

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ МЕНЕДЖМЕНТУ ТОВАРІВ У
ІНТЕРНЕТ МАГАЗИНІ ІГОР
DEVELOPMENT OF A WEB APPLICATION TO MANAGE GOODS IN AN
ONLINE GAME STORE

Виконав: студент 4 курсу, групи КНТ-210сп
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ГОРДЄСВ Д.Л.

(ПРИЗВИЩЕ та ініціали)

Керівник ГЛАДКОВА О.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 «Комп'ютерні науки»
(код і найменування)
Освітня програма (спеціалізація) «Комп'ютерні науки»
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ГОРДЕЄВА Данила Леонідовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка вебзастосунку для менеджменту товарів у інтернет магазині ігор. Development of a Web Application to Manage Goods in an Online Game Store

керівник проєкту (роботи) к.т.н., ГЛАДКОВА Ольга Миколаївна,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “04” квітня 2022 року № 87

2. Строк подання студентом проєкту (роботи) 01 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області та постановка завдання. 2. Вибір інструментарію розробки. 3. Основні рішення щодо реалізації системи. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ГЛАДКОВА О.М., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст.викладач		

7. Дата видачі завдання “05” травня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Срок виконання етапів проекту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Данило ГОРДЕЄВ
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проекту (роботи)

_____ Ольга ГЛАДКОВА
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 69 с., 29 рис., 2 дод., 20 джерел.

ІНТЕРНЕТ-МАГАЗИН, ІГРИ, КОНТРОЛЕРИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, DJANGO.

Об'єкт дослідження – процес та методи ведення обліку товарів у магазинах.

Предмет дослідження – програмної реалізації складу інтернет-магазину.

Метою роботи є розроблення вебзастосунку для ведення обліку в інтернет-магазині ігор.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування Python, фреймворк Django, візуальні складові застосунку написані за допомогою HTML і CSS.

Результати. Розроблено програмне забезпечення у вигляді вебзастосунку для ведення обліку товарів в інтернет-магазині ігор.

Практична цінність роботи полягає у розробці зручного програмного забезпечення для ведення обліку товарів в інтернет-магазині ігор у вигляді вебзастосунку.

Висновки. Розроблено вебзастосунок для ведення обліку товарів в інтернет-магазині ігор.

Галузь використання – ігрова індустрія.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 69 pages, 29 figures, 2 appendixes, 20 sources.

ONLINE STORE, GAMES, CONTROLLERS, SOFTWARE, DJANGO.

The object of the study is the process and methods of store accounting.

The subject of the study is the software implementation of the online store.

The purpose of the work is to develop a web application for accounting in an online game store.

Materials, methods and techniques: object-oriented programming, Python programming language.

The results. Software in the form of a web application for keeping records of goods in an online game store has been developed.

The practical value of the work lies in the development of convenient software for keeping track of goods in the online game store in the form of a web application.

Conclusions. A web application for keeping records of goods in an online game store has been developed.

The field of use is the gaming industry.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області та постановка завдання	11
1.1 Поняття інтернет-магазину ігор.....	11
1.2 Управління товарним асортиментом та вимоги до менеджменту товарів у галузі ігрової індустрії.....	13
1.3 Аналіз наявних аналогічних застосунків	15
1.3.1 Застосунок “Magento”.....	16
1.3.2 Застосунок “Shopify”	18
1.3.3 Застосунок “PrestaShop”	20
1.4 Постановка завдання	22
1.4.1 Функціональність системи.....	23
1.4.2 Вимоги до інтерфейсу	23
1.4.3 Вимоги до продуктивності.....	23
1.4.4 Вимоги до безпеки	24
1.4.5 Експлуатаційні вимоги	24
1.4.6 Не функціональні вимоги до продукту.....	24
1.5 Висновки за розділом 1	25
2 Вибір інструментарію розробки.....	26
2.1 Вибір мови програмування.....	26
2.2 Вибір та обґрунтування Framework.....	28
2.3 Вибір системи керування базами даних	31
2.4 Проектування бази даних	35
2.5 Висновки за розділом 2	38
3 Основні рішення щодо реалізації системи.....	39
3.1 Реалізація структурної схеми та основних форм застосунку	39
3.2 Проектування інтерфейсу.....	43
3.3 Висновки за розділом 3.....	45

4 Експлуатація, тестування та експериментальне дослідження програми.....	46
4.1 Призначення й умови застосування програми.....	46
4.2 Вимоги до інтерфейсу.....	47
4.3 Розгортання застосунку.....	47
4.3.1 Звернення до програми.....	48
4.3.2 Вхідні й вихідні дані.....	49
4.4 Методика та результати тестування.....	49
Висновки	51
Перелік джерел посилань	52
Додаток А Опис програми.....	54
Додаток Б Слайди презентації	64

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CSS – Cascading Style Sheets (каскадні таблиці стилю);

ER – Entry Relationship (діаграма відображення концептуальної моделі бази даних);

UI – інтерфейс користувача;

БД – база даних;

ВЗ – вебзастосунок;

ІМ – інтернет-магазин;

ІМІ – інтернет-магазин ігор;

ІТС – інформаційно-телекомунікаційних систем;

ОС – операційна система;

ПК – персональний комп'ютер;

СКБД – система керування базами даних.

ВСТУП

Сучасний світ – світ комп'ютерних технологій. Люди не уявляють своє життя без гаджетів та зв'язку. Але сьогодні, комп'ютери, телефони, Інтернет дуже сильно полегшують наше життя. Ми можемо замовити таксі, їжу, показати свої документи, заплатити рахунки і зробити покупки, навіть не докладаючи зусиль.

Особливо нові можливості для розширення бізнесу та залучення більшого кола клієнтів надав розвиток електронної комерції. Особливої популярності набули інтернет-магазини, які забезпечують зручний та швидкий доступ до різноманітних товарів і послуг. Однак, власники інтернет-магазинів зустрічаються з новими викликами в управлінні товарами та підтриманні рівня якості зі зростанням конкуренції і збільшенням асортименту товарів.

Зокрема, у сегменті інтернет-магазинів ігор, які пропонують широкий вибір ігрових продуктів, виникає потреба в ефективному менеджменті товарів. Забезпечення актуальної інформації про товари, оновлення асортименту та швидке реагування на зміни на ринку стають вирішальними факторами для залучення та утримання клієнтів.

Актуальність розробки вебзастосунку для менеджменту товарів у інтернет-магазині ігор є актуальною задачею, що має на меті поліпшити процес управління товарним асортиментом та забезпечити зручний користувацький досвід для клієнтів.

Метою даної дипломної роботи є розробка такого вебзастосунку, який надає ефективний інструментарій для менеджменту товарів у інтернет-магазині ігор. Цей вебзастосунок має забезпечити можливість швидкого та зручного додавання, редагування та видалення товарів, оновлення їх параметрів, а також надати інші функціональні можливості, що сприятимуть підвищенню ефективності менеджменту товарів у інтернет-магазині ігор.

Для досягнення цієї мети необхідно вивчити сучасні тенденції та методи управління товарним асортиментом в інтернет-магазинах, проаналізувати потреби та вимоги клієнтів, а також розробити відповідну архітектуру та функціонал вебзастосунку.

Ця робота висвітлить основні аспекти розробки вебзастосунку для менеджменту товарів у інтернет-магазині ігор. Результати даної роботи можуть бути використані в практичній діяльності інтернет-магазинів для поліпшення процесів управління товарами та підвищення конкурентоспроможності на ринку.

Далі в роботі буде проведений аналіз потреб клієнтів та інтернет-магазинів, розглянуті методи та інструменти, використані для розробки вебзастосунку, описана архітектура та функціональні можливості розробленого вебзастосунку та зроблені висновки щодо комфорту та ефективності використання даного вебзастосунку.

Отже, дана дипломна робота має важливе практичне значення, оскільки вона спрямована на розробку вебзастосунку, який забезпечує ефективне управління товарами у інтернет-магазині ігор, що дозволить поліпшити конкурентоспроможність магазину та задовольнити потреби його клієнтів, власників інтернет-магазинів ігор.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

У першому розділі дипломної роботи буде зосереджено увагу на вивченні предметної області інтернет-магазинів ігор. У розділі буде розглянуто основні аспекти цієї області, проведено аналіз наявних аналогічних застосунків, проаналізовано їх плюси та мінуси, а також визначено функціональні вимоги для розробки та подальшої реалізації проекту.

1.1 Поняття інтернет-магазину ігор

Інтернет-магазини стали популярними завдяки зручності та широкому вибору товарів, які доступні для покупців в будь-який час та з будь-якого місця з Інтернет-підключенням.

У п. 8 ч. 1 ст. 3 Закону № 675 ІМ визначено як засіб для представлення або реалізації товару, роботи чи послуги шляхом вчинення електронного правочину [1].

Електронні правочини вчиняють у сфері електронної торгівлі, якою вважають господарську діяльність у сфері електронної купівлі-продажу, реалізації товарів дистанційним способом покупцю шляхом вчинення електронних правочинів із використанням інформаційно-телекомунікаційних систем (п. 2 ч. 1 ст. 3 Закону № 675) [1].

Електронна торгівля входить до поняття "електронна комерція", яке визначено в п. 1 ч. 1 ст. 3 Закону № 675 як відносини, спрямовані на отримання прибутку, що виникають під час вчинення правочинів щодо набуття, зміни або припинення цивільних прав й обов'язків, здійснені дистанційно з використанням ІТС, унаслідок чого в учасників таких відносин виникають права й обов'язки майнового характеру [1].

ІМ (також використовується онлайн-магазин або електронний магазин) - це віртуальна платформа або веб-сайт, який дозволяє покупцям

придбати товари або послуги через Інтернет. ІМ забезпечують можливість здійснення покупок онлайн, без необхідності фізичного відвідування звичайних магазинів.

ІМ надають різноманітні послуги та можливості. Клієнти можуть переглядати каталог товарів або використовувати пошукову функцію для знаходження конкретного товару або категорії товарів. Звичайно, людина може обрати бажані товари, додати їх до кошика і зробити замовлення, вказавши необхідну кількість, розмір, колір тощо.

ІМ зазвичай пропонують різні методи оплати, такі як кредитні картки, електронні платіжні системи, банківські перекази або післяплату. Після вибору або здійснення оплати замовлення, товари доставляються клієнту за допомогою кур'єрів або поштовими компаніями.

Крім того, ІМ забезпечують обробку замовлень, відстеження статусу доставки та надають підтримку клієнтам через електронну пошту, онлайн-чат або телефон.

Розвиток цифрового ринку та збільшення кількості людей, які грають у відеоігри та готові витратити на них кошти, зробив комп'ютерну гру таким самим звичним товаром, який люди купляють онлайн.

ІМІ також входять у сферу електронної комерції, де продаж товарів та послуг здійснюється в онлайн-режимі, з використанням електронних платіжних систем та доставки. ІМІ суттєво відрізняється від звичайних ІМ тим, що несе в собі суто розважальний характер, тому що тут люди можуть купити те, що принесе їм задоволення. В таких онлайн-магазинах клієнти можуть придбати різноманітні ігрові продукти, включаючи комп'ютерні ігри, консольні ігри, аксесуари тощо.

Станом на середину 2020 року у світі майже 3,1 млрд гравців. Це означає, що близько 40% населення грають в ігри у тій чи іншій формі. У дослідження були включені лише ті споживачі, які платять за ігри. Більшість гравців використовують кілька платформ: гральні приставки, комп'ютери та смартфони. З 3,1 млрд гравців тільки 8% є спеціалізованими користувачами

ігрових приставок. У цієї групи витрати в розрахунку на одного споживача найвищі. 1,5 млрд або 48% користувачів відеоігор надають перевагу комп'ютерним іграм [2].

У дослідженні DFC також є інформація, скільки геймерів проживає у конкретній частині світу. В Азії їх найбільше — 1,42 млрд платних користувачів ігор. Європа займає друге місце — 383 млн, у Північній Америці — 261 млн. [2].

Саме тому, головною перевагою ІМІ є те, що вони дозволяють клієнтам з усього світу мати доступ до широкого асортименту ігор та товарів, доступність покупки з будь-якого пристрою з Інтернет-підключенням, можливість зробити покупку в будь-який час, наявність різних акцій та знижок, незалежно від їх місця проживання.

1.2 Управління товарним асортиментом та вимоги до менеджменту товарів у галузі ігрової індустрії

Ще декілька років тому відсутність новинок, різновиди товарів, надмірність або неконкурентоспроможність асортименту могли взагалі ніяк не вплинути на онлайн-магазин, але з більшим розвитком онлайн-торгівлі постійна робота над асортиментом є необхідною функцією для конкуренції на ринку та прибутковості бізнесу. Робота з асортиментом дозволяє бізнесу не просто вижити, але процвітати в умовах жорсткої конкуренції та загальної доступності порівнянних товарів [3].

Асортимент - це різноманітний, але упорядкований перелік товарів у магазині за споживчими властивостями: видом, призначенням та характеристиками, такими як вага, колір, розмір, технічні характеристики, бренд [3].

Робота кожного ІМ починається з формування асортименту товарів і послуг. Це завдання вирішує модуль управління асортиментом товарів і послуг на основі інформації від постачальників та виробників товарів і

послуг. Основні функції модуля управління асортиментом товарів і послуг: відстеження стану ринків; моніторинг цін на товари і послуги; прийняття рішень про формування цін; прийняття рішень про формування асортименту; оновлення і відстеження даних про асортимент товарів і послуг конкретних ІМ. Управління кожним ІМ (модуль управління ІМ) визначається вимогами до його функціонування з боку бізнесу. Ці вимоги є характерною особливістю конкретного ІМ і представлені умовами формування асортименту, умовами оплати і доставки товарів, умовами ведення торгівлі, маркетинговими і рекламними акціями, змістовним наповненням вітрин [4].

У сучасному світі саме галузь ігрової індустрії є однією з найшвидше зростаючих галузей, тому менеджмент товарів повинен враховувати постійну зміну та розширення асортименту ігор.

Цифрові та онлайн ігри мають значний вплив на сучасний ринок ігрової індустрії. Вони стали надзвичайно популярними серед гравців різного віку та забезпечують гравцям можливість насолоджуватися грою в будь-який зручний для них час і місце, але в цей же час народжують нові можливості та потреби для управління товарним асортиментом у сфері ІМІ. Ця популярність створює великий попит на ігрові продукти, що впливає на менеджмент товарів у ІМІ.

Популярність цифрових та онлайн ігор створює виклики та можливості для менеджменту товарів у галузі ігрової індустрії. ІМІ повинні враховувати популярні жанри та платформи гри, оновлювати свій асортимент з урахуванням найновіших випусків ігор, а також надавати можливість клієнтам швидко знаходити та замовляти бажані товари. Управління товарним асортиментом повинно враховувати популярність ігрових продуктів та аксесуарів, пов'язаних з цими технологіями.

Більше того, розробники ігр постійно пропонують додатковий контент, доповнення та інші ігрові елементи, які можна придбати в магазинах. Це створює додаткові можливості для ІМІ управляти та пропонувати різні типи товарів для гравців.

Таким чином, цифрові та онлайн ігри мають значний вплив на сучасний ринок, популярність та менеджмент товарів у галузі ІМІ. Варто ретельно аналізувати ці тенденції та враховувати їх при розробці та управлінні асортиментом товарів.

Ці ідеї можна деталізувати та розширити відповідно до специфіки вашої дипломної роботи та актуальних досліджень у галузі ігрової індустрії.

1.3 Аналіз наявних аналогічних застосунків

Зростаючий ринок ІМ збільшують потребу у розробці застосунків для ведення обліку товарів. Існуючі вебзастосунки для ведення обліку асортиментів у ІМІ мають свої переваги та недоліки.

З переваг можна, звичайно, виділити те, що такі ВЗ допомагають організувати та контролювати товарний асортимент ІМ. Вони надають можливість додавати, видаляти та оновлювати товари, керувати їх характеристиками, цінами та наявністю.

Також вони добре оптимізують процес замовлення тим, що дозволяють автоматизувати процес замовлення товарів, включаючи обробку замовлень, відстеження статусу, підтвердження оплати та організацію доставки. Це сприяє збільшенню ефективності та швидкості обробки замовлень, а також поліпшенню задоволеності клієнтів.

Крім того, такі ВЗ дозволяють збирати та аналізувати дані про продажі, запаси, популярність товарів та інші показники. Саме це дозволяє зробити обґрунтовані рішення щодо управління товарним асортиментом, маркетингових акцій та стратегії розвитку бізнесу.

В цілому, розробка ВЗ для ведення обліку товарів у ІМ сприяє покращенню управління товарним асортиментом, оптимізації процесу замовлення, аналізу даних та покращенню взаємодії з клієнтами, що впливає на ефективність та успішність бізнесу в галузі ІМІ.

Незважаючи на переваги існуючих ВЗ для ведення обліку товарів у ІМ, вони також мають деякі недоліки.

По-перше, хочеться зазначити, що деякі ВЗ можуть бути складними у використанні, особливо для некваліфікованих користувачів. Це може вимагати додаткової підготовки або навчання персоналу, що може займати багато часу та бути досить витратним.

По-друге, частина ВЗ мають обмежений функціонал або можливості настроювання, що в свою чергу створює ряд обмежень для бізнесу, який потребує специфічних функцій або процесів, як от наприклад ІМІ.

Та основний недолік, що більшість ВЗ можуть бути вразливими до кібератак або злому. Тому в край важливо забезпечити належний рівень безпеки, щоб захистити конфіденційну інформацію про товари та клієнтів.

Враховуючи всі недоліки описані вище, розробка нового ВЗ для ведення обліку товарів у ІМ може сприяти усуненню цих проблем та врахуванню унікальних потреб бізнесу.

Але, перш за все, для розробки конкурентоспроможного застосунку слід проаналізувати вже існуючі аналогічні ВЗ та розглянути їх детально.

1.3.1 Застосунок “Magento”

Magento - одна з найкращих платформ для електронної комерції. Її творці створили її з використанням технології з відкритим вихідним кодом. Це дає користувачам можливість контролювати зовнішній вигляд, функціональність і контент свого ІМ без шкоди для зручності покупців. Крім того, Magento надає своїм користувачам безліч корисних інструментів і функцій. Сюди входять інструменти маркетингу, пошукової оптимізації та управління каталогом. Magento базується на PHP та використовує MySQL для збереження даних [5].

Загалом, платформа може задовольнити потреби магазину будь-якого розміру. Незалежно від того, обслуговується кілька клієнтів чи мільйон,

Magento пропонує як безкоштовний Magento Open Source, так і багатофункціональний Magento Commerce, що означає, що платформа може рости і розвиватися разом з вашим бізнесом. Часто компаніям доводиться змінювати платформи в міру зростання. Але Magento дозволяє легко залишатися на одному місці і продовжувати розвиватися [5].

Платформа також пропонує своїм клієнтам різноманітні плагіни та теми, які відповідають тому, що вони хочуть робити, і тому, як вони хочуть, щоб виглядав їхній веб-сайт. Це також може покращити клієнтський досвід та покращити стосунки з клієнтами [5].

Magento включає велику кількість сучасних та необхідних технологій, які покращують оптимізацію та кастомізацію сайту. Фактично, розробники сформували абсолютно новий стек технологій для Magento. Серед таких можливі виділити:

- Composer - дозволяє повторно використовувати сторонні бібліотеки без роботи з вихідним кодом, зменшує конфлікти розширень;
- NGINX - слугує зворотнім проксі, HTTP-кешем та балансувальником навантаження;
- Redis - використовує кеш бази даних і брокер для повідомлень;
- Symfony - полегшує управління контентом, покращує функціональність і зовнішній вигляд сайту;
- Varnish - збільшує швидкість роботи сайту [5].

Нарешті, слід розглянути плюси і мінуси платформи Magento, щоб зрозуміти та підбити підсумки, чи варто взагалі розглядати Magento як платформу для електронної комерції. Слід зазначити, що будуть розглядатись переваги та недоліки Magento 2, враховуючи, що це більш активна і рекомендована версія Magento на даний момент.

Серед плюсів Magento слід виділити такі:

- платформа гнучка для користувачів;
- платформа пропонує безкоштовні сервіси;

- спільнота підтримує і допомагає користувачам, що заплуталися, зорієнтуватися в проблемних питаннях, які у них можуть виникнути [5].

Варто наголосити на таких значних мінусах Magento як:

- додаткові великі витрати на хостинг та безпеку;
- платна підтримка клієнтів [5].

Magento спрощує для компаній будь-якого розміру створення, налаштування та управління каналами онлайн-продажів завдяки своїм потужним функціям та інтуїтивно зрозумілому інтерфейсу. Незалежно від того, чи ви тільки починаєте, чи вже маєте усталений бізнес, система може допомогти вам вивести ваші онлайн-продажі на новий рівень [5].

1.3.2 Застосунок “Shopify”

Shopify - це повноцінна комерційна платформа, яка дозволяє створювати, розвивати та керувати бізнесом.

У технічному плані Shopify - це платформа для продажу програмного забезпечення як послуги (SaaS), що базується на підписці. Усі стандартні плани підтримують фірмовий ІМ і повний набір інструментів для особистого продажу та онлайн-продажів [6].

Shopify дозволяє створювати та налаштовувати ІМ, продавати товари в різних місцях, включаючи веб, мобільні пристрої, соціальні мережі, онлайн-маркети, традиційні магазини та флеш-роздрібну торгівлю, керувати асортиментом, запасами, оплатою та доставкою [6].

Shopify повністю працює у хмарному середовищі та розміщується на хостингу, а це означає, що не потрібно турбуватися про оновлення або обслуговування програмного забезпечення чи веб-серверів. Це дає користувачам гнучкість у доступі та управлінні бізнесом практично з будь-якого місця, включаючи мобільні пристрої [6].

Не дивлячись на те, що Shopify пропонує безліч варіантів продажів і функцій управління бізнесом, щоб зрозуміти чи підходить цей сервіс для використання треба виокремити важливі переваги та недоліки.

Для початку, варто звернути увагу на плюси використання Shopify, а саме:

- загальна простота використання, що дозволяє швидко почати працювати та вести свій ІМ;
- низька щомісячна плата для тарифних планів початкового рівня та конкурентоспроможні пропозиції для вищого рівня;
- повністю керована платформа Shopify обробляє всі технічні деталі автоматично, не потребуючи втручання людини;
- повністю інтегровані функції мобільних і роздрібних кас дозволяють здійснювати особисті продажі будь-де та будь-коли;
- можливість наповнювання свого ІМ практично без початкових інвестицій, використовуючи вбудовані мережі дропшипінгу та оптової торгівлі Shopify;
- Shopify має вбудований автоматизований email-маркетинг, що допомагає збільшенню продажам одним кліком [6].

Серед мінусів використання Shopify треба виділити:

- Shopify пропонує чотири стандартні плани підписки для магазинів, вартість яких починається від 1 994 рупій на місяць, що є досить великовартісним, особливо для починаючих ІМ;
- якщо ви не використовуєте Shopify Payments, користувач має сплачувати комісію за кожну транзакцію, здійснену за допомогою кредитної або дебетової картки;
- теми Shopify, як правило, досить прості, а додавання розширених функцій і можливостей вимагає спеціалізованого кодування, тобто є досить обмеженими;

– IM Shopify мають базові функції SEO, а також функції ведення блогу в Shopify досить обмежені порівняно з WordPress, Squarespace та Wix [6].

1.3.3 Застосунок “PrestaShop”

PrestaShop - це вільнодоступна платформа електронної комерції з відкритим кодом. Це одна з найкращих платформ електронної комерції з відкритим вихідним кодом, яка пропонує безліч функцій, що роблять її ще сильнішою. Вона працює на PHP і може допомогти підприємцям створювати комплексні веб-сайти електронної комерції з дивовижною функціональністю [7].

PrestaShop - одна з найпотужніших платформ для створення веб-сайтів електронної комерції, яка має інтуїтивно зрозумілий інтерфейс для ефективного управління ІМ. Він пропонує сотні модулів для підвищення ефективності цього програмного забезпечення [7].

Функції PrestaShop, такі як розширені схеми резервного копіювання, висока надійність і легка доступність, дозволяють користувачам масштабувати свій бізнес і підвищувати довіру до сайту. Деякі численні функції PrestaShop можна віднести до переваг цієї платформи електронної комерції.

PrestaShop має понад триста вбудованих функцій, які дозволяють керувати способами оплати, доставкою, переліком товарів, постачальниками та покупцями. Використовує систему веб-шаблонів, яка дозволяє вам налаштовувати тему магазину та використовувати додаткові модулі.

PrestaShop - це абсолютно безкоштовна платформа з відкритим вихідним кодом, яка дозволяє створити повністю налаштований сайт за лічені хвилини.

Можна використовувати розширені фрагменти, щоб ваш сайт відрізнявся від ваших конкурентів. Можна знайти доповнення на маркетплейсі PrestaShop, яке спрощує саме цей процес.

PrestaShop пропонує можливість додавати кастомні функції на ваш сайт, щоб зробити його більш широким. Серед них - режим знижок, безкоштовна доставка, спеціальні пропозиції, email-маркетинг, партнерське програмування, подарункова упаковка, купонні коди та багато іншого.

PrestaShop підтримує безліч платіжних систем, таких як PayPal, Amazon Pay, Worldpay, Discover Global Network, First Data та Authorize.Net. Крім того, він має понад 250 платіжних провайдерів в якості додатків [7].

Вище було названо багато різних плюсів та позитивних сторін PrestaShop, але це не означає, що вона немає недоліків:

- щоб керувати PrestaShop якнайкраще, адміністратор магазину повинен володіти певними знаннями розробника та навичками, яких він вимагає для роботи на ній;
- PrestaShop є безкоштовним, але за реєстрацію домену та хостинг доведеться заплатити певну суму, втім, рішення про хостинг повністю залежить від бажань користувача, отже, людина може зменшити ці витрати, також виділяється додаткова вартість послуг на налаштування;
- оскільки доповнення для PrestaShop створюються різними командами розробників, їхня функціональність може заважати один одному, тут найкращим рішенням є лише метод проб і помилок, і, звичайно, потрібно мати запасний варіант для таких випадків [8].

Порівняння існуючих аналогів було зведено і зображено на рисунку 1.1.

Порівняння існуючих аналогів				
Назва застосунку Критерії порівняння	Magento	Shopify	PrestaShop	Розроблений вебзастосун.
Зручність навігації	+	+	-	+
Простий інтерфейс	+	-	-	+
Вартість	Безк.	Підписка	Підписка	Безк.
Вимоги до ПК	+	+	+	Незначні
Додаткове навчання для корист.	+	-	+	-
Обов'язова авторизція	+	+	+	+

Рисунок 1.1 – Порівняння існуючих аналогів з розробленим ВЗ

Висновок: Кожна платформа для розробки електронної комерції має як свої плюси, так і мінуси. Користувачі повинні вибрати ту, яка відповідає потребам їх бізнесу та цілям, яких вони хочемо досягти. Звертаючи увагу на всі вище вказані недоліки та переваги, можна вже чітко визначити подальші вимоги до розробки проекту програмного забезпечення у вигляді додатку для ведення обліку товарів в ІМІ.

1.4 Постановка завдання

Метою даної дипломної роботи є розробка такого ВЗ, який надає ефективний інструментарій для менеджменту товарів у ІМІ. Цей ВЗ має забезпечити можливість швидкого та зручного додавання, редагування та видалення товарів, оновлення їх параметрів, а також надати інші функціональні можливості, що сприятимуть підвищенню ефективності менеджменту товарів у ІМІ.

ВЗ забезпечить ефективне управління товарами у ІМІ, що дозволить поліпшити конкурентоспроможність магазину та задовольнити потреби його клієнтів, власників ІМІ.

1.4.1 Функціональність системи

Функціональні вимоги:

- можливість пошуку товару за деякими критеріями;
- можливість перегляду всього доступного асортименту;
- можливість перегляду детальної інформації про товар;
- можливість додавання нових та редагування існуючих товарів;
- можливість видалення обраного товару;
- можливість перегляду всіх товарів, які мають спільний жанр або платформу;
- можливість перегляду детальної інформації про обраний товар і зміна певної інформації про нього.

1.4.2 Вимоги до інтерфейсу

Вимоги до інтерфейсу ВЗ для менеджменту товарів у ІМІ:

- інтерфейс повинен бути в виразних кольорах;
- інтерфейс повинен бути простим у використанні і мати зрозумілу навігацію;
- пошук потрібних функцій не повинен займати багато часу у користувача;
- кожен товар повинен мати окрему комірку з основною інформацією про нього.

1.4.3 Вимоги до продуктивності

Для ІМІ важливо подбати про надійність системи, вони повинні витримувати високі навантаження через велику кількість товару , що постійно оновлюється. Крім того , система має бути простою та зрозумілою в використанні для будь-якого користувача.

Застосунок повинен запускатися і завантажуватися швидко для забезпечення ефективного використання користувачами. Це означає оптимізацію коду, мінімізацію запитів до сервера і швидкий рендерінг веб-сторінок.

Система повинна надавати безпроблемний доступ до бажаного товару за лічений час.

1.4.4 Вимоги до безпеки

Безпека ВЗ займає не останнє місце серед важливих аспектів успішного ІМІ.

Авторизовані користувачі можуть переглядати сайт, перевірити наявність будь-якого товару для клієнта, вносити зміни в будь-який товар , додавати та видаляти його , якщо товар відсутній чи тимчасово недоступний.

В майбутньому планується додати двофакторну аутентифікацію та розробити механізм захисту від різних типів кібератак.

1.4.5 Експлуатаційні вимоги

Для використання ВЗ не існує конкретних вимог до ПК користувача , потрібно лише стабільне підключення до мережі Internet та встановлений веббраузер. Для того щоб ВЗ функціонував без перебоїв – потрібен хоча б один ПК-сервер та людина яка буде його контролювати. База даних ІМ буде знаходитися на сервері.

1.4.6 Не функціональні вимоги до продукту

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- встановлена та налагоджена оперативна система – Linux або Windows;
- підтримка Python і Django;
- персональний комп'ютер;
- підключення до мережі Internet;
- встановлений браузер;
- процесор із тактовою частотою від 1,2 GHz;
- обсяг оперативної пам'яті – від 2GB.

1.5 Висновки за розділом 1

Розкрита актуальність теми. З'ясовано суть поняття інтернет-магазинів. Визначено сучасні вимоги до менеджменту товарів у Інтернет-магазині ігор та управління товарним асортиментом. Проаналізовано переваги та недоліки існуючих застосунків для ведення обліку товарів загалом та виокремлено деякі застосунки які є конкурентоспроможними в цій сфері. Визначено можливості для подальших досліджень та розширень проекту. Виконано аналіз та документування вимог до ПЗ. Визначено функціональність системи для розроблення та затвердження технічного завдання.

2 ВИБІР ІНСТРУМЕНТАРІЮ РОЗРОБКИ

У першому розділі відбулося дослідження предметної області . Було розглянуто сучасні вимоги щодо проблемної області. Після цього було проаналізовано існуючі застосунки та виявлено їх недоліки та переваги.

У наступному розділі буде обрано та затверджено технічні засоби розробки такі як: мова програмування та framework, система баз даних MySQL, технологія реалізації вебзастосунку. Завдяки порівнянню та аналізу переваг та недоліків можна зробити остаточний вибір.

2.1 Вибір мови програмування

Python — це інтерпретована, інтерактивна, об'єктно-орієнтована та високорівнева мова програмування загального призначення з динамічною суворою типізацією та автоматичним управлінням пам'яттю, орієнтована на підвищення продуктивності розробника, читабельність коду, а також на забезпечення портування написаних на ній програм [9].

Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато менша за обсягом коду за свого аналога, написаного на іншій мові [9].

Ця мова програмування є багатоплатформовою. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін [9].

Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами [9].

Як відомо, Python дуже часто використовується для:

- для аналізу даних;
- для візуалізації даних;
- для машинного навчання;
- для розробки програмного забезпечення;

- для веброзробки;
- для автоматизації задач/скриптингу [9].

Переваги Python зробили цю мову дуже зручною для використання не тільки професіоналами вебпрограмування, а й для новачків.

З основних переваг Python варто відмітити наступні:

- відкритий вихідний код та безкоштовне використання;
- у Python мало ключових слів, проста структура та чітко визначений синтаксис, тобто його легко вчити та читати ;
- велика стандартна бібліотека - велика частина бібліотеки Python легко портується і є кросплатформною, сумісна з Unix, Windows та macOS;
- гнучка мова загального призначення, яка повністю підтримує як процедурне, так і об'єктно-орієнтоване програмування;
- програми Python зазвичай набагато коротші, ніж аналогічні програми написані на інших мовах програмування, це чудовий вибір для швидкого створення прототипів в сфері швидкої розробки програмного забезпечення;
- Python автоматично виділяє пам'ять, а збирач сміття "перероблює" пам'ять із відкинутих об'єктів. Це означає, що розробникам не потрібно турбуватися про витоки пам'яті, некоректні посилання чи вказівники та розмір кожного об'єкта [6].

Звичайно, Python не ідеальна мова програмування, бо має свої недоліки.

Невисока швидкість роботи. Python є інтерпретованою мовою. Тобто її код виконується без попередньої компіляції машинної мови. Тому швидкість такого виконання суттєво поступається іншим мовам, що компілюються, таким як C++, Objective-C і т.д. Ще один фактор, що «уповільнює», полягає в тому, що Python — високорівнева мова. Тобто вона більше схожа на людську, ніж на машинну, тому її обробка виконується довше.

Неефективне витрачання пам'яті. Python використовує динамічну типізацію, тобто перевірка типів змінних в ній виконується під час виконання, а не в момент оголошення. Це спрощує роботу розробника і робить мову програмування більш гнучкою, але збільшує споживання пам'яті та уповільнює роботу програми.

Взаємодія з базами даних. Працювати з базами даних у Python складніше. Python не має такого потужного і простого у використанні інтерфейсу, як Java Database Connectivity (JDBC). Його все ще можна використовувати, якщо операції читання та запису до баз даних відносно прості. Але це не найкращий вибір для програм, які мають складні взаємодії з великою корпоративною базою даних [9].

Порівняння мов програмування зображена на рисунку 2.1.

Порівняння мов програмування			
Критерії порівняння \ Назва	Python	PHP	C ++
Продуктивність та швидкодія	±	±	+
Синтаксис і зручність використання	+	+	–
Розширюваність та модульність	+	+	+
Обробка помилок та безпека	+	±	±

Рисунок 2.1 – Порівняння мов програмування

2.2 Вибір та обґрунтування Framework

Django – це високорівневий веб-фреймворк Python, який заохочує швидку розробку та чистий, прагматичний дизайн. Створений досвідченими розробниками, він бере на себе більшу частину клопоту з веб-розробкою,

тому ви можете зосередитися на написанні вашого додатку без необхідності винаходити велосипед. Він безкоштовний і з відкритим кодом [10].

Django дотримується філософії "Batteries included" і надає майже все, що може знадобитися розробникам "з коробки". Оскільки все, що вам потрібно, є частиною одного "продукту", все це бездоганно працює разом, відповідає єдиним принципам проектування та має вичерпну та актуальну документацію [11].

Django може використовуватися (і використовувався) для створення практично будь-якого типу веб-сайтів - від систем управління контентом і вікі до соціальних мереж і новинних сайтів. Він може працювати з будь-яким фреймворком на стороні клієнта і може передавати вміст майже у будь-якому форматі (включаючи HTML, RSS-стрічки, JSON і XML) [11].

Внутрішньо, хоча він надає вибір для майже будь-якої функціональності, яка вам може знадобитися (наприклад, кілька популярних баз даних, шаблонізаторів тощо), він також може бути розширений для використання інших компонентів, якщо це необхідно [11].

Django допомагає розробникам уникнути багатьох поширених помилок безпеки, надаючи фреймворк, який був спроектований так, щоб "робити правильні речі" для автоматичного захисту веб-сайту. Наприклад, Django надає безпечний спосіб керувати обліковими записами та паролями користувачів, уникаючи поширених помилок, таких як розміщення інформації про сеанс у файлах cookie, де вона є вразливою (натомість файли cookie містять лише ключ, а реальні дані зберігаються в базі даних), або безпосереднє зберігання паролів, а не їх хеш-суміші [11].

Хеш пароля - це значення фіксованої довжини, створене шляхом надсилання пароля через криптографічну хеш-функцію. Django може перевірити правильність введеного пароля, прогнавши його через хеш-функцію і порівнявши результат зі збереженим хеш-значенням. Однак через "односторонній" характер функції, навіть якщо збережене хеш-значення скомпрометоване, зловмиснику буде складно відновити оригінальний пароль.

Django за замовчуванням забезпечує захист від багатьох вразливостей, включаючи SQL-ін'єкції, міжсайтовий скриптинг, підробку міжсайтових запитів та клікджекінг (докладніше про такі атаки див. розділ Безпека веб-сайтів).

Django використовує компонентну архітектуру "shared-nothing" (кожна частина архітектури не залежить від інших, а отже може бути замінена або змінена за потреби). Чітке розділення між різними частинами означає, що він може масштабуватися для збільшення трафіку шляхом додавання обладнання на будь-якому рівні: серверів кешування, серверів баз даних або серверів додатків. Деякі з найбільш завантажених сайтів успішно масштабували Django для задоволення своїх потреб (наприклад, Instagram та Disqus) [11].

Зручність у підтримці. Код Django написаний з використанням принципів та патернів проектування, які сприяють створенню коду, який можна підтримувати та використовувати повторно. Зокрема, він використовує принцип "Не повторюйся" (DRY), щоб уникнути непотрібного дублювання, зменшуючи кількість коду. Django також сприяє групуванню пов'язаної функціональності у "додатки", які можна використовувати повторно, а на нижчому рівні групує пов'язаний код у модулі (за зразком патерну Model View Controller (MVC)).

Django написано мовою Python, яка працює на багатьох платформах. Це означає, що ви не прив'язані до якоїсь конкретної серверної платформи і можете запускати свої додатки на багатьох різновидах Linux, Windows і macOS. Крім того, Django добре підтримується багатьма хостинг-провайдерами, які часто надають спеціальну інфраструктуру та документацію для розміщення сайтів на Django [11].

Висновок: Django є чудовим доповненням до проєктів, які потребують обробки великих об'ємів контенту, взаємодії з користувачами або великого трафіку, або мають справу зі складними функціями чи технологіями (наприклад, машинним навчанням). Водночас вона досить проста для

невеликих проєктів, або якщо ви маєте намір масштабувати свій проєкт на значно вищий рівень. Ось чому Django використовується багатьма компаніями, різними за розміром та цілями [12].

Порівняння вищенаведеного фреймворка з аналогом зображено на рисунку 2.2.

Порівняння Framework		
Критерії порівняння \ Назва	Django	Ruby on Rails
Мова програмування	Python	Ruby
Архітектура та концепції	MVC або MTV	MVC
Швидкодія та масштабованість	+	±
Продуктивність розробки	Забезпечує широкий спектр інструментів та бібліотек для швидкої розробки, включаючи вбудовану адмін-панель, ORM (Object-Relational Mapping) та систему маршрутизації.	Зосереджений на збільшенні продуктивності розробки шляхом використання конвенцій над конфігурацією та готових рішень для багатьох задач, таких як міграції бази даних та автентифікація.

Рисунок 2.2 – Порівняння Framework

2.3 Вибір системи керування базами даних

База даних - це організований набір структурованої інформації або даних, які зазвичай зберігаються в електронному вигляді в комп'ютерній системі. Зазвичай базою даних керує система керування базами даних (СКБД). Разом дані та СКБД, а також додатки, які з ними пов'язані, називаються системою баз даних, яку часто скорочують до просто бази даних [13].

Дані в найпоширеніших типах баз даних, що використовуються сьогодні, зазвичай моделюються у вигляді рядків і стовпців у серії таблиць, щоб зробити обробку та запити до даних ефективними. Тоді до даних можна легко отримати доступ, керувати ними, змінювати, оновлювати,

контролювати та організовувати. Більшість баз даних використовують мову структурованих запитів (SQL) для запису і запитів до даних [13].

MySQL - це реляційна система управління базами даних з відкритим вихідним кодом, заснована на мові SQL. Вона була розроблена та оптимізована для веб-додатків і може працювати на будь-якій платформі. З появою в Інтернеті нових і різноманітних вимог MySQL стала платформою вибору для веб-розробників і веб-додатків. Оскільки вона призначена для обробки мільйонів запитів і тисяч транзакцій, MySQL є популярним вибором для підприємств електронної комерції, яким потрібно управляти численними грошовими переказами. Гнучкість на вимогу - головна особливість MySQL [13].

MySQL - це СКБД, на якій працюють деякі з найпопулярніших веб-сайтів і веб-додатків у світі, включаючи Airbnb, Uber, LinkedIn, Facebook, Twitter і YouTube [13].

Оскільки MySQL широко використовується в багатьох галузях, бізнес-користувачі - від веб-майстрів-початківців до досвідчених менеджерів - повинні прагнути зрозуміти її основні характеристики. Рішення про використання цієї технології та ефективне інформування про неї починається з огляду базової доступності, структури, філософії та зручності використання MySQL [14].

Хоча MySQL часто асоціюється з інтернет-додатками або веб-сервісами, вона була розроблена для широкої сумісності з іншими технологіями та архітектурами. СКБД працює на всіх основних обчислювальних платформах, включаючи операційні системи на базі Unix, такі як численні дистрибутиви Linux або Mac OS, а також Windows [14].

Клієнт-серверна архітектура MySQL означає, що вона може підтримувати різні бекенди, а також різні програмні інтерфейси. Дані можна безпосередньо мігрувати з MySQL до її форків (наприклад, MariaDB), а також до більшості інших СУБД завдяки архітектурній та мовній схожості [14].

Існуючі інструменти міграції від Oracle та сторонніх розробників також дозволяють MySQL переміщувати дані до та з широкого набору загальних систем зберігання даних, незалежно від того, чи вони призначені для локального або хмарного зберігання. MySQL можна розгортати у віртуалізованих середовищах, розподілених або централізованих, і навіть у вигляді портативних автономних бібліотек для навчальних цілей, тестування або невеликих додатків [14].

Широка сумісність MySQL з усіма цими системами та програмним забезпеченням робить її особливо практичним вибором СКБД у більшості ситуацій [14].

Бази даних MySQL є реляційними. Основний фактор, що відрізняє реляційні бази даних від інших цифрових сховищ, полягає в тому, як дані організовані на високому рівні. Такі бази даних, як MySQL, містять записи в декількох окремих таблицях з високим рівнем кодифікації, на відміну від єдиного всеохоплюючого сховища або колекцій напів- або неструктурованих документів [14].

Це дозволяє СКБД краще оптимізувати такі дії, як пошук даних, оновлення інформації або більш складні дії, такі як агрегування. Логічна модель визначається для всього вмісту бази даних, описуючи, наприклад, допустимі значення в окремих стовпчиках, характеристики таблиць і представлень або те, як пов'язані між собою індекси з двох таблиць [14].

Реляційні моделі залишаються популярними з кількох причин. Вони надають користувачам інтуїтивно зрозумілі, декларативні мови програмування - по суті, вказуючи базі даних, який результат потрібен, мовою, схожою або принаймні зрозумілою на англійську, замість того, щоб ретельно кодувати кожен крок процедури, що веде до цього результату. Це перекладає більшу частину роботи на СУБД та механізми SQL, що краще забезпечує дотримання логічних правил та економить цінні ресурси та робочу силу [14].

MySQL з відкритим вихідним кодом. Будь-яка фізична чи юридична особа може вільно використовувати, модифікувати, публікувати та розширювати відкриту кодову базу MySQL від Oracle. Програмне забезпечення випускається на умовах ліцензії GNU General Public License (GPL) [14].

Якщо код MySQL потрібно інтегрувати або включити в комерційний додаток (або якщо програмне забезпечення з відкритим кодом не є пріоритетом), підприємства можуть придбати комерційну ліцензовану версію від Oracle [14].

Знову ж таки, ці варіанти надають організаціям додаткову гнучкість, якщо вони вирішили працювати з MySQL. Публічний характер випусків з відкритим вихідним кодом збагачує документацію MySQL та культуру онлайн-підтримки, а також гарантує, що підтримувані або нові можливості ніколи не відходять надто далеко від поточних потреб користувачів [14].

MySQL проста у використанні. Хоча реляційна природа MySQL і пов'язані з нею жорсткі структури зберігання даних можуть здатися обмежувальними, таблична парадигма є, мабуть, найбільш інтуїтивно зрозумілою і, в кінцевому рахунку, забезпечує більшу зручність використання [14].

Насправді, MySQL робить багато поступок для підтримки якомога ширшого розмаїття структур даних, від стандартних, але багатих логічних, числових, алфавітно-цифрових, датових і часових типів, до більш просунутих JSON або геопросторових даних. Окрім типів даних і широкого набору вбудованих функцій, екосистема MySQL також включає різноманітні інструменти, що полегшують роботу з усіма процесами - від керування сервером до створення звітів і аналізу даних [14].

Незалежно від загальної архітектури СУБД, користувачі завжди можуть знайти функцію MySQL, яка дозволить їм моделювати та кодифікувати дані так, як вони бажають. MySQL залишається однією з найпростіших у вивченні та використанні технологій баз даних [14].

2.4 Проектування бази даних

Моделювання даних (data modeling) - це процес створення моделі даних для даних, які будуть зберігатися в базі даних. Ця модель даних є концептуальним представленням об'єктів даних, асоціацій між різними об'єктами даних і правил.

Моделювання даних допомагає у візуальному представленні даних і забезпечує дотримання бізнес-правил, нормативних вимог та державної політики щодо даних. Моделі даних забезпечують узгодженість іменування, значень за замовчуванням, семантики, безпеки, забезпечуючи при цьому якість даних.

Основною метою використання моделі даних є забезпечення точного представлення всіх об'єктів даних, необхідних для бази даних. Відсутність даних призведе до створення помилкових звітів та отримання невірних результатів.

Модель даних допомагає проектувати базу даних на концептуальному, фізичному та логічному рівнях.

Структура моделі даних допомагає визначити реляційні таблиці, первинні та зовнішні ключі і збережені процедури.

Вона дає чітке уявлення про дані бази даних і може бути використана розробниками бази даних для створення фізичної бази даних.

Вона також корисна для виявлення відсутніх і надлишкових даних.

Хоча початкове створення моделі даних є трудомістким і займає багато часу, в довгостроковій перспективі це робить оновлення та обслуговування вашої ІТ-інфраструктури дешевшим і швидшим [15].

ER-модель (Entity-relationship model або Entity-relationship diagram) – це семантична модель даних, яка призначена для спрощення процесу проектування бази даних. З ER-моделі можуть бути породжені всі види баз даних: реляційні, ієрархічні, мережні, об'єктні. В основі ER-моделі лежать поняття “сутність”, “зв’язок” та “атрибут”[16].

Для великих баз даних побудова ER-моделі дозволяє уникнути помилок проектування, які надзвичайно важко виправляти, особливо, якщо база даних вже експлуатується чи на стадії тестування. Помилки в розробці структури бази даних може призвести до перебудови коду програмного забезпечення, що керує цією базою даних. У результаті час, кошти та людські ресурси будуть використані неефективно [16].

ER-модель – це представлення бази даних у вигляді наочних графічних діаграм (рис.2.3). ER-модель візуалізує процес, що визначає деяку предметну область. Діаграма “сутність-зв’язок” – це діаграма, яка представляє в графічному вигляді сутності, атрибути і зв’язки [16].

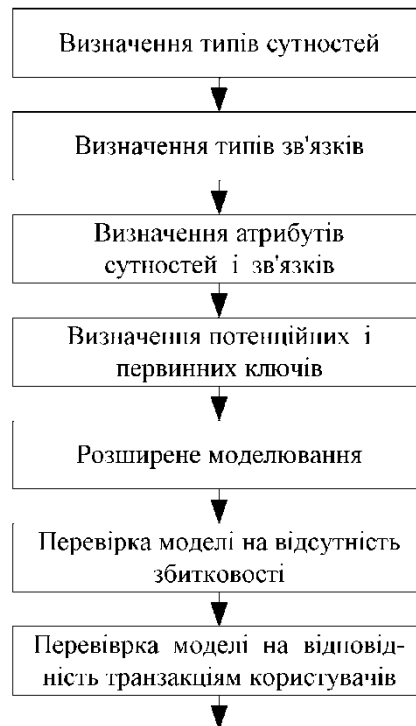


Рисунок 2.3– Послідовність проведення ER-моделювання.

Сутність в БД – це будь-який об’єкт в БД, який можна виділити виходячи з суті предметної області для якої розробляється ця БД. Розробник БД повинен вміти правильно визначати сутності [16].

Наприклад, предметна область ВЗ для менеджменту товарів в ІМІ включає в себе такі сутності:

- «Games»;
- «Controllers»;
- «Category»;
- «Users».

Сутність «Games» служить для збереження даних про ігри, які є в наявності. Має в собі такі атрибути:

- id – ключовий атрибут;
- id_category – атрибут зв'язку між сутностями «Games» та «Category»;
- platform – атрибут збереження назви платформи;
- genre – атрибут збереження інформації про жанр гри;
- price – атрибут зберігає вартість гри;
- balance – атрибут зберігає дані про кількість ігор, які є на складі;
- name – атрибут збереження назви для гри.

Сутність «Controllers» служить для зберігання про контролери, які є в наявності. Має в собі такі атрибути:

- id – ключовий атрибут;
- name – зберігає назву контролера;
- id_category – атрибут зв'язку між сутностями «Category» та «Controllers»;
- price – атрибут зберігає вартість контролера;
- platform – атрибут збереження назви платформи;
- balance – атрибут зберігає дані про кількість контролерів, які є на складі.

Сутність «Category» служить для зберігання інформації про категорію до якої відноситься товар. Сутність має в собі такі атрибути:

- id – ключовий атрибут;
- cat_name – атрибут зберігає назву категорії.

Сутність «Users» служить для зберігання інформації про користувачів. Сутність має в собі такі атрибути:

- id – ключовий атрибут;
- username – атрибут зберігає ім'я для входу в систему користувача;
- password – атрибут зберігає пароль користувача.

Під час концептуального проєктування була розроблена модель бази даних. Розроблена БД зображена на ER-діаграмі (рис. 2.4).

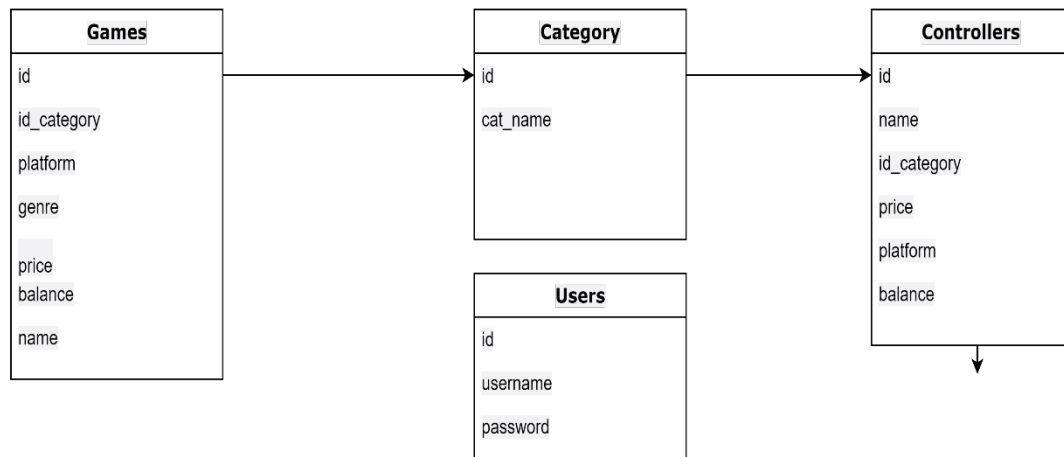


Рисунок 2.4 – ER-діаграма

2.5 Висновки за розділом 2

В другому розділі було обрано та затверджено технічні засоби розробки так як: технологія реалізації застосунку, мова програмування та система керування базами даних. Завдяки аналізу переваг та недоліків можемо зробити впевнений вибір: мова Python, Framework Django.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ СИСТЕМИ

3.1 Реалізація структурної схеми та основних форм застосунку

Функціональна вимога - це опис того, як повинна поводитися система. Вона визначає, що система повинна робити, щоб задовольнити потреби або очікування користувача. Функціональні вимоги можна уявити як функції, які виявляє користувач. Вони відрізняються від нефункціональних вимог, які визначають, як система повинна працювати внутрішньо (наприклад, продуктивність, безпека тощо) [17].

Функціональні вимоги складаються з двох частин: функції та поведінки. Функція - це те, що робить система (наприклад, "розрахувати ціну товару для покупця"). Поведінка - це те, як система це робить (наприклад, "система повинна обчислювати товару для покупця, помноживши ціну товару на відсоток націнки") [17].

Розроблена загальна структурна схема застосунку зображена на рисунку 3.1.

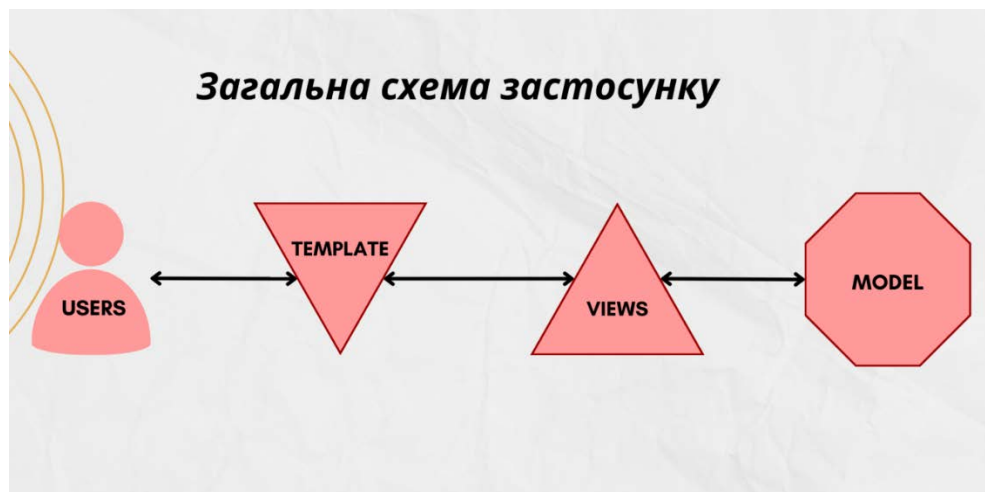


Рисунок 3.1 – Загальна структурна схема застосунку

В даній системі було розроблено форми, кожна з яких виконує свою власну функцію.

Перша форма відповідає за авторизацію користувача у систему (рис.3.2).

Увійдіть у систему

Логін

Пароль

Увійти

Рисунок 3.2 – Форма авторизації

Друга форма – це головна форма, яка відображає всі товари, які є в наявності (рис.3.3).

Головна Ігри Контролери Вихід

Перелік доступних товарів

Пошук...

Назва товару	Платформа	Ціна	Жанр
Devil May Cry	Playstation 4	1240	Shooter
Far Cry 5	Xbox One	1719	First Person Shooter
Far Cry 5	Playstation 5	2199	First Person Shooter
Far Cry 3	Playstation 4	1399	First Person Shooter
Devil May Cry	Playstation 3	1700	Shooter
Devil May Cry	Xbox One	1500	Shooter
God Of War	Playstation 5	2999	Action-Adventure
God Of War	Playstation 4	1799	Action-Adventure
Far Cry 4	Xbox One	1100	First Person Shooter
The Last Of Us I	Playstation 4	1650	Survival
The Last Of Us I Remastered	Playstation 5	2300	Survival
The Last Of Us II	Playstation 5	2999	Survival
Horizon Zero Dawn	Playstation 4	1169	Action RPG
Horizon Zero Dawn	Playstation 5	1970	Action RPG
Horizon Forbidden West	Playstation 5	2199	Action RPG
FIFA 22	Playstation 4	1670	Sport Simulator
FIFA 22	Playstation 5	2100	Sport Simulator
FIFA 22	Xbox One	1750	Sport Simulator
FIFA 22	Playstation 3	2799	Sport Simulator
FIFA 23	Playstation 4	1900	Sport Simulator
FIFA 23	Xbox One	1790	Sport Simulator
Skay	Playstation 5	1949	Action-Adventure
Skay	Playstation 4	1399	Action-Adventure
Skay	Xbox One	1190	Action-Adventure
The Last Of Us I Remastered	Playstation 4	1900	Survival
Call Of Duty Modern Warfare II	Playstation 5	2799	First Person Shooter
Call Of Duty Modern Warfare II	Playstation 4	1999	First Person Shooter
Call Of Duty Modern Warfare II	Xbox One	1799	First Person Shooter
Grand Theft Auto V	Playstation 5	2399	Action-Adventure
Grand Theft Auto V	Playstation 4	1599	Action-Adventure

Рисунок 3.3 – Головна форма

У меню також присутня кнопка «Ігри», натиснувши на цю кнопку ми переходимо на нову форму , яка виводить нам всі ігри які є в наявності (рис.3.4).



Назва товару	Платформа	Ціна	Жанр
Devil May Cry	Playstation 4	1240	Slasher
Far Cry 6	Xbox One	1789	First Person Shooter
Far Cry 6	Playstation 5	2199	First Person Shooter
Far Cry 5	Playstation 4	1399	First Person Shooter
Devil May Cry	Playstation 5	1700	Slasher
Devil May Cry	Xbox One	1500	Slasher
God Of War	Playstation 5	2999	Action-Adventure
God Of War	Playstation 4	1799	Action-Adventure
Far Cry 5	Xbox One	1100	First Person Shooter
The Last Of Us I	Playstation 4	1650	Survival
The Last Of Us I Remastered	Playstation 5	2300	Survival
The Last Of Us II	Playstation 5	2999	Survival
Horizon Zero Dawn	Playstation 4	1169	Action RPG
Horizon Zero Dawn	Playstation 5	1970	Action RPG
Horizon Forbidden West	Playstation 5	2199	Action RPG
FIFA 22	Playstation 4	1670	Sport Simulator
FIFA 22	Playstation 5	2100	Sport Simulator
FIFA 22	Xbox One	1750	Sport Simulator
FIFA 23	Playstation 5	2799	Sport Simulator
FIFA 23	Playstation 4	1900	Sport Simulator
FIFA 23	Xbox One	1790	Sport Simulator
Stray	Playstation 5	1949	Action-Adventure
Stray	Playstation 4	1399	Action-Adventure
Stray	Xbox One	1190	Action-Adventure
The Last Of Us I Remastered	Playstation 4	1900	Survival
Call Of Duty: Modern Warfare II	Playstation 5	2799	First Person Shooter
Call Of Duty: Modern Warfare II	Playstation 4	1999	First Person Shooter
Call Of Duty: Modern Warfare II	Xbox One	1799	First Person Shooter
Grand Theft Auto V	Playstation 5	2399	Action-Adventure
Grand Theft Auto V	Playstation 4	1599	Action-Adventure
Grand Theft Auto V	Xbox One	1400	Action-Adventure
Red Dead Redemption II	Playstation 5	2999	Action-Adventure
Red Dead Redemption II	Playstation 4	1999	Action-Adventure

Рисунок 3.4 – Форма «Ігри»

У меню також присутня кнопка «Контролери», натиснувши на цю кнопку ми переходимо на нову форму, яка виводить нам всі контролери які є в наявності (рис.3.5).



Назва товару	Платформа	Ціна
Dual Sense White	Playstation 5	3799
Dual Sense Black	Playstation 5	3559
DualShock 4 V2 White	Playstation 4	1240
DualShock 4 V2 Black	Playstation 4	1140
DualShock 4 V2 Camouflage	Playstation 4	1390
Xbox One W/C Black	Xbox One	975
Xbox One W/C White	Xbox One	1200
Xbox One W/C Camouflage	Xbox One	1390

Рисунок 3.5 – Форма «Контролери»

Наступна форма яка реалізована в проєкті – «Створити новий контролер», в неї ми можемо перейти з форми «Контролери» (рис.3.6).

Після заповнення всіх полів натискаємо кнопку «Зберегти».

Рисунок 3.6 – Форма «Створити новий контролер»

Повернувшись на форму «Ігри» можна натиснути кнопку «Створити нову гру» після чого відкриється наступна форма (рис.3.7).

Рисунок 3.7 – Форма «Створити нову гру»

Після заповнення всіх полів треба натиснути кнопку «Зберегти». Після чого користувач переходить на головну форму (3.8).

Назва товару	Платформа	Ціна	Жанр
Devil May Cry	Playstation 4	1240	Slasher
Far Cry 6	Xbox One	1789	First Person Shooter
Far Cry 6	Playstation 5	2199	First Person Shooter
Far Cry 5	Playstation 4	1599	First Person Shooter
Devil May Cry	Playstation 5	1700	Slasher
Devil May Cry	Xbox One	1500	Slasher
God Of War	Playstation 5	2999	Action-Adventure
God Of War	Playstation 4	1799	Action-Adventure
Far Cry 3	Xbox One	1100	First Person Shooter
The Last Of Us I	Playstation 4	1650	Survival
The Last Of Us I Remastered	Playstation 5	2300	Survival
The Last Of Us II	Playstation 5	2999	Survival
Horizon Zero Dawn	Playstation 4	1169	Action RPG
Horizon Zero Dawn	Playstation 5	1970	Action RPG
Horizon Forbidden West	Playstation 5	2199	Action RPG
FIFA 22	Playstation 4	1670	Sport Simulator
FIFA 22	Playstation 5	2100	Sport Simulator
FIFA 22	Xbox One	1750	Sport Simulator
FIFA 23	Playstation 5	2799	Sport Simulator
FIFA 22	Playstation 4	1900	Sport Simulator
FIFA 23	Xbox One	1790	Sport Simulator
Stray	Playstation 5	1949	Action-Adventure
Stray	Playstation 4	1399	Action-Adventure
Stray	Xbox One	1190	Action-Adventure
The Last Of Us I Remastered	Playstation 4	1900	Survival
Call Of Duty Modern Warfare II	Playstation 5	2799	First Person Shooter
Call Of Duty Modern Warfare II	Playstation 4	1999	First Person Shooter
Call Of Duty Modern Warfare II	Xbox One	1799	First Person Shooter
Grand Theft Auto V	Playstation 5	2399	Action-Adventure
Grand Theft Auto V	Playstation 4	1599	Action-Adventure

Рисунок 3.8 – Головна форма

З будь-якої форми («Головна» , «Ігри», «Контролери») можна натиснути на будь-який товар і відкриється нова форма яка покаже всі дані про товар (рис.3.9), там буде доступно дві кнопки «Редагувати товар» і «Видалити товар».



Рисунок 3.9 – Форма «Інформація про обраний товар»

Натиснувши кнопку «Редагувати товар» можливо змінити будь-яку інформацію про товар (рис.3.10), яка буде виводитися в формах «Головна» , «Ігри», «Контролери».

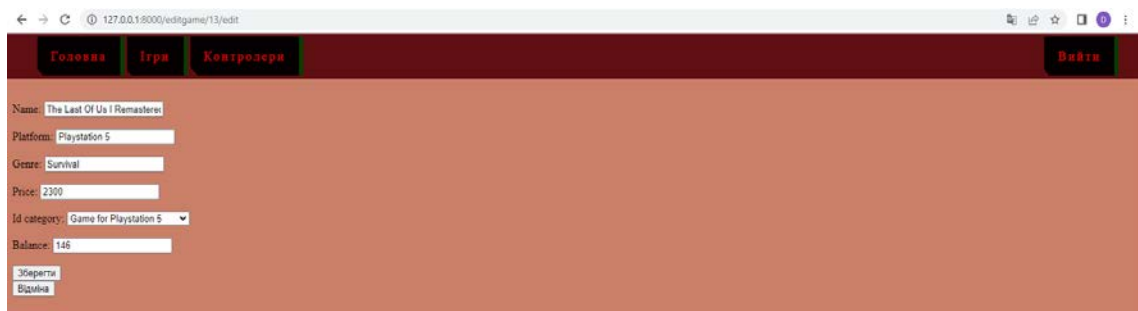


Рисунок 3.10 – Форма «Редагувати товар»

3.2 Проектування інтерфейсу

Дизайн інтерфейсу користувача (UI) - це процес, який дизайнери використовують для створення інтерфейсів у програмному забезпеченні або комп'ютеризованих пристроях, зосереджуючись на зовнішньому вигляді та стилі. Дизайнери прагнуть створювати інтерфейси, які користувачі вважають простими у використанні та приємними. UI дизайн стосується графічних

інтерфейсів та інших форм, наприклад, інтерфейсів з голосовим управлінням [18].

Користувацькі інтерфейси - це точки доступу, де користувачі взаємодіють з дизайном. Вони бувають трьох форматів:

- графічні інтерфейси користувача (GUI) - користувачі взаємодіють з візуальними зображеннями на цифрових панелях управління;
- інтерфейси з голосовим управлінням (VUI) - користувачі взаємодіють з ними за допомогою голосу, більшість розумних помічників - наприклад, Siri на iPhone і Alexa на пристроях Amazon - є VUI;
- інтерфейси на основі жестів - користувачі взаємодіють з 3D-простором дизайну за допомогою рухів тіла, наприклад, в іграх віртуальної реальності (VR) [18].

Щоб спроектувати користувацький інтерфейс найкращим чином, слід врахувати наступне:

- користувачі швидко оцінюють дизайн і дбають про зручність та привабливість;
- їм важливий не ваш дизайн, а те, щоб їхні завдання виконувалися легко і з мінімальними зусиллями.

Тому ваш дизайн повинен бути "невидимим": Користувачі повинні зосередитися не на ньому, а на виконанні завдань [18].

Отже, зрозумійте контекст і потоки завдань ваших користувачів (які ви можете знайти, наприклад, на картах подорожей клієнтів), щоб налаштувати найкращі, найбільш інтуїтивно зрозумілі інтерфейси, які забезпечать безперебійну роботу з ними [18].

Користувацький інтерфейс також повинен бути приємним (або, принаймні, задовольняти і не викликати розчарувань).

Коли ваш дизайн передбачає потреби користувачів, вони можуть насолоджуватися більш персоналізованим і захоплюючим досвідом. Порадуйте їх, і вони будуть повертатися.

Там, де це доречно, елементи гейміфікації можуть зробити ваш дизайн більш цікавим [18].

Користувацький інтерфейс повинен передавати цінності бренду і зміцнювати довіру користувачів.

Хороший дизайн - це емоційний дизайн. Користувачі асоціюють хороші почуття з брендами, які промовляють до них на всіх рівнях і зберігають магію приємного, безперешкодного досвіду [19].

Нижче наведено низку фундаментальних принципів дизайну інтерфейсу, які можна - і навіть потрібно - застосовувати майже до кожного проекту дизайну інтерфейсу, над яким ви працюєте:

- зробіть дизайн інтерфейсу простим;
- прогнозуйте і випереджайте;
- поставте користувача на місце водія;
- будьте методичними та послідовними;
- уникайте зайвої складності;
- надавати чіткі вказівники;
- будьте толерантні до помилок;
- надавати релевантний зворотній зв'язок;
- розставляти пріоритети у функціях;
- розробляйте інтерфейс користувача для доступності[19].

3.3 Висновки за розділом 3

За допомогою інструментів проектування завершується фізичний і логічний дизайн веб-додатку. На основі визначальних принципів розробки графічного інтерфейсу було прийнято рішення щодо подальшого впровадження програмного забезпечення.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Головним завданням роботи є створення вебзастосунку для менеджменту товарів у ІМІ.

Система менеджменту товарів у ІМІ – це комплексний інструмент або платформа, розроблена для ефективного управління асортиментом товарів у магазині, їх інформацією та доступністю для клієнтів. Система допомагає власнику або оператору ІМ забезпечити організоване та ефективне управління продукцією, що продається в Інтернеті.

Основні функції системи менеджменту товарів ІМІ:

- додавання та редагування товарів: система дозволяє власникам магазинів додавати нові продукти до каталогу, змінювати інформацію про продукт, таку як назва, опис, зображення, ціна, вага, виробник та інші характеристики;
- управління категоріями та підкатегоріями: система дозволяє організовувати продукти за готовими продуктами та підкатегоріями, щоб полегшити клієнту навігацію та покращити взаємодію з користувачем.

Автоматизована система управління (АСУ) - інформаційна система, призначена для автоматизованого здійснення управлінських процесів [20].

До АСУ пред'являється ряд загальних вимог:

- у першу чергу, повинна бути забезпечена сумісність елементів один з одним, а також з автоматизованими системами, взаємопов'язаними з даною АСУ;
- автоматизована система повинна бути пристосована до модернізації, розвитку і розширенню з урахуванням майбутніх перспектив;
- АСУ повинна мати достатній ступінь надійності для досягнення встановлених цілей функціонування системи при заздалегідь заданих умовах її застосування;

- автоматизована система управління повинна володіти достатньою адаптивністю до змін умов її використання, при цьому ступінь зміни умов застосування системи, як правило, спеціально обмовляється заздалегідь;
- в АСУ повинні бути передбачені контроль правильності виконання автоматизуються функцій і діагностування із зазначенням місця, виду та причини виникнення порушень правильності функціонування системи.

В автоматизованій системі управління повинні бути передбачені заходи захисту від неправильних дій персоналу, приводять до аварійного стану об'єкта або системи управління, від випадкових змін і руйнування інформації і програм, а також від несанкціонованого втручання і витоку інформації [20].

4.2 Вимоги до інтерфейсу

Вимоги до інтерфейсу ВЗ для менеджменту товарів у ІМІ:

- інтерфейс повинен бути в виразних кольорах;
- інтерфейс повинен бути простим у використанні і мати зрозумілу навігацію;
- пошук потрібних функцій не повинен займати багато часу у користувача;
- кожен товар повинен мати окрему комірку з основною інформацією про нього.

4.3 Розгортання застосунку

Структура вебзастосунка на основі використання обраного Framework Django зображена на рис.4.1.

Архітектура серверу, на якому розгортається застосунок. Операційна система: Windows. Сервер бази даних: MySQL. Версія Python: 3.11.2.

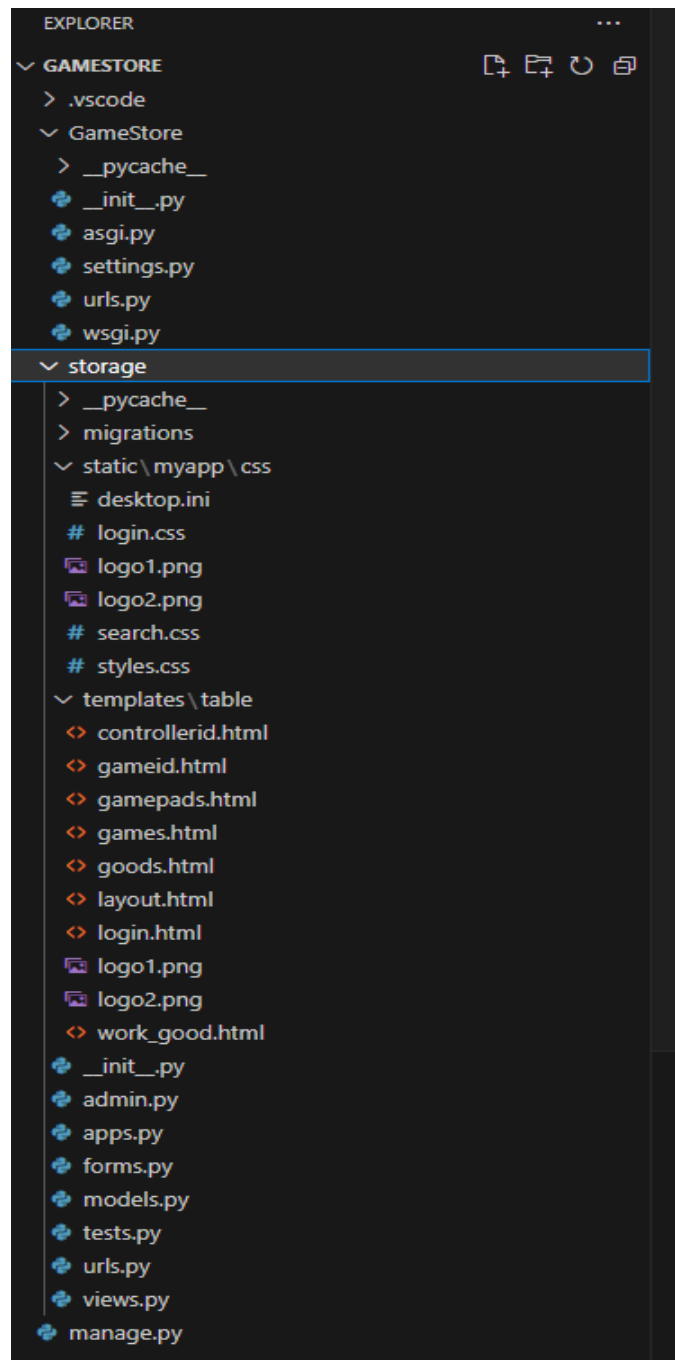


Рисунок 4.1 – Структура ВЗ

4.3.1 Звернення до програми

Організуйте доступ користувачів до програм, надсилаючи запити на відвідування сайтів за допомогою адресного рядка веб-браузера. Програма також доступна, коли адміністратори змінюють налаштування або виконують дії з користувачами.

4.3.2 Вхідні й вихідні дані

В якості вхідних даних використовується база даних наведена на рисунку 4.2

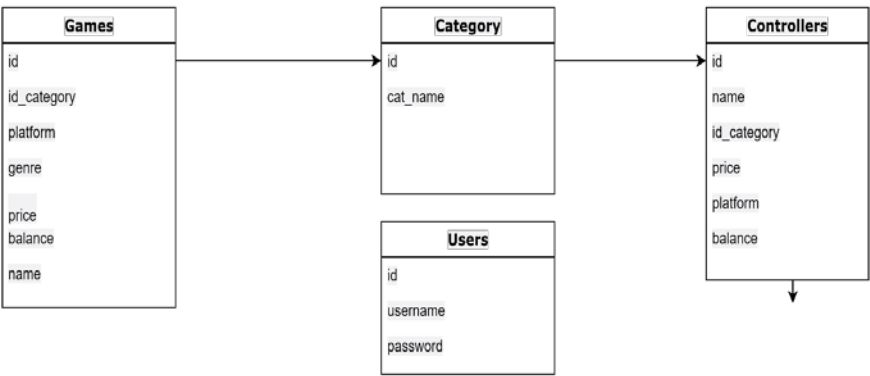


Рисунок 4.2 – База даних ВЗ для менеджменту товарів у ІМП

4.4 Методика та результати тестування

Під час випробувань було обрано стратегію покрокової реалізації алгоритму за допомогою програми.

На рис. 4.3 наведено зовнішній вигляд головної сторінки.

Назва товару	Платформа	Ціна	Жанр
Devil May Cry	Playstation 4	2240	Shooter
Far Cry 3	Xbox One	1795	First Person Shooter
Far Cry 4	Playstation 3	2199	First Person Shooter
Far Cry 5	Playstation 4	1199	First Person Shooter
Devil May Cry	Playstation 3	1700	Shooter
Devil May Cry	Xbox One	1700	Shooter
Call of War	Playstation 3	2999	Action-Adventure
Call of War	Playstation 4	1799	Action-Adventure
Far Cry 3	Xbox One	1100	First Person Shooter
The Last of Us I	Playstation 4	1850	Survival
The Last of Us I Remastered	Playstation 5	2100	Survival
The Last of Us II	Playstation 5	2999	Survival
Horizon Zero Dawn	Playstation 4	1199	Action-RPG
Horizon Zero Dawn	Playstation 5	1970	Action-RPG
Horizon Forbidden West	Playstation 5	2399	Action-RPG
FIFA 21	Playstation 4	1670	Sport Simulator
FIFA 22	Playstation 5	2100	Sport Simulator
FIFA 23	Xbox One	1720	Sport Simulator
FIFA 24	Playstation 5	2799	Sport Simulator
FIFA 23	Playstation 4	1800	Sport Simulator
FIFA 23	Xbox One	1700	Sport Simulator
Star Wars	Playstation 3	1949	Action-Adventure
Star Wars	Playstation 4	1199	Action-Adventure
Star Wars	Xbox One	1100	Action-Adventure
The Last of Us I Remastered	Playstation 4	1900	Survival
Call of Duty: Modern Warfare II	Playstation 5	2799	First Person Shooter
Call of Duty: Modern Warfare II	Playstation 4	1999	First Person Shooter
Call of Duty: Modern Warfare II	Xbox One	1799	First Person Shooter
Grand Theft Auto V	Playstation 5	2299	Action-Adventure
Grand Theft Auto V	Playstation 4	1799	Action-Adventure

Рисунок 4.3 – Зовнішній вигляд головної сторінки

На рис. 4.4 наведено сторінку детальної інформації про один з товарів. На рис. 4.5 показано інтерфейс сторінки «Ігри». На рис. 4.6 показано інтерфейс сторінки «Контролери».



Рисунок 4.4– Сторінка детальної інформації про один з товарів



Рисунок 4.5 – Інтерфейс сторінки «Ігри»

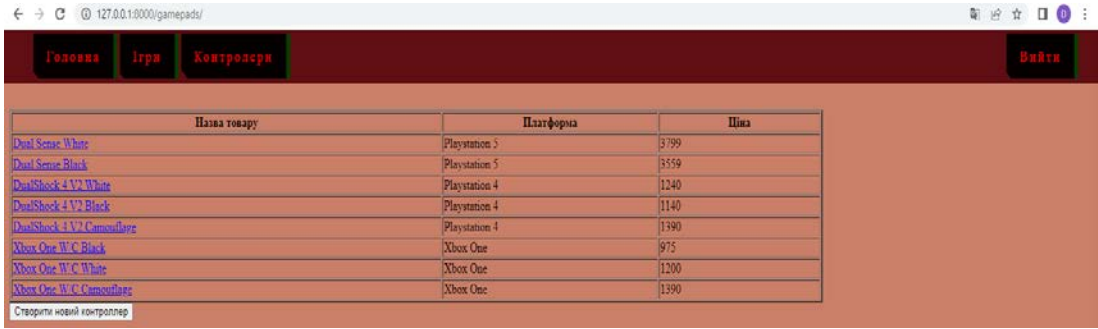


Рисунок 4.6 – Інтерфейс сторінки «Контролери»

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було реалізовано головну мету – програмну реалізацію вебзастосунку для менеджменту товарів у інтернет магазині ігор. Для цього було виконано ряд завдань:

- виконана постановка завдання на розробку;
- обрано інструментарій розробки;
- виконано проєктування нового вебзастосунку;
- виконано розробку та тестування вебзастосунку на основі проєктування.

Структура вебзастосунку була реалізована із використанням Framework Django та мови програмування Python .

Був розроблений сценарій та проведено тестування вебзастосунку. Тестування вебзастосунку було пройдено успішно. Отриманий результат свідчить про успішність проєкту.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки. Розроблена система є зручною для використання користувачем та виконує всі необхідні функції.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Загальні засади функціонування електронної комерції в Україні [Електрон. ресурс]. — Режим доступу: https://ips.ligazakon.net/document/view/t150675?an=51&ed=2020_01_14.
2. Близько 40% населення планети грають у відеоігри [Електрон. ресурс]. — Режим доступу: <https://nzl.theukrainians.org/grayut-u-videoigry.html>.
3. Управління асортиментом товарів [Електрон. ресурс]. — Режим доступу: <https://torgsoft.ua/articles/gid-po-torgsoft/assortiment-tovarov/>.
4. Функціональна модель Web-підприємства з мережею Інтернет-магазинів [Електрон. ресурс]. — Режим доступу: https://www.problecon.com/export_pdf/problems-of-economy-2015-2_0-pages-166_171.pdf.
5. Magento is one of the top ecommerce platforms in the United States and increasingly around the world too [Electronic source]. – Access mode: <https://www.rapyd.net/blog/what-is-magento/>
6. What Is Shopify And How Does It Work (2023 Guide) [Electronic source]. — Access mode: https://www.forbes.com/advisor/in/business/software/what-is-shopify/#what_is_shopify
7. What is PrestaShop? Tutorial For Beginners [Electronic source]. – Access mode: <https://blog.templatetoaster.com/what-is-prestashop/>
8. What Is PrestaShop? A Brief Introduction [Electronic source]. – Access mode: https://www.emizentech.com/blog/what-is-prestashop.html#Disadvantages_Of_Prestashop.
9. Python: плюси і мінуси мови, які завдання вирішує і чи варто вивчати [Електрон. ресурс]. — Режим доступу: <https://avada-media.ua/ua/services/python-plyusy-i-minusy-yazyka-kakiye-zadachi-reshayet-i-stoit-li-izuchat/>

10. Meet Django [Electronic source]. – Access mode: <https://www.djangoproject.com>
11. Django introduction [Electronic source]. – Access mode: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>
12. / Why We Use Django Framework & What Is Django Used For [Electronic source]. – Access mode: <https://djangostars.com/blog/why-we-use-django-framework/>
13. What Is a Database? [Electronic source]. – Access mode: <https://www.oracle.com/database/what-is-database/>
14. What is MySQL? Everything You Need to Know [Electronic source]. – Access mode: <https://ua.talend.com/resources/what-is-mysql/>
15. What is Data Modelling? Types (Conceptual, Logical, Physical). [Electronic source]. – Access mode: <https://www.guru99.com/data-modelling-conceptual-logical.html>
16. Бази даних. Поняття еr-моделі. Поняття сутності (entity). Атрибути. Види атрибутів [Електрон. ресурс]. — Режим доступу: <https://www.bestprog.net/uk/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ua/#q02>
17. What are Functional Requirements: Examples, Definition, Complete Guide [Electronic source]. – Access mode: <https://visuresolutions.com/blog/functional-requirements/>
18. User Interface (UI) Design [Electronic source]. – Access mode: <https://www.interaction-design.org/literature/topics/ui-design>
19. 10 Fundamental UI Design Principles You Need to Know [Electronic source]. – Access mode: <https://dribbble.com/resources/ui-design-principles>
20. Поняття "автоматизовані системи управління" [Електрон. ресурс]. — Режим доступу: https://stud.com.ua/21186/informatika/ponyattya_avtomatizovani_sistemi_upravlinnya

ДОДАТОК А
Текст програми

A.1 Файл `views.py`

```

from django.shortcuts import render
from django.http import HttpResponseRedirect
from .models import Games, Sales, Category, Controllers, Goods
from django.contrib.auth import authenticate, login, logout
from django.http import HttpResponseRedirect, HttpResponseRedirect
from django.shortcuts import render, get_object_or_404, redirect
from django.urls import reverse
from .forms import GamesForm, ControllersForm
import random
from django.db.models import Q

# Create your views here.

def index(request):
    if not request.user.is_authenticated:
        return HttpResponseRedirect(reverse("login"))
    query = request.GET.get("q")
    if query:
        games=Games.objects.filter(
            Q(name=query ) | Q(genre=query ))
        controllers=Controllers.objects.filter(
            Q(name=query ) | Q(platform=query ))
    else:
        games=Games.objects.all()
        controllers=Controllers.objects.all()
    return render(request, "table/goods.html", {
        "games": games, "controllers": controllers
    })

```

```
def gameid(request, id):
    game = get_object_or_404(Games, id=id)
    return render(request, 'table/gameid.html', {
        'game': game
    })

def controllerid(request, id):
    controller = get_object_or_404(Controllers, id=id)
    return render(request, 'table/controllerid.html', {
        'controller': controller
    })

def edit_controller(request, id):
    controller = get_object_or_404(Controllers, id=id)
    if request.method == 'POST':
        form = ControllersForm(request.POST, instance=controller)
        if form.is_valid():
            form.save()
            return redirect('controllerid', id=controller.id)
    else:
        form = ControllersForm(instance=controller)
    return render(request, 'table/work_good.html', {'form': form})

def new_controller(request):
    if request.method == 'POST':
        form = ControllersForm(request.POST)
        if form.is_valid():
            form.save()
```

```
        return redirect('index')
    else:
        form = ControllersForm()
        return render(request, 'table/work_good.html', {'form': form})

def delete_game(request, id):
    record = Games.objects.get(id=id)

    # Удаляем запись из базы данных
    record.delete()

    # Перенаправляем пользователя на другую страницу
    return redirect('index')

def delete_controller(request, id):
    record = Controllers.objects.get(id=id)

    # Удаляем запись из базы данных
    record.delete()

    # Перенаправляем пользователя на другую страницу
    return redirect('index')

def new_game(request):
    if request.method == 'POST':
        form = GamesForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('index')
    else:
```

```
        form = GamesForm()
    return render(request, 'table/work_good.html', {'form': form})

def games(request):
    return render(request, "table/games.html", {
        "games": Games.objects.all()
    })

def gamepads(request):
    return render(request, "table/gamepads.html", {
        "gamepads": Controllers.objects.all()
    })

def edit_game(request, id):
    game = get_object_or_404(Games, id=id)
    if request.method == 'POST':
        form = GamesForm(request.POST, instance=game)
        if form.is_valid():
            form.save()
            return redirect('gameid', id=game.id)
    else:
        form = GamesForm(instance=game)
    return render(request, 'table/work_good.html', {'form': form})

def login_view(request):
    if request.method == "POST":
        username = request.POST["username"]
        password = request.POST["password"]
```

```

        user = authenticate(request, username=username,
password=password)
        if user is not None:
            login(request, user)
            return HttpResponseRedirect(reverse("home"))
        else:
            return render(request, "table/login.html", {
                "message": "Invalid credentials."
            })
    return render(request, "table/login.html")

def logout_view(request):
    logout(request)
    return render(request, "table/login.html", {
        "message": "Logged out."
    })

def login_view(request):
    if request.method == "POST":
        username = request.POST["username"]
        password = request.POST["password"]
        user = authenticate(request, username=username,
password=password)
        if user is not None:
            login(request, user)
            return HttpResponseRedirect(reverse("index"))
        else:
            return render(request, "table/login.html", {
                "message": "Invalid credentials."
            })

```

```
return render(request, "table/login.html")
```

```
def goods(request):
    if not request.user.is_authenticated:
        return HttpResponseRedirect(reverse("login"))
```

A.2 Файл forms.py

```
from django import forms
from .models import Games, Category, Controllers
from django.contrib.auth.models import User

class GamesForm(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(attrs={"class":"form-
control"}))
    platform = forms.CharField(widget=forms.TextInput(attrs={"class":"form-
control"}))
    genre = forms.CharField(widget=forms.TextInput(attrs={"class":"form-
control"}))
    price = forms.IntegerField()
    id_category = forms.ModelChoiceField(queryset=Category.objects.all())
    balance = forms.IntegerField()

    class Meta:
        model = Games
        fields = ['name', 'platform', 'genre', 'price', 'id_category', 'balance']

class ControllersForm(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(attrs={"class":"form-
control"}))
```

```

        platform = forms.CharField(widget=forms.TextInput(attrs={"class":"form-control"}))

        price = forms.IntegerField()
        id_category = forms.ModelChoiceField(queryset=Category.objects.all())
        balance = forms.IntegerField()

    class Meta:
        model = Controllers
        fields = ['name', 'platform', 'price', 'id_category', 'balance']

```

Третій файл models.py

```
From django.db import models
```

```
From django.contrib.auth.models import user
```

```
# create your models here.
```

```
#table category
```

```
Class category(models.model):
```

```
    cat_name=models.charfield(max_length=100, null=true, blank=false)
```

```
    def __str__(self):
```

```
        return self.cat_name
```

```
#table games
```

```
Class games(models.model):
```

```
    name=models.charfield(max_length=100, null=true, blank=false)
```

```
    platform=models.charfield(max_length=20, null=true, blank=false)
```

```
    genre=models.charfield(max_length=40, null=true, blank=false)
```

```
    price=models.integerfield(null=true, blank=false)
```

```

        id_category=models.foreignkey(category, on_delete=models.cascade,
null=true, blank=false)
        balance=models.integerfield(null=true, blank=false)

```

```

#table controllers

```

```

Class controllers(models.model):
    name=models.charfield(max_length=100, null=true, blank=false)
    platform=models.charfield(max_length=20, null=true, blank=false)
    price=models.integerfield(null=true, blank=false)
    id_category=models.foreignkey(category, on_delete=models.cascade,
null=true, blank=false)
    balance=models.integerfield(null=true, blank=false)

```

```

#table sales

```

```

Class sales(models.model):
    number_check=models.integerfield(null=true, blank=false)
    date=models.datefield(null=true, blank=false)
    games_code=models.foreignkey(games, on_delete=models.cascade,
null=true, blank=true)
    controllers_code=models.foreignkey(controllers,
on_delete=models.cascade, null=true, blank=true)
    personnel_code=models.foreignkey(user, on_delete=models.cascade,
null=true, blank=false)
    quantity=models.integerfield(null=true, blank=false)
    total=models.integerfield(null=true, blank=false)

```

```

#table goods

```

```

Class goods(models.model):

```

```
        controllers_code=models.foreignkey(controllers,  
on_delete=models.cascade, null=true, blank=true)  
        games_code=models.foreignkey(games, on_delete=models.cascade,  
null=true, blank=true)
```

ДОДАТОК Б
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна робота бакалавра на тему:

"РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ
МЕНЕДЖМЕНТУ ТОВАРІВ У ІНТЕРНЕТ
МАГАЗИНІ ІГОР"



Розробив
студент
гр. КНТ-210СП

Керівник
к.т.н., доцент

Данило ГОРДЕЄВ

Ольга ГЛАДКОВА

Рисунок Б.1 – Слайд 1

<u>Об'єкт дослідження</u>	процес та методи ведення обліку товарів у магазинах.
<u>Предмет дослідження</u>	програмної реалізації складу інтернет-магазину.
<u>Мета роботи</u>	розроблення вебзастосунку для ведення обліку в інтернет-магазині ігор.

01

Рисунок Б.2 – Слайд 2

Постановка завдання

Метою даної дипломної роботи є розробка такого вебзастосунку, який надає ефективний інструментарій для менеджменту товарів у інтернет-магазині ігор. Цей вебзастосунок має забезпечити можливість швидкого та зручного додавання, редагування та видалення товарів, оновлення їх параметрів, а також надати інші функціональні можливості, що сприятимуть підвищенню ефективності менеджменту товарів у інтернет-магазині ігор.

Вебзастосунок забезпечує ефективне управління товарами у інтернет-магазині ігор, що дозволить поліпшити конкурентоспроможність магазину та задовольнити потреби його клієнтів, власників інтернет-магазинів ігор.

02

Рисунок Б.3 – Слайд 3

Порівняння існуючих аналогів

Критерії порівняння \ Назва застосунку	Magento	Shopify	PrestaShop	Розроблений вебзастосун.
Зручність навігації	+	+	-	+
Простий інтерфейс	+	-	-	+
Вартість	Безк.	Підписка	Підписка	Безк.
Вимоги до ПК	+	+	+	Незначні
Додаткове навчання для корист.	+	-	+	-
Обов'язова авторизація	+	+	+	+

03

Рисунок Б.4 – Слайд 4

Порівняння мов програмування			
Критерії порівняння \ Назва	Python	PHP	C ++
Продуктивність та швидкодія	±	±	+
Синтаксис і зручність використання	+	+	–
Розширюваність та модульність	+	+	+
Обробка помилок та безпека	+	±	±

04

Рисунок Б.5 – Слайд 5

Порівняння Framework		
Критерії порівняння \ Назва	Django	Ruby on Rails
Мова програмування	Python	Ruby
Архітектура та концепції	MVC або MTV	MVC
Швидкодія та масштабованість	+	±
Продуктивність розробки	Забезпечує широкий спектр інструментів та бібліотек для швидкої розробки, включаючи вбудовану адмін-панель, ORM (Object-Relational Mapping) та систему маршрутизації.	Зосереджений на збільшенні продуктивності розробки шляхом використання конвенцій над конфігурацією та готових рішень для багатьох задач, таких як міграції бази даних та автентифікація.

05

Рисунок Б.6 – Слайд 6

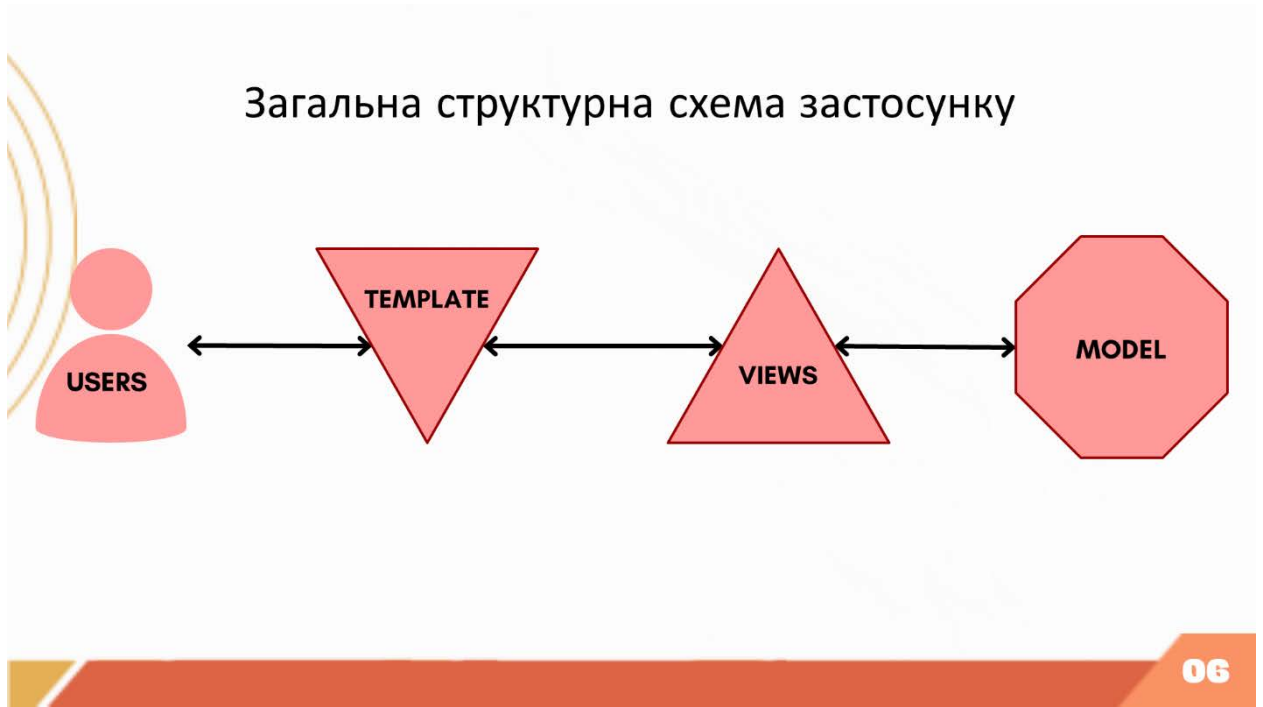


Рисунок Б.7 – Слайд 7



Рисунок Б.8 – Слайд 8

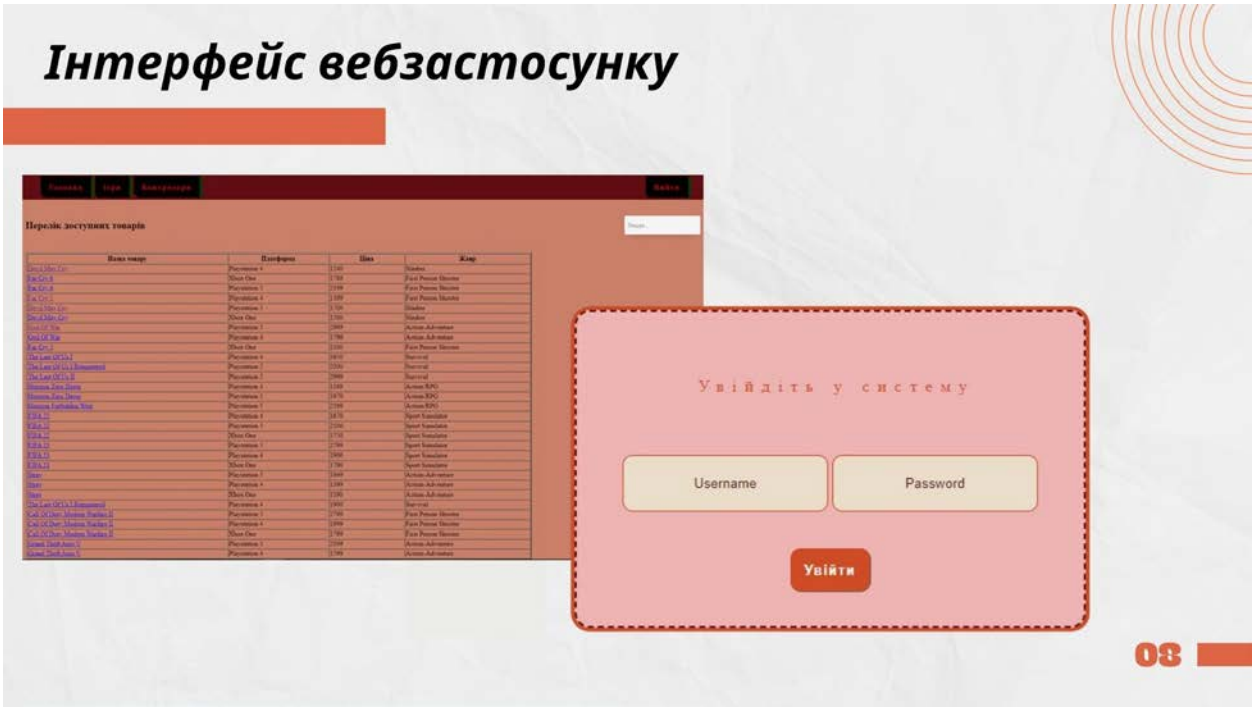


Рисунок Б.9 – Слайд 9



Рисунок Б.10 – Слайд 10