

Перша сторінка містить товари, які відвідувач може переглянути, вибрати та замовити.



Рисунок 3.2 - Сторінка перегляду коміксу

Далі відвідувач може як зареєструватись, так і увійти у власний акаунт та змінити інформацію за потребою (рис. 3.3).

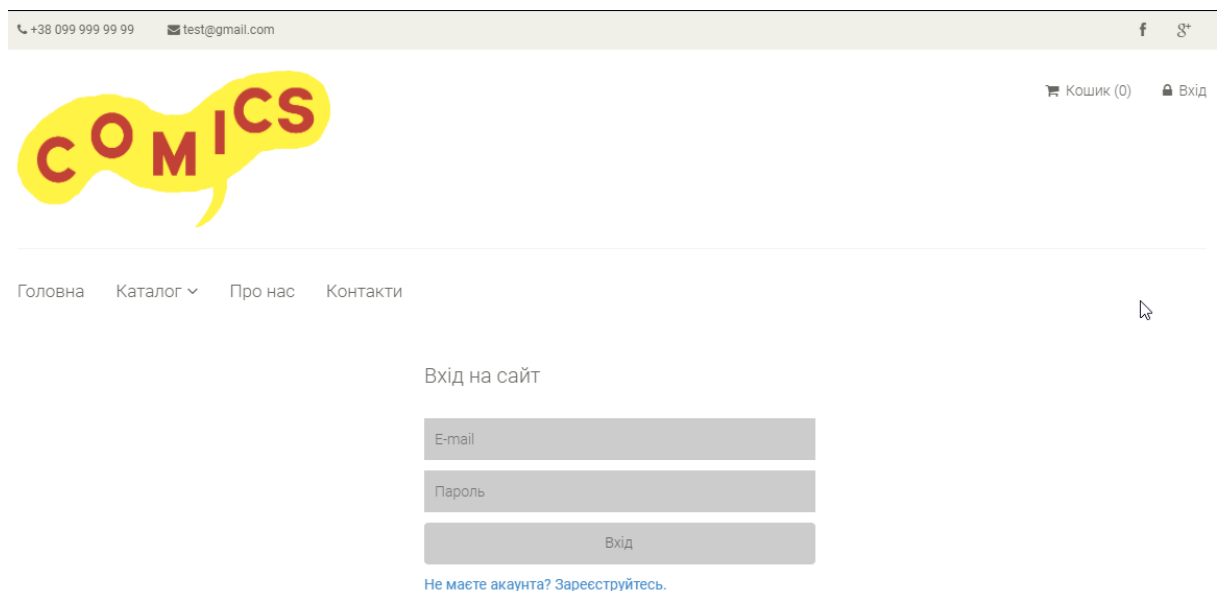


Рисунок 3.3 - Вхід на сайт

Одразу після реєстрації користувач заходить у власний кабінет, де він бачить привітання та посилання для редагування даних та на панель адміністратора (рис. 3.4).

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

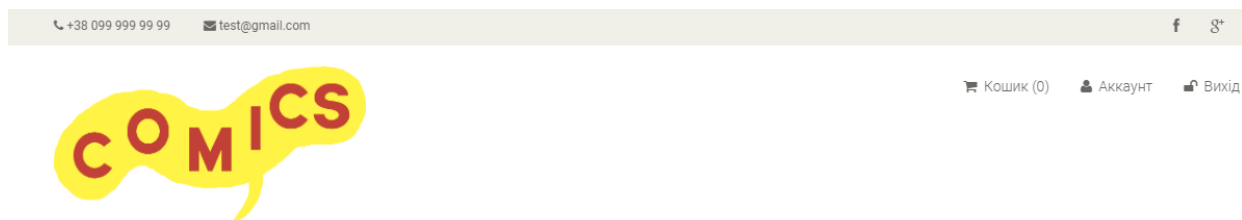


Рисунок 3.4 - Власний кабінет користувача сайту

Якщо до акаунту заходить адміністратор, то він може перейти на спеціальну сторінку (рис. 3.5):

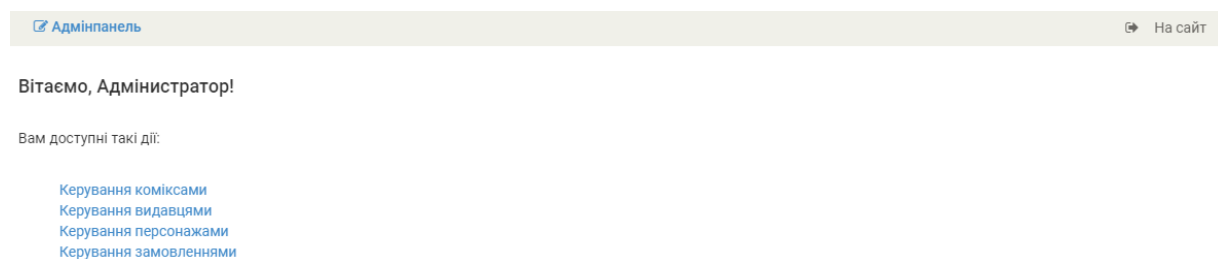


Рисунок 3.5 - Панель адміністратора

Якщо повернутись до загальної складової структури сайту, то переглядаючи вкладку «Каталог» відвідувачу пропонуються до його уваги всі комікси, наявні в каталозі.

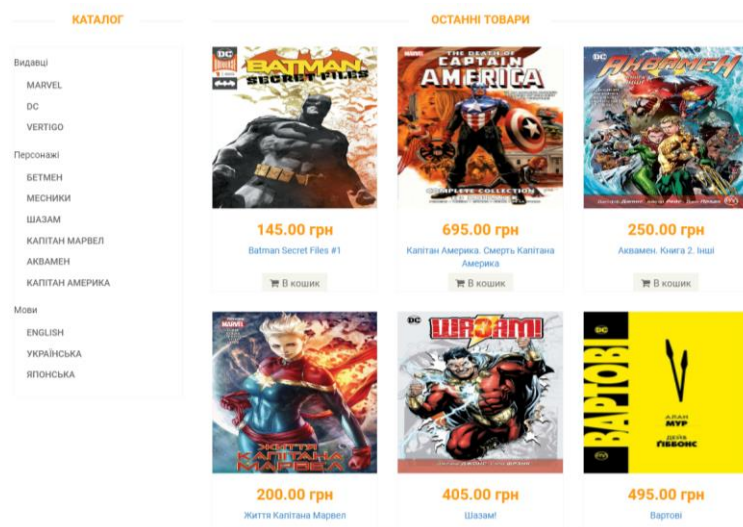


Рисунок 3.6 - Вкладка «Каталог»

		Осадчук Є.Р.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці «Про нас» користувач бачить інформацію про каталог.

Головна Каталог ▾ Про нас Контакти

Інформація про нас

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eget finibus nibh. Mauris ultricies, enim non scelerisque laoreet, libero lorem facilisis dui, eget semper magna tellus vel erat. Nullam in mi velit. Sed eleifend maximus velit, a scelerisque ante mollis vitae. Aliquam erat volutpat. In vel vulputate orci, sit amet rhoncus magna. Mauris enim massa, bibendum quis faucibus eu, pharetra vitae magna. Suspendisse et convallis est. Phasellus ac purus est. Maecenas egestas libero non condimentum laoreet. Aliquam ornare eleifend ex non viverra. Duis tincidunt semper ante a aliquam. Nunc convallis mollis sapien sed scelerisque. Donec vel dui sit amet justo efficitur egestas quis vel quam. Vivamus eget efficitur nunc, sit amet imperdiet quam. Proin sodales tortor vel porttitor auctor. Aliquam at mi ac diam euismod condimentum. Ut et vehicula urna, vel

Рисунок 3.7 - Сторінка «Про нас»

Якщо відвідувач хоче написати якесь повідомлення, то він має можливість зробити це на вкладці «Контакти».

Головна Каталог ▾ Про нас Контакти

Зворотній зв'язок

Є питання? Напишіть нам!

Ваша пошта

E-mail

Повідомлення

Повідомлення

Відправити

Рисунок 3.8 - Вкладка «Контакти»

Коли користувач визначився з коміксами та додав їх в кошик, він може перейти до нього та оформити замовлення.

Головна Каталог ▾ Про нас Контакти

КАТАЛОГ

КОШИК

Видавці

MARVEL

DC

VERTIGO

Персонажі

БЕТМЕН

МЕСНИКИ

Ви обрали такі товари:

Назва	Вартість, грн	Кількість, шт	Видалити	
Капітан Америка. Смерть Капітана Америка	695.00	1	✕	
Batman Secret Files #1	145.00	1	✕	
Загальна вартість, грн:				840

🛒 Оформити замовлення

Рисунок 3.9 - Оформлення замовлення

		Осадчук Є.Р.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

Обрано товарів: 2, на суму: 840, грн

Для оформлення замовлення
заповніть форму. Наш менеджер
зв'яжеться з Вами.

Ваше Ім'я

admin

Номер телефона

Коментар до замовлення

Повідомлення

Оформити

Рисунок 3.10 - Форма замовлення

3.2 Тестування системи

Переходимо безпосередньо до тестування розробленого сайту. На меті проведення дослідження є використання створеного програмного продукту в реальних умовах та перевірка отриманих результатів роботи.

Розглянемо замовлення, відправимо дані про замовлення (рис. 3.11).

КОШИК

Обрано товарів: 2, на суму: 840, грн

Для оформлення замовлення
заповніть форму. Наш менеджер
зв'яжеться з Вами.

Ваше Ім'я

Тест

Номер телефона

1111111111

Коментар до замовлення

тест

Оформити

Рисунок 3.11 - Відправляємо замовлення

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

Можна побачити, що дані про замовлення додалися в базу даних (рис. 3.12).

		id	user_name	user_phone	user_comment	user_id	date	comics	status
<input type="checkbox"/>		1	Тест	990999999	ТЕСТОВЕ ПОВІДОМЛЕННЯ	1	2024-05-18 14:09:18	{"2":1}	1
<input type="checkbox"/>		2	Тест	990999999	test	1	2024-05-18 15:23:06	{"2":1}	1
<input type="checkbox"/>		3	Юра	665656565	test	1	2024-05-18 14:03:06	{"1":1}	1
<input type="checkbox"/>		4	ТЕСТ	737373737	ТЕСТУЄМО ВІДПРАВЛЕННЯ ЗАМОВЛЕННЯ	1	2024-05-19 10:32:26	{"8":1,"7":1}	1
<input type="checkbox"/>		5	Тест	111111111	тест	9	2024-05-18 19:24:04	{"9":2,"8":1}	1

Рисунок 3.12 – Таблиця з замовленнями

Далі перейдемо до сторінки реєстрації, перевіряємо створення в базі даних нового запису (рис. 3.13).

Як бачимо на рис. 3.14, нового користувача з відповідними даними успішно створено.

Реєстрація на сайті

test

testtest@test.test

Реєстрація

Рисунок 3.13 - Спроба реєстрації
















 			<u>id</u>	<u>name</u>	<u>email</u>	<u>password</u>	<u>role</u>
<input type="checkbox"/>	   	9	admin	admin@gmail.com	123456	admin	
<input type="checkbox"/>	   	10	test	test@gmail.com	123456		
<input type="checkbox"/>	   	11	test	testtest@test.test	testtest		

Рисунок 3.14 - Поява відповідного запису в БД

Також спробуємо створити новий комікс (рис. 3.15).

[Адмінпанель](#)[На сайт](#)

Адмінпанель

Керування товарами

Додати товар

Список товарів

ID товара	Артикул	Назва товара	Ціна		
1	Нові Месники. Том 1. Все помирає	200.00	✎	✕	
2	Doomsday Clock #1	140.00	✎	✕	
3	Бетмен. Повернення темного лицаря	440.00	✎	✕	
4	Вартові	495.00	✎	✕	
5	Шазам!	405.00	✎	✕	
6	Життя Капітана Марвел	200.00	✎	✕	
7	Аквамен. Книга 2. Інші	250.00	✎	✕	
8	Капітан Америка. Смерть Капітана Америка	695.00	✎	✕	
9	Batman Secret Files #1	145.00	✎	✕	

Рисунок 3.15 - Керування коміксами

[Адмінпанель](#)

Адмінпанель

Керування товарами

Редагувати товар

Додати новий товар

Назва комікса

Бетмен. Проклятий

Перепліт

Твердий

Вартість, грн

750

Видавець

DC

Мова

Українська

Персонаж

Бетмен

Кількість сторінок

152

Формат

23,4x30 см

Кількість

13

Рік випуску

2019

Рисунок 3.16 - Параметри коміксу

Як можна побачити, комікс відобразився на головній сторінці (рис. 3.17).

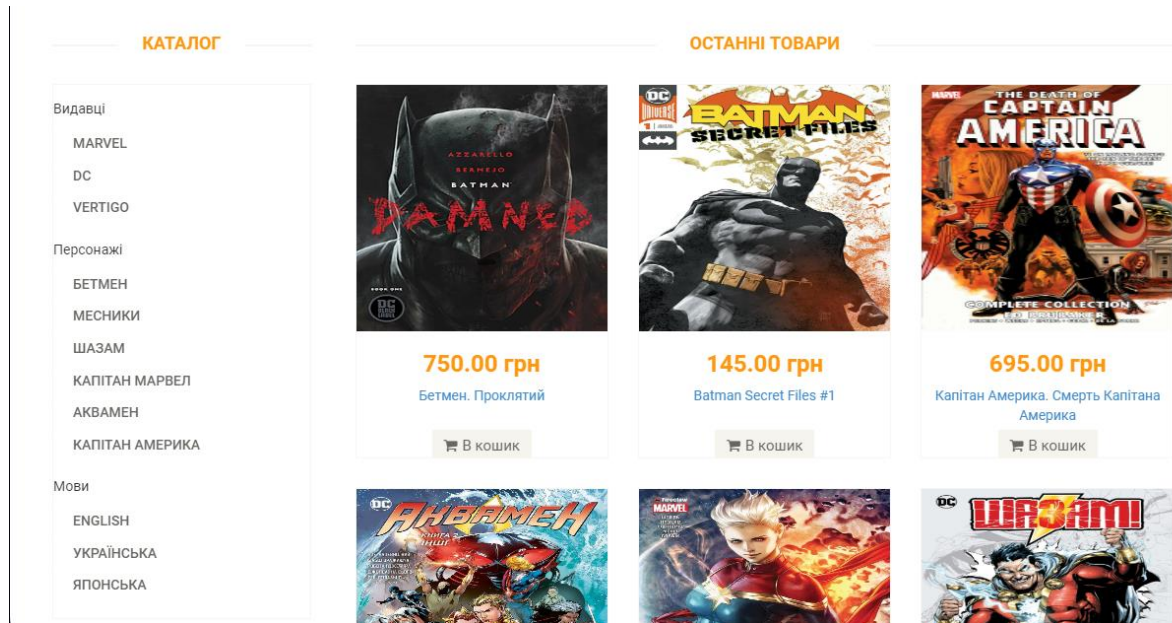


Рисунок 3.17 - Головна сторінка

Висновки до третього розділу

Було описано методологію користування створеним сайтом, та проведено тестування створеного додатку.

Опис роботи з сайтом включає в себе текстову та ілюстративну частину, які пояснюють принципи роботи з сайтом та опис елементів інтерфейсу сайту. При описі було роз'яснено призначення кожного елемента інтерфейсу у всіх вікнах.

При проведенні тестування додаток був запущений на виконання у реальних умовах, при яких показав стабільну і коректну роботу.

		Осадчук Є.Р.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У першому розділі випускної кваліфікаційної роботи бакалавра на тему «Розробка інтерактивного каталогу коміксів» було проаналізовано сайти схожого функціоналу та повністю сформовано уявлення про сайт, що розробляється.

Далі було визначено необхідні інструменти для реалізації програмного забезпечення, а також інтерфейсу та функціоналу програми, що розробляється.

Отримано теоретичний фундамент для подальшого написання програмного коду, використовуючи ООП, патерн MVC, мови програмування PHP та JavaScript, базу даних MySQL та бібліотеку jQuery.

У другому розділі було проведено розробку загального та функціональних алгоритмів системи, описано принципи виконання основних методів роботи. Розроблено базу даних для інтерактивного каталогу, яка містить шість таблиць. Також представлено деталізовану розробку сторінок.

Розроблено загальний алгоритм функціонування роботи програми. Була розроблена інтерфейсна частина, змодельовані вкладки сайту, та спроектовано програмний код сайту.

В результаті було отримано готовий програмний продукт з функціональністю, який виконує всі поставлені задачі та відповідає всім вимогам.

У третьому розділі було описано методологію користування створеним сайтом, та проведено тестування створеного додатку.

Опис роботи з сайтом включає в себе текстову та ілюстративну частину, які пояснюють принципи роботи з сайтом та опис елементів інтерфейсу сайту. При описі було роз'яснено призначення кожного елемента інтерфейсу у всіх вікнах.

При проведенні тестування додаток був запущений на виконання у реальних умовах, при яких показав стабільну і коректну роботу.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

В результаті виконання кваліфікаційної роботи було розроблено сайт, що повністю відповідає поставленим вимогам. Сайт є простим у користуванні, зі зрозумілим інтерфейсом.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://cosmic.com.ua> [Електронний ресурс] – Режим доступу до ресурсу: <https://cosmic.com.ua>.
2. <https://worldofcomics.ua> [Електронний ресурс] – Режим доступу до ресурсу: <https://worldofcomics.ua>.
3. Сесіл Мартін Р. Чиста архітектура / Роберт Сесіл Мартін., 2019. – 368 с.
4. Сесіл Мартін Р. Чистий код: створення, аналіз, рефакторинг / Роберт Сесіл Мартін., 2019. – 416 с.
5. Плюси і мінуси об'єктно-орієнтованого програмування [електронний ресурс] - 2015. Режим доступу: <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogo-programuvannya/>
6. DeBarros A. Practical SQL: A Beginner's Guide to Storytelling with Data / Anthony DeBarros., 2018. – 392 с.
7. Shields W. SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL (Coding & Programming - QuickStart Guides / Walter Shields., 2019. – 242 с.
8. Matt Zandstra. PHP Objects, Patterns, and Practice, 2013. Режим доступу: <https://www.apress.com/la/book/9781430260325>
9. Васильєв О. Програмування мовою PHP : навч. посіб. Збільшений формат В5. / Олексій Васильєв., 2022. – 368 с.
10. Документація бібліотеки JQuery [Електронний ресурс] // JQuery incorp. – 2016. – Режим доступу: <https://jquery.com>
11. SQL [електронний ресурс] - 2016. Режим доступу: <https://uk.wikibooks.org/wiki/SQL>
12. Duckett J. PHP & MySQL: Server-side Web Development / Jon Duckett., 2022. – 672 с.

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Чепюк Л.О.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Осадчук Є.Р.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Чепюк Л.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Допоміжний програмний код

```
.htaccess
```

```
AddDefaultCharset utf-8
```

```
RewriteEngine on
```

```
RewriteBase /
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule ^(.*)$ index.php
```

```
index.php
```

```
<?php
```

```
ini_set('display_errors',1);
```

```
error_reporting(E_ALL);
```

```
session_start();
```

```
define('ROOT', dirname(__FILE__));
```

```
require_once(ROOT.'/components/Autoload.php');
```

```
$router = new Router();
```

```
$router->run();
```

					ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Програмний код моделей

Клас Character

<?php

class Character

{

public static function getCharactersList()

{

\$db = Db::getConnection();

```
$result = $db->query('SELECT id, name FROM characters ORDER BY
id ASC');
```

\$charactersList = array();

\$i = 0;

while (\$row = \$result->fetch(PDO::FETCH_ASSOC)) {

```
$charactersList[$i]['id'] = $row['id'];
```

```
$charactersList[$i]['name'] = $row['name'];
```

\$i++;

}

return \$charactersList;

}

public static function getCharactersListAdmin()

{

\$db = Db::getConnection();

```
$result = $db->query('SELECT id, name FROM characters ORDER BY
id ASC');
```

\$charactersList = array();

		Осадчук Є.Р.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Чепюк Л.О.				56
Змн.	Арк.	№ докум.	Підпис	Дата		


```

    $i = 0;
    while ($row = $result->fetch(PDO::FETCH_ASSOC)) {
        $charactersList[$i]['id'] = $row['id'];
        $charactersList[$i]['name'] = $row['name'];
        $i++;
    }
    return $charactersList;
}

public static function deleteCharacterById($id)
{
    $db = Db::getConnection();
    $sql = 'DELETE FROM characters WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    return $result->execute();
}

public static function updateCharacterById($id, $name)
{
    $db = Db::getConnection();
    $sql = "UPDATE characters
        SET
            name = :name
        WHERE id = :id";
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $name, PDO::PARAM_STR);

    return $result->execute();
}

```

		Осадчук Є.Р.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Чепюк Л.О.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public static function getCharacterById($id)
{
    $db = Db::getConnection();
    $sql = 'SELECT * FROM characters WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->setFetchMode(PDO::FETCH_ASSOC);
    $result->execute();
    return $result->fetch();
}

public static function createCharacter($name)
{
    $db = Db::getConnection();
    $sql = 'INSERT INTO characters (name) '
        . 'VALUES (:name)';
    $result = $db->prepare($sql);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
    return $result->execute();
}
}

```

Клас Comics

<?php

```
class Comics
{
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 58 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
const SHOW_BY_DEFAULT = 6;
```

```
public static function getLatestComics($count = self::SHOW_BY_DEFAULT)
```

```
{
```

```
    $db = Db::getConnection();
```

```
    $sql = 'SELECT id, name, price FROM comics '
```

```
        . 'ORDER BY id DESC '
```

```
        . 'LIMIT :count';
```

```
    $result = $db->prepare($sql);
```

```
    $result->bindParam(':count', $count, PDO::PARAM_INT);
```

```
    $result->setFetchMode(PDO::FETCH_ASSOC);
```

```
    $result->execute();
```

```
    $i = 0;
```

```
    $comicsList = array();
```

```
    while ($row = $result->fetch()) {
```

```
        $comicsList[$i]['id'] = $row['id'];
```

```
        $comicsList[$i]['name'] = $row['name'];
```

```
        $comicsList[$i]['price'] = $row['price'];
```

```
        $i++;
```

```
    }
```

```
    return $comicsList;
```

```
}
```

```
public static function getComicsListByPublisher($publisherId, $page = 1)
```

```
{
```

```
    $limit = Comics::SHOW_BY_DEFAULT;
```

```
    $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
```

```
    $db = Db::getConnection();
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 59 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

$sql = 'SELECT id, name, price FROM comics '
      . 'WHERE publisher_id = :publisher_id '
      . 'ORDER BY id ASC LIMIT :limit OFFSET :offset';

$result = $db->prepare($sql);
$result->bindParam(':publisher_id', $publisherId, PDO::PARAM_INT);
$result->bindParam(':limit', $limit, PDO::PARAM_INT);
$result->bindParam(':offset', $offset, PDO::PARAM_INT);
$result->execute();

$i = 0;
$comics = array();
while ($row = $result->fetch()) {
    $comics[$i]['id'] = $row['id'];
    $comics[$i]['name'] = $row['name'];
    $comics[$i]['price'] = $row['price'];
    $i++;
}
return $comics;
}

```

```

public static function getComicsListByLanguage($languageId, $page = 1)
{
    $limit = Comics::SHOW_BY_DEFAULT;
    $offset = ($page - 1) * self::SHOW_BY_DEFAULT;

    $db = Db::getConnection();

    $sql = 'SELECT id, name, price FROM comics '
          . 'WHERE language_id = :language_id '

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 60 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        . 'ORDER BY id ASC LIMIT :limit OFFSET :offset';

$result = $db->prepare($sql);
$result->bindParam(':language_id', $languageId, PDO::PARAM_INT);
$result->bindParam(':limit', $limit, PDO::PARAM_INT);
$result->bindParam(':offset', $offset, PDO::PARAM_INT);

$result->execute();

$i = 0;
$comics = array();
while ($row = $result->fetch()) {
    $comics[$i]['id'] = $row['id'];
    $comics[$i]['name'] = $row['name'];
    $comics[$i]['price'] = $row['price'];
    $i++;
}
return $comics;
}

public static function getComicsListByCharacter($charactersId, $page = 1)
{
    $limit = Comics::SHOW_BY_DEFAULT;
    $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
    $db = Db::getConnection();
    $sql = 'SELECT id, name, price FROM comics '
        . 'WHERE character_id = :character_id '
        . 'ORDER BY id ASC LIMIT :limit OFFSET :offset';

    $result = $db->prepare($sql);

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 61 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $result->bindParam(':character_id', $charactersId,
PDO::PARAM_INT);

        $result->bindParam(':limit', $limit, PDO::PARAM_INT);
        $result->bindParam(':offset', $offset, PDO::PARAM_INT);

        $result->execute();

        $i = 0;
        $comics = array();
        while ($row = $result->fetch()) {
            $comics[$i]['id'] = $row['id'];
            $comics[$i]['name'] = $row['name'];
            $comics[$i]['price'] = $row['price'];
            $i++;
        }
        return $comics;
    }

    public static function getComicsById($id)
    {
        $db = Db::getConnection();
        $sql = 'SELECT * FROM comics WHERE id = :id';
        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $result->execute();
        return $result->fetch();
    }

    public static function getTotalComicsInPublisher($publisherId)
    {

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 62 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

$db = Db::getConnection();
$sql = 'SELECT count(id) AS count FROM comics WHERE pub-
lisher_id = :publisher_id';
$result = $db->prepare($sql);
$result->bindParam(':publisher_id', $publisherId, PDO::PARAM_INT);
$result->execute();
$row = $result->fetch();
return $row['count'];
}

```

```

public static function getTotalComicsInLanguage($languageId)
{
    $db = Db::getConnection();
    $sql = 'SELECT count(id) AS count FROM comics WHERE lan-
guage_id = :language_id';
    $result = $db->prepare($sql);
    $result->bindParam(':language_id', $languageId, PDO::PARAM_INT);
    $result->execute();
    $row = $result->fetch();
    return $row['count'];
}

```

```

public static function getTotalComicsInCharacter($characterId)
{
    $db = Db::getConnection();
    $sql = 'SELECT count(id) AS count FROM comics WHERE charac-
ter_id = :character_id';
    $result = $db->prepare($sql);
    $result->bindParam(':character_id', $characterId, PDO::PARAM_INT);
    $result->execute();
}

```

```

        $row = $result->fetch();
        return $row['count'];
    }

    public static function getComicsByIds($idsArray)
    {
        $db = Db::getConnection();
        $idsString = implode(',', $idsArray);
        $sql = "SELECT * FROM comics WHERE id IN ($idsString)";
        $result = $db->query($sql);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $i = 0;
        $comics = array();
        while ($row = $result->fetch()) {
            $comics[$i]['id'] = $row['id'];
            $comics[$i]['name'] = $row['name'];
            $comics[$i]['price'] = $row['price'];
            $i++;
        }
        return $comics;
    }
}

    public static function getRecommendedComics()
    {
        $db = Db::getConnection();
        $sql = 'SELECT id, name, price FROM comics '
            . 'WHERE is_recommended = "1" '
            . 'ORDER BY id DESC';
        $query = $db->query($sql);
        $comicsList = array();
    }

```



```

    $i = 0;
    while ($row = $query->fetch(PDO::FETCH_ASSOC)) {
        $comicsList[$i]['id'] = $row['id'];
        $comicsList[$i]['name'] = $row['name'];
        $comicsList[$i]['price'] = $row['price'];
        $i++;
    }
    return $comicsList;
}

public static function getComicsList()
{
    $db = Db::getConnection();
    $result = $db->query('SELECT id, name, price FROM comics ORDER
BY id ASC');
    $comicsList = array();
    $i = 0;
    while ($row = $result->fetch()) {
        $comicsList[$i]['id'] = $row['id'];
        $comicsList[$i]['name'] = $row['name'];
        $comicsList[$i]['price'] = $row['price'];
        $i++;
    }
    return $comicsList;
}

public static function deleteComicsById($id)
{
    $db = Db::getConnection();
    $sql = 'DELETE FROM comics WHERE id = :id';
    $result = $db->prepare($sql);

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 65 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

$result->bindParam(':id', $id, PDO::PARAM_INT);
return $result->execute();
}

public static function updateComicsById($id, $options)
{
    $db = Db::getConnection();
    $sql = "UPDATE comics
        SET
            name = :name,
            publisher_id = :publisher_id,
            cover_type = :cover_type,
            release_date = :release_date,
            pages = :pages,
            format = :format,
            language_id = :language_id,
            character_id = :character_id,
            description = :description,
            count = :count,
            price = :price,
            is_new = :is_new,
            is_recommended = :is_recommended
        WHERE id = :id";
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $options['name'], PDO::PARAM_STR);
    $result->bindParam(':publisher_id', $options['publisher_id'],
        PDO::PARAM_INT);
    $result->bindParam(':cover_type', $options['cover_type'],
        PDO::PARAM_STR);

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 66 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $result->bindParam(':release_date',          $options['release_date'],
PDO::PARAM_STR);

        $result->bindParam(':pages', $options['pages'], PDO::PARAM_INT);
        $result->bindParam(':format', $options['format'], PDO::PARAM_STR);
        $result->bindParam(':language_id',          $options['language_id'],
PDO::PARAM_INT);

        $result->bindParam(':character_id',          $options['character_id'],
PDO::PARAM_INT);

        $result->bindParam(':description',          $options['description'],
PDO::PARAM_STR);

        $result->bindParam(':count', $options['count'], PDO::PARAM_INT);
        $result->bindParam(':price', $options['price'], PDO::PARAM_STR);
        $result->bindParam(':is_new', $options['is_new'], PDO::PARAM_INT);
        $result->bindParam(':is_recommended',      $options['is_recommended'],
PDO::PARAM_INT);

        $result->execute();
    }

    public static function createComics($options)
    {
        $db = Db::getConnection();
        $sql = 'INSERT INTO comics '
            . '(name, publisher_id,cover_type, release_date,pages,format,lan-
guage_id,
            character_id,description,count,price,is_new, is_recommended)'
            . 'VALUES '
            . '(:name,      :publisher_id,:cover_type,:release_date,:pages,:for-
mat,:language_id,
            :character_id,:description,:count,:price,:is_new, :is_recommended)';
        $result = $db->prepare($sql);
        $result->bindParam(':name', $options['name'], PDO::PARAM_STR);

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 67 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $result->bindParam(':publisher_id',          $options['publisher_id'],
PDO::PARAM_INT);

        $result->bindParam(':cover_type',            $options['cover_type'],
PDO::PARAM_STR);

        $result->bindParam(':release_date',          $options['release_date'],
PDO::PARAM_STR);

        $result->bindParam(':pages', $options['pages'], PDO::PARAM_INT);
        $result->bindParam(':format', $options['format'], PDO::PARAM_STR);
        $result->bindParam(':language_id',          $options['language_id'],
PDO::PARAM_INT);

        $result->bindParam(':character_id',          $options['character_id'],
PDO::PARAM_INT);

        $result->bindParam(':description',          $options['description'],
PDO::PARAM_STR);

        $result->bindParam(':count', $options['count'], PDO::PARAM_INT);
        $result->bindParam(':price', $options['price'], PDO::PARAM_STR);
        $result->bindParam(':is_new', $options['is_new'], PDO::PARAM_INT);
        $result->bindParam(':is_recommended',      $options['is_recommended'],
PDO::PARAM_INT);

        if ($result->execute()) {
            return $db->lastInsertId();
        }
        return 0;
    }

    public static function getImage($id)
    {
        $noImage = 'no-image.jpg';
        $path = '/upload/images/comics/';
        $pathToProductImage = $path . $id . '.jpg';
    }

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 68 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        if (file_exists($_SERVER['DOCUMENT_ROOT'].$pathToProductImage)) {
            return $pathToProductImage;
        }
        return $path . $noImage;
    }

}

```

Клас Language

```
<?php
```

```
class Language
```

```
{
```

```
    public static function getLanguagesList()
```

```
    {
```

```
        $db = Db::getConnection();
```

```
        $result = $db->query('SELECT id, name FROM language ORDER BY
id, name ASC');
```

```
        $i = 0;
```

```
        $languagesList = array();
```

```
        while ($row = $result->fetch()) {
```

```
            $languagesList[$i]['id'] = $row['id'];
```

```
            $languagesList[$i]['name'] = $row['name'];
```

```
            $i++;
```

```
        }
```

```
        return $languagesList;
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ПІЗ.КР.Б-121-24-ПІЗ | Арк. |
| | | Чепюк Л.О. | | | | 69 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
}
```

```
public static function getLanguagesListAdmin()
```

```
{
```

```
    $db = Db::getConnection();
```

```
    $result = $db->query('SELECT id, name FROM language ORDER BY  
id ASC');
```

```
    $languagesList = array();
```

```
    $i = 0;
```

```
    while ($row = $result->fetch(PDO::FETCH_ASSOC)) {
```

```
        $languagesList[$i]['id'] = $row['id'];
```

```
        $languagesList[$i]['name'] = $row['name'];
```

```
        $i++;
```

```
    }
```

```
    return $languagesList;
```

```
}
```

```
public static function deleteLanguageById($id)
```

```
{
```

```
    $db = Db::getConnection();
```

```
    $sql = 'DELETE FROM language WHERE id = :id';
```

```
    $result = $db->prepare($sql);
```

```
    $result->bindParam(':id', $id, PDO::PARAM_INT);
```

```
    return $result->execute();
```

```
}
```

```
public static function updateLanguageById($id, $name)
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 70 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

{
    $db = Db::getConnection();
    $sql = "UPDATE language
        SET
            name = :name
        WHERE id = :id";

    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $name, PDO::PARAM_STR);

    return $result->execute();
}

```

```

public static function getLanguageById($id)
{
    $db = Db::getConnection();
    $sql = 'SELECT * FROM language WHERE id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->setFetchMode(PDO::FETCH_ASSOC);
    $result->execute();
    return $result->fetch();
}

```

```

public static function createLanguage($name)
{
    $db = Db::getConnection();

    $sql = 'INSERT INTO language (name) '

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 71 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
. 'VALUES (:name)';
```

```
$result = $db->prepare($sql);
```

```
$result->bindParam(':name', $name, PDO::PARAM_STR);
```

```
return $result->execute();
```

```
}
```

```
}
```

Клас Order

```
<?php
```

```
class Order
```

```
{
```

```
    public static function save($userName, $userPhone, $userComment,  
$userId, $comics)
```

```
{
```

```
    $db = Db::getConnection();
```

```
    $sql = 'INSERT INTO comics_order (user_name, user_phone,  
user_comment, user_id, comics) '
```

```
        . 'VALUES (:user_name, :user_phone, :user_comment, :user_id,  
:comics)';
```

```
    $result = $db->prepare($sql);
```

```
    $result->bindParam(':user_name', $userName, PDO::PARAM_STR);
```

```
    $result->bindParam(':user_phone', $userPhone, PDO::PARAM_STR);
```

```
    $result->bindParam(':user_comment', $userComment,  
PDO::PARAM_STR);
```

```
    $result->bindParam(':user_id', $userId, PDO::PARAM_STR);
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 72 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |


```

$result->bindParam(':comics', $comics, PDO::PARAM_STR);

return $result->execute();
}

public static function getOrdersList()
{

    $db = Db::getConnection();

    $result = $db->query('SELECT id, user_name, user_phone, date, status
FROM comics_order ORDER BY id DESC');
    $ordersList = array();
    $i = 0;
    while ($row = $result->fetch()) {
        $ordersList[$i]['id'] = $row['id'];
        $ordersList[$i]['user_name'] = $row['user_name'];
        $ordersList[$i]['user_phone'] = $row['user_phone'];
        $ordersList[$i]['date'] = $row['date'];
        $ordersList[$i]['status'] = $row['status'];
        $i++;
    }
    return $ordersList;
}

public static function getStatusText($status)
{
    switch ($status) {
        case '1':
            return 'Новый заказ';

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ПІЗ.КР.Б-121-24-ПІЗ | Арк. |
| | | Чепюк Л.О. | | | | 73 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        break;
    case '2':
        return 'В обработке';
        break;
    case '3':
        return 'Доставляется';
        break;
    case '4':
        return 'Закрыт';
        break;
    }
}

public static function getOrderById($id)
{
    $db = Db::getConnection();

    $sql = 'SELECT * FROM comics_order WHERE id = :id';

    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->setFetchMode(PDO::FETCH_ASSOC);
    $result->execute();
    return $result->fetch();
}

public static function deleteOrderById($id)
{
    $db = Db::getConnection();

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 74 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $sql = 'DELETE FROM comics_order WHERE id = :id';
        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        return $result->execute();
    }

```

```

    public static function updateOrderByid($id, $userName, $userPhone, $userComment, $date, $status)
    {
        $db = Db::getConnection();
        $sql = "UPDATE comics_order
            SET
                user_name = :user_name,
                user_phone = :user_phone,
                user_comment = :user_comment,
                date = :date,
                status = :status
            WHERE id = :id";

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        $result->bindParam(':user_name', $userName, PDO::PARAM_STR);
        $result->bindParam(':user_phone', $userPhone, PDO::PARAM_STR);
        $result->bindParam(':user_comment', $userComment, PDO::PARAM_STR);
        $result->bindParam(':date', $date, PDO::PARAM_STR);
        $result->bindParam(':status', $status, PDO::PARAM_INT);
        return $result->execute();
    }

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 75 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
}
```

Клас User

```
<?php
```

```
class User
```

```
{
```

```
    public static function register($name, $email, $password)
```

```
    {
```

```
        $db = Db::getConnection();
```

```
        $sql = 'INSERT INTO user (name, email, password) '
```

```
            . 'VALUES (:name, :email, :password)';
```

```
        $result = $db->prepare($sql);
```

```
        $result->bindParam(':name', $name, PDO::PARAM_STR);
```

```
        $result->bindParam(':email', $email, PDO::PARAM_STR);
```

```
        $result->bindParam(':password', $password, PDO::PARAM_STR);
```

```
        return $result->execute();
```

```
    }
```

```
    public static function edit($id, $name, $password)
```

```
    {
```

```
        $db = Db::getConnection();
```

```
        $sql = "UPDATE user
```

```
            SET name = :name, password = :password
```

```
            WHERE id = :id";
```

```
        $result = $db->prepare($sql);
```

```
        $result->bindParam(':id', $id, PDO::PARAM_INT);
```

```
        $result->bindParam(':name', $name, PDO::PARAM_STR);
```

```
        $result->bindParam(':password', $password, PDO::PARAM_STR);
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 76 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        return $result->execute();
    }

    public static function checkUserData($email, $password)
    {
        $db = Db::getConnection();

        $sql = 'SELECT * FROM user WHERE email = :email AND password
= :password';

        $result = $db->prepare($sql);
        $result->bindParam(':email', $email, PDO::PARAM_INT);
        $result->bindParam(':password', $password, PDO::PARAM_INT);
        $result->execute();
        $user = $result->fetch();

        if ($user) {
            return $user['id'];
        }
        return false;
    }

    public static function auth($userId)
    {
        $db = Db::getConnection();
        $_SESSION['user'] = $userId;
    }

    public static function checkLogged()
    {
        $db = Db::getConnection();
        if (isset($_SESSION['user'])) {
            return $_SESSION['user'];
        }
    }

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ПІЗ.КР.Б-121-24-ПІЗ | Арк. |
| | | Чепюк Л.О. | | | | 77 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

    }

    header("Location: /user/login");
}

public static function isGuest()
{
    if (isset($_SESSION['user'])) {
        return false;
    }
    return true;
}

public static function checkName($name)
{
    if (strlen($name) >= 2) {
        return true;
    }
    return false;
}

public static function checkPhone($phone)
{
    if (strlen($phone) >= 10) {
        return true;
    }
    return false;
}

public static function checkPassword($password)

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ПІЗ.КР.Б-121-24-ПІЗ | Арк. |
| | | Чепюк Л.О. | | | | 78 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

{
    if (strlen($password) >= 6) {
        return true;
    }
    return false;
}

public static function checkEmail($email)
{
    if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
        return true;
    }
    return false;
}

public static function checkEmailExists($email)
{
    $db = Db::getConnection();
    $sql = 'SELECT COUNT(*) FROM user WHERE email = :email';
    $result = $db->prepare($sql);
    $result->bindParam(':email', $email, PDO::PARAM_STR);
    $result->execute();

    if ($result->fetchColumn())
        return true;
    return false;
}

public static function getUserById($id)
{

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 79 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

$db = Db::getConnection();
$sql = 'SELECT * FROM user WHERE id = :id';
$result = $db->prepare($sql);
$result->bindParam(':id', $id, PDO::PARAM_INT);
$result->setFetchMode(PDO::FETCH_ASSOC);
$result->execute();
return $result->fetch();
}

}

```

Клас ComicsController

<?php

```

class ComicsController
{
    public function actionView($comicsId)
    {
        $publishers = Publisher::getPublishersList();
        $character = Character::getCharactersList();
        $language = Language::getLanguagesList();
        $comics = Comics::getComicsById($comicsId);
        require_once(ROOT . '/views/comics/view.php');
        return true;
    }
}

```

Контроллер SiteController

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 80 |


```
<?php
```

```
class SiteController
```

```
{
```

```
    public function actionIndex()
```

```
    {
```

```
        $publishers = Publisher::getPublishersList();
```

```
        $character = Character::getCharactersList();
```

```
        $language = Language::getLanguagesList();
```

```
        $latestComics = Comics::getLatestComics(6);
```

```
        $sliderComics = Comics::getRecommendedComics();
```

```
        require_once(ROOT . '/views/site/index.php');
```

```
        return true;
```

```
    }
```

```
    public function actionContact()
```

```
    {
```

```
        $userEmail = false;
```

```
        $userText = false;
```

```
        $result = false;
```

```
        if (isset($_POST['submit'])) {
```

```
            $userEmail = $_POST['userEmail'];
```

```
            $userText = $_POST['userText'];
```

```
            $errors = false;
```

```
            if (!User::checkEmail($userEmail)) {
```

```
                $errors[] = 'Неправильный email';
```

```
            }
```

```
            if ($errors == false) {
```

```
                $adminEmail = 'wirtuozombi@gmail.com';
```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | 81 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $message = "Текст: {$userText}. От {$userEmail}";
        $subject = 'Тема письма';
        $result = mail($adminEmail, $subject, $message);
        $result = true;
    }
}
require_once(ROOT . '/views/site/contact.php');
return true;
}

public function actionAbout()
{
    require_once(ROOT . '/views/site/about.php');
    return true;
}
}

```

| | | | | | | |
|------|------|--------------|--------|------|---------------------|------|
| | | Осадчук Є.Р. | | | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
| | | Чепюк Л.О. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 82 |