

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»

на тему:

«База даних магазину побутової техніки»

студента II курсу групи ІПЗ-21-5
спеціальності 121 «Інженерія
програмного забезпечення»

Волкова Олександра Максимовича

(прізвище, ім'я та по-батькові)

Керівник ст. викл., Олексій
ЧИЖМОТРА

Дата захисту: " 08 " _____ червня 2023 р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

(підпис)

Ольга КОРОТУН

(прізвище та ініціали)

(підпис)

Світлана КРАВЧЕНКО

(прізвище та ініціали)

(підпис)

Інна СУГОНЯК

(прізвище та ініціали)

(підпис)

Олексій ЧИЖМОТРА

(прізвище та ініціали)

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет інформаційно-комп'ютерних технологій
Кафедра інженерії програмного забезпечення
Освітній рівень: бакалавр
Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
В.о.зав. кафедри
_____ А.В.Морозов
“ ____ ” _____ 20__ р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ
Волкову Олександрю Максимовичу

1. Тема роботи: База даних магазину побутової техніки,
керівник курсової роботи: ст. викл. Олексій ЧИЖМОТРЯ
2. Строк подання студентом: “ 8 ” червня 2023 р.
3. Вихідні дані до роботи: розробити базу даних для магазину побутової техніки.
4. Зміст розрахунково-пояснювальної записки(перелік питань, які підлягають розробці)
 1. Постановка завдання
 2. Аналіз існуючих БД
 3. Алгоритми роботи програми
 4. Опис роботи програми
 5. Програмне дослідження
5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 1. Презентація до КП
 2. Посилання на репозиторій: <https://gitlab.com/2021-2025/ipz-21-5/volkov-alexander/databasecourswork/-/tree/master>
6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	Олексій ЧИЖМОТРЯ ст. викладач каф. ІПЗ	22.02.2023	22.02.2023

7. Дата видачі завдання “ 22 ” лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
1	Постановка задачі	22.02-25.02	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	26.02 – 28.02	Виконано
3	Формулювання технічного завдання	01.03 – 10.03	Виконано
4	Опрацювання літературних джерел	11.03 – 15.03	Виконано
5	Проектування структури	16.03 – 31.03	Виконано
6	Написання програмного коду	01.04 – 01.06	Виконано
7	Налагодження	02.06 – 04.06	Виконано
8	Написання пояснювальної записки	05.06 – 16.06	Виконано
9	Захист	08.06.2023	

Студент

(підпис)

Олександр ВОЛКОВ

(прізвище та ініціали)

Керівник роботи

(підпис)

Олексій ЧИЖМОТРЯ

(прізвище та ініціали)

РЕФЕРАТ

Завданням на курсової роботи було створення бази даних магазину побутової техніки.

Пояснювальна записка до курсової роботи на тему “База даних магазину побутової техніки” складається з вступу, чотирьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 59 сторінках друкованого тексту.

Пояснювальна записка має 14 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 42 рисунки. Загальний обсяг роботи – 74 сторінок.

Ключові слова: магазин, веб-додаток, БД, PHP, CMS, MySQL

					ДУ «Житомирська політехніка».23.121.07.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	«База даних магазину побутової техніки»	Літ.	Арк.	Аркушів
Розроб.		Олександр ВОЛКОВ						
Перевір.		Олексій ЧИЖМОТЯ					4	75
Керівник						ФІКТ Гр. ІПЗ-21-5[2]		
Н. контр.								
Зав. каф.								

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	8
1.1. Аналіз інформаційних потреб та визначення предметної області дослідження	9
1.2. Обґрунтування вибору засобів реалізації.....	10
Висновки до першого розділу.....	12
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
2.1. Аналіз інформаційних процесів	13
2.2. Проектування структури бази даних за напрямком курсової роботи.....	15
2.3. Розробка математичної моделі та алгоритмів обробки даних системи.....	26
Висновки до другого розділу	28
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ЗА НАПРЯМКОМ “БАЗА ДАНИХ МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ”	29
3.1. Проектування інтерфейсу магазину побутової техніки.....	29
3.2. Реалізація операцій обробки даних в БД	44
3.3. Реалізація звітності системи	47
Висновки до третього розділу	52
РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ	53
4.1. Розробка заходів захисту інформації в БД	53
4.2. Налаштування параметрів роботи SQL-сервера	55
Висновки до четвертого розділу.....	57
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТКИ.....	60

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних.

КП – Курсовий проект.

ПЗ – Програмне забезпечення.

SQL (Structured Query Language) - мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних.

СУБД - система управління базами даних.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми

Тема курсової роботи “База даних магазину побутової техніки” є дуже актуальною в сьогодення, оскільки ринок побутової техніки постійно розвивається та стрімко розширюється. Більшість компаній, що продають побутову техніку, використовують бази даних для управління своїм бізнесом: зберігання інформації про товари, замовлення, клієнтів та для іншої інформації.

Курсова робота на тему "База даних магазину побутової техніки" може включати створення бази даних, яка містить інформацію про різноманітні товари, що продаються в магазині, їх характеристики, ціни, стан наявності на складі, інформацію про клієнтів, що здійснюють покупки, їх замовлення, інформацію про поставки товарів та інше.

Створення такої бази даних допоможе зберігати, збирати і аналізувати інформацію, яка є важливою для ефективного управління бізнесом. Більш того, створення бази даних може покращити ефективність роботи магазину та допомогти забезпечити краще обслуговування клієнтів.

Таким чином, можна зробити висновок, що обрана тема є дуже актуальною в сучасному світі, оскільки бази даних допомагають бізнесу збирати, певним чином обробляти та аналізувати великі обсяги інформації, що є важливим для ефективного функціонування інтернет-магазину та його управління, а також для забезпечення високої якості обслуговування клієнтів.

Об’єктом дослідження є методи та засоби проектування баз даних для магазину побутової техніки.

Предметом дослідження є можливості застосування концепції БД та СУБД для забезпечення інформаційних потреб магазину.

Метою курсової роботи є дослідження особливостей проектування та реалізації баз даних магазину побутової техніки.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Завданням на курсову роботу є:

- аналіз теоретичних засад проектування та реалізації систем на основі баз даних;
- визначення інформаційних потреб магазину побутової техніки;
- аналіз напрямку ризиків інформаційних потоків та їх структури;
- проектування бази даних магазину побутової техніки;
- розробка математичної та алгоритмічної моделі функціонування системи на основі БД;
- реалізація БД та інтерфейсних засобів інформаційної системи.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1. Аналіз інформаційних потреб та визначення предметної області дослідження

Задача полягає у створенні бази даних для магазину побутової техніки. Для аналізу інформаційних потреб та визначення предметної області дослідження бази даних магазину побутової техніки, необхідно з'ясувати, які дані повинні бути збережені в базі даних та які є потреби клієнтів магазину щодо цих даних.

Наприклад, база даних магазину побутової техніки може містити наступні дані: назва товару, його опис (повний та короткий), характеристики (колір, країна виробника, марка, розмір та інші характеристики, в залежності від категорії, до якої належить товар), ціна, кількість товарів в наявності, зображення товару.

Важливо, щоб база даних зберігала інформацію про тих користувачів, які пройшли реєстрацію на сайті магазину і тому необхідно врахувати збереження наступних даних (прізвище, ім'я та по-батькові користувача, його логін, в якості якого виступає електронна пошта, пароль, який повинен бути обов'язково в захешованому вигляді, для суттєвого зменшення ризику несанкціонованого доступу до системи, а також БД повинна містити поле, яке зберігає визначену роль користувача у системі).

Важливим є наявність системи фільтрів, які допомагають полегшити пошук необхідного товару, тому є необхідність у створенні гнучкої системи, що дозволить додавати або видаляти наявні в БД фільтри з товарів і при цьому програма не зазнає краху. Для вирішення цієї проблеми потрібно реалізувати таблиці, які описують певну характеристику товарів (містить id та відповідне до нього певне значення), реалізувати таблицю, яка буде зберігати співвідношення категорії товару та всіх можливих назв таблиць фільтрів, які притаманні даній категорії, а також таблицю, яка буде

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		9

зберігати ідентифікатор товару, значення яке охарактеризовує товар, та ідентифікатор фільтру з якого ми можемо дізнатися в іншій таблиці назву таблиці до якої належить значення, яке охарактеризовує товар.

Щодо інформаційних потреб клієнтів, вони можуть бажати отримувати інформацію про товари, що продаються, їх характеристики, вартість, зображення. Крім того, клієнти можуть мати інтерес до відгуків попередніх покупців про товари, а також бажання залишати відгуки самостійно, тому необхідно створити таблицю, яка буде зберігати відгук покупця та рейтинг, яким він оцінить свої враження про придбаний товар.

З огляду на вищенаведене, предметною областю дослідження бази даних магазину побутової техніки можна вважати зберігання, оновлення та доступ до інформації про товари, що продаються, про користувачів, що зареєстровані, про фільтри та відгуки притаманні товарам.

Для реалізації магазину побутової техніки я обрав паттерн MVC (Model View Controller) реалізований вручну на мові PHP та веб-додаток PhpMyAdmin для керуваннями базами даних MySQL.

1.2. Обґрунтування вибору засобів реалізації

СУБД (Система управління базами даних) є програмним забезпеченням, яке дозволяє створювати, зберігати, організовувати, керувати та забезпечувати доступ до бази даних. Вона надає зручний інтерфейс для взаємодії з базою даних і виконання операцій з даними.

СУБД виконує різноманітні завдання, такі як створення таблиць для організації даних, введення, оновлення і видалення даних, виконання запитів для отримання потрібної інформації, забезпечення цілісності та безпеки даних, оптимізація роботи з базою даних для забезпечення ефективності та інші.

Для вибору реалізації бази даних в сьогодення існує велика кількість СУБД. Всі вони мають певні переваги та недоліки, тож перш ніж вибрати конкретну СУБД, складу для них порівняльну таблицю в якій розгляну основні переваги та недоліки.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Таблиця 1.1.

СУБД	Переваги	Недоліки
MySQL	<ul style="list-style-type: none"> • Простий у використанні • Висока продуктивність • Масштабованість і гнучкість • Відкритий код • Широка підтримка 	<ul style="list-style-type: none"> • Обмежена підтримка деяких розширень • Відносно обмежені можливості для обробки даних, порівняно з деякими іншими СУБД
Microsoft SQL Server	<ul style="list-style-type: none"> • Інтеграція з іншими продуктами Microsoft • Широкі можливості для бізнес-аналітики та звітності • Гнучкість в розгортанні та керуванні 	<ul style="list-style-type: none"> • Комерційна ліцензія та високі витрати • Обмежена підтримка для деяких платформ та мов програмування
MongoDB	<ul style="list-style-type: none"> • Гнучка структура даних • Хороша масштабованість • Підтримка географічних даних • Висока продуктивність в роботі з великими обсягами даних 	<ul style="list-style-type: none"> • Відсутність підтримки транзакцій • Обмежена підтримка SQL-запитів
SQLite	<ul style="list-style-type: none"> • Простота використання • Невеликі розміри • Портативність 	<ul style="list-style-type: none"> • Обмежена масштабованість • Погана продуктивність з великими обсягами даних

Таблиця 1.2.

Характеристика	MySQL	Microsoft SQL Server	SQLite	MongoDB
Модель даних	Реляційна	Реляційна	Реляційна	Документно-орієнтована
Адміністративне керування	Відмінно	Добре	Добре	Добре

Характеристика	MySQL	Microsoft SQL Server	SQLite	MongoDB
Графічні інструменти	Добре	Відмінно	Обмежено	Обмежено
Простота обслуговування	Відмінно	Відмінно	Відмінно	Добре
Робота з декількома ЦП	Добре	Задовільно	Добре	Відмінно
Одночасний доступ декількох користувачів	Відмінно	Добре	Добре	Відмінно
Підключення до Web	Добре	Задовільно	Добре	Добре
Побудова БД	Відмінно	Добре	Відмінно	Добре
Мова SQL	Відмінно	Відмінно	Відмінно	Відсутня
Вбудована мова програмування	Добре	Задовільно	Добре	Відмінно
Підтримка об'єктноорієнтованих парадигм	Відмінно	Задовільно	Відмінно	Добре
Інтеграція з іншими СУБД	Добре	Добре	Відмінно	Добре
Масштаб застосування	Великий	Великий	Невеликий	Великий

Провівши порівняльний аналіз переваг та недоліків вирішив реалізувати базу даних для інтернет магазину побутової техніки за допомогою MySQL та для його керування обрав веб-додаток phpMyAdmin.

Висновки до першого розділу

У ході виконання першого розділу було проведено аналіз інформаційних потреб та визначено предметну область дослідження. Було проведено порівняльний аналіз чотирьох популярних СУБД та на його основі визначено необхідний інструмент для реалізації баз даних інтернет магазину побутової техніки.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Аналіз інформаційних процесів

Загальна структура системи магазину побутової техніки:

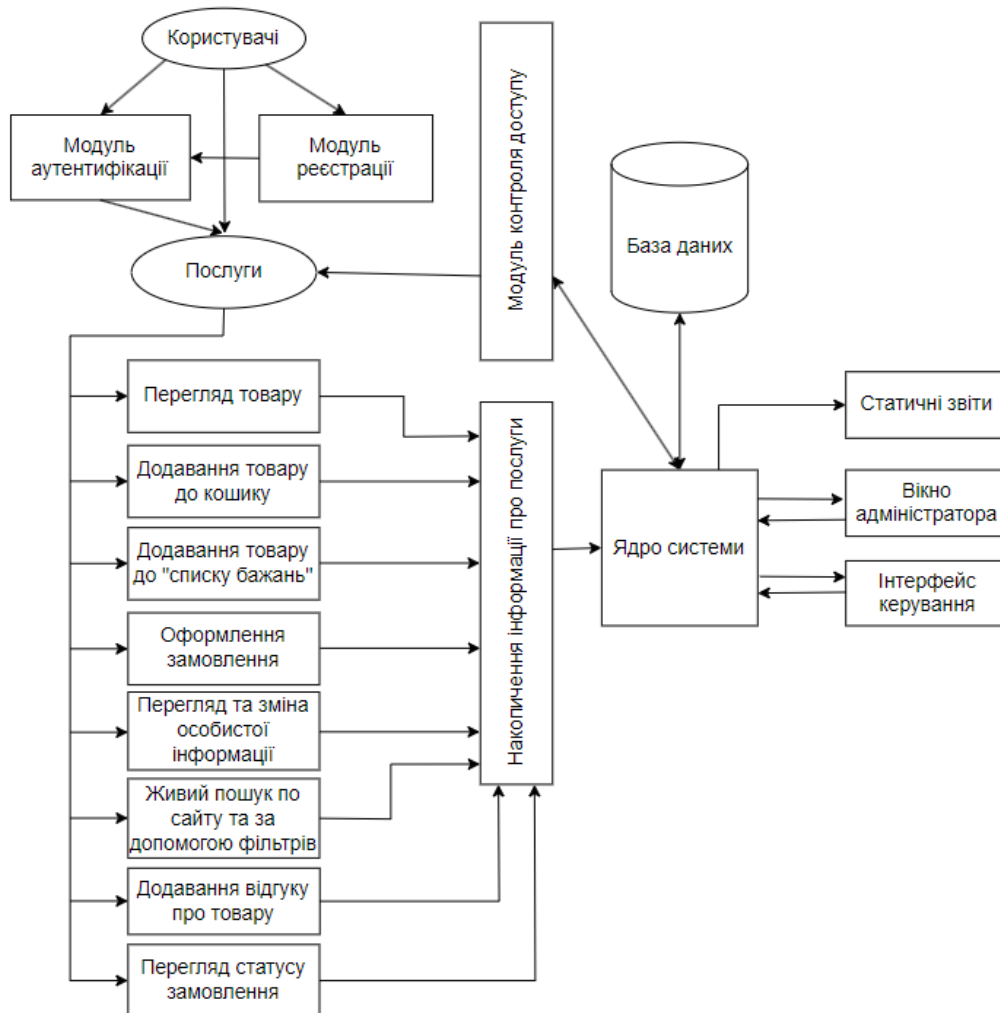


Рис. 2.1. Загальна структура роботи магазину побутової техніки

Маючи загальний алгоритм роботи програми, опишу конкретний перелік можливих дій для кожного типу користувача (неавторизованого користувача, авторизованого користувача який не є адміністратором та авторизованого користувача який є адміністратором).

Після того як користувач зайшов на сайт магазину, програма очікує дій.

Користувачі можуть виконати наступні дії у вкладці входу:

1. Вести логін та пароль, та за умови правильного вводу даних, натиснувши кнопку “Увійти”, в залежності від отриманих прав, перейти в головне меню магазину або на сторінку адміністратора.
2. Якщо користувач ще не зареєстрований, він має змогу пройти реєстрацію натиснувши кнопку “Реєстрація”.
3. Якщо користувач не бажає здійснювати реєстрацію, він має змогу перейти в головне меню магазину.

Користувач може виконати наступні дії у головному меню інтернет-магазину (коли обліковий запис не має прав адміністратора):

1. Погортати рекламний слайдер.
2. Обрати необхідну категорію товарів, після чого відкриється сторінка з товарами, які належать даній категорії.
3. Здійснити перегляд кошику та за наявності товарів оформити замовлення.
4. Здійснити перегляд історії покупок (за умови, що користувач є авторизованим в інтернет-магазині).
5. Здійснити перегляд товарів, які віднесені до списку охочих купити (за умови, що користувач є авторизованим в інтернет-магазині).
6. Написати коментар для товару з вказанням рейтингу (за умови що товар придбаний користувачем і користувач є авторизованим в системі)
7. Здійснити налаштування профілю (змінити особисті дані, логін, пароль або видалити поточний акаунт за умови що користувач є авторизованим в системі).

Користувач може виконати наступні дії у головному меню інтернет-магазину (коли обліковий запис є з правами адміністратора):

1. Переглянути точки видачі розетки на міні-карті (для логістики).
2. Додати, змінити або видалити категорію.
3. Додати товар до нової категорії або змінити та видалити товар в певній категорії.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		14

4. Змінити фото слайдеру та посилання на ці фото.
5. Змінити статус замовлення клієнта.
6. Змінити права облікового запису.
7. Переглянути статистичні дані (обіг товарів за конкретний рік, обсяг проданого товару за конкретний місяць та рік для кожної категорії, рейтинг оплати товарів готівковим та безготівковим способом, рейтинг товару для усієї категорії за результатами залишених оцінок від покупців в відгуках до товару).
8. Переглянути кількість залишених товарів на складі та їх загальну суму для кожної категорії.
9. Здійснити резервне копіювання БД за вказаним шляхом для зберігання.
10. Здійснити налаштування профілю (змінити особисті дані, логін, пароль або видалити поточний акаунт).

2.2. Проектування структури бази даних за напрямком курсової роботи

Основним заданням курсового проекту було створення БД, яка забезпечить успішне функціонування магазину побутової техніки. Для реалізації цієї цілі було використано СУБД MySQL, веб-додаток PhpMyAdmin. Загальна кількість таблиць створена в базі даних "SHOPCMS" складає 25:

- basket
- category_filter
- category_product
- color
- comment_products
- country
- destination
- energy_consumption
- filter
- operation_memory

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		15

- power
- product
- product_filter_value
- product_mark
- product_photo
- slider
- spin_drum_count
- towns
- typePayment
- user
- volume_refrigerating_chamber
- width_cm
- wish_list
- _order_

Розберемо структуру кожної таблиці:

Таблиця “basket” призначена для збереження вмісту кошику для аутентифікованих користувачів. Структура таблиці наведена нижче:

Таблиця 2.1.

Структура таблиці “basket”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
id_user	int(11)	-	+	Ідентифікаційний номер користувача (зв’язаний з таблицею user)
count	int(11)	-	-	Кількість одиниць товару
id_product	int(11)	-	+	Ідентифікаційний номер товару (зв’язаний з таблицею product)

Таблиця “category_filter” призначена для співставлення категорій товарів та фільтрів, які належать цій категорії. Структура таблиці наведена нижче:

Таблиця 2.2.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Структура таблиці “category_filter ”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
category_id	int(11)	-	+	Ідентифікаційний номер категорії (зв'язаний з таблицею category_product)
filter_id	int(11)	-	+	Ідентифікаційний номер фільтру (зв'язаний з таблицею filter)

Таблиця “category_product” призначена для збереження інформації про категорію товару. Структура таблиці наведена нижче:

Таблиця 2.3.

Структура таблиці “category_product”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Назва категорії
photo	varchar(255)	-	-	Назва фото, яке належить даній категорії

Таблиця “color” призначена для збереження переліку кольорів. Структура таблиці наведена нижче:

Таблиця 2.4.

Структура таблиці “color”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	varchar(255)	-	-	Назва кольору

Таблиця “comment_products” призначена для збереження коментаря про товар, рейтингу, його переваг та недоліків, повне ім'я користувача, під яким він бажає залишити відгук, ідентифікаційний номер користувача, ідентифікаційний номер товару, дата залишення коментаря. Структура таблиці наведена нижче:

Таблиця 2.5.

Структура таблиці “comment_products”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
comment	text	-	-	Коментар до товару
reiting	int(11)	-	-	Рейтинг товару
advantages	text	-	-	Переваги товару
disadvantages	text	-	-	Недоліки товару
user_name	varchar(255)	-	-	Повне ім'я (нікнейм) під яким користувач бажає залишити відгук
id_user	int(11)	-	+	Ідентифікаційний номер користувача, який залишив відгук (зв'язаний з таблицею user)
id_product	int(11)	-	+	Ідентифікаційний номер товару, для якого залишено відгук (зв'язаний з таблицею product)
date	date	-	-	Дата написання відгуку

Таблиця “country” призначена для збереження переліку країн-виробників товару. Структура таблиці наведена нижче:

Таблиця 2.6.

Структура таблиці “country”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	varchar(255)	-	-	Назва країни

Таблиця “destination” призначена для збереження пунктів видачі замовлень, створеного магазину побутової техніки. Структура таблиці наведена нижче:

Таблиця 2.7.

Структура таблиці “destination”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Адреса пункту видачі
id_town	int(11)	-	+	Ідентифікаційний номер міста пункту видачі (зв'язаний з таблицею towns)

Таблиця “energy_consumption” призначена для збереження переліку класів розподілу товарів за споживанням електроенергії. Структура таблиці наведена нижче:

Таблиця 2.8.

Структура таблиці “energy_consumption”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	varchar(255)	-	-	Назва класу у відповідності до споживання електроенергії

Таблиця “filter” призначена для збереження переліку таблиць та опису інформації яку вони зберігають і які використовуються у системі фільтрування товарів за певною характеристикою. Структура таблиці наведена нижче:

Таблиця 2.9.

Структура таблиці “filter”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Опис інформації яку зберігає таблиця
table_name	varchar(255)	-	-	Назва таблиці

Таблиця “operation_memory” призначена для збереження переліку можливих значень вмісту оперативної пам'яті. Структура таблиці наведена нижче:

Таблиця 2.10.

Структура таблиці “operation_memory”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	int(11)	-	-	Кількість оперативної пам'яті

Таблиця “power” призначена для збереження переліку можливих значень потужності товару. Структура таблиці наведена нижче:

Таблиця 2.11.

Структура таблиці “power”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	int(11)	-	-	Кількість одиниць потужності

Таблиця “product” призначена для збереження інформації про товар. Структура таблиці наведена нижче:

Таблиця 2.12.

Структура таблиці “product”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
count	int(11)	-	-	Кількість товару
name	varchar(255)	+	-	Назва товару (зв'язаний з таблицею product_mark)
price	decimal	-	-	Ціна товару
id_category	int(11)	-	+	Ідентифікаційний номер категорії (зв'язаний з таблицею category_product)
brief_description	text	-	-	Короткий опис товару
full_description	text	-	-	Повний опис товару
visibility	int(11)	-	-	Видимість товару користувачам
id_mark	int(11)	-	+	Ідентифікаційний номер марки товару (зв'язаний з таблицею product_mark)

Таблиця “product_filter_value” призначена для збереження переліку можливих значень потужності товару. Структура таблиці наведена нижче:

Таблиця 2.13.

Структура таблиці “power”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	int(11)	-	-	Кількість одиниць потужності

Таблиця “product_mark” призначена для збереження інформації про список марок (компаній) до яких належить конкретний товар. Структура таблиці наведена нижче:

Таблиця 2.14.

Структура таблиці “product_mark”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Назва марки (компанії) товару

Таблиця “product_photo” призначена для збереження назви фотографій, які належать конкретному товару. Структура таблиці наведена нижче:

Таблиця 2.15.

Структура таблиці “product_photo”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Назва фотографії
product_name	varchar(255)	-	+	Назва товару, якому належить дана фотографія (зв’язаний з таблицею product)

Таблиця “slider” призначена для збереження назви фотографій, які належать слайдеру та посилань, які належать конкретному зображенню. Структура таблиці наведена нижче:

Таблиця 2.16.

Структура таблиці “ slider ”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Назва фотографії
first_image	int(11)	-	-	Поле яке вказує, чи є дане зображення першим у слайдері
url	varchar(255)	-	-	Посилання, яке належить даному зображенню

Таблиця “spin_drum_count” призначена для збереження списку можливих кількостей обертів барабану, який належить до категорії товарів “пральні машинки”. Структура таблиці наведена нижче:

Таблиця 2.17.

Структура таблиці “spin_drum_count”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	int(11)	-	-	Значення кількості обертів барабану

Таблиця “towns” призначена для збереження списку міст, одне з яких необхідно обрати для успішного здійснення замовлення і яке буде слугувати точкою доставки товару. Структура таблиці наведена нижче:

Таблиця 2.18.

Структура таблиці “towns”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Назва міста

Таблиця “typePayment” призначена для збереження списку можливих типів оплат замовлення. Структура таблиці наведена нижче:

Таблиця 2.19.

Структура таблиці “typePayment”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
name	varchar(255)	-	-	Назва типу оплати

Таблиця “user” призначена для збереження інформації про користувача.
Структура таблиці наведена нижче:

Таблиця 2.20.

Структура таблиці “user”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
firstName	varchar(255)	-	-	Ім'я користувача
middleName	varchar(255)	-	-	Прізвище користувача
lastName	varchar(255)	-	-	По-батькові користувача
login	varchar(255)	-	-	Логін користувача
password	varchar(255)	-	-	Пароль користувача
typeAccess	int(11)	-	-	Тип доступу користувача

Таблиця “volume_refrigerating_chamber” призначена для збереження списку діапазону значень, яким може належати об'єм камери холодильника.
Структура таблиці наведена нижче:

Таблиця 2.21.

Структура таблиці “volume_refrigerating_chamber”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	varchar(255)	-	-	Діапазон можливого значення об'єму камери холодильника

Таблиця “water_consumption” призначена для збереження списку значень, яким може належати кількість спожитої води пральною машинкою за один цикл прання. Структура таблиці наведена нижче:

Таблиця 2.22.

Структура таблиці “water_consumption”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	double	-	-	Значення, якому може дорівнювати кількість спожитої води пральною машинкою за один цикл прання

Таблиця “width_cm” призначена для збереження списку можливих діапазонів ширини в сантиметрах, яким можуть належати габаритні розміри техніки. Структура таблиці наведена нижче:

Таблиця 2.23.

Структура таблиці “width_cm”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
value	varchar(255)	-	-	Діапазон, межах якого може належати ширина побутової техніки

Таблиця “wish_list” призначена для збереження списку товарів, які користувач можливо захоче придбати в найближчому майбутньому або для швидкого їх знаходження в разі наступного входу в систему. Структура таблиці наведена нижче:

Таблиця 2.24.

Структура таблиці “wish_list”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
id_product	int(11)	-	+	Ідентифікатор товару, який знаходиться в “списку бажань” (зв’язаний з таблицею product)
id_user	int(11)	-	+	Ідентифікатор користувача, у якого знаходиться даний товар у “списку бажань” (зв’язаний з таблицею user)

Таблиця “_order_” призначена для збереження замовлених товарів користувачів та інформацію про замовлення та замовників. Структура таблиці наведена нижче:

Таблиця 2.25.

Структура таблиці “_order_”

Назва	Тип даних	ПК	ЗК	Опис поля
id	int(11) identity	+	-	Ідентифікаційний номер запису
id_product	int(11)	-	+	Ідентифікатор товару, який був замовлений (зв'язаний з таблицею product)
id_destination	int(11)	-	+	Ідентифікатор місця призначення замовлення (зв'язаний з таблицею destination)
id_user	int(11)	-	+	Ідентифікатор користувача який замовив товар (зв'язаний з таблицею user)
mobile	varchar(255)	-	-	Мобільний телефон замовника
firstName	varchar(255)	-	-	Ім'я замовника (якщо користувач не аутентифікований)
middleName	varchar(255)	-	-	Прізвище замовника (якщо користувач не аутентифікований)
login	varchar(255)	-	-	Логін замовника (якщо користувач не аутентифікований)
typePayment_id	int(11)	-	+	Тип оплати замовлення (зв'язаний з таблицею typePayment)
count	int(11)	-	-	Кількість одиниць замовленого товару даного типу
status	int(11)	-	-	Статус замовлення
date	date	-	-	Дата замовлення

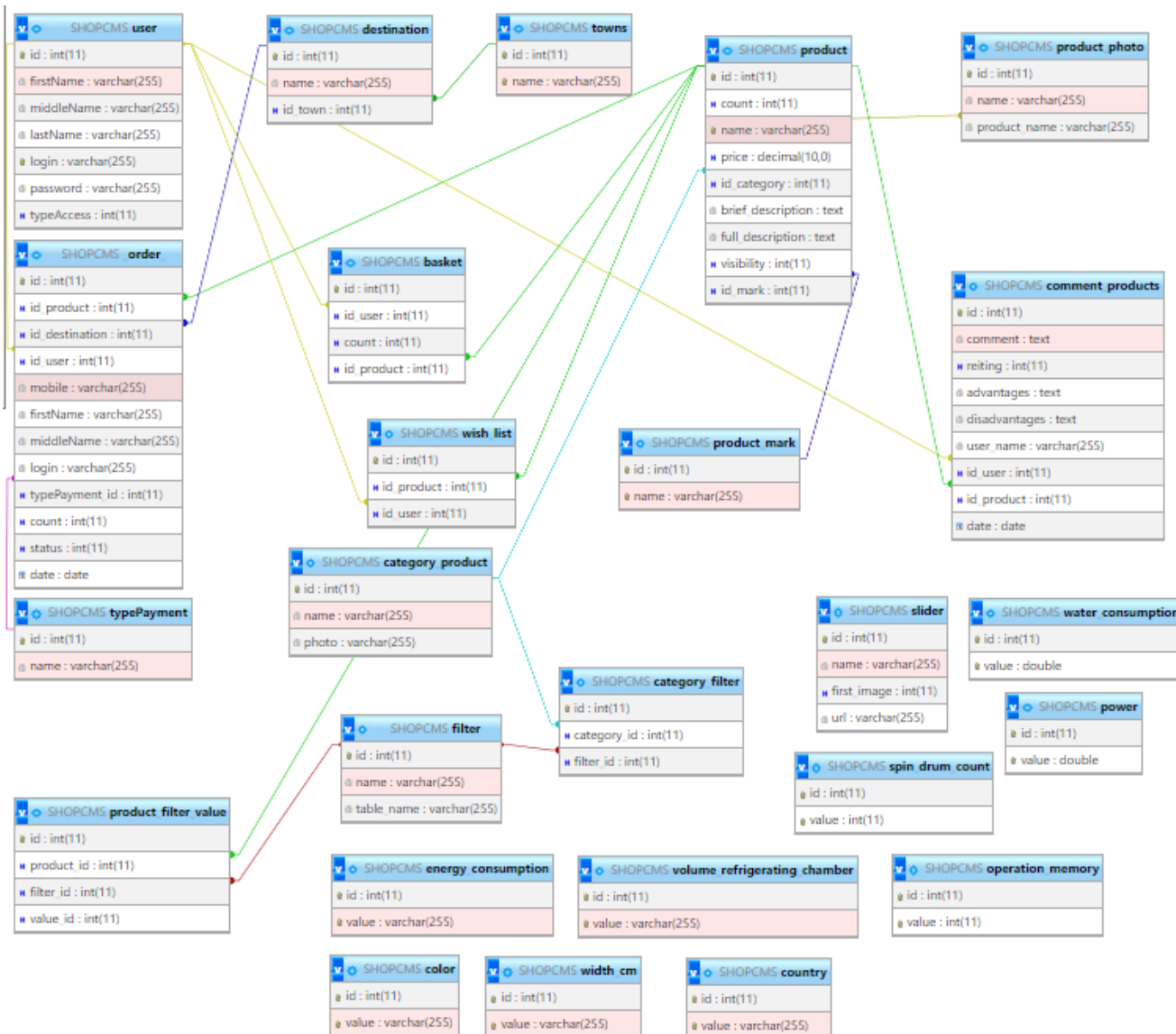


Рис. 2.2. Діаграма бази даних “SHOPCMS”

2.3. Розробка математичної моделі та алгоритмів обробки даних системи

Розглянемо алгоритми обробки статистичних даних:

1) Обіг продукції протягом року по місяцям.

Для знаходження всіх місяців року у яких було зроблено продаж товару та суму отриманого доходу використав наступний алгоритм:

```
$date = Order::getAllOrders([
    'YEAR(date)' => $_POST['year']
], [
    'MONTH(date) AS month', 'SUM(_order_.count*product.price) AS sum'
], null, ['month'], null, null, null, null, "JOIN product on _order_.id_product = product.id");
```

Вибірка виконується з таблиці замовлень – “_order_”. Першим параметром в метод Order::getAllOrders передав умову - рік який обрав користувач і для якого потрібно вибрати обіг продукції, другим параметром передав параметр для вибірки: місяці даного року та формулу для знаходження обігу продукції, який обраховується для кожного знайденого місяця за наступним алгоритмом:

`SUM(_order_.count*product.price)`, де “_order_.count” – кількість продуктів в замовленні, “product.price” – ціна за одиницю товару в замовленні, “SUM(_order_.count*product.price)” – обчислює суму всіх проданих товарів за відповідний місяць, оскільки 4 параметром передано значення, яке відповідає за групування отриманих даних по місяцям. Останній параметр відповідає за поєднання таблиць.

Далі для представлення даних у потрібній формі виконується наступний код:

```
$chartData = [];
if(!empty($date)){
    foreach ($date as $key=>$value){
        $chartData[$value['month']] = $value['sum'];
    }
}
```

В данному алгоритмі відбувається перевірка отриманих даних на порожнечу і якщо масив не порожній, то відбувається циклічне його проходження та отримання з нього необхідних даних (місяця та суми продажу (обігу) за даний місяць) з наступним записом його в допоміжний масив, який передається в представлення.

- 2) Рейтинг для кожної категорії товару, який обчислюється в результаті обробки рейтингів залишининими покупцями до товарів, які належить цій категорії.

Для знаходження всіх категорій товарів та оцінки для них використав наступний алгоритм:

```
$values = Comment::selectComments(['id_category',
'sum(comment_products.reiting)/count(comment_products.reiting) as
mark'],null,null,'product.id_category',null,null,null,null,'JOIN product on
product.id=comment_products.id product');
```

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				27
Змн.	Арк.	№ докум.	Підпис	Дата		

В даний метод першим параметром передається масив даних для вибірки, другим елементом якого є знаходження рейтингу за категорією `sum(comment_products.reiting)/count(comment_products.reiting) as mark'` де відбувається знаходження середнього арифметичного між сумою всіх оцінок товару категорії та кількості оцінок за допомогою агрегатних функцій, до того ж відбувається групування даних по id категорії і в данному запиті використано об'єднання таблиць за допомогою команди "join". Наступним етапом є організація даних у зручному для обробки вигляді за допомогою наступного алгоритму:

```
foreach ($values as $key=>$value) {
    $categoriesMarks[$categoriesArray[$value["id_category"]]]=$value["mark"];
}
```

Відбувається циклічне проходження по масиву даних, отриманих з БД та запис в якості ключа масива "\$categoriesMarks" назву категорії, а в якості значення – оцінка категорії.

Висновки до другого розділу

У ході виконання другого розділу, було проведено аналіз інформаційних процесів та сформовано загальний алгоритм роботи програми, спроектовано базу даних для інтернет магазину побутової техніки, описана кожна таблиця в БД та зв'язки між ними, розглянуто основні алгоритми обробки статистичних даних в системі.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		28

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ЗА НАПРЯМКОМ “БАЗА ДАНИХ МАГАЗИНУ ПОБУТОВОЇ ТЕХНІКИ”

3.1. Проектування інтерфейсу магазину побутової техніки

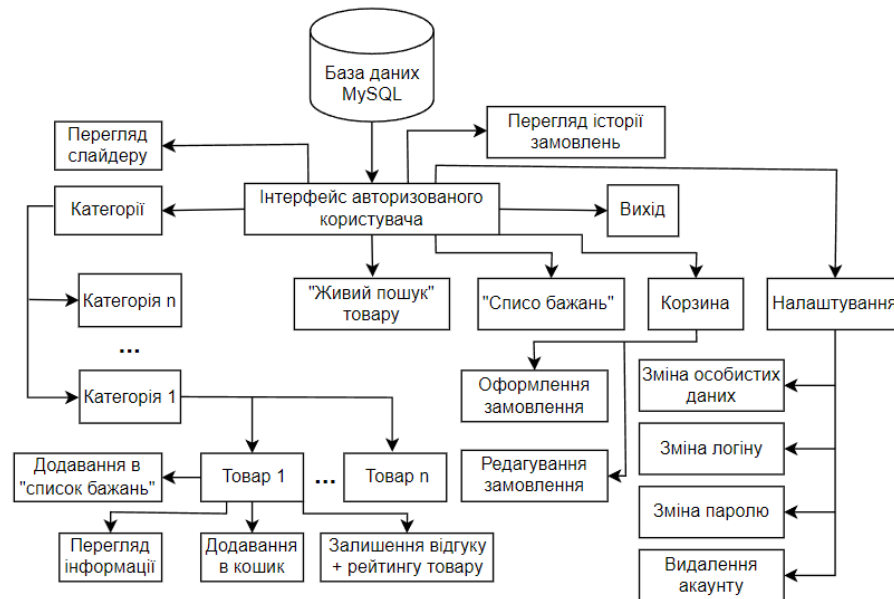


Рис. 3.1. Структура інтерфейсу магазину з облікового запису авторизованого користувача (не адміна)

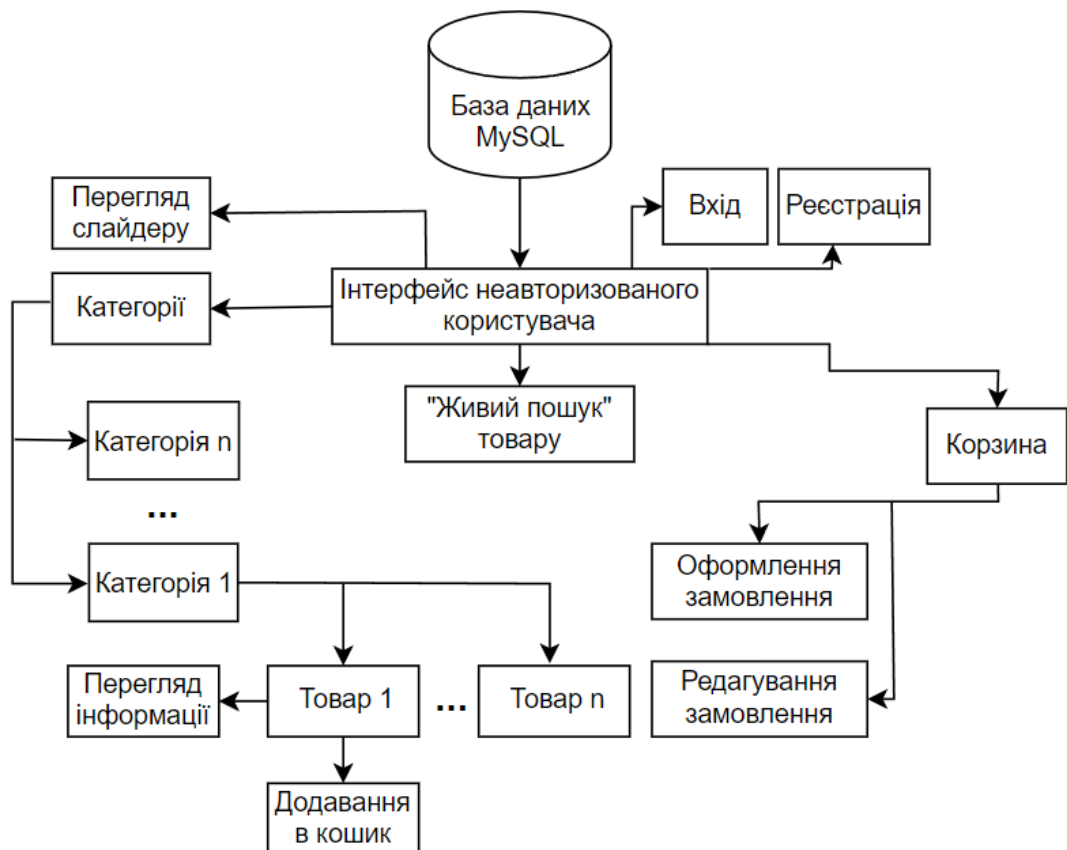


Рис. 3.2. Структура інтерфейсу магазину зі сторони неавторизованого користувача

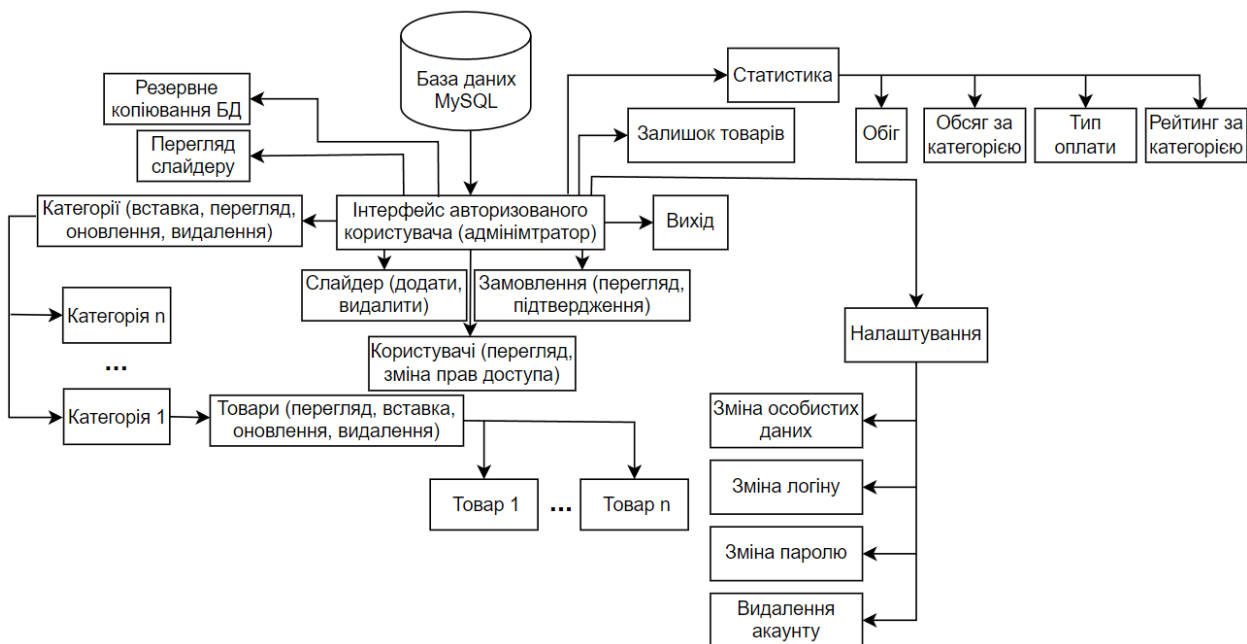


Рис. 3.3. Структура інтерфейсу магазину з облікового запису адміністратора

Розгляну спроектований інтерфейс магазину побутової техніки на прикладі побудованого сайту.

Після переходу неавторизованого користувача на сайт інтернет-магазину, перед ним з'являється вікно, де він може пройти авторизацію (рис. 3.4)

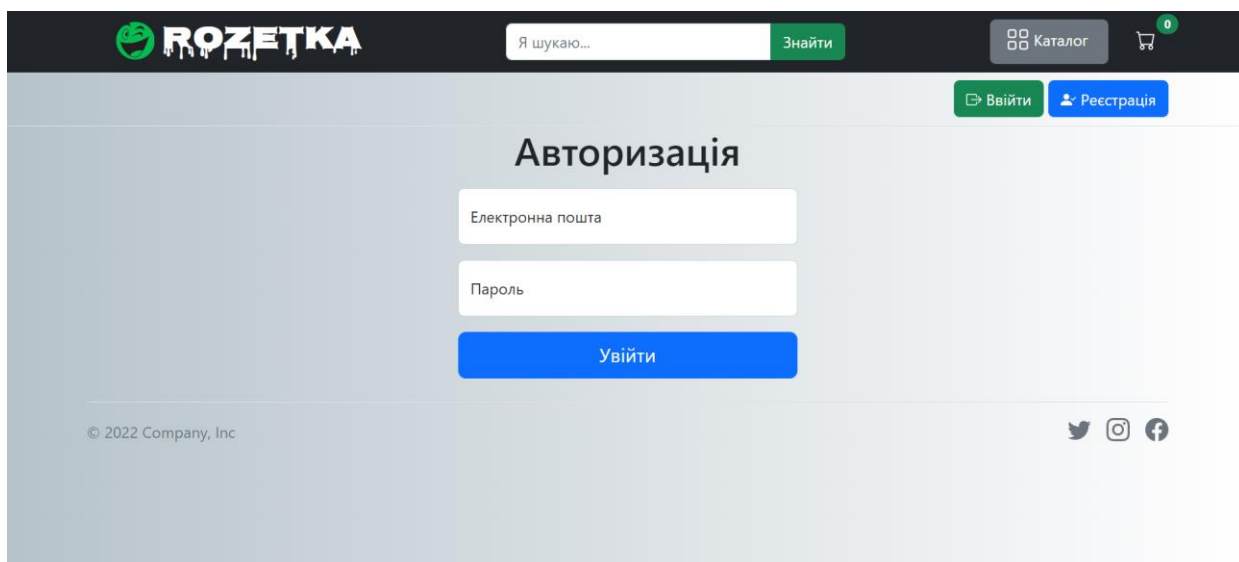


Рис. 3.4. Вікно авторизації користувача

Якщо користувач бажає здійснити реєстрацію, він повинен натиснути кнопку “Реєстрація”, після чого перед ним з'явиться відповідне вікно (рис. 3.5)

Рис. 3.5. Вікно реєстрації користувача

Для здійснення успішної реєстрації користувачу слід вказати прізвище, ім'я та по-батькові, які повинні обов'язково починатися з великої літери та бути записаними літерами кирилиці. Електронна пошта повинна відповідати стандартам синтаксису та не повинна бути вже використана в реєстрації раніше, а пароль слід придумати такий, який повинен складатися з великих та малих літер англійського алфавіту, з чисел, та не повинен бути довжиною менше 8 символів і повинен співпадати зі своїм повторенням.

Якщо користувач введе неправильно дані при реєстрації, то система сповістить користувача про помилку, інакше він буде перекинутий на форму авторизації, для проходження якої, йому потрібно вказати правильно логін та пароль. Якщо користувач не є адміністратором, то його буде перекинуто на наступну сторінку (3.6)

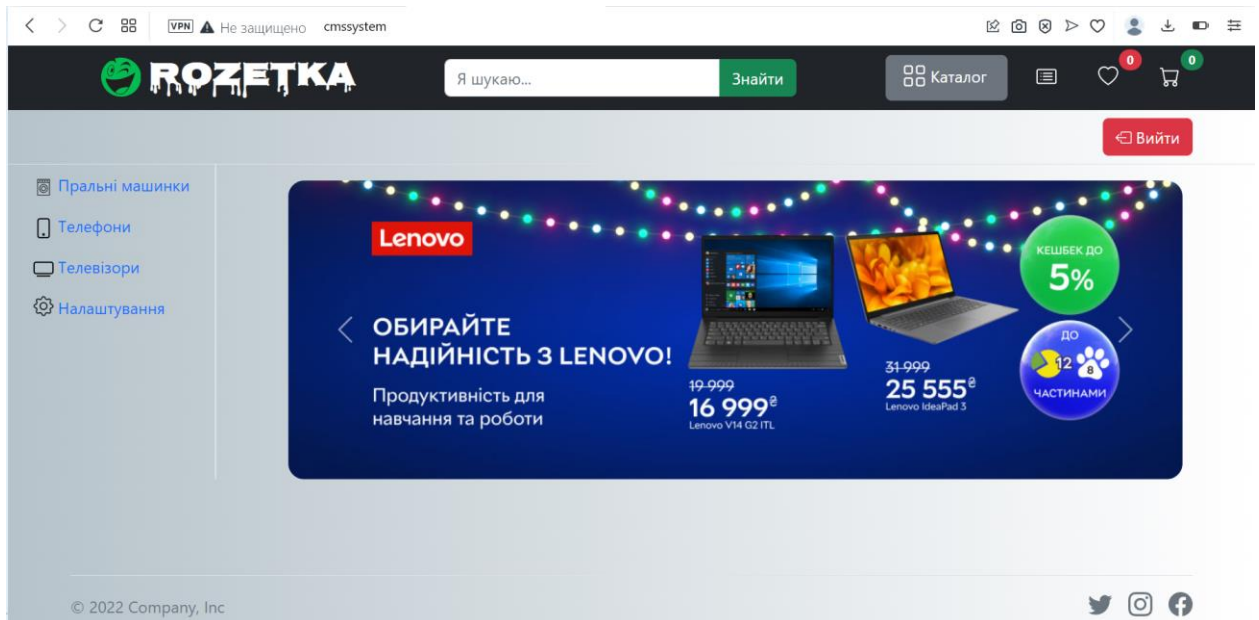


Рис. 3.6. Головне вікно інтернет-магазину (для користувача що не є адміністратором і є авторизованим)

Перед користувачем з'являється можливість погортати рекламний слайдер та за бажанням натиснути на відповідну рекламу товару, після чого він буде перекинутий на відповідну сторінку товару.

Зліва, в головному меню інтернет-магазину, розміщені категорії товарів, при кліку на які, користувач буде перекинутий на сторінку з відображенням усіх товарів, що належать даній категорії. (рис. 3.7)

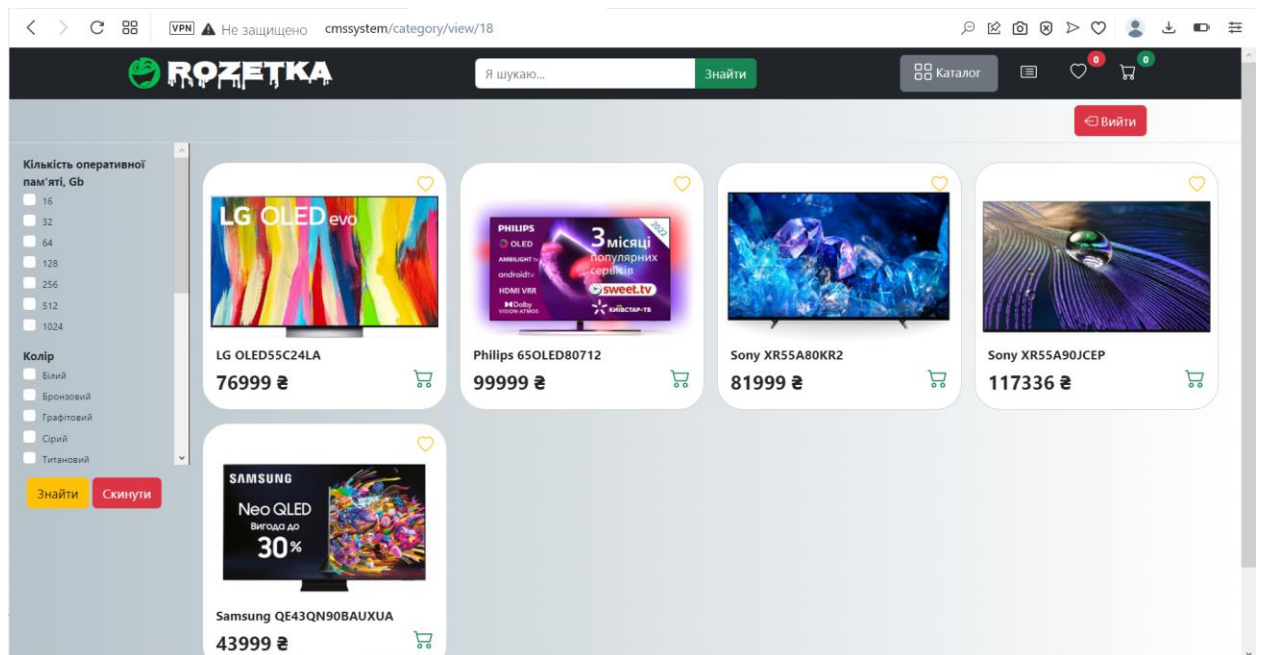


Рис. 3.7. Сторінка з відображенням товарів які належать до категорії “Телевізори”

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		32

На цій сторінці користувач може знайти необхідний товар за допомогою використання “фільтрів товару”, додати товар в кошик, натиснувши зелену кнопку, яка розміщена в правому нижньому кутку карточки товару, після чого дана кнопка стане повністю зелена і з’явиться галочка, яка символізує, що даний товар знаходиться в кошику, якщо кнопка на даному місці є сірого кольору, то це сигналізує, що товару немає в наявності і тому користувач не зможе його додати до кошику. Також користувач може додати товари в список бажань, натиснувши на кнопку в вигляді “сердечка”, після чого вона стане повністю замальована в жовтий колір, що є сигналом для користувача про знаходження даного товару в списку бажань. Натиснувши на зображення будь-якого з представлених товарів, користувача буде перекинуто на сторінку з інформацією про товар (рис. 3.8)

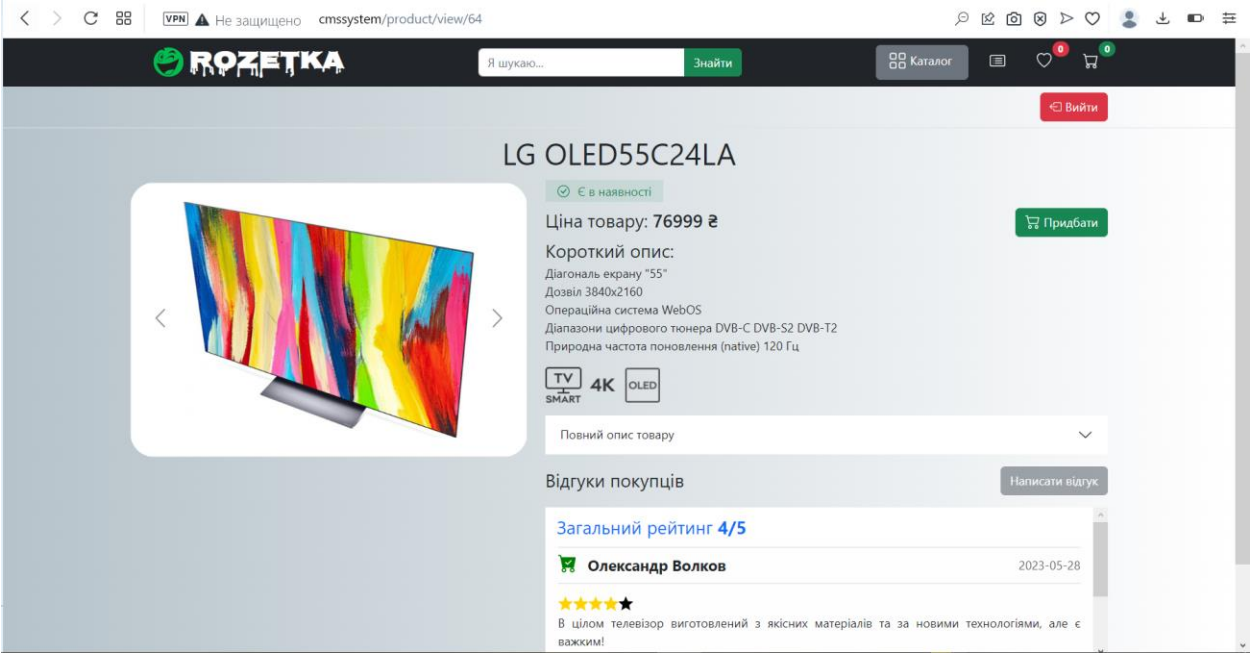


Рис. 3.8. Сторінка з відображенням інформації про товар та коментарів

Перед користувачем з’являється опис товару, а також список коментарів (за наявності). Якщо користувач здійснював покупку даного товару раніше та ще не залишав відгук, він має змогу це зробити натиснувши кнопку “Написати відгук”, після чого перед ним з’явиться спеціальне модальне вікно, де він може це зробити (рис 3.9)

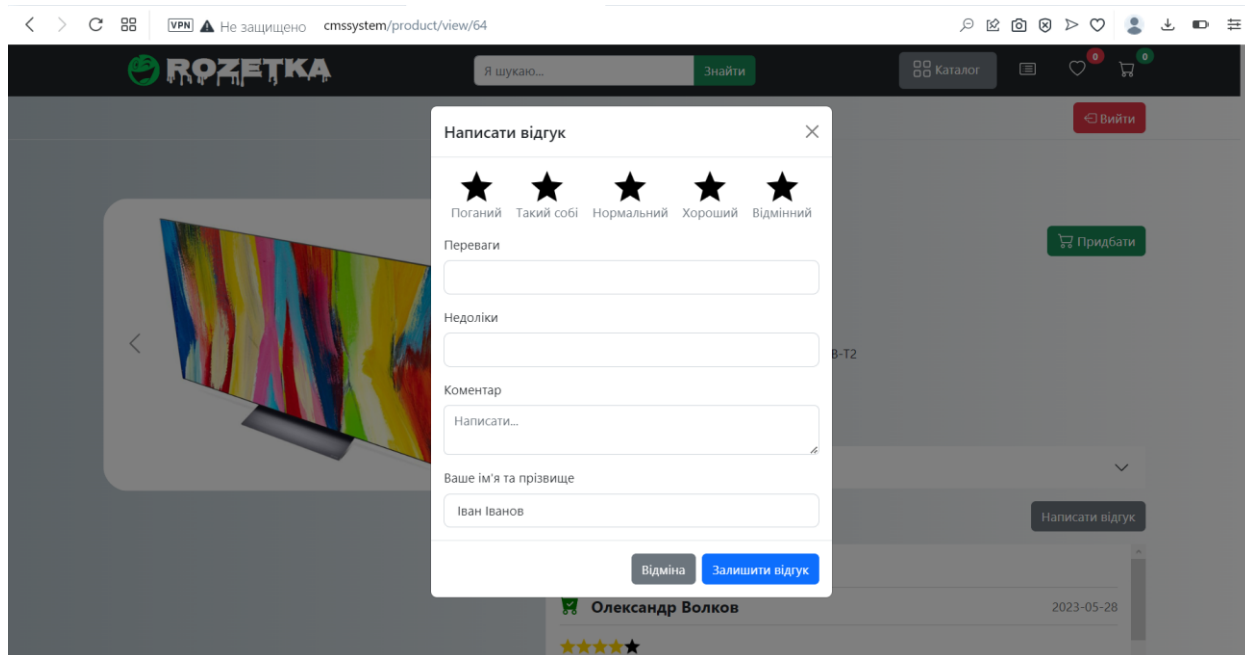


Рис. 3.9. Сторінка з модальним вікном для написання відгуку

В даному вікні потрібно вказати кількість зірок рейтингу, переваги та недоліки (за бажанням), загальний коментар, що є обов'язковим, а також прізвище та ім'я, після чого натиснути кнопку “Залишити відгук” для його надсилання або натиснути кнопку “Відміна”, якщо користувач не хоче надсилати відгук.

У верхньому меню користувача (заголовку) знаходиться кнопка, у вигляді списку, після натискання якої, відкривається список замовлень та статус їх обробки, розгорнувши який, можна побачити опис замовлення (рис. 3.10)

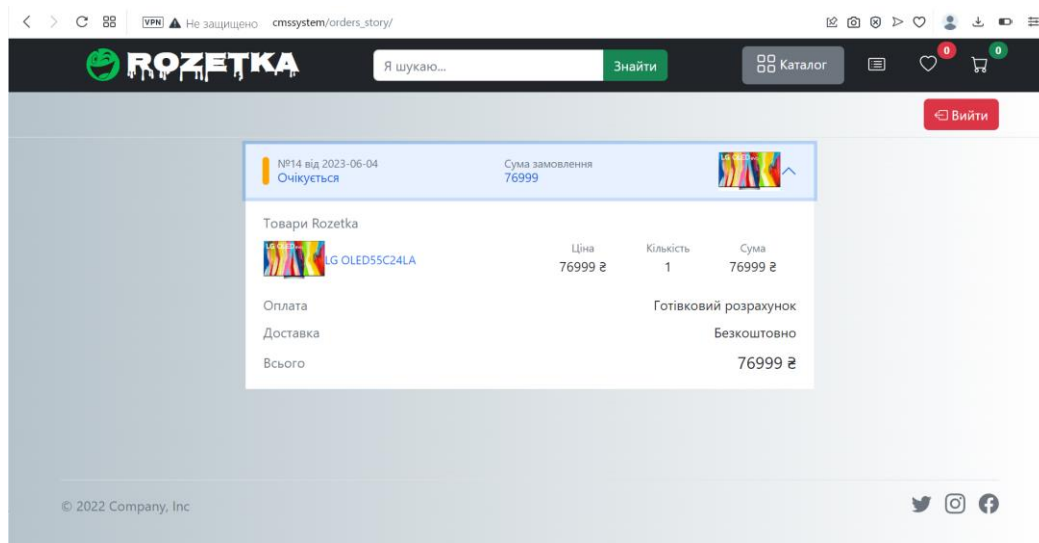


Рис. 3.10. Сторінка з відображенням списку замовлень (історії замовлень)

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		34

При натисканні кнопки у вигляді сердечка, у верхньому меню відкриється список товарів, які користувач відправив до списку бажань (рис. 3.11).

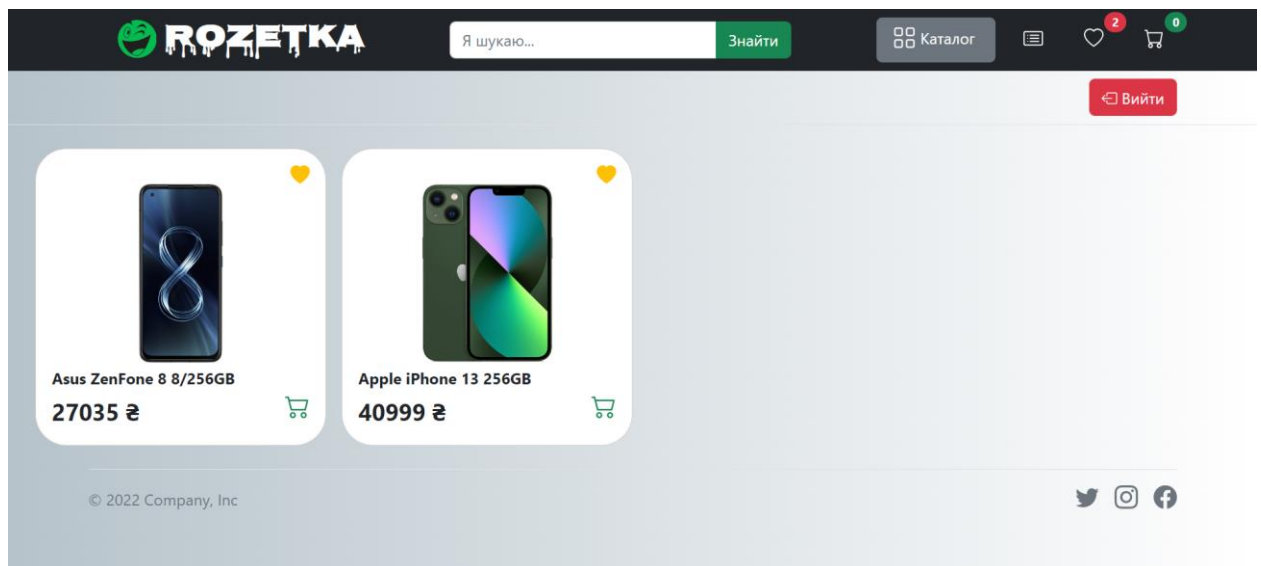


Рис. 3.11. Сторінка з відображенням товарів, які належать до списку бажань

При натисканні кнопки, у вигляді кошика, яка розміщена у верхньому навігаційному меню, відкриється кошик, де можна переглянути товари, які наявні в ньому, а при потребі редагувати потрібну кількість чи взагалі видалити (рис 3.12).

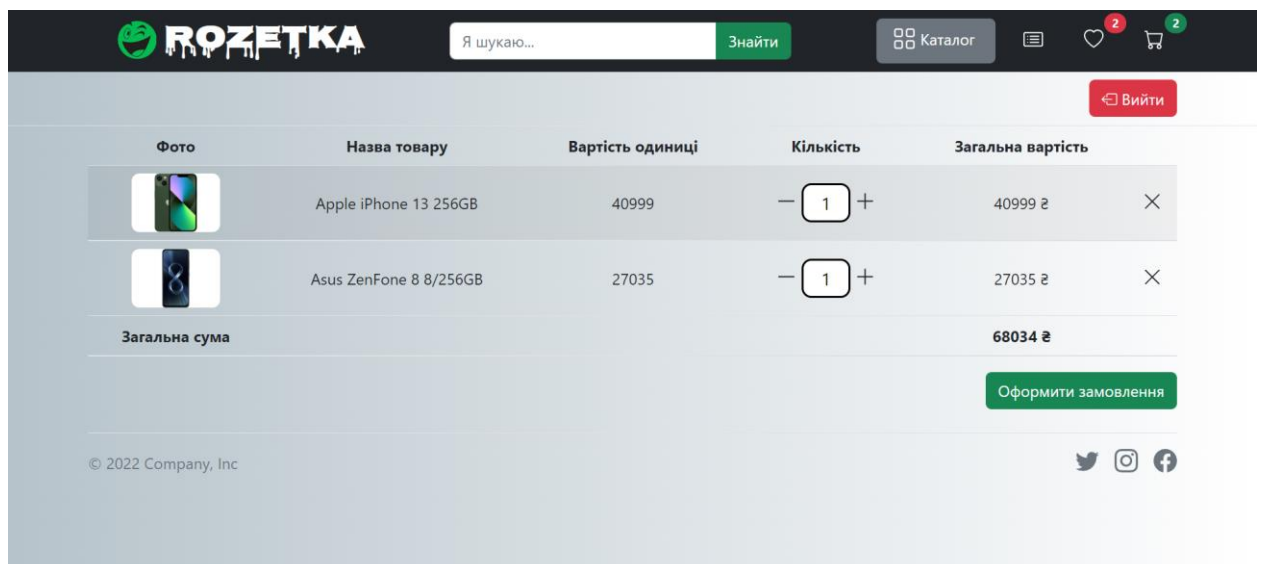


Рис. 3.12. Сторінка з відображенням товарів, які знаходяться в кошику

При натисканні кнопки “оформити замовлення” відбувається перехід на сторінку (рис. 3.13)

Оформлення замовлення

① Ваші контактні дані

Ім'я: Дмитро

Прізвище: Шевченко

Мобільний телефон: [input field]

Електронна пошта: volkov.aliksandr03@gmail.com

Замовлення на суму: 68034 ₴

② Товари продавця Rozetka

Товар	Ціна	Кількість	Сума
Apple iPhone 13 256GB	40999 ₴	1	40999 ₴
Asus ZenFone 8 8/256GB	27035 ₴	1	27035 ₴

③ Доставка

Оберіть ваше місто: [dropdown]

Оберіть пункт видачі Rozetka: [dropdown]

③ Оплата

☒ Готівковий розрахунок (при отриманні товару)

☐ Безготівковий розрахунок (при отриманні товару)

Всього

2 товари на суму: 68034 ₴

Вартість доставки: 0 ₴

До сплати: 68034 ₴

Замовлення підтверджую

Отримання замовлення від 5 000 ₴ тільки за паспортом (Закон від 06.12.2019 № 361-IX)

Підтверджуючи замовлення, я приймаю умови:

- положення про збирання та захист персональних даних
- користувальницька угода

Рис. 3.13. Сторінка призначена для оформлення замовлення

Також в головному меню користувача розміщений пункт “Налаштування”, при натисканні на який, можна перейти на сторінку налаштувань (рис 3.14)

Налаштування

Особисті дані

Логін

Пароль

Дії з акаунтом

© 2022 Company, Inc

Рис. 3.14. Сторінка призначена для налаштування профілю

Розгорнувши пункт налаштувань “Особисті дані”, можна редагувати ім’я, прізвище та по-батькові. Пункт “Логін” призначений для редагування логіну (електронної пошти) облікового запису. Пункт “Пароль” призначений для зміни паролю і останній пункт “Дії з акаунтом” призначений для видалення поточного акаунта даного магазину.

Якщо авторизований користувач є адміністратором, то перед ним з’явиться наступне вікно головного меню (рис 3.15)

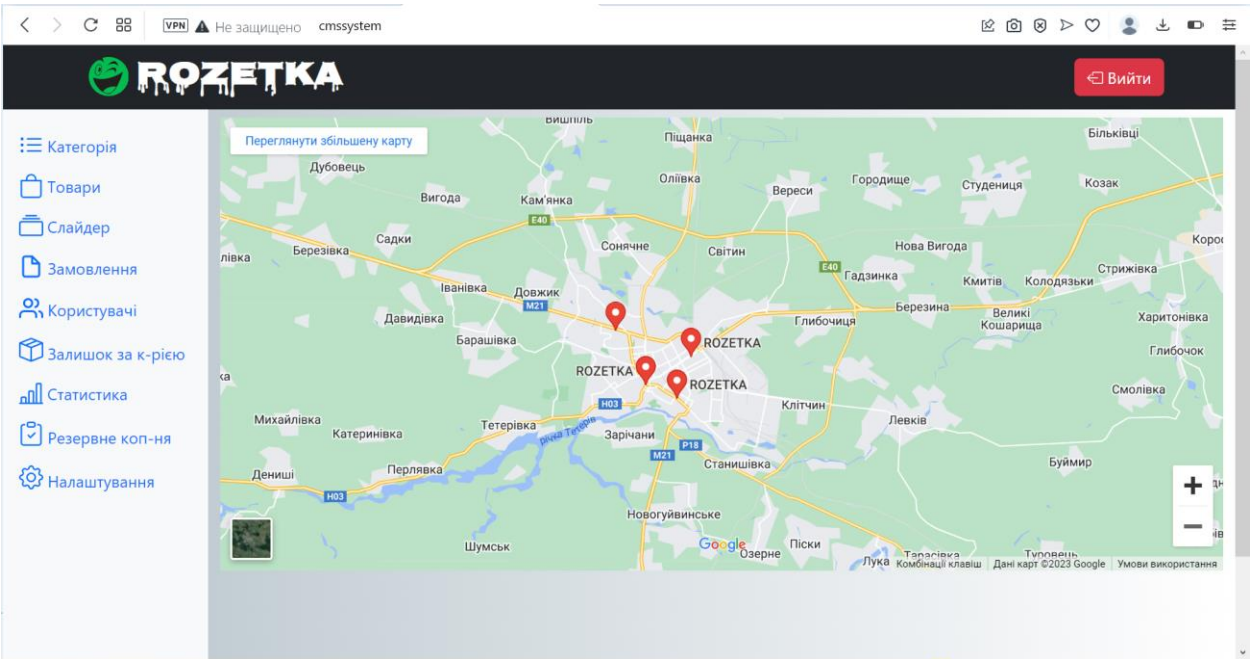


Рис. 3.15. Головне меню адміністратора

На даній сторінці розміщено карта пунктів видачі “Rozetka” з логістичних мотивів та меню дій. Натиснувши пункт меню “Категорія” користувач має можливість додати, змінити або видалити категорію (рис. 3.16), (рис. 3.17), (рис. 3.18) відповідно:

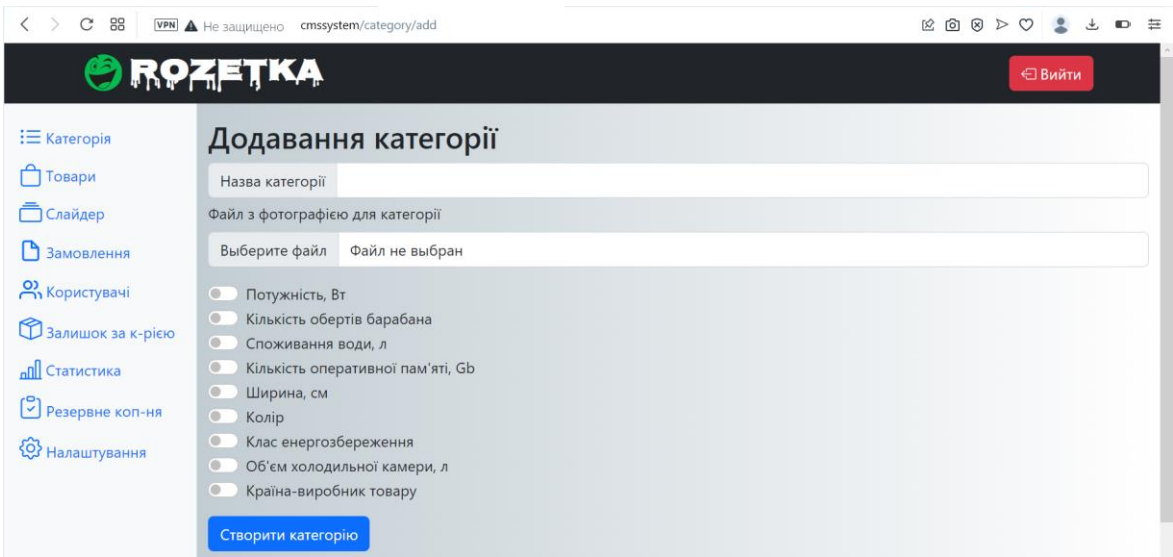


Рис. 3.16. Меню додавання категорії товару

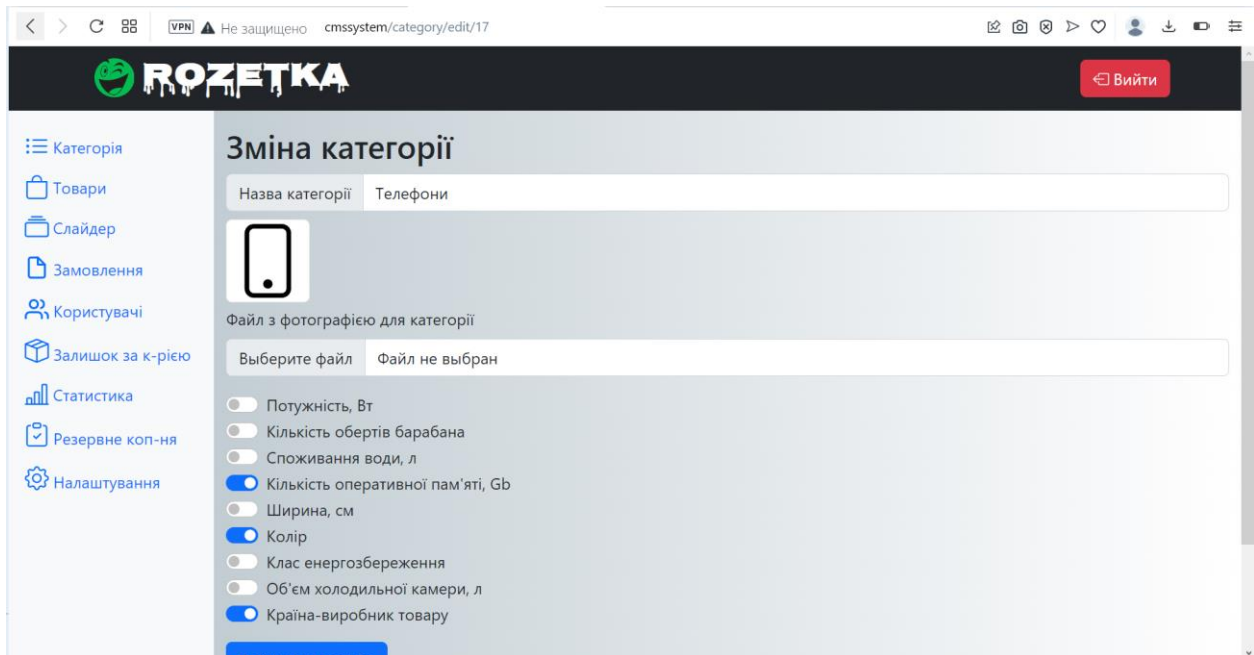


Рис. 3.17. Приклад зміни категорію товару

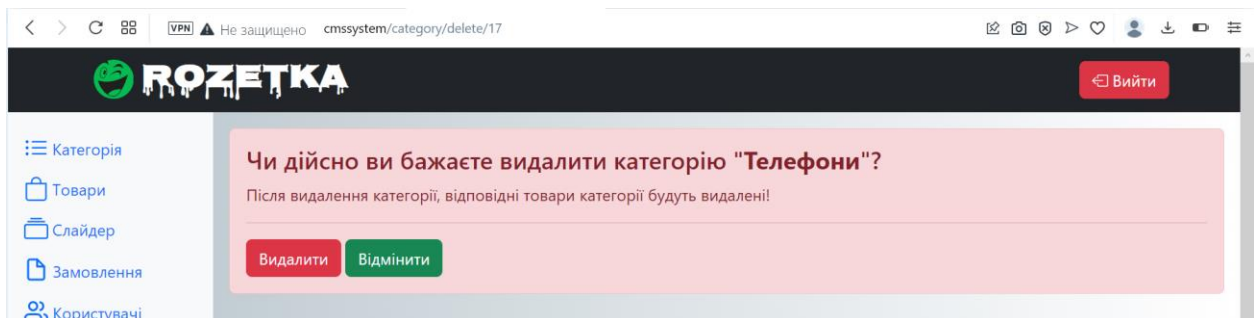


Рис. 3.18. Приклад видалення категорії товару

В меню, вибравши пункт “Товари”, можна змінити, додати або видалити товар в категорії (рис. 3.19), (рис. 3.20), (рис. 3.21) відповідно:

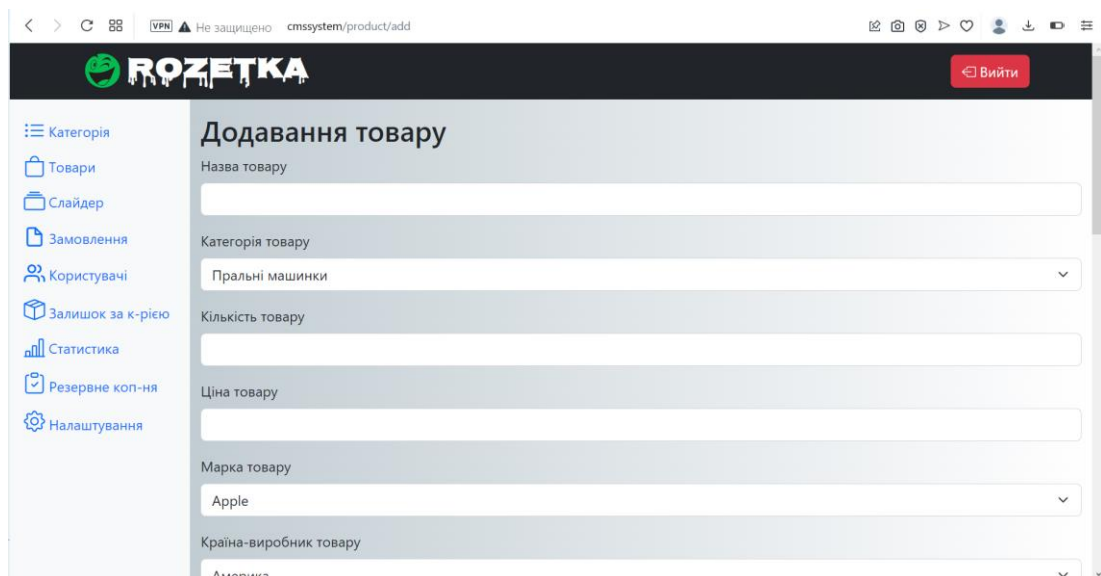


Рис. 3.19. Вікно для додавання товару в категорію

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				38
Змн.	Арк.	№ докум.	Підпис	Дата		

Скріншот веб-інтерфейсу сайту ROZETKA. Вигляд сторінки "Зміна товару".

Ліве меню (сайдбар):

- Категорія
- Товари
- Слайдер
- Замовлення
- Користувачі
- Залишок за к-рією
- Статистика
- Резервне коп-ня
- Налаштування

Головний контент:

Зміна товару

Назва товару:

Категорія товару:

Кількість товару:

Ціна товару:

Марка товару:

Країна-виробник товару:

Верхня панель: Вийти

Рис. 3.20. Приклад зміни товару в категорії

Скріншот веб-інтерфейсу сайту ROZETKA. Вигляд сторінки "Видалення товару".

Ліве меню (сайдбар):

- Категорія
- Товари
- Слайдер
- Замовлення
- Користувачі
- Залишок за к-рією
- Статистика
- Резервне коп-ня
- Налаштування

Головний контент:

Чи дійсно ви бажаєте видалити товар "LG OLED55C24LA"?

Після видалення товару, відповідні фото товару будуть видалені!

Видалити Відмінити

Верхня панель: Вийти

Рис. 3.21. Приклад видалення товару в категорії

В пункті меню “Слайдер” можна додати або видалити фото і посилання при кліку на них (рис. 3.22), (рис. 3.23) відповідно:

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		39

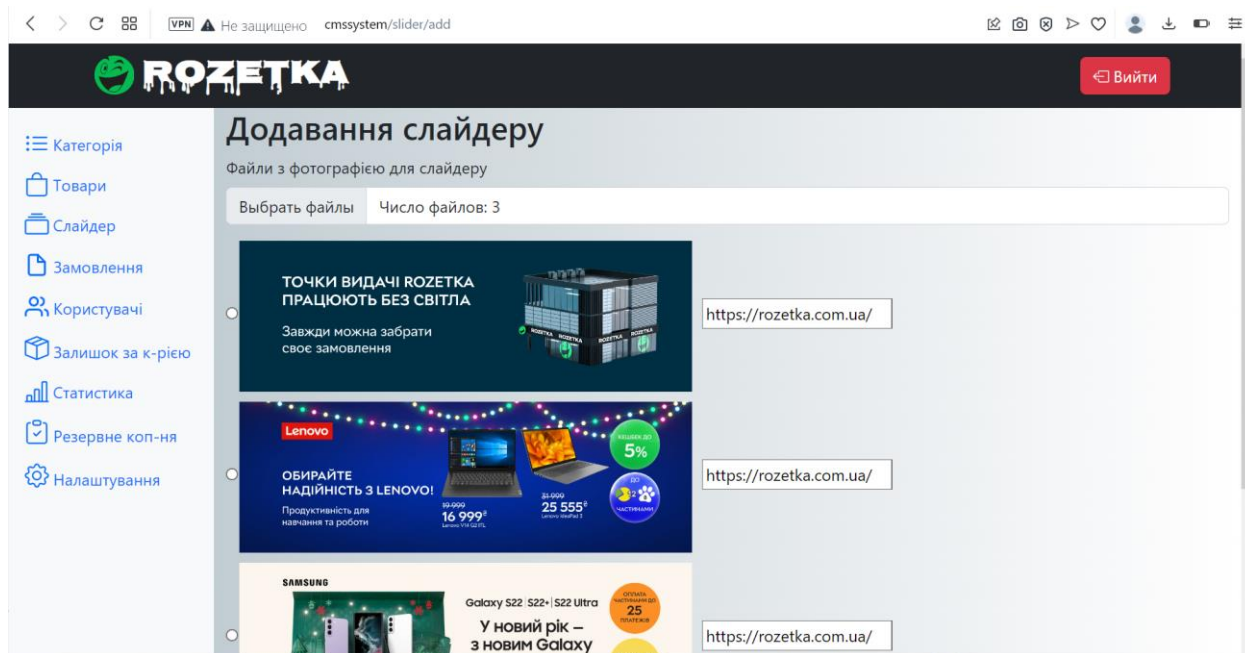


Рис. 3.22. Приклад додавання наповнення слайдеру

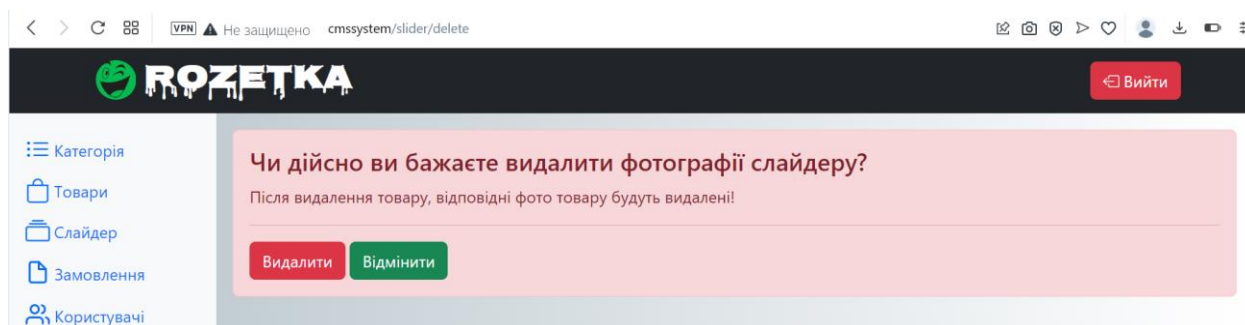


Рис. 3.23. Приклад видалення наповнення слайдеру

У пункті меню “Замовлення” можна переглянути замовлення, які надійшли до магазину та інформацію про замовника та місце доставки (рис. 3.24):

#	Ім'я замовника	Прізвище замовника	Електронна пошта замовника	Номер телефону замовника	Тип оплати	Адреса доставки	Назва товару	Кількість товару	Сума замовлення	
1	Олександр	Максимович	volkov.aliksandr0204@gmail.com	+380987778821	Готівковий розрахунок	Київська 106	LG OLED55C24LA	1	76999 ₪	✓
2	Іван	Іванович	ivan@gmail.com	+380987778821	Готівковий розрахунок	Київська 106	LG OLED55C24LA	1	76999 ₪	✓

Рис. 3.24. Демонстрація таблиці з усіма замовленнями

При натисненні зеленої кнопки в останній колонці таблиці можна змінити статус замовлення (з очікується на виконано).

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		
						40

В пункті меню “Користувачі” можна переглянути всіх зареєстрованих користувачів та інформацію про них, а за потреби змінити їм тип доступу, вибравши його в випадаючому списку (передостання колонка таблиці) та натиснувши зелену кнопку в останній колонці таблиці застосувати зміни (рис. 3.25):

#	Ім'я	Прізвище	По-батькові	Логін	Тип доступу	
1	Олександр	Максимович	Волков	volkov.aliksandr0203@gmail.com	Адміністратор	✓
2	Олександр	Максимович	Волков	volkov.aliksandr0204@gmail.com	Клієнт	✓
3	Іван	Іванович	Іванов	ivan@gmail.com	Клієнт	✓

Рис. 3.25. Таблиця з усіма користувачами зареєстрованих в системі

В пункті меню “Залишок за категорією” можна переглянути кількість товарів за категорією, які залишилися на складі та загальну суму цих товарів у відповідності до категорії (рис. 3.26):

#	Категорія товару	Загальна кількість (од.)	Загальна сума (грн.)
1	Пральні машинки	115	7750185
2	Телефони	97	2279403
3	Телевізори	71	5633614

Рис. 3.26. Таблиця з списком товарів, які залишилися на складі

В пункті меню “Статистика” можна переглянути статистичні дані (обіг, обсяг проданого товару за категорією, типи оплат, загальний рейтинг за категорією) відповідно (рис. 3.27), (рис. 3.28), (рис. 3.29), (рис. 3.30):



Рис. 3.27. Діаграма обігу продукції за 2023 рік

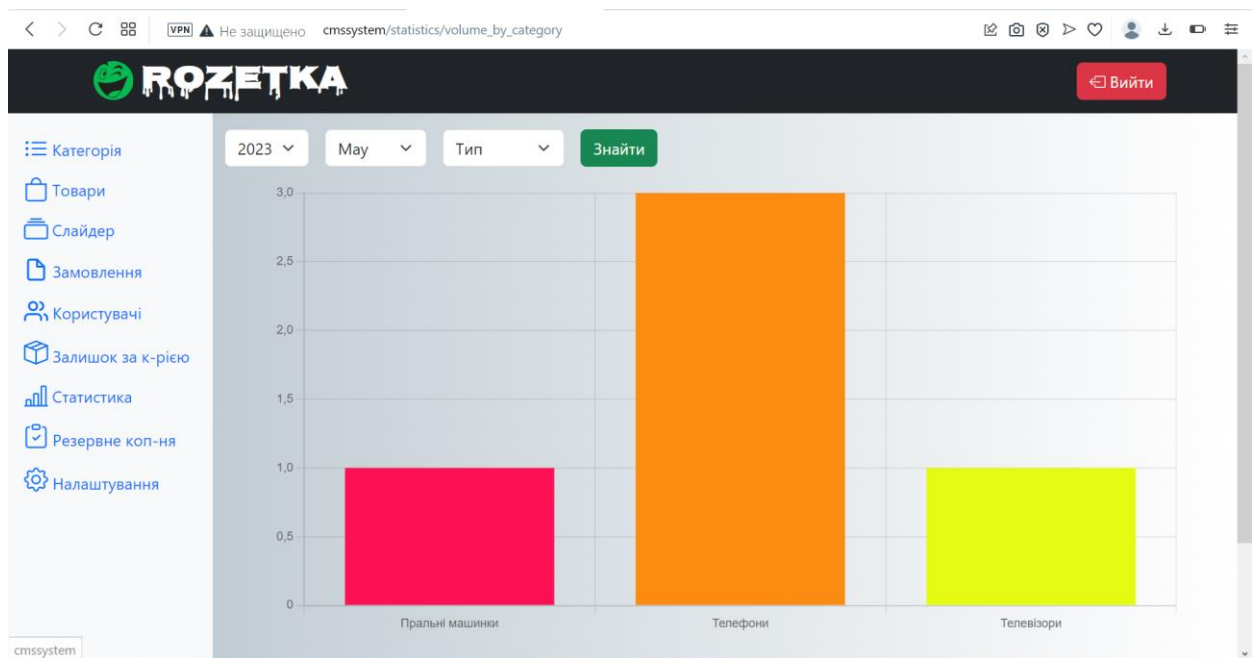


Рис. 3.28. Діаграма обігу продукції за травень 2023 року

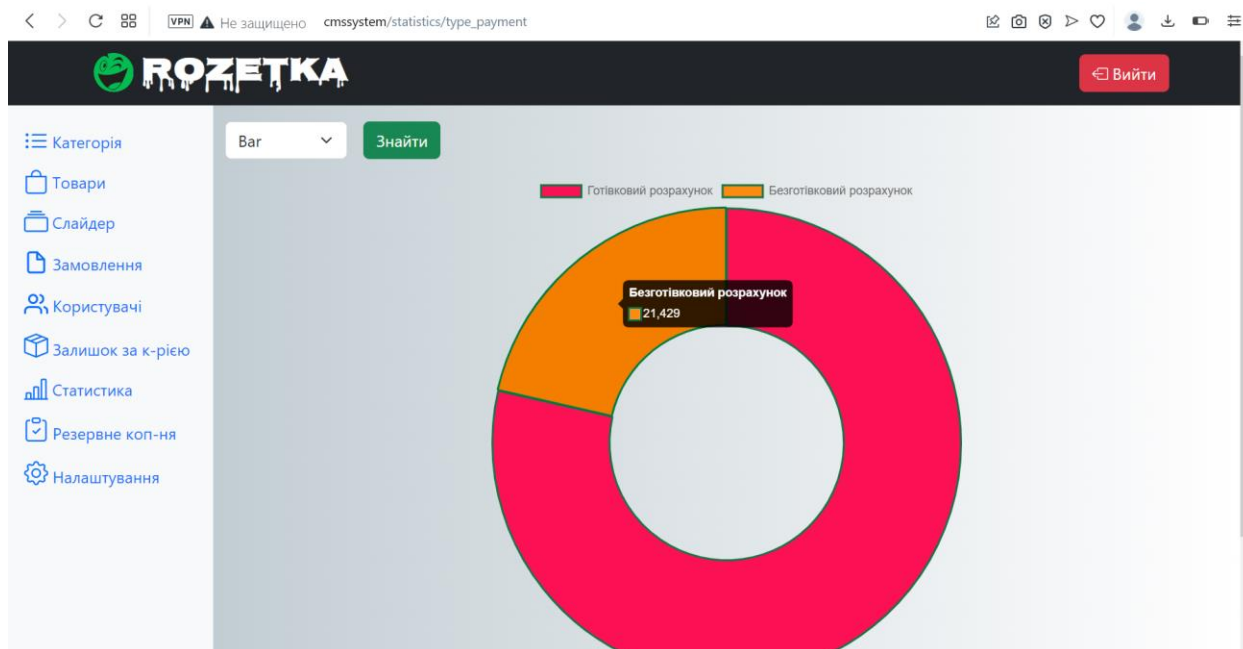


Рис. 3.29. Діаграма відображення статистики за типом оплати у відсотковому співвідношенні



Рис. 3.30. Діаграма відображення загального рейтингу для категорії

В пункті “Резервне копіювання” можна здійснити резервну копію бази даних “SHOPCMS” за вказаним шляхом на диску (рис. 3.31):

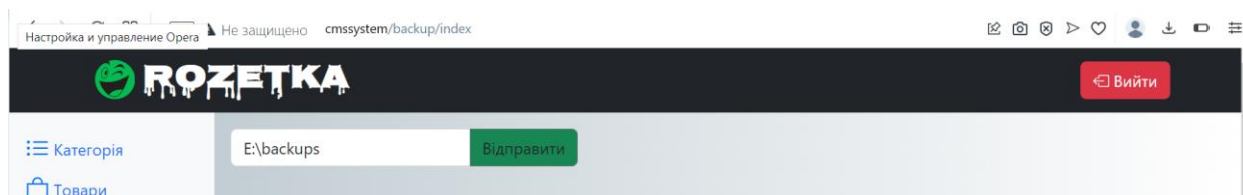


Рис. 3.31. Сторінка сайту для здійснення резервної копії БД

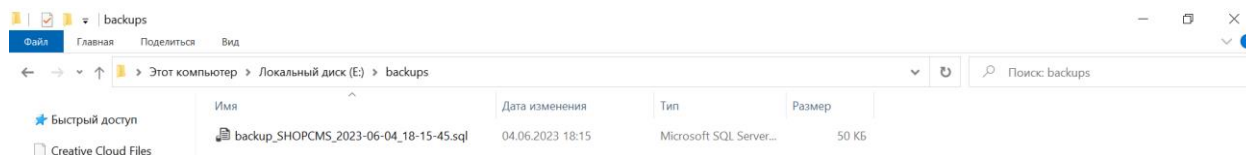


Рис. 3.32. Результат виконання резервного копіювання БД

В пункті “Налаштування” можна здійснити налаштування профілю аналогічно описаного вище (налаштування профіля користувача, який не є адміністратор).

3.2. Реалізація операцій обробки даних в БД

Для підключення до БД створив та використав конструктор “__construct” класу “DB”:

```
public function __construct($server, $login, $password, $database)
{
    $this->pdo = new \PDO("mysql:host={$server};dbname={$database};", $login,
    $password);
}
```

Для здійснення вставки інформації в БД, було створено метод “insert” в класі “DB”:

```
public function insert($tableName, $fieldsList)
{
    $fieldsArray = array_keys($fieldsList);
    $fieldsListString = implode(' ', $fieldsArray);
    $paramsArray = [];
    foreach ($fieldsList as $key => $value) {
        $paramsArray [] = ':' . $key;
    }
    $valuesListString = implode(' ', $paramsArray);
    $res = $this->pdo->prepare("INSERT INTO {$tableName} ({$fieldsListString})
VALUES ({$valuesListString})");
    $res->execute($fieldsList);
}
```

Для здійснення читання інформації з бази даних, було створено метод “select” в класі “DB”:

```
public function select($tableName, $fieldsList = "*", $conditionsArray =
null, array $conditionLikeArray = null, $groupByArray=null, $havingArray =
null, $orderByArray = null, $limit = null, $offset = null, $joinCondition =
null)
{
    $fieldsStr = "*";
    if (is_string($fieldsList)) {
        $fieldsStr = $fieldsList;
    }
    if (is_array($fieldsList)) {
        $fieldsStr = implode(' ', $fieldsList);
    }
    $sql = "SELECT {$fieldsStr} FROM {$tableName}";
    if (!empty($joinCondition)) {
        $sql.=" {$joinCondition}";
    }
}
```

```

    }
    list($key, $value, $sql) = $this->extracted($conditionsArray, $sql);

    if (empty($conditionsArray) && !empty($conditionLikeArray)) {
        list($key, $value, $sql) = $this->extractedLike($conditionLikeArray,
$mysql);
    }

    if(!empty($groupByArray)){
        if (is_array($groupByArray)&&count($groupByArray) > 0) {
            $groupByParts = [];
            foreach ($groupByArray as $key => $value) {
                $groupByParts [] = "{$value}";
            }
            $sql .= ' GROUP BY ' . implode(' ', $groupByParts);
        }
        if (is_string($groupByArray)) {
            $sql .= ' GROUP BY ' . $groupByArray;
        }
    }

    if(isset($havingArray)){
        list($key, $value, $sql) = $this->extractedHaving($havingArray,
$mysql);
    }

    if (is_array($orderByArray)) {
        $orderByParts = [];
        foreach ($orderByArray as $key => $value) {
            $orderByParts [] = "{$value}";
        }
        $sql .= ' ORDER BY ' . implode(' ', $orderByParts);
    }

    if (!empty($limit)) {
        if (!empty($offset)) {
            $sql .= " LIMIT {$offset}, {$limit}";
        } else {
            $sql .= " LIMIT {$limit}";
        }
    }
    $res = $this->pdo->prepare($sql);

    if (is_array($conditionsArray) && count($conditionsArray) > 0)
        $res->execute(array_values($conditionsArray));
    else if (isset($conditionLikeArray) && count($conditionLikeArray) > 0)
        $res->execute(array_values($conditionLikeArray));
    else
        $res->execute();
    return $res->fetchAll(PDO::FETCH_ASSOC);
}

```

В даній функції використовуються допоміжні функції: “extracted”, “extractedLike”, “extractedHaving”.

Функція “extracted” призначена для формування умови запиту в залежності від того, чи запит має багато операцій порівняння, чи лише одну. Лістинг даної функції наступний:

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИДКОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		45

```

public function extracted($conditionsArray, string $sql): array
{
    if (is_array($conditionsArray) && count($conditionsArray) > 0) {
        $whereParts = [];
        foreach ($conditionsArray as $key => $value) {
            $whereParts[] = "{$key} = ?";
        }
        $whereStr = implode(' AND ', $whereParts);
        $sql .= ' WHERE ' . $whereStr;
    }

    if (is_string($conditionsArray))
        $sql .= ' WHERE ' . $conditionsArray;
    return array($key, $value, $sql);
}

```

Функція “extractedLike” призначена для формування умови запиту, який складається з оператора “LIKE” і в залежності від того, чи запит має багато рядків та шаблонів для порівняння, чи лише один шаблон і рядок. Лістинг даної функції наступний:

```

public function extractedLike($conditionsArray, string $sql): array
{
    if (is_array($conditionsArray) && count($conditionsArray) > 0) {
        $whereParts = [];
        foreach ($conditionsArray as $key => $value) {
            $whereParts[] = "{$key} LIKE ?";
        }
        $whereStr = implode(' AND ', $whereParts);
        $sql .= ' WHERE ' . $whereStr;
    }

    if (is_string($conditionsArray))
        $sql .= ' WHERE ' . $conditionsArray;
    return array($key, $value, $sql);
}

```

Функція “extractedHaving” призначена для формування умови запиту для фільтрування груп в залежності від того, чи запит має багато операцій порівняння, чи лише одну. Лістинг даної функції наступний:

```

public function extractedHaving($havingArray, string $sql): array
{
    if (is_array($havingArray) && count($havingArray) > 0) {
        $whereParts = [];
        foreach ($havingArray as $key => $value) {
            $whereParts[] = "{$key} {$value}";
        }
        $whereStr = implode(' AND ', $whereParts);
        $sql .= ' HAVING ' . $whereStr;
    }

    if (is_string($havingArray))
        $sql .= ' HAVING ' . $havingArray;
    return array($key, $value, $sql);
}

```

Для здійснення оновлення інформації в БД, було створено метод “update” в класі “DB”:

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

public function update($table, $fieldsList, $conditionArray)
{
    $sql = "UPDATE {$table} SET ";
    $setParts = [];
    $paramsArr = [];
    foreach ($fieldsList as $key => $value) {
        $setParts[] = "{$key} = ?";
        $paramsArr[] = $value;
    }
    $sql .= implode(' ', $setParts);
    if (is_array($conditionArray) && count($conditionArray) > 0) {
        $whereParts = [];
        foreach ($conditionArray as $key => $value) {
            $whereParts[] = "{$key} = ?";
            $paramsArr[] = $value;
        }
        $whereStr = implode(' AND ', $whereParts);
        $sql .= ' WHERE ' . $whereStr;
    }
    if (is_string($conditionArray))
        $sql .= ' WHERE ' . $conditionArray;
    $res = $this->pdo->prepare($sql);
    $res->execute($paramsArr);
}

```

Для здійснення видалення інформації з БД, було створено метод “delete” в класі “DB”:

```

public function delete($table, $conditionArray = null)
{
    $sql = "DELETE FROM {$table}";
    list($key, $value, $sql) = $this->extracted($conditionArray, $sql);
    $res = $this->pdo->prepare($sql);
    if (is_array($conditionArray) && count($conditionArray) > 0)
        $res->execute(array_values($conditionArray));
    else
        $res->execute();
}

```

Для здійснення резервної копії БД, було створено метод “createBackup” в класі “DB”:

```

public function createBackup($backupFilePath, $username, $password, $database): bool
{
    $backupFileName = 'backup_' . "{$database}_" . date('Y-m-d_H-i-s') . '.sql';
    $command = "mysqldump --user={$username} --password={$password} --databases {$database} > {$backupFilePath}/{$backupFileName}";
    system($command, $result);
    if ($result === 0) {
        return true;
    } else {
        return false;
    }
}

```

3.3. Реалізація звітності системи

Звітність передбачає:

- Перегляд обігу продукції за конкретний рік:

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				47
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.33. Обіг продукції

Контроллер, який збирає дані для її виведення:

```
public function indexAction()
{
    if (!User::isUserAdmin()) {
        return $this->error(403);
    }
    $years = Order::getAllOrders([
        'status'=>1
    ], [
        "YEAR(date) as year"
    ], null, ['year']);

    if (Core::getInstance()->requestMethod == 'POST') {
        if ($_POST['diagramType'] == 'default') {
            $_POST['diagramType'] = 'bar';
        }
        $date = Order::getAllOrders([
            'YEAR(date)' => $_POST['year']
        ], [
            'MONTH(date) AS month', 'SUM( order_.count*product.price) AS sum'
        ], null, ['month'], null, null, null, null, "JOIN product on
_order_.id_product = product.id");

        $chartData = [];
        if (!empty($date)) {
            foreach ($date as $key=>$value) {
                $chartData[$value['month']] = $value['sum'];
            }
        }

        $type = $_POST['diagramType'];
        $year = $_POST['year'];
        return $this->render(null, [
            'years'=>$years,
            'date'=>$chartData,
            'type'=>$type,
            'year'=>$year
        ]);
    }
}
```



```
return $this->render(null, [
    'years'=>$years,
]);
}
```

- Кількість проданого товару за конкретний рік та місяць по категоріям товару:



Рис. 3.34. Кількість проданого товару за травень 2023 року

Контроллер, який збирає дані для її виведення:

```
public function volume_by_categoryAction()
{
    if (!User::isUserAdmin()) {
        return $this->error(403);
    }
    $monthsLabels =
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'];

    if (Core::getInstance()->requestMethod == 'GET' && isset($_GET['year'])) {
        $months = Order::getAllOrders([
            'status'=>1,
            'YEAR(date)'=>$_GET['year']
        ], [
            "MONTH(date) as month"
        ]);
        $resultArray = [];
        foreach ($months as $key=>$value) {
            $resultArray[$value['month']] = $monthsLabels[$value['month']-1];
        }
        exit(json_encode($resultArray));
    }

    $years = Order::getAllOrders([
        'status'=>1
    ], [
        "YEAR(date) as year"
    ], null, ['year']);
}
```

```

if (Core::getInstance()->requestMethod == 'POST') {
    if ($_POST['diagramType']=='default') {
        $_POST['diagramType'] = 'bar';
    }
    $date = Order::getAllOrders([
        'YEAR(date)'=>$_POST['year'],
        'MONTH(date)'=>$_POST['month']
    ], [
        'id_category',
        'sum( order .count) AS count'
    ], null, ['id_category'], 'count>0', null, null, null, "JOIN product on
_order_.id_product = product.id");

    $months = [];
    $monthArr = Order::getAllOrders([
        'status'=>1,
        'YEAR(date)'=>$_POST['year']
    ], [
        "MONTH(date) as month"
    ]);
    foreach ($monthArr as $key=>$value) {
        $months[$value['month']] = $monthsLabels[$value['month']-1];
    }

    $chartData = [];
    $categoriesName = [];
    if (!empty($date)) {
        foreach ($date as $key=>$value) {
            $chartData[$value['id_category']] = $value['count'];
            $categoriesName[$value['id_category']] = Category::getCategoryById($value['id_category'])['name'];
        }
    }

    $type = $_POST['diagramType'];
    $year = $_POST['year'];
    $month = $_POST['month'];

    return $this->render(null, [
        'years'=>$years,
        'data'=>$chartData,
        'type'=>$type,
        'year'=>$year,
        'categoriesName'=>$categoriesName,
        'months'=>$months,
        'month'=>$month
    ]);
}

return $this->render(null, [
    'years'=>$years,
]);
}

```

- Статистика по типу оплати:

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИДКОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		50

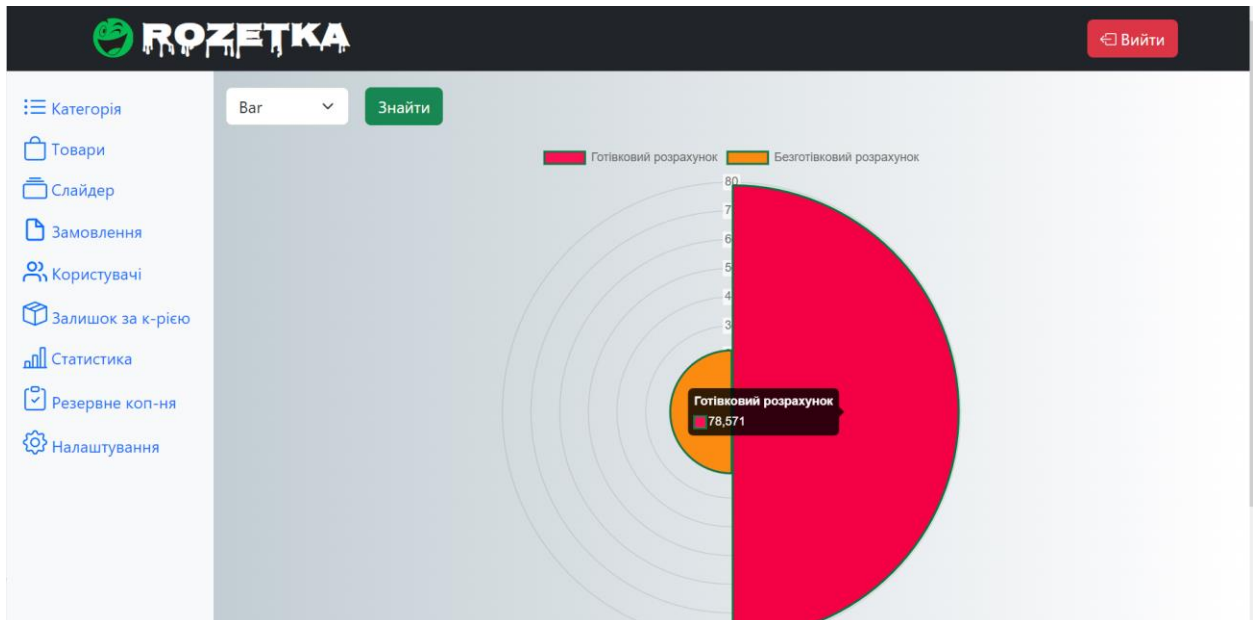


Рис. 3.35. Статистика по типу оплати

Контроллер, який збирає дані для її виведення:

```
public function type_paymentAction() {
    if (!User::isUserAdmin()) {
        return $this->error(403);
    }
    $orderWithCashCount = count(Order::getAllOrders([
        'typePayment_id'=>1
    ]));
    $orderWithNoCashCount = count(Order::getAllOrders([
        'typePayment_id'=>2
    ]));
    $sum = $orderWithCashCount+$orderWithNoCashCount;
    $orderWithCashCountPercent = $orderWithCashCount*100/$sum;
    $orderWithNoCashCountPercent = 100 - $orderWithCashCountPercent;
    $orderWithCashCount = ["Готівковий розрахунок"=>$orderWithCashCountPercent];
    $orderWithNoCashCount = ["Безготівковий розрахунок"=>$orderWithNoCashCountPercent];

    if (Core::getInstance()->requestMethod == 'POST') {
        $type = $_POST['diagramType'];
        return $this->render(null, [
            'cashPayment'=>$orderWithCashCount,
            'noCashPayment'=>$orderWithNoCashCount,
            'type'=>$type
        ]);
    }

    return $this->render(null, [
        'cashPayment'=>$orderWithCashCount,
        'noCashPayment'=>$orderWithNoCashCount,
        'type'=>'bar'
    ]);
}
```

- Рейтинг за категорією товарів:

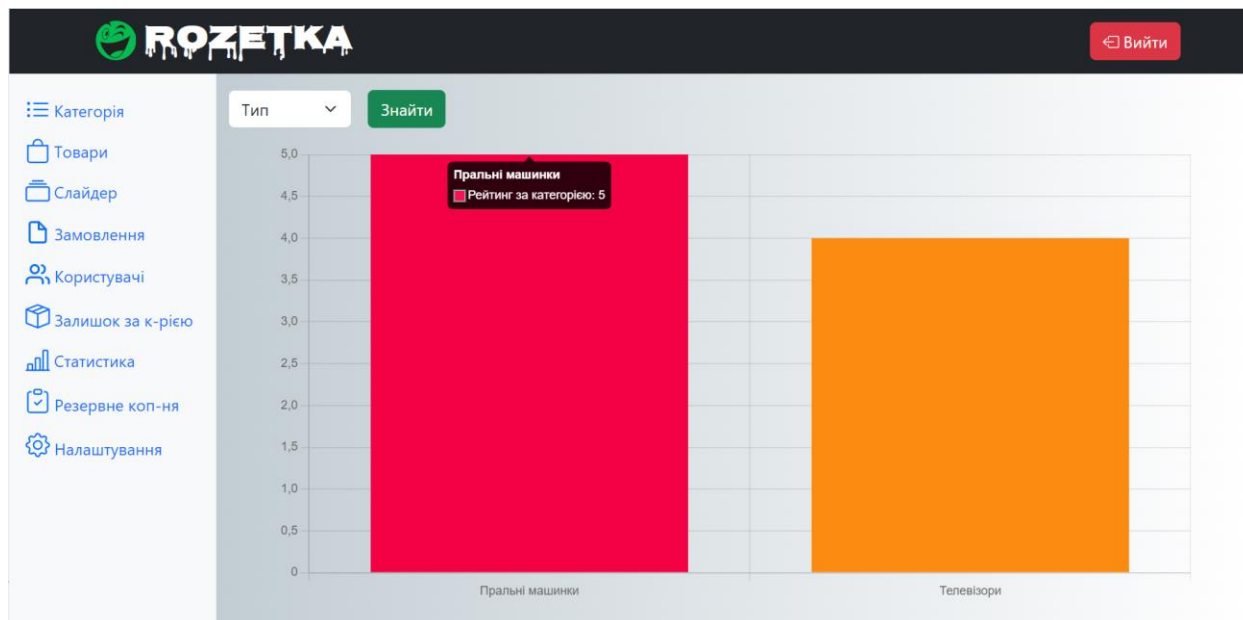


Рис. 3.36. Статистика рейтингу за категорією

Контроллер, який збирає дані для її виведення:

```
public function reiting_by_categoryAction()
{
    if (!User::isUserAdmin()) {
        return $this->error(403);
    }
    $categories = Category::getCategories(['id', 'name']);
    $categoriesArray = [];
    foreach ($categories as $key=>$value) {
        $categoriesArray[$value['id']]=$value['name'];
    }
    $values = Comment::selectComments(['id_category',
    'sum(comment_products.reiting)/count(comment_products.reiting) as
    mark'], null, null, 'product.id_category', null, null, null, null, 'JOIN product on
    product.id=comment_products.id_product');
    $categoriesMarks = [];
    foreach ($values as $key=>$value) {
        $categories-
        Marks[$categoriesArray[$value['id_category']]]=$value['mark'];
    }

    if (Core::getInstance()->requestMethod=='POST') {
        return $this->render(null, [
            'categoriesMarks'=>$categoriesMarks,
            'type'=>$_POST['diagramType']
        ]);
    }
    return $this->render(null, [
        'categoriesMarks'=>$categoriesMarks
    ]);
}
```

Висновки до третього розділу

Було спроектовано інтерфейс обробки даних магазину побутової техніки, реалізовано та описано опереції обробки даних в БД, а також реалізовано та продемонстровано реалізовану звітність в системі магазину.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИДКОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		52

РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

4.1. Розробка заходів захисту інформації в БД

Для захисту інформації в БД було створено наступний функціонал:

- Реалізовано хешування пароля за допомогою вбудованої функції “md5()”

```
public static function hashPassword($password)
{
    return md5($password);
}
```

- Реалізовано 3 типи користувачів (авторизований користувач, неавторизований користувач, адміністратор) на рівні веб-додатку:

Таблиця 4.1.

Таблиці/полі	Матриця доступу на рівні додатку		
	Авторизований користувач (не адміністратор)	Не авторизований користувач	Адміністратор
basket	5	5	0
category_filter	1	1	5
category_product	1	1	5
color	1	1	1
comment_products	1,2	1	1
country	1	1	1
destination	1	1	1
energy_consumption	1	1	1
filter	1	1	1
operation_memoty	1	1	1
power	1	1	1
product	1	1	5
product_filter_value	0	0	5
product_mark	1	1	1
product_photo	1	1	5
slider	1	1	5
spin_drum_count	1	1	1

towns	1	1	1
typePayment	1	1	1
user	5	0	5
volume_refrigigerating_chamber	1	1	1
water_consumption	1	1	1
width_cm	1	1	1
wish_list	5	0	0
order	1,2	2	1,3

Де 0 – немає доступу 1 – читання 2 – вставка 3 – редагування 4 – видалення
5 – повний доступ

- Реалізовано користувачів на рівні БД:

Таблиця 4.2.

Таблиці/ролі	Матриця доступу на рівні БД	
	root	admin
basket	5	5
category_filter	5	5
category_product	5	5
color	5	5
comment_products	5	5
country	5	5
destination	5	5
energy_consumption	5	5
filter	5	5
operation_memoty	5	5
power	5	5
product	5	5
product_filter_value	5	5
product_mark	5	5
product_photo	5	5

Таблиці/ролі	Матриця доступу на рівні БД	
	root	root
slider	5	5
spin_drum_count	5	5
towns	5	5
typePayment	5	5
user	5	5
volume_refrigigerating_chamber	5	5
water_consumption	5	5
width_cm	5	5
wish_list	5	5
order	5	5
Доступ до інших БД	5	0

Де 0 – немає доступу 1 – читання 2 – вставка 3 – редагування 4 – видалення
5 – повний доступ

- Використання в PHP параметризованих запитів для захисту інформації від SQL-ін'єкції

4.2. Налаштування параметрів роботи SQL-сервера

Для розробки, тестування та відлагодження веб-додатку використав платформу Open Server та веб-сервер Apache. Вона дозволяє нам працювати з веб-застосунком PhpMyAdmin, через який ми можемо здійснити управління БД MySQL.

Для роботи з веб-сервером вибрав наступні налаштування (рис. 4.1)

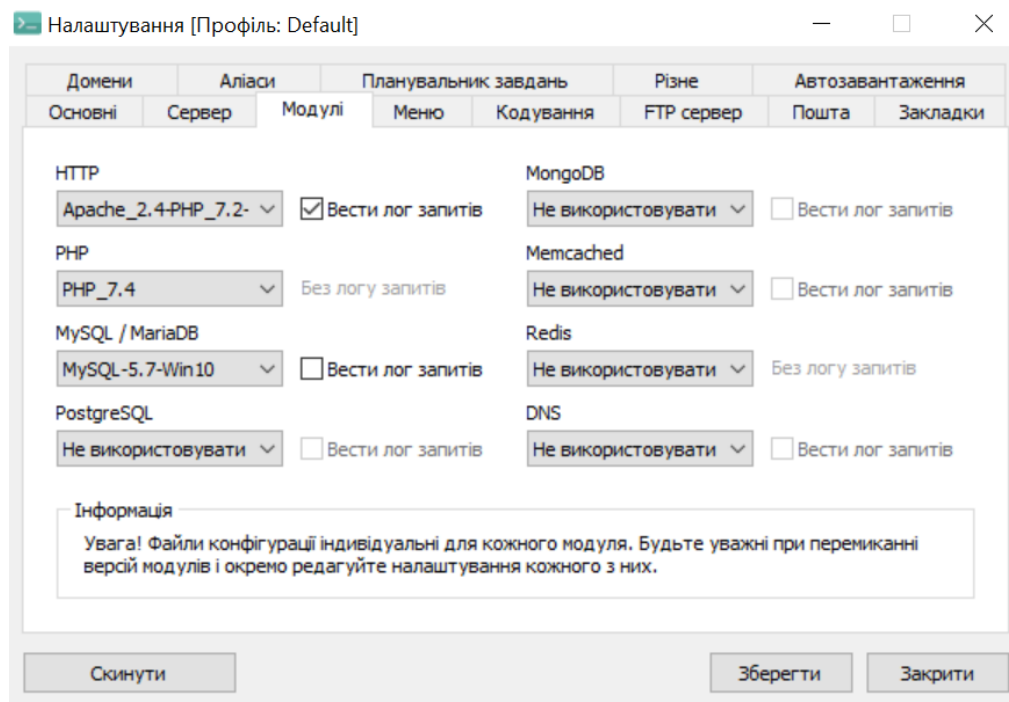


Рис. 4.1. Налаштування веб-сервера

Для запуску створеного веб-проекту необхідно запустити програму Open Server Panel, після чого в правій частині панелі завдань або у прихованих віджетах з'явиться червоний прапорець, який буде сигналізувати, що веб-сервер вимкнений (рис. 4.2).

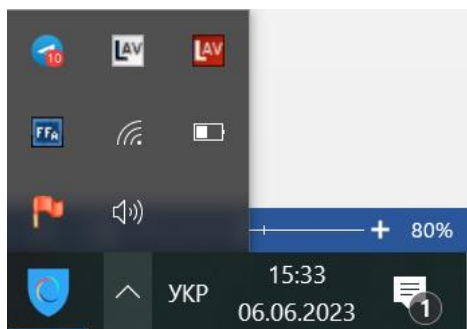


Рис. 4.2. Відображення стану веб-сервера як вимкнений

Для увімкнення веб-сервера, слід натиснути правою кнопкою на червоний прапорець та вибрати пункт "Запустити" (рис. 4.3), після чого прапорець в панелі завдань поміняє колір з червоного на зелений.

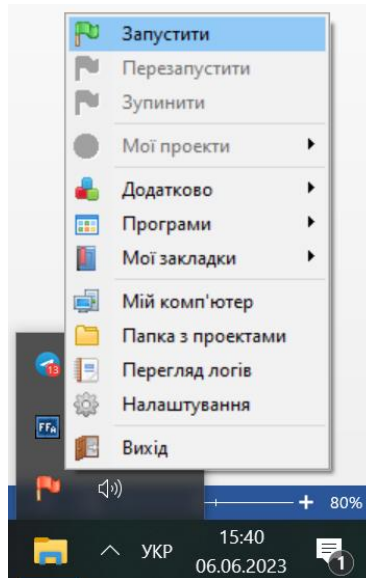


Рис. 4.3. Запуск веб-сервера

Для відкриття веб-додатку PhpMyAdmin слід натиснути на відповідний пункт (рис. 4.4).

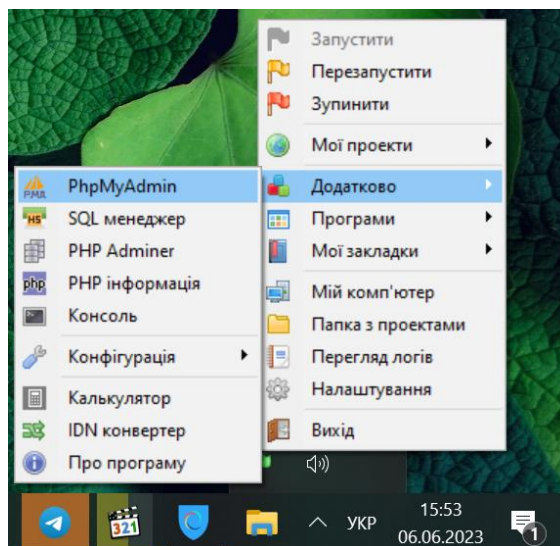


Рис. 4.4. Відкриття PhpMyAdmin

Після проходження авторизації, в PhpMyAdmin нам буде доступний профіль користувача, в якому ми можемо здійснити управління БД.

Висновки до четвертого розділу

У данному розділі було розроблено заходи захисту інформації в БД (сформовано матриці доступу, як на рівні додатку так і на рівні БД, реалізовано хешування пароля, а також захищено запити від SQL-ін'єкції). Було налаштовано та продемонстровано запуск SQL-сервера.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				57
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання курсової роботи було отримано практичні навички побудови бази даних інтернет магазину побутової техніки за допомогою БД MySQL, отримано досвід використання мов SQL, PHP, JS, HTML та таблиці стилей CSS.

В першому розділі було детально проведено аналіз поставленої задачі та обрано засоби і методи її рішення, провів порівняльний аналіз існуючих СУБД, навів їхні переваги та недоліки та на основі отриманих результатів обрав СУБД для реалізації проекту.

В другому розділі було спроектовано загальний алгоритм роботи програми, що допоможе краще зрозуміти принцип дії програмного продукту, спроектовано БД, її таблиці та встановлено необхідні зв'язки між ними, наведено діаграму створеної БД, розроблено математичну модель та алгоритми обробки даних в системі.

В третьому розділі розроблено та описано інтерфейс обробки даних, який використовується в розробленій системі, реалізовано та наведено операції обробки даних в БД для магазину побутової техніки, а також реалізовано звітність системи

В четвертому розділі розроблено засоби захисту інформації в БД для запобігання несанкціонованого доступу до інформації. Налаштовано параметри роботи SQL-сервера.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А.Зандстра, Ф.Матт. PHP Objects, Patterns, and Practice, 2013. Режим доступу до ресурсу: <https://www.apress.com/la/book/9781430260325>
2. “Работа с mysql через PDO,” На главную, <https://metanit.com/php/mysql/2.1.php>
3. Руководство по MySQL [Электронный ресурс]: [Веб-сайт]. Режим доступу до ресурсу: <https://metanit.com/sql/mysql/>
4. Руководство по JavaScript [Электронный ресурс]. Режим доступу до ресурсу: <https://metanit.com/web/javascript/>
5. Руководство по PHP [Электронный ресурс]. Режим доступу до ресурсу: <https://metanit.com/php/tutorial/>
6. Сучасний підручник з JavaScript [Електронний ресурс]. Режим доступу до ресурсу: <https://uk.javascript.info/>
7. Дэвид Скляр, Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов, 2017. Режим доступу до ресурсу: https://library-it.com/wp-content/uploads/2021/05/izuchaem_php_7_rukovodstvo.pdf
8. J. T. Mark Otto, “Bootstrap,” Bootstrap · The most popular HTML, CSS, and JS library in the world., <https://getbootstrap.com/>
9. J. Boggiano, “Packagist the PHP package repository,” Packagist, <https://packagist.org/packages/nnnick/chartjs>
10. “HTML підручник,” HTML Підручник. Початок. Уроки для початківців. [Електронний ресурс]. Режим доступу до ресурсу: <https://w3schoolsua.github.io/html/index.html#gsc.tab=0>

ДОДАТКИ

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		60

Технічне завдання

1 Загальне положення

1.1. Найменування програмного засобу

Повне найменування програмної системи: "База даних магазину побутової техніки" (надалі "веб-додаток"). Коротка назва програмної системи - "Інтернет-магазин"

1.2. Призначення розробки та область застосування

Програмна система "База даних магазину побутової техніки" призначена для збору, обробки та відображення даних про товари, користувачів магазину, реклами. В програмі передбачено використання механізму пошуку товарів за категорією, за буквосполученням в назві товару, а також за допомогою фільтрів, які можуть реалізовуватися для певної категорії товарів.

Система може бути впроваджена у роботу магазину побутової техніки з кількістю співробітників до 20 осіб та базою товарів до 100000 записів.

1.3. Мета

Програмна система "База даних магазину побутової техніки" дозволить швидко додавати, оновлювати або видаляти інформацію про товари, та забезпечить як адміністратора так і покупця своєчасною та повною інформацією про актуальний стан обробки замовлень магазину, а також надасть користувачам можливість вибору побутової техніки різних типів, в залежності від приналежності їх до певної категорії. В доповнення до вище сказаного, адміністратор сайту має можливість змінювати статус зареєстрованої особи, надавати користувачу права адміністратора.

1.4. Найменування розробника та замовника

Розробник даного продукту - студент групи ІПЗ-21-5 Волков Олександр Максимович (надалі "розробник").

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		61

Замовник програмного продукту – кафедра інженерії програмного забезпечення Державного університету «Житомирська політехніка» в межах виконання курсової з дисципліни «Бази даних» (надалі замовник)

2. Підстава для розробки

2.1. Документ на підставі якого ведеться розробка

Розробка ведеться на підставі навчального плану за напрямом 121 «Інженерія програмного забезпечення» наказу про закріплення тем курсових робіт за студентами.

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик

3.1.1. Загальні вимоги

Програмна система має забезпечувати:

- можливість дистанційної роботи з робочих станцій локальної та глобальної мережі підприємства;
- інтерфейс користувача, що не залежить від операційної системи;
- постійний доступ користувача до БД;
- оптимальне збереження даних (за обсягом та структурою);
- аутентифікацію користувачів та захист інформації від несанкціонованого доступу;
- надійне збереження даних та можливість відновлення даних у випадку непередбачуваних збоїв системи;
- створення архівів даних, що не використовуються протягом визначеного терміну;
- можливість модернізації системи через зміну функціональних потреб користувача або модернізацію обладнання;

3.1.2. Склад виконуваних функцій

Розробити базу даних магазину побутової техніки, що підтримує виконання наступних транзакцій:

1. Реєстрацію користувачів;
2. Пошук товарів за категоріями.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		62

3. Збереження, оновлення та видалення категорій та товарів.
4. Збереження, оновлення та видалення інформації про товар та категорії.
5. Можливість зміни налаштування профілю (зміна паролю, електронної пошти, власних даних, а також змога видалити профіль при необхідності).
6. Пошук та відбір товарів по мультикритеріями: (країна виробника, колір та інші, в залежності від категорії товару).
7. “Живий пошук товарів” по буквосполученню слова.
8. Вивід статистичної інформації.
9. Виконання резервних копій БД

3.1.3. Організація вхідних і вихідних даних

Вхідними даними є інформація про товари (назва, ціна, кількість, повний опис товару, короткий опис товару, категорія товару і відображення товару користувачу), категорії товарів, користувачів магазину, рекламні оголошення та про процес пов'язаний з замовленням товарів.

Організація вхідних і вихідних даних повинна відповідати інформаційній структурі виконуваних операцій, вхідним та вихідним паперовим документами.

Введення оперативних даних повинно виконуватися з використанням діалогових екранних форм, побудованих на основі візуальних компонентів. Введення даних виконується на основі затверджених форм документів: анкета, заява, інформаційна довідка та в режимі online оператором зі слів користувача.

3.1.4. Часові характеристики і розмір пам'яті, необхідної для роботи програми.

Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с.

Час виконання команд меню не більше 1 с.

Відображення масивів даних за запитом не більше 3 хвилин.

Доступність БД – 90% цілодобово.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		63

Операції з'єднання з БД не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи програми не менше 1Гб. Дисковий простір, необхідний для збереження програми і файлів даних не більше 300 Мбат для робочої станції та 20 ГБайт..

Інсталяційний пакет програми, що містить у складі БД не повинні перевищувати 100 Мбайт.

3.2. Вимоги до надійності

3.2.1. Вимоги до надійності функціонування

Програма повинна нормально функціонувати при безперебійній роботі ПК. Доступність БД 90% при одночасному доступі 30 користувачів.

При апаратних збоях, відновлення нормальної роботи програми повинне виконуватися після:

- а) апаратні збої сервера - перезавантаження ОС сервера, запуск сервера БД (запуску резервного сервера, використання технологій RAID для збереження даних);
- б) апаратні збої робочої станції – перезавантаження ОС ПК, запуск виконуваного файлу програми.
- в) БД повинна повертатись в найближчий несуперечний стан – передбачити точки відновлення.

При збоях програмного забезпечення:

- а) система повинна забезпечувати можливість відновлення даних та фіксацію і «відкат» транзакцій.
- б) в системі має бути реалізована коректна обробка виняткових ситуацій.

3.2.2. Контроль вхідної та вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора:

- Перевірка відповідності даних доменам інформаційних атрибутів;
- Використанням механізму авто заповнення та вибору за переліком для зв'язаних даних;
- Захист від помилок оператора (залипань, випадкових символів тощо).

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування.

Для вихідної інформації передбачити:

- відсутність блокування даних через багато користувальницький доступ;
- постійне оновлення даних у відображених на екрані звітних формах.
- повідомлення про використання режиму монопольного доступу супервайзерами БД

3.2.3. Час відновлення після відмови

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезавантаження операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.3. Умови експлуатації і збереження

Програма використовується у багато користувальницькому середовищі. Регламенті операції проводити за графіком:

- створення резервних копій даних: 1 раз на тиждень;
- збереження резервної копії журналу транзакцій – 1 раз на тиждень;
- архівування даних – 1 раз на місяць;
- обслуговуючі функції (прибирання “сміття”, де фрагментація тощо) - щоденне;
- оновлення системи ідентифікації та аутентифікації користувачів – 1 раз на місяць та за потребою

Збереження архівних копій - на визначених носіях. Збереження резервних копій – на визначеному диску без перезапису. Знищення копій за регламентом.

3.4. Вимоги до інформаційної і програмної сумісності

3.4.1. Вимоги до інформаційних структур на вході і виході

Формат відображення даних має дозволяти імпорт даних в додатки MS Office для редагування та перегляду.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		65

3.4.2. Вимоги до методів рішення і мов програмування

Вибір методів рішення здійснюється розробникам без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

3.4.3. Вимоги до інформаційних структур на вході і виході

Вимоги до програмного забезпечення сервера:

ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником).

Вимоги до програмного забезпечення робочої станції:

ОС - родина Windows не нижче Windows XP.

Драйвери периферійних пристроїв - введення/виводу визначаються та встановлюються при встановленні ОС в залежності від конфігурації робочої станції.

Пакет додатків MS Office.

Робоча станція є клієнтом мереж Microsoft.

3.5. Вимоги до складу і параметрів технічних засобів

Вимоги до складу технічних засобів:

1. Сервер:

- сервер у базовій конфігурації із підтримкою RAID;
- з'ємний запам'ятовуючий пристрій для архівування даних;
- принтер для друку;
- засоби для під'єднання до локальної мережі.

2. Робоча станція:

- ПК на базі процесорів Intel, AMD у стандартній комплектації;
- периферійні пристрої друку;
- засоби для під'єднання до локальної мережі. засоби для під'єднання до локальної мережі.
- Pentium III з тактовою частотою 750 МГц;
- оперативна пам'ять - 128Мбайт;

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТЯ				
Змн.	Арк.	№ докум.	Підпис	Дата		66

- об'єм дискової пам'яті - 20Гбайт.

4. Вимоги до програмної документації

Програмна документація повинна включати наступні відомості:

1. "Інструкція по інсталяції (встановленню) програми", складається з опису інсталяційного пакету, переліку етапів інсталяції та їх послідовності, опису встановлених програмних компонентів та режимів їх роботи після інсталяції. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться у 4 розділі.
2. Керівництво користувача" складається з опису послідовності завантаження програми, основних режимів роботи, опису основних екранних форм, переліку виняткових ситуацій та реакції користувача на них, порядку виконання завдань в системі. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 3 розділі курсової роботи.
3. "Керівництво адміністратора баз даних", складається з опису складу таблиць бази даних та правил доступів до них, опису та послідовності робіт по обслуговуванню бази даних (архівування, резервне копіювання, з вказівкою періодичності виконання та засобів, що для цього використовуються), порядку налаштування серверу та клієнтських додатків. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 2,3 та 4 розділах курсової роботи згідно плану.

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		67

Програмний код

Core.php

```

<? php

namespace core;

use controllers\SiteController;
use models\User;

/**
 * Головний клас ядра системи
 * SingleTone
 */
class Core
{
    private static ? Core $instance = null;
    public array $application;
    public DB $db;
    public $requestMethod;
    public array $pageParams;

    private function __construct()
    {
        global $pageParams;
        $this->application = [];
        $this->pageParams = $pageParams;
    }

    /**
     * Повертає екземпляр ядра системи
     * @return Core
     */
    public static function getInstance()
    {
        if (empty(self::$instance))
        {
            self::$instance = new Core();
        }
        return self::$instance;
    }

    /**
     * Ініціалізація системи
     */
    public function initialize()
    {
        session_start();
        $this->db = new DB(DATABASE_HOST, DATABASE_LOGIN, DATABASE_PASSWORD, DATA-
BASE_BASENAME);
        $this->requestMethod = $_SERVER['REQUEST_METHOD'];
    }

    /**
     * Виконує основний процес роботи CMS системи
     */
    public function run()
    {
        session_start();
        $path = $_GET['path'];
        $pathParts = explode('/', $path);

```

		Олександр ВОЛКОВ			ДУ «Житомирська політехніка».23.121.07.000 - ПЗ	Арк.
		Олексій ЧИЖМОТРА				
Змн.	Арк.	№ докум.	Підпис	Дата		68

```

$moduleName = strtolower(array_shift($pathParts));
$actionName = strtolower(array_shift($pathParts));
$statuscode = 200;
if (empty($moduleName))
    $moduleName = "site";
if (empty($actionName))
    if (User::isUserAdmin())
    {
        $actionName = "admin";
    }
    else
    {
        $actionName = "index";
    }

$this->application['moduleName'] = $moduleName;
$this->application['actionName'] = $actionName;
$controllerName = '\controllers\\'.ucfirst($moduleName). 'Controller';
$actionName = $actionName. 'Action';
if (class_exists($controllerName))
{
    $controller = new $controllerName();
    if (method_exists($controller, $actionName))
    {
        $actionResult = $controller->$actionName($pathParts);
        if ($actionResult instanceof Error)
            $statusCode = $actionResult->code;
        $this->pageParams['content'] = $actionResult;
    }
    else
    {
        $statusCode = 404;
    }
}
else
{
    $statusCode = 404;
}
$statuscodeType = intval($statusCode / 100);
if ($statusCodeType == 4 || $statusCodeType == 5) {
    $siteController = new SiteController();
    $this->pageParams['content'] = $siteController->errorAction($statusCode);
}
}

/**
 * Завершення роботи системи та виведення результату
 */
public function done()
{
    $pathToMainPage = 'themes/light/mainPage.php';
    $tpl = new Template($pathToMainPage);
    $tpl->setParams($this->pageParams);
    $html = $tpl->getHTML();
    echo $html;
}
}

```

DB.php

<? php

namespace core;

/**

* Клас до виконання запитів до БД

| | | | | | | |
|------|------|------------------|--------|------|---|------|
| | | Олександр ВОЛКОВ | | | ДУ «Житомирська політехніка».23.121.07.000 - ПЗ | Арк. |
| | | Олексій ЧИЖМОТРА | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 69 |

```

*/
class DB
{
    protected $pdo;

    public function __construct($server, $login, $password, $database)
    {
        $this->pdo = new \PDO("mysql:host={$server};dbname={$database};", $login,
        $password);
    }

    /**
     * @param $conditionsArray
     * @param string $sql
     * @return array
     */
    public function extracted($conditionsArray, string $sql): array
    {
        if (is_array($conditionsArray) && count($conditionsArray) > 0) {
            $whereParts = [];
            foreach ($conditionsArray as $key => $value) {
                $whereParts[] = "{$key} = ?";
            }
            $whereStr = implode(' AND ', $whereParts);
            $sql.= ' WHERE ' . $whereStr;
        }

        if (is_string($conditionsArray))
            $sql.= ' WHERE ' . $conditionsArray;
        return array($key, $value, $sql);
    }
    public function extractedHaving($havingArray, string $sql): array
    {
        if (is_array($havingArray) && count($havingArray) > 0) {
            $whereParts = [];
            foreach ($havingArray as $key => $value) {
                $whereParts[] = "{$key} {$value}";
            }

            $whereStr = implode(' AND ', $whereParts);
            $sql.= ' HAVING ' . $whereStr;
        }
        if (is_string($havingArray))
            $sql.= ' HAVING ' . $havingArray;
        return array($key, $value, $sql);
    }
}

/**
 * Виконання запиту на отримання даних з вказаної таблиці БД
 * @param string $tableName Назва таблиці бази даних
 * @param string|array $fieldsList Список полів
 * @param array|null $conditionArray Асоціативний масив з умовою для пошуку
 * @return array|false
 */
public function select($tableName, $fieldsList = "*", $conditionsArray = null,
array $conditionLikeArray = null, $groupByArray= null, $havingArray = null, $order-
ByArray = null, $limit = null, $offset = null, $joinCondition = null)
{
    $fieldsStr = "*";
    if (is_string($fieldsList))
    {
        $fieldsStr = $fieldsList;
    }
    if (is_array($fieldsList))
    {
        $fieldsStr = implode(', ', $fieldsList);
    }

```

```

    }
    $sql = "SELECT {$fieldsStr} FROM {$tableName}";
    if (!empty($joinCondition))
    {
        $sql.= " {$joinCondition}";
    }
    list($key, $value, $sql) = $this->extracted($conditionsArray, $sql);

    if (empty($conditionsArray) && !empty($conditionLikeArray))
    {
        list($key, $value, $sql) = $this->extractedLike($conditionLikeArray, $sql);
    }

    if (!empty($groupByArray))
    {
        if (is_array($groupByArray) && count($groupByArray) > 0)
        {
            $groupByParts = [];
            foreach ($groupByArray as $key => $value) {
                $groupByParts[] = "{$value}";
            }
            $sql.= ' GROUP BY '.implode(', ', $groupByParts);
        }
        if (is_string($groupByArray))
        {
            $sql.= ' GROUP BY '. $groupByArray;
        }
    }

    if (isset($havingArray))
    {
        list($key, $value, $sql) = $this->extractedHaving($havingArray, $sql);
    }

    if (is_array($orderByArray))
    {
        $orderByParts = [];
        foreach ($orderByArray as $key => $value) {
            $orderByParts[] = "{$value}";
        }
        $sql.= ' ORDER BY '.implode(', ', $orderByParts);
    }

    if (!empty($limit))
    {
        if (!empty($offset))
        {
            $sql.= " LIMIT {$offset}, {$limit}";
        }
        else
        {
            $sql.= " LIMIT {$limit}";
        }
    }

    $res = $this->pdo->prepare($sql);

    if (is_array($conditionsArray) && count($conditionsArray) > 0)
        $res->execute(array_values($conditionsArray));
    else if (isset($conditionLikeArray) && count($conditionLikeArray) > 0)
        $res->execute(array_values($conditionLikeArray));
    else
        $res->execute();
    return $res->fetchAll(\PDO::FETCH_ASSOC);
}

public function insert($tableName, $fieldsList)

```

| | | | | | | |
|------|------|------------------|--------|------|---|------|
| | | Олександр ВОЛКОВ | | | ДУ «Житомирська політехніка».23.121.07.000 - ПЗ | Арк. |
| | | Олексій ЧИЖМОТЯ | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 71 |

```

{
    $fieldsArray = array_keys($fieldsList);

    $fieldsListString = implode(', ', $fieldsArray);

    $paramsArray = [];
    foreach ($fieldsList as $key => $value) {
        $paramsArray[] = ':' . $key;
    }

    $valuesListString = implode(', ', $paramsArray);
    $res = $this->pdo->prepare("INSERT INTO {$tableName} ($fieldsListString)
VALUES($valuesListString)");
    $res->execute($fieldsList);
}

public function delete($table, $conditionArray = null)
{
    $sql = "DELETE FROM {$table}";
    list($key, $value, $sql) = $this->extracted($conditionArray, $sql);
    $res = $this->pdo->prepare($sql);
    if (is_array($conditionArray) && count($conditionArray) > 0)
        $res->execute(array_values($conditionArray));
    else
        $res->execute();
}

public function update($table, $fieldsList, $conditionArray)
{
    $sql = "UPDATE {$table} SET ";
    $setParts = [];
    $paramsArr = [];
    foreach ($fieldsList as $key => $value) {
        $setParts[] = "{$key} = ?";
        $paramsArr[] = $value;
    }
    $sql .= implode(', ', $setParts);
    if (is_array($conditionArray) && count($conditionArray) > 0) {
        $whereParts = [];
        foreach ($conditionArray as $key => $value) {
            $whereParts[] = "{$key} = ?";
            $paramsArr[] = $value;
        }
        $whereStr = implode(' AND ', $whereParts);
        $sql .= ' WHERE ' . $whereStr;
    }
    if (is_string($conditionArray))
        $sql .= ' WHERE ' . $conditionArray;
    $res = $this->pdo->prepare($sql);
    $res->execute($paramsArr);
}

public function extractedLike($conditionsArray, string $sql): array
{
    if (is_array($conditionsArray) && count($conditionsArray) > 0) {
        $whereParts = [];
        foreach ($conditionsArray as $key => $value) {
            $whereParts[] = "{$key} LIKE ?";
        }
        $whereStr = implode(' AND ', $whereParts);
        $sql .= ' WHERE ' . $whereStr;
    }

    if (is_string($conditionsArray))
        $sql .= ' WHERE ' . $conditionsArray;
    return array($key, $value, $sql);
}

```

| | | | | | | |
|------|------|------------------|--------|------|---|------|
| | | Олександр ВОЛКОВ | | | ДУ «Житомирська політехніка».23.121.07.000 - ПЗ | Арк. |
| | | Олексій ЧИЖМОТЯ | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 72 |


```

    public function createBackup($backupFilePath, $username, $password, $database):
bool
    {
        $backupFileName = 'backup_' . "{$database}_" . date('Y-m-d_H-i-s') . '.sql';
        $command = "mysqldump --user=$username --password=$password --databases $da-
tabase > $backupFilePath/$backupFileName";
        system($command, $result);
        if ($result === 0) {
            return true;
        } else {
            return false;
        }
    }
}

```

Controller.php

```

<? php

namespace core;

/**
 * Базовий клас для всіх контролерів
 * @package core
 */
class Controller
{
    protected string $viewPath;
    protected $moduleName;
    protected $actionName;

    public function __construct()
    {
        $this->moduleName = Core::getInstance()->application['moduleName'];
        $this->actionName = Core::getInstance()->application['actionName'];
        $this->viewPath = "views/{$this->moduleName}/{$this->actionName}.php";
    }

    public function render($viewPath = null, $params = null)
    {
        if (empty($viewPath))
            $viewPath = $this->viewPath;
        $tpl = new Template($viewPath);
        if (!empty($params))
            $tpl->setParams($params);
        return $tpl->getHTML();
    }

    public function renderView($viewName)
    {
        $path = "views/{$this->moduleName}/{$viewName}.php";
        $tpl = new Template($path);
        if (!empty($params))
        {
            $tpl->setParams($params);
        }
        return $tpl->getHTML();
    }

    public static function redirect($url)
    {
        header("Location: {$url}");
        die;
    }
}

```

| | | | | | | |
|------|------|------------------|--------|------|---|------|
| | | Олександр ВОЛКОВ | | | ДУ «Житомирська політехніка».23.121.07.000 - ПЗ | Арк. |
| | | Олексій ЧИЖМОТРА | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 73 |

```

        public static function error($type, $message = null): Error
        {
            return new Error($type, $message);
        }
    }
}

```

Template.php

```

<? php

namespace core;
/**
 *Клас шаблонізатора
 * @package core
 */
class Template
{
    protected array $parameters;
    protected $path;
    public function __construct($path)
    {
        $this->parameters = [];
        $this->path = $path;
    }

    public function setParam($name, $value)
    {
        $this->parameters[$name] = $value;
    }

    public function setParams($params)
    {
        foreach ($params as $name => $value){
            $this->setParam($name, $value);
        }
    }

    public function getHTML()
    {
        ob_start();
        extract($this->parameters);
        include($this->path);
        $html = ob_get_contents();
        ob_end_clean();
        return $html;
    }
}

```

| | | | | | | |
|------|------|------------------|--------|------|---|------|
| | | Олександр ВОЛКОВ | | | ДУ «Житомирська політехніка».23.121.07.000 - ПЗ | Арк. |
| | | Олексій ЧИЖМОТЯ | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 74 |