

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ
ФУТБОЛЬНИХ ТОВАРІВ
DEVELOPMENT OF AN ONLINE FOOTBALL GOODS STORE

Виконав: студент(ка) 4 курсу, групи КНТ-219сп
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Боровик Д.І.

(прізвище та ініціали)

Керівник Степаненко О.О.

(прізвище та ініціали)

Рецензент Поляков М.О.

(прізвище та ініціали)

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

Інститут, факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2022 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Боровика Данила Ігоровича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка інтернет-магазину з продажу футбольних товарів. Development of an Online Football Goods Store

керівник проєкту (роботи) Степаненко Олександр Олексійович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 07 червня 2022 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Характеристика об'єкта проєктування. 2. Вибір і обґрунтування структури системи. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво системного адміністратора. 6. Керівництво користувача.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-6 Основна частина	Степаненко О.О., доцент		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “04” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент

(підпис) Боровик Д.І.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Степаненко О.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 104 с., 39 рис., 5 табл., 3 дод., 13 джерел.

ФУТБОЛ, ІНТЕРНЕТ-МАГАЗИН, ПРОДАЖ, БАЗА ДАНИХ, SQL – ЗАПИТИ, КЛЮЧОВЕ ПОЛЕ, ОПЕРАЦІЙНА СИСТЕМА WINDOWS, ВИТРАТИ НА РОЗРОБКУ ПРОГРАМНОГО ПРОДУКТУ, ОХОРОНА ПРАЦІ.

Об'єкт розробки - Інтернет магазин із продажу футбольних товарів.

Мета роботи - розробка програмного продукту, який реалізує функції Інтернет магазину з індивідуальним наповненням та працює в мережі Інтернет.

Електронний магазин автоматизує процес вибору й покупки футбольних товарів, дозволяє користувачеві здійснювати покупки, не виходячи з будинку, розширює клієнтську базу.

Вхідними даними для Інтернет-магазин є інформація про товари, які продаються в інтернет-магазині футбольних товарів.

Вихідними даними для Інтернет-магазин є сконфігуровані сторінки сайту з інформацією про категорії футбольних товарів, які продаються в інтернет - магазині, й замовлення, зроблені користувачем.

Інтернет магазин розроблений мовою програмування PHP із застосуванням системи керування базами даних MySQL. Електронний магазин коректно працює й відображається у всіх сучасних браузерях.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 104 pages, 39 figures, 5 tables, 3 appendixes, 13 sources.

FOOTBALL, ONLINE STORE, SALES, DATABASE, SQL - QUERY, KEY FIELD, WINDOWS OPERATING SYSTEM, COSTS OF SOFTWARE DEVELOPMENT, OCCUPATIONAL SAFETY AND HEALTH.

The object of development is an online store selling football goods.

The purpose of the work is to develop a software product that implements the functions of an online store with individual content and works on the Internet.

The e-shop automates the process of choosing and buying football products, allows the user to make purchases without leaving home, expands the customer base.

The input data for the online store is information about the products sold in the online store of football products.

The source data for the online store is the configured pages of the site with information about the categories of football products sold in the online store and orders made by the user.

The online store is developed in the PHP programming language using the MySQL database management system. The e-shop works correctly and is displayed in all modern browsers.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Характеристика об'єкта проектування	11
1.1 Поняття інтернет-магазину	11
1.2 Основні способи ведення бізнесу за допомогою системи онлайн-покупок.....	14
1.3 Основні функціональні можливості інтернет-магазину	18
1.4 Висновки за розділом	19
2 Вибір і обґрунтування структури системи	20
2.1 Вибір середовища програмування для серверної частини сайту.....	20
2.2 Вибір середовища програмування для клієнтської частини сайту.....	26
2.3. Обґрунтування вибору середовища функціонування системи	28
2.4 Бази даних	31
2.5 Функціональна схема роботи інтернет-магазину	36
2.6 Висновки за розділом	36
3 Основні рішення щодо реалізації компонентів системи.....	38
3.1 Основні відомості про розробку Інтернет-магазину	38
3.2. Прицип роботи кошика товарів у інтернет-магазині.	39
3.3 Алгоритм роботи головної сторінки інтернет-магазину.....	43
3.4 Алгоритм функціонування адміністративної частин	45
3.5 Висновки за розділом	48
4 Керівництво програміста.....	49
4.1 Призначення й умови використання Інтернет-магазину з продажу футбольних товарів.....	49
4.2 Характеристики розробленого продукту.....	49
4.3. Структура бази даних	57
4.4. Структурна схема компонентів інтернет-магазину.....	58
4.5.Висновки за розділом	58

5 Керівництво системного адміністратора	60
5.1 Розгортання інтернет-магазину	60
5.2 Створення віртуального хоста на сервері.....	60
5.3 Розгортання бази даних вебсайту.....	61
5.4 Адміністративна панель інтернет магазину	62
5.5 Висновки за розділом	64
6 Керівництво користувача	65
6.1 Призначення сайту.....	65
6.2 Вхід до вебсайту та процес авторизації й замовлення товару.....	65
6.3 Реалізація зворотнього зв'язку з клієнтами інтернет-магазину.....	67
6.4 Висновки за розділом	69
Висновки	70
Перелік джерел посилання	71
Додаток А Технічне завдання	73
Додаток Б Текст програми	76
Додаток В Слайди презентації.....	95

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

B2C – Business-to-consume;

C2B – Consumer-to-business;

HTML – HyperText Markup Language;

IE – Internet Explorer;

JS – JavaScript;

PHP – Hypertext Preprocessor;

АРМ – автоматизоване робоче місце;

БД – база даних;

ЖД – жорсткий диск;

НД – нормативна документація;

ОС – операційна система;

ПО – програмне забезпечення;

СКБД – система керування базами даних;

ЦП – центральний процесор.

ВСТУП

На сьогоднішній день за допомогою сучасної електронно-обчислювальної техніки здійснюється автоматизація розв'язання майже всіх задач, що виникають в процесі роботи підприємства чи організації. Автоматизація включає застосування керувальних пристроїв, які використовують електронні девайси і комп'ютерну техніку.

Будь-яке підприємство має за мету збільшення прибутку під час діяльності. Це досягається за рахунок підвищення продуктивності праці. Саме тому спостерігається інтенсивне впровадження у виробництво досягнень науково-технічного прогресу, застосування новітніх революційних технологій.

Ще й ми можемо бачити що на сьогоднішній день значна кількість підприємців потерпіли збитки від вимушених державних рішень про карантинні заходи у зв'язку з епідеміологічною ситуацією в країні. Коли кожний бізнес, який був розташований лише у фізичному просторі, і не був автоматизований, був вимушений взагалі припинити свою діяльність, а всі підприємці, які хотіли заробляти хоч якісь кошти, одразу ж почали процес автоматизації своїх магазинів.

Виходячи з цього можна зрозуміти, що в нинішній час автоматизація - це необхідна річ, оскільки вона полегшує працю робітників магазину. Також дає можливість підприємцю заощаджувати на масштабуванні свого бізнесу оскільки тепер йому не має різниці де саме відкривати новий філіал, оскільки тепер покупець може обрати його товар з будь-якої точки світу. Є можливість заощаджень на зарплаті працівників магазину, оскільки значну частину їх функцій виконує автоматизована система продажів і допомагає збільшити кількість продажу товару, тим самим покращуючи рентабельність.

Також при роботі інтернет-магазину ще одною важливою перевагою є популяризація магазину і збільшення охоплення активної аудиторії, що дає значно більше можливостей для підприємців котрі стають незалежними від

локації магазину і часу роботи, бо система замовлень працює автоматично і незалежить від того чи відчинений магазин, для того щоб була можливість прийняти замовлення.

В рамках дипломної кваліфікаційної роботи було прийнято рішення розробити власний інтернет-магазин з продажу футбольних товарів. Щоб не втрачати час і не роз'їжджати по місту в пошуках одного товару, всі необхідні товари будуть зібрані в одному місці, в інтернет-магазині.

Інтернет-магазин здійснюватиме швидкий продаж товарів, оптимізуватиме роботу працівників магазину, надаватиме можливість отримання потрібної інформації про товар який цікавить покупця, матиме функції фільтрації даних для більш зручного користування магазином покупцем.

Інтернет-магазин розроблений за допомогою мови програмування PHP та СКБД MySQL оскільки саме ця мова програмування та СКБД зарекомендували себе роками праці і ефективно виконує вимоги до роботи і надійності. Ці інструменти надійні і ефективні [1].

1 ХАРАКТЕРИСТИКА ОБ'ЄКТА ПРОЄКТУВАННЯ

1.1 Поняття інтернет-магазину

Інтернет-магазин – місце в інтернеті, де відбувається прямий продаж товарів споживачеві (юридичній або фізичній особі), враховуючи доставку. При цьому розміщення споживацької інформації, замовлення товару і угода відбуваються там само як і в середині мережі (на сайті інтернет-магазину).

Інтернет-покупки – це діяльність або дія, пов'язана з купівлею товарів чи послуг через Інтернет. Це означає зайти в Інтернет, зайти на вебсайт продавця, вибрати щось і організувати його доставку. Покупець оплачує товар або послугу онлайн за допомогою кредитної чи дебетової картки або при доставці.

Термін включає не лише купівлю речей в Інтернеті, а й пошук їх в Інтернеті.

Інтернет-магазини існують вже близько двадцяти п'яти років. Його популярність значно зросла.

Сьогодні ми можемо купити майже все в Інтернеті. Насправді, експерти роздрібної торгівлі кажуть, що онлайн-шопінг скоро випередить традиційні покупки в грошовому еквіваленті.

Електронна комерція - сфера економіки, що включає всі торгові та фінансові транзакції, що здійснюються за допомогою комп'ютерних мереж, а також всі бізнес-процеси, що стосуються проведення цих транзакцій.

Інтернет-магазини є частиною "E-Commerce", що означає електронна комерція.

При покупці товару через інтернет покупець робить електронний правочин через інтернет.

Електронний правочин – дія особи, спрямована на придбання, зміну або припинення цивільних прав та обов'язків, здійснену з використанням інформаційно-телекомунікаційних систем (пп. 7 ч. 1 ст. 3 Закону № 675).

Електронні угоди здійснюються у сфері електронної торгівлі, якою вважається господарська діяльність у сфері електронної купівлі-продажу, реалізації товарів дистанційним способом покупцю шляхом укладання електронних угод з використанням інформаційно-телекомунікаційних систем.

У свою чергу, електронна торгівля вписується у поняття "електронна комерція", визначене у пп. 1 год. 1 ст. 3 Закону № 675: відносини, спрямовані на отримання прибутку, що виникають при скоєнні угод щодо придбання, зміни або припинення цивільних прав та обов'язків, здійснені дистанційно з використанням інформаційно-телекомунікаційних систем, унаслідок чого в учасників таких відносин виникають права та обов'язки майнового характеру.

Покупки в Інтернеті відбуваються, коли клієнт купує через цифрову платформу. Прикладом такої платформи є Bikerringshop.com, онлайн-магазин байкерської біжутерії з Таїланду. Вам не потрібно жити в цій країні, щоб придбати один із продуктів, які продає цей магазин. Глобальне покриття є однією з головних переваг онлайн-покупок.

Головна привабливість полягає в тому, що ми можемо знайти та купити потрібні нам речі, не виходячи з дому.

На сьогоднішній день ми можемо бачити, як серед лідуючих країн на ринку електронної комерції - Індію.

Більшість інтернет-магазинів в Індії розробили передові системи розподілу, які допомагають споживачам і роздрібним продавцям, забезпечуючи своєчасну доставку та відстеження замовлень. Споживачі можуть відстежувати свої замовлення, а роздрібні продавці можуть забезпечити доставку в певний час, тим самим зменшуючи логістичні затримки.

З метою стимулювання електронної комерції в Індії з 2016 року дозволено 100% прямих іноземних інвестицій у роздрібну торгівлю. Крім того, за прогнозами, до 2026 року ринок електронної комерції Індії досягне цифри в 200 мільярдів. За прогнозами, Індійський ринок електронної

комерції перевищить американський і може стати другим за величиною ринком електронної комерції до 2034 року (IBEF, 2021). Це може бути зараховано до мереж 4G і проникнення смартфонів. Уряд Індії ініціював кампанію Digital India з метою створення трильйонної онлайн-економіки до 2025 року. Керівний комітет був створений для вивчення розвитку платформи електронної комерції, ініційованої урядом Індії. Цей комітет має на меті створити інфраструктуру для розвитку онлайн-доставки продуктів для покращення логістичної інфраструктури в селах та сільській місцевості. Незважаючи на ці зусилля уряду Індії, серед 560 мільйонів користувачів Інтернету лише 20 мільйонів є активними онлайн-покупцями [2].

Forrester Research (2020) оцінює різке зростання продажів електронної комерції на 7-8 відсотків у 2020 році, але це зростання не може перетворитися на респектабельні цифри покупок. Це викликає занепокоєння щодо покращення клієнтської бази в Інтернеті та потреби визначити й зрозуміти фактори, які допомагають утримувати споживачів. Крім того, Індія також відстає від сусідньої країни Китаю в поширенні електронної комерції, яка наприкінці 2019 року досягла 14 % онлайн-роздрібної торгівлі. За оцінками експертів ООН з торгівлі та розвитку, сектор електронної комерції пережив «драматичний» збільшити свою частку в усіх роздрібних продажах з 16 відсотків до 19 відсотків у 2020 році. У Великобританії також спостерігається сплеск онлайн-транзакцій за той же період з 15,8 до 23,3 відсотка; також Китай (з 20,7 до 24,9%), США (11-14%), Австралія (6,3-9,4%), Сінгапур (5,9-11,7%) і Канада (3,6-6,2%). З іншого боку, сектор електронної комерції Індії включає лише 3,4 відсотка загального ринку роздрібної торгівлі з 100-110 мільйонами користувачів і валовою вартістю онлайн-товарів близько 30 мільйонів. Таким чином, країни, що розвиваються, і розвинені країни відрізняються за своїми соціально-економічними та нормативними аспектами. Тому необхідною вимогою є дослідження змінних, які сприяють електронним роздрібним торговцям

залучати додаткових споживачів у контексті економіки, що розвивається, як от індійська [2].

1.2 Основні способи ведення бізнесу за допомогою системи онлайн-покупок

Існує п'ять найпоширеніших способів ведення бізнесу за допомогою системи онлайн-покупок: B2C (від бізнесу до споживача), B2B (від бізнесу до бізнесу), B2B2C (розроблений як комбінація цих двох методів), C2C (споживач до споживача), C2B (споживач до бізнесу).

Термін бізнес-споживач (B2C) відноситься до процесу продажу продуктів і послуг безпосередньо між бізнесом і споживачами, які є кінцевими користувачами його продуктів або послуг. Більшість компаній, які продають безпосередньо споживачам, можна назвати компаніями B2C.

B2C став надзвичайно популярним під час буму доткомів наприкінці 1990-х років, коли він в основному використовувався для позначення інтернет-магазинів, які продавали продукти та послуги споживачам через Інтернет.

Як бізнес-модель бізнес-споживач суттєво відрізняється від моделі бізнес-бізнес (B2B), яка відноситься до торгівлі між двома або більше підприємствами.

«Бізнес-споживач» – це процес продажу товарів і послуг підприємствам безпосередньо споживачам без середніх осіб.

B2C зазвичай відноситься до інтернет-магазинів, які продають товари та послуги споживачам через Інтернет.

Інтернет B2C став загрозою для традиційних роздрібних продавців, які отримували прибуток від додавання надбавки до ціни.

Однак такі компанії, як Amazon, eBay і Priceline, процвітали, зрештою ставши руйнівниками галузі.

B2C традиційно називали покупки в торгових центрах, харчування в ресторанах, фільми з оплатою за перегляд та рекламні ролики. Однак розвиток Інтернету створив абсолютно новий бізнес-канал B2C у формі електронної комерції або продажу товарів і послуг через Інтернет.

Хоча багато компаній B2C стали жертвами подальшого краху доткомів, оскільки інтерес інвесторів до сектору зменшився, а фінансування венчурного капіталу перестало існувати, лідери B2C, такі як Amazon і Priceline, пережили перетрясіння і відтоді досягли величезного успіху [3].

Будь-який бізнес, який покладається на продажі B2C, повинен підтримувати добрі відносини зі своїми клієнтами, щоб забезпечити їх повернення. На відміну від бізнес-бізнесу (B2B), чий маркетингові кампанії спрямовані на демонстрацію цінності продукту чи послуги, компанії, які покладаються на B2C, зазвичай викликають емоційну реакцію на свій маркетинг у своїх клієнтів.

Вітрини B2C проти Інтернет-магазинів.

Традиційно багато виробників продавали свою продукцію роздрібним продавцям із фізичним розташуванням. Роздрібні торговці отримували прибуток від націнки, яку вони додали до ціни, сплаченої виробнику. Але це змінилося, коли з'явився Інтернет. Виникли нові підприємства, які обіцяли продавати безпосередньо споживачеві, таким чином виключивши посередника – роздрібного продавця – і знизивши ціни. Під час краху буму доткомів у 1990-х роках підприємства боролися за забезпечення присутності в Інтернеті. Багато роздрібних торговців були змушені закрити свої двері та припинили свою діяльність.

Через десятиліття після революції доткомів B2C-компанії з вебприсутністю продовжують домінувати над своїми традиційними звичайними конкурентами. Такі компанії, як Amazon, Priceline та eBay, пережили ранній бум дот-комів. Вони продовжували розширювати свій ранній успіх, щоб стати руйнівниками галузі.

Інтернет В2С можна розділити на п'ять категорій: прямі продавці, онлайн-посередники, В2С на основі реклами, на основі спільноти та платні.

В2С в цифровому світі [3].

Зазвичай існує п'ять типів бізнес-моделей онлайн В2С, які більшість компаній використовує онлайн для націлювання на споживачів:

- прямі продавці. Це найпоширеніша модель, в якій люди купують товари в інтернет-магазинів. Сюди можуть входити виробники, малі підприємства або просто онлайн-версії універмагів, які продають товари різних виробників;
- онлайн-посередники. Це зв'язки або посередники, які насправді не володіють продуктами чи послугами, які об'єднують покупців і продавців. До цієї категорії належать такі сайти, як Expedia, Trivago та Etsy;
- В2С на основі реклами. Ця модель використовує безкоштовний вміст для залучення відвідувачів на вебсайт. Ці відвідувачі, у свою чергу, стикаються з цифровими або онлайн-реклами. Великі обсяги вебтрафіку використовуються для продажу реклами, яка продає товари та послуги. Одним із прикладів є медіа-сайти, такі як Huffington Post, сайт із високою відвідуваністю, який змішує рекламу зі своїм рідним вмістом;
- спільнота. Такі сайти, як Meta (раніше Facebook), які створюють онлайн-спільноти на основі спільних інтересів, допомагають маркетологам і рекламодавцям просувати свої продукти безпосередньо споживачам. Вебсайти, як правило, націлені на оголошення на основі демографічного та географічного розташування користувачів;
- платний. Сайти, орієнтовані безпосередньо на споживача, такі як Netflix, стягують плату, щоб споживачі могли отримати доступ до їх вмісту. Сайт може також пропонувати безкоштовний, але обмежений вміст, хоча більшість із них платна. The New York Times та інші великі газети часто використовують платну бізнес-модель В2С.

В2В2С – це бізнес-модель, де онлайн або електронна комерція, підприємства та портали виходять на нові ринки та клієнтів, співпрацюючи з

компаніями, орієнтованими на споживача продуктів і послуг. Підприємство, що розробляє продукт, послугу або рішення, співпрацює з іншим підприємством, щоб використовувати певну послугу, наприклад вебсайт електронної комерції, портал або блог. Ці два бізнеси об'єднують зусилля та рекламують взаємовигідні продукти, послуги та/або рішення [3].

Як приклад моделі B2B2C, компанія А платить компанії В за користувачів, потенційних клієнтів або продажі, отримані від бізнесу або вебсайту компанії В. Потім компанія А використовує канали бізнесу В для пошуку потенційних клієнтів. Бізнес В надає своїм клієнтам нові та відповідні послуги, сприяння збільшенню клієнтської бази та отримання доходу від проданих продуктів і послуг.

C2C (Consumer-To-Consumer) – це комерційна діяльність між приватними особами (споживачами). Ця бізнес-модель може бути реалізована безпосередньо, а також через третю сторону (посередника).

Спочатку модель C2C передбачає прямі продажі між фізичними особами. Сьогодні мова йде більше про онлайн-продажі між фізичними особами. У цьому випадку виникає потреба в сторонній стороні (маркетплейси та сайти з оголошеннями, такі як eBay, Craigslist або Gumtree). Таким чином, якщо ви купуєте iPad на eBay у фізичної особи, а не у підприємця, це класична модель C2C. Якщо ви продаєте бензопилу своєму сусіду, це також є прикладом C2C, однак, офлайн типу «старої школи».

C2B (Consumer-To-Business) – трохи незвичайна модель електронної комерції. Споживачі самостійно визначають ціни на товари та послуги (пропоновані підприємствами). Типовим прикладом C2B є американська компанія Priceline.com. Таким чином, компанія більше схожа на брокера, який шукає фірми, які готові продати товари чи послуги за пропозицією (замовником) цінами [3].

1.3 Основні функціональні можливості інтернет-магазину

Інтернет-магазин є вебсайтом в якому розроблені функціональні можливості для автоматизації інтернет покупок користувача. При розробці інтернет-магазину розробник повинен звертати увагу на функціональність, зручність користування, і надійність. В розроблену інтернет магазині реалізовані наступні можливості:

- відображення категорій товарів;
- зручний та швидкий пошук інформації за заданими критеріями;
- можливість придбання товару та додавання його у кошик;
- можливість вибору валюти;
- можливість вибору способу доставки товару;
- можливість вибору способу оплати товару;
- формування чеку;
- можливість сортування товарів у їх категоріях.

Це перелік ключових можливостей інтернет-магазину.

Взагалі ринок сфери інформаційних технологій дуже стрімко розвивається і на сьогоднішній день коли існують такі технології як наприклад безконтактний спосіб оплати чи ідентифікація користувача за допомогою відбитку пальця, чи ідентифікації обличчя, перехід фізичного магазину до it є найлогічнішим рішенням. Оскільки описані вище можливості дають змогу не тільки конкурувати з фізичним магазином, а й вирватися уперед. Принцип функціонування інтернет-магазину дає такі можливості:

- залишити відгуки о товарах;
- ставити через електронну форму питання та отримати відповідь;
- замовлення товару в будь який день так будь який час;
- економити власнику на оренді та зарплаті персоналу;
- робити продаж товару без фактичної наявності.

Виходячи з того що продаючи товар ми робимо послугу для людей що його купують, а в послугі головне зручність та корисність, ми можемо

зробити висновок що за допомогою перелічених переваг та можливостей, клієнту більше не потрібно спочатку шукати де йому потрібно купити той чи інший товар, потім стояти в черзі, та порівнювати ціни одного товару з іншими, чи питати відгуки у знайомих, а потрібно всього лиш перейти за інтернет-посиланням у зручний для користувача час, та ознайомитися з різноманітним асортиментом, за потреби поставити запитання в режимі онлайн чи прочитати відгуки, й врешті решт оформити замовлення обрав найзручніший спосіб оплати та доставки, знаходячись у будь-якій точці світу [4].

1.4 Висновки за розділом

Було проведено повноцінне дослідження об'єкта проєктування, сформовано набір термінів, які б характеризували основні процеси електронної комерції та принципи функціонування інтернет-магазинів. Були розглянуті основні етапи ведення бізнесу за допомогою систем онлайн-покупок.

Також були розглянуті основні функціональні можливості створення інтернет-магазину, які в свою чергу дають перевагу над звичайними магазинами. На основі вищесказаного, було вирішено створити інтернет-магазин через впершу чергу актуальність, зручність та ефективність ведення бізнесу саме у режимі онлайн та можливість масштабування існуючого бізнесу за допомогою існуючих інструментів онлайн розробки.

2 ВИБІР І ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ

2.1 Вибір середовища програмування для серверної частини сайту

Інтернет-магазин переважно було розроблено за допомогою текстового редактора.

Для розробки мого інтернет-магазину я використав мову PHP – це мова сценаріїв на стороні сервера, який використовується для розробки статичних або динамічних вебсайтів чи вебдодатків. PHP означає Hypertext Pre-processor, що раніше означало Personal Home Pages.

Таблицю порівняння мови PHP з іншими відомими мовами програмування такими як JAVA та Python наведено нижче таблиця 2.1.

Таблиця 2.1 – Порівняння основних можливостей Python, PHP та JAVA.

№ п/п	Можливості	Python	PHP	Java
1	2	3	4	5
1	Використання фреймворків	Python включає велику кількість фреймворків, таких як Django, Flask	Користувачі PHP мають доступ до таких фреймворків як Laravel, Zend, Yii	Користувачі Java мають доступ до фреймворків, як Spring, Spring boot, Hybernate
2	Популярність використання	Широко використовується в галузі штучного інтелекту, науки про дані і, отже, наукової спільноти	Вибір мови для веброзробки	Широко використовується в додатках безпеки, додатках Andoid, вебдодатках AI
3	Підключення до бази даних	Надійне підключення до бази даних	Можливий доступ до більш ніж 20 різних баз даних.	Можливий доступ майже до всіх баз даних.

Продовження таблиці 2.1

1	2	3	4	5
4	Крива здатності освоєння мови	Python найкраще підходить для довгострокових проєктів.	PHP відрізняється дуже низькою кривою навчання, легко почати. Вивчати	Більше підходить для довгострокових проєктів
5	Майбутнє мови програмування	В довгостроковій перспективі	Довготривале, понад 70% вебсайтів створено за допомогою PHP	Безперечно довготривале завдяки своїй безпеці та швидкості
6	Читабельність	Python використовує досить суворе застосування відступів.	Мова PHP не дуже типізована і дотримується класичного підходу	Нелегко читати.
7	Спеціалізація мови	Мова програмування загального призначення	Спеціалізована мова для програмування веброзробки.	Мова програмування Java – це універсальна, одночасна, строго типізована, об'єктно-орієнтована мова на основі класів.
8	Синтаксис	Дуже чіткий і стислий синтаксис кодів	Вбудована бібліотека містить широкий вибір умов імен	Синтаксис чистий, але не простий, як Python
9	Відома компанія, яка використовує цю технологію	Google, Facebook, Netflix, Spotify	Facebook, Wikipedia, slack, wordpress	Google, Uber, Instagram

Порівнюючи ці мови було обрано мовою розробки саме PHP, через легкість засвоєння, широкий вибір серед можливих використання баз даних, та орієнтованість мови на веброзробку.

Сценарій - це набір інструкцій програмування, який інтерпретується під час виконання.

Мова сценаріїв – це мова, яка інтерпретує сценарії під час виконання. Скрипти зазвичай вбудовуються в інші програмні середовища.

Метою скриптів зазвичай є підвищення продуктивності або виконання рутинних завдань для програми.

Скрипти на стороні сервера інтерпретуються на сервері, тоді як сценарії на стороні клієнта інтерпретуються клієнтською програмою [5].

PHP – це сценарій на стороні сервера, який інтерпретується на сервері, тоді як JavaScript є прикладом сценарію на стороні клієнта, який інтерпретується клієнтським браузером. І PHP, і JavaScript можна вбудувати в сторінки HTML.

Ілюстрацію функціонування мови PHP на рисунку 2.1 [5].



Рисунок 2.1 – Блок-схема функціонування мови PHP

Традиційність.

Мова PHP здається знайомою програмістам, що працюють у різних сферах. Багато мовних конструкцій запозичені з C, Perl.

Код PHP дуже схожий на той, що міститься в типових програмах на C або Pascal. Це значно зменшує початкові зусилля при вивченні PHP. PHP – це мова, яка поєднує в собі переваги Perl і C і спеціально націлена на роботу в Інтернеті, мова з універсальним (хоча з деякими застереженнями) і зрозумілим синтаксисом [6].

І хоча PHP є досить молодого мовою, вона набула такої популярності серед вебпрограмістів, що на даний момент є чи не найпопулярнішою мовою для створення вебдодатків (скриптів).

Простота.

PHP-скрипт може мати 10 000 рядків або один рядок, залежно від особливостей вашого завдання. Вам не потрібно завантажувати бібліотеки, вказувати спеціальні параметри компіляції чи щось подібне. Механізм PHP просто починає виконувати код після першої escape-послідовності (<?) і продовжує виконання, поки не зустрине парну escape-послідовність (?>). Якщо код має правильний синтаксис, він виконується точно так, як вказав програміст.

PHP - це мова, яка може бути вбудована безпосередньо в html-код сторінок, який, у свою чергу, буде правильно оброблятися інтерпретатором PHP. Ми можемо використовувати PHP для написання CGI-скриптів і позбутися багатьох незручних операторів виведення тексту. Ми можемо використовувати PHP для створення документів HTML без багатьох зовнішніх викликів сценаріїв.

Велика різноманітність функцій PHP позбавить вас проблем із написанням багаторядкових функцій, визначених користувачем на C або Pascal.

Ефективність.

Ефективність має виняткове значення під час програмування для багатокористувацьких середовищ, які включають Інтернет.

Дуже важлива перевага PHP полягає в його «двигуна». «Двигун» PHP не є ні компілятором, ні інтерпретатором. Це мовний перекладач. Такий пристрій «двигун» PHP дозволяє обробляти скрипти з досить високою швидкістю.

За деякими оцінками, більшість PHP-скриптів (особливо не дуже великих) обробляються швидше, ніж їхні аналоги, написані на Perl. Однак, що б не робили розробники PHP, скомпільовані виконувані файли працюватимуть набагато швидше – у десятки, а іноді й сотні разів. Але продуктивності PHP цілком достатньо для створення досить серйозних вебдодатків. Подробиці про пристрій і характеристики «двигуна» PHP можна знайти тут [7].

Безпека.

PHP надає розробникам і адміністраторам гнучкі та потужні інструменти безпеки, які приблизно поділяються на дві категорії: інструменти системного рівня та інструменти рівня програми:

Перший рівень. Функції безпеки на системному рівні.

PHP реалізує механізми безпеки, якими керують адміністратори; При правильному налаштуванні PHP забезпечує максимальну свободу дій і безпеку. PHP може працювати в так званому безпечному режимі, який обмежує можливість користувачів використовувати PHP кількома важливими способами. Наприклад, ви можете обмежити максимальний час виконання та використання пам'яті (неконтрольоване споживання пам'яті негативно впливає на продуктивність сервера). Подібно до `cgi-bin`, адміністратор також може встановлювати обмеження на каталоги, де користувач може переглядати та виконувати PHP-скрипти, а також використовувати PHP-скрипти для перегляду конфіденційної інформації на сервері (наприклад, файлу `passwd`).

Другий рівень. Безпека прикладного рівня.

У стандартній комплектації PHP є ряд потужних механізмів шифрування. PHP також сумісний з багатьма програмами сторонніх розробників, завдяки чому його легко інтегрувати з безпечними технологіями електронної комерції. Ще одна перевага полягає в тому, що PHP-скрипти не можна переглядати в браузері, оскільки скрипт компілюється перед відправкою користувачеві. Реалізація PHP на стороні сервера запобігає крадіжці нетривіальних скриптів користувачами, які навіть мають достатньо знань для виконання команди `View Source`.

Гнучкість.

Оскільки PHP є вбудованою мовою, вона надзвичайно гнучка по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати разом з HTML, він так само добре інтегрується в JavaScript, WML, XML та інші мови. Крім того, добре структуровані програми PHP

легко розширювати за потреби (до речі, це стосується всіх основних мов програмування).

Немає також проблем із залежністю браузера, оскільки PHP-скрипти повністю компілюються на стороні сервера, перш ніж вони надсилаються клієнту. Фактично, PHP-скрипти можуть бути доставлені на будь-який пристрій із підтримкою браузера, включаючи мобільні телефони, електронні блокноти, пейджери та портативні комп'ютери, не кажучи вже про традиційні ПК. Програмісти, що займаються допоміжними утилітами, можуть запускати PHP як командного рядка.

Оскільки PHP не містить коду, орієнтованого на конкретний вебсервер, користувачі не обмежуються певними серверами (можливо незнайомими для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold та Zeus - PHP працює на всіх перерахованих серверах. Оскільки ці сервери працюють на різних платформах, PHP загалом є платформно-незалежною мовою та існує на таких платформах, як UNIX, Solaris, FreeBSD та Windows 95/98/NT/2000/XP/2003.

Зрештою, засоби PHP дозволяють програмісту працювати із зовнішніми компонентами, такими як Enterprise Java Beans або COM-об'єкти Win32. Завдяки цим новим можливостям PHP займає гідне місце серед сучасних технологій та забезпечує масштабування проєктів до необхідних меж.

Безкоштовне поширення.

Стратегія Open Source, і розповсюдження вихідних текстів програм у масах, справило безсумнівно сприятливий вплив на багато проєктів, насамперед – Linux, хоча успіх проєкту Apache сильно підкріпив позиції прихильників Open Source. Сказане стосується і історії створення PHP, оскільки підтримка користувачів з усього світу виявилася дуже важливим фактором у розвитку проєкту PHP.

Прийняття стратегії Open Source та безкоштовне розповсюдження вихідних текстів PHP надало неоціненну послугу користувачам. До того ж

чуйна спільнота користувачів PHP є своєрідною «колективною службою підтримки», і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

Більшість серверів вебхостингу за замовчуванням підтримують PHP, на відміну від інших мов, таких як ASP, які потребують IIS. Це робить PHP економічно ефективним вибором.

PHP регулярно оновлюється, щоб бути в курсі останніх технологічних тенденцій.

Іншою перевагою, яку ви отримуєте від PHP, є те, що це мова сценаріїв на стороні сервера; це означає, що вам потрібно лише встановити його на сервері, а на клієнтських комп'ютерах, які запитують ресурси від сервера, не потрібно інсталиувати PHP; було б достатньо лише веббраузера.

PHP має вбудовану підтримку для роботи рука об руку з MySQL; це не означає, що ви не можете використовувати PHP з іншими системами керування базами даних. Ви все ще можете використовувати PHP з Postgres, Оракул, MS SQL Server, ODBC, тощо [8].

2.2 Вибір середовища програмування для клієнтської частини сайту

Вебдодаток – розподілений застосунок, в якому клієнтом виступає браузер, а сервером виступає вебсервер. Браузер може бути реалізацією так званих тонких клієнтів, логіка застосунку зосереджується на сервері, а функція браузера полягає переважно у зображенні інформації, завантаженої мережею з сервера, і передачі назад даних користувача. Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому вебзастосунки є міжплатформовими сервісами[9].

Вебдодаток отримує запит від клієнта і виконує обчислення, після цього формує вебсторінку і відправляє її клієнтові мережею з використанням протоколу HTTP. Саме вебзастосунок може бути клієнтом інших служб,

наприклад, бази даних або стороннього вебзастосунку, розташованого на іншому сервері. Яскравим прикладом вебзастосунку є система управління вмістом статей Вікіпедії: безліч її учасників можуть брати участь у створенні мережевої енциклопедії, використовуючи для цього браузери своїх операційних систем (Microsoft Windows, GNU/Linux або будь-якої іншої операційної системи) без завантаження додаткових виконуваних модулів для роботи з базою даних статей[9].

Для більшої інтерактивності й продуктивності розроблений новий підхід до розробки вебзастосунків, названий AJAX, і який нині є стандартним де-факто. При використанні Ajax сторінки вебзастосунку здатні відправляти вебзапити до сервера у фоновому режимі, і не перезавантажуються цілком, а лише довантажують необхідні дані з сервера, що значно пришвидшує роботу і робить її зручнішою[9].

Клієнтська частина сайту – це частина сайту яку бачить безпосередньо користувач сайту, тобто той код що безпосередньо виконується на стороні клієнта[9].

У ході роботи для реалізації клієнтської частини було обрано мову гіпертекстової розмітки HTML, таблицю каскадних стилів CSS, та місцями для прискорення роботи сайту було використано JavaScript[9].

HTML – стандартизована мова розмітки документів для перегляду вебсторінок у браузері. Веббраузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відображатиметься на екрані монітора.

CSS (Каскадні таблиці стилів) – це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних[9].

JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування, що додає інтерактивність вебсайту [9].

2.3 Обґрунтування вибору середовища функціонування системи

Для середовища функціонування інтернет-магазину використовуються веббраузери. На сьогоднішній день на кожному ПК чи іншому пристрої з можливістю виходу в інтернет є веббраузер. Веббраузер необхідний для перегляду сторінок вебсайтів, тому й для відображення й взаємодії з інтернет магазином теж необхідний веббраузер. Сьогодні існує великий вибір серед веббраузерів, найпопулярнішими є: Google Chrome, Opera, Mozilla Firefox, Internet Explore.

Google Chrome – це безкоштовний вебпереглядач, розроблений компанією Google на основі вебпереглядача з відкритим кодом Chromium та іншого відкритого програмного забезпечення. Його програмна архітектура була створена з чистого аркуша (однак використовуючи інше програмне забезпечення з відкритим кодом, у тому числі компоненти WebKit та Mozilla) з метою задоволення поточних потреб користувачів. Нова архітектура насамперед визнає той факт, що на сьогодні більшість вебсайтів є не просто вебсторінками, але вебпрограмами. Заявленими перевагами цієї архітектури є підвищена стабільність, швидкість, безпека, а також чистий, простий та ефективний інтерфейс користувача [10].

Opera вебпереглядач виробництва норвезької компанії Opera Software. Вперше випущений у 1994 році групою дослідників з норвезької компанії Telenor. З 1995 року продукт компанії Opera Software, створеної авторами першої версії браузера. До 2013 року Opera включала додаткові можливості для роботи з інтернетом (пошта і низка інших служб), але після переходу з власного браузерного рушія на відкритий рушій Blink додаткові функції були виокремлені з браузера в супутні проєкти.[10]

Mozilla Firefox – вільний безкоштовний вебоглядач з відкритим кодом, заснований на ядрі Gecko (Проект Mozilla).

Після випуску 9 листопада 2004 року версії 1.0 одержав схвальні відгуки від багатьох видань, включно із «Форбс» і «The Wall Street Journal». З

більш ніж 25 мільйонами завантажень з головного сайту в перші 100 днів після випуску версії 1.0, Firefox став однією із найуживаніших програм із відкритим кодом. 19 листопада 2005 року (за 344 дні після випуску версії 1.0) Firefox було завантажено 100-мільйонний раз. Станом на квітень 2015 року Firefox має частку 18,46% поміж десктопними браузерами і посідає третю позицію після Google Chrome та Internet Explorer. В Україні Firefox займає 2 місце після Google Chrome за популярністю з часткою 18,48% станом на 2015 рік. 30 червня 2009 кількість завантажень Firefox різних версій перетнула межу в 1 мільярд [10].

Існують версії для Windows, Mac OS X, Linux, OS/2, FreeBSD, Solaris й інших операційних систем.

Microsoft Internet Explorer – графічний вебглядач (браузер), який розробляє корпорація Microsoft.

Найперша версія побачила світ 16 серпня 1995 року. Вона була заснована на браузері Mosaic, права на який корпорація Microsoft придбала у компанії Spyglass. Починаючи з версії 3.0, Microsoft поширювала Internet Explorer у складі операційної системи Windows, що дозволило йому швидко збільшити ринкову частку і навіть витіснити провідний колись вебглядач Netscape Navigator [10].

Нижче, в таблиці 2.2. представлена інформація про підтримку операційних систем браузерами.

Таблиця 2.2 – Підтримка операційних систем браузерами

Internet Explorer	Mozilla FireFox	Opera	Chrome
Працює під Windows, але існують версії для Mac і Unix	Windows 95/98/NT/2000/XP; Macintosh Classic, Mac OS X; Linux UNIX, BeOS, OS/2	Windows 95/98/NT/2000/XP; Mac Classic, Mac OS X; Linux QNX, EPOC,	Windows 95/98/NT/2000/XP; Mac OS X; Linux

Нижче в таблиці 2.3 представлена інформація про поширені можливості браузерів.

Таблиця 2.3 – Функціональність браузерів

Можливості	Internet Explorer	Mozilla FireFox	Opera	Chrome
Вкладки	Так (з 7-ї версії)	Так	Так	Так
Менеджер Закачувань	Так	Так	Так	Так
Панель пошуку	Так	Так	Так	Так
Перевірка Орфографії	Частково	Так	Так	Так
Блокування Банерів	Так	Так	Частково	Частково
Блокування pop-up	Так	Так	Так	Так

Існують різні стандарти, технології та протоколи. В ідеалі всі браузери повинні однаково відображати Вебсторінки. Якщо взяти який-небудь простий приклад, то і Internet Explorer, і Netscape Navigator, і Opera, і інші браузери покажуть одну і ту ж інтерпретацію коду [11].

Але якщо мова йтиме про складний проєкт з багаторівневими вкладеними таблицями, фреймами, стилями, скриптами, то кінцевий результат може дуже сильно відрізнятись [11].

Також кожна компанія-розробник намагається розвивати додаткові технології, часто не сумісні з продуктами конкурентів. Навіть Java, здавалося б, універсальна мова, і та нерідко обробляється браузерами різних фірм по-своєму. Те ж відноситься і до набору скриптових мов [11].

Тому для якісного відображення вебсторінок слід враховувати, які вебстандарти, технології та протоколи підтримує браузер. Інформація представлена нижче в таблиці 2.4.

Таблиця 2.4 – Підтримка вебтехнологій і протоколів

Вебстандарти і технології	Internet Explorer	Mozilla FireFox	Opera	Chrome
CSS2	Так	Так	Так	Так
Фрейми	Так	Так	Так	Так
Java	Так	Так	Так	Плагін
JavaScript	Так	Так	Так	Так
XHTML	Немає	Так	Так	Так
RSS	Немає	Так	Так	Так

Рекомендовано використовувати саме Google Chrome, оскільки він вважається найоптимальнішим серед інших популярних браузері, може чітко і без проблем відображувати сторінки будь-якого вебсайту зокрема й представленого інтернет-магазину [11].

2.4 Базы даних

Для реалізації бази даних випускої кваліфікаційної роботи було обрано СКБД mysql версії 4.1

MySQL – популярна система керування базами даних (СКБД), яка часто використовується разом із PHP.

База даних являє собою структурований набір даних. Ці дані можуть бути будь-якими: від простого списку майбутніх покупок до списку експонатів у художній галереї або величезної кількості інформації в корпоративній мережі. Для запису, отримання та обробки даних, що зберігаються в комп'ютерній базі даних, потрібна система управління базою даних, яка є програмним забезпеченням MySQL. Оскільки комп'ютери

чудово обробляють великі обсяги даних, управління базою даних відіграє центральну роль у обчислювальній роботі. Такий контроль може бути реалізований різними способами - як у вигляді окремих утиліт, так і у вигляді коду, що входить до складу інших програм.

MySQL – це система управління реляційною базою даних. У реляційній базі даних дані зберігаються не в масі, а в окремих таблицях, що дозволяє збільшити швидкість і гнучкість. Таблиці пов'язуються між собою за допомогою зв'язків, що дає можливість об'єднати дані з кількох таблиць під час виконання запиту. SQL як частину системи MySQL можна описати як мову структурованих запитів плюс найпоширенішу стандартну мову, яка використовується для доступу до баз даних.

MySQL – програмне забезпечення з відкритим вихідним кодом. Будь-хто може використовувати та змінювати його. Таке програмне забезпечення можна отримати з Інтернету та використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його відповідно до своїх потреб. Використання програмного забезпечення MySQL регулюється GPL (GNU General Public License), <http://www.gnu.org/licenses/>, яка визначає, що можна, а що не можна робити з програмним забезпеченням у різних ситуаціях.

MySQL дуже швидкий, надійний і простий у використанні. Якщо вам потрібні ці якості, спробуйте працювати з цим сервером. MySQL також має ряд зручних функцій, розроблених у тісній співпраці з користувачами. Спочатку сервер MySQL був розроблений для управління великими базами даних, щоб забезпечити швидшу роботу, ніж існуючі на той момент аналоги. І вже кілька років цей сервер успішно використовується в промислових середовищах з високим попитом. Незважаючи на те, що MySQL постійно вдосконалюється, він уже сьогодні надає широкий спектр корисних функцій. Завдяки своїй доступності, швидкості та безпеці MySQL дуже добре підходить для доступу до баз даних через Інтернет.

Технічні можливості СКБД MySQL MySQL являє собою клієнт-серверну систему, яка містить багатопотоковий SQL-сервер, який забезпечує підтримку різних комп'ютерів баз даних, а також кілька різних клієнтських програм і бібліотек, інструменти адміністрування та широкий спектр інтерфейсів програмування (API). Ми також надаємо сервер MySQL як багатопотокову бібліотеку, яку можна підключити до спеціальної програми для меншого, швидшого та легшого в управлінні продукту. Існує також велика кількість програмного забезпечення MySQL, більшість з яких безкоштовна.

Устрій MySQL.

MySQL складається з двох частин: сервера і клієнта.

На комп'ютері постійно працює сервер MySQL. Клієнтські програми (наприклад, PHP-скрипти) надсилають SQL-запити на сервер MySQL через механізм сокетів (тобто за допомогою мережових інструментів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера бази даних. Потім сервер бази даних надсилає відповідь (результат) клієнту (скрипт).

Розмір результуючого набору даних може бути занадто великим, і для його передачі по мережі знадобиться занадто багато часу. Так, і рідко, коли це трапляється, вам потрібно отримати весь висновок запиту відразу (тобто всі записи, які задовольняють вираз запиту). Наприклад, нам може знадобитися лише підрахувати, скільки записів задовольняють певній умові, або ми можемо вибрати лише перші 10 записів із даних. Механізм сокетів має на увазі технологію клієнт-сервер, тобто в системі має бути запущена спеціальна програма – сервер MySQL, який приймає та обробляє запити від програм. Оскільки вся робота фактично виконується на одній машині, накладні витрати на мережу є незначними (налаштування та підтримка з'єднання з сервером MySQL досить дешеві).

Структура MySQL трирівнева: бази даних - таблиці - записи. Бази даних і таблиці MySQL фізично представлені файлами з розширеннями `frm`,

MYD, MYI. Логічно - таблиця - це набір записів. Запис – це сукупність полів різних типів. Ім'я бази даних MySQL унікальне в системі, таблиці в базі даних і поля в таблиці. Один сервер MySQL може підтримувати кілька баз даних, доступ до яких можна отримати за логіном та паролем. Знаючи ці логіни та паролі, можна працювати з конкретною базою даних. Наприклад, ви можете створити або видалити в ній таблицю, додати записи тощо. Як правило, ідентифікатор і пароль призначаються постачальниками хостингу, які надають підтримку MySQL для своїх користувачів.

MySQL значно покращив контроль доступу, щоб запобігти несанкціонованому доступу до системи. Цей контроль заснований на п'ятирівневій ієрархії привілеїв, яка визначає загальні правила доступу для обробки запитів користувачів:

- з'єднання з сервером здійснюється лише тоді, коли суб'єкт доступу відповідає правилам доступу, які становлять основу системи привілеїв MySQL. Ці правила можуть бути встановлені користувачами та/або хостом, щоб обмежити доступ із зовнішніх хостів, певних підмереж і певних діапазонів IP-адрес. Підключення до сервера можливе лише в одному випадку - якщо користувач введе правильний пароль;

- після встановлення з'єднання система починає стежити за кожним кроком користувача, щоб він не вийшов за межі своїх привілеїв. Користувач може бути обмежений у роботі лише з певними базами даних. MySQL постійно контролює всі запити користувачів до бази даних, таблиці або поля.

Безпека системи була значно покращена за рахунок організації одностороннього шифрування пароля користувача. Наприклад, схема кодування була значно покращена в MySQL 4.x. Останні версії MySQL підтримують протокол SSL, який дозволяє шифрувати інформацію, яка проходить через з'єднання клієнт/сервер для кращої безпеки [12].

Однією з найбільших переваг є підтримка багатьох типів даних.

Типи даних, що підтримуються:

- TINYINT: дуже маленьке ціле;

- SMALLINT: невелике ціле;
- MEDIUMINT: ціле середній розмір;
- INT чи INTEGER: ціле нормального розміру.
- BIGINT: велике ціле;
- FLOAT: знакове число з плаваючою комою одинарної точності;
- DOUBLE, DOUBLE PRECISION, REAL: знакове число з плаваючою комою подвійної точності;
- DECIMAL, NUMERIC: знакове число з плаваючою комою;
- DATE: Дата;
- DATETIME: комбінація дати та часу;
- TIMESTAMP: позначка часу;
- TIME: час;
- YEAR: рік у форматі YY або YYYY;
- CHAR: рядок фіксованого розміру, що доповнюється праворуч пробілами до максимальної довжини;
- VARCHAR: Рядок змінної довжини;
- TINYBLOB, TINYTEXT: BLOB- або TEXT-стовпець довжиною максимум $2^8 - 1$ символів;
- BLOB, TEXT: BLOB- або TEXT-стовпець довжиною максимум $2^{16} - 1$ символів;
- MEDIUMBLOB, MEDIUMTEXT: BLOB- або TEXT-стовпець довжиною максимум $2^{24} - 1$ символів;
- LONGBLOB, LONGTEXT: BLOB- або TEXT-стовпець довжиною максимум $2^{32} - 1$ символів;
- ENUM: перерахування;
- SET: множини.

Переваги використання СКБД MySQL:

- простота - MySQL легко встановлюється. Існує багато сторонніх інструментів, включаючи візуальні, що полегшують початок роботи з БД;

- багато функцій - MySQL підтримує більшу частину функціоналу SQL;
- безпека - MySQL має багато функцій безпеки;
- потужність і масштабованість - MySQL може працювати з дійсно великими обсягами даних, і непогано схожий на масштабованих додатків.
- швидкість - обхід деяких стандартів дозволяє MySQL працювати продуктивніше, місцями зрізаючи на поворотах.

Недоліки роботи з СКБД MYSQL:

- відомі обмеження - за визначенням MySQL не може зробити все, що завгодно, і в ній присутні певні обмеження функціональності;
- питання надійності - деякі операції реалізовані менш надійно, ніж інших СКБД;
- застій у створенні - хоча MySQL і є open-source продуктом, робота над нею сильно загальмована. Тим не менш, існує кілька баз даних, повністю заснованих на MySQL (наприклад, MariaDB) [13].

2.5 Функціональна схема роботи інтернет-магазину

Схематично відображаючи принцип функціонування роботи в інтернет-магазині ми можемо казати про певний алгоритм роботи а саме: покупець спочатку входить до самого магазину потім або одразу ж реєструється, або спочатку обирає товар, а згодом реєструється, далі товар додається у кошик і формується замовлення, після чого замовлення було прийнято магазином, товар резервується на складі та формується до відправки після чого йде оплата. Якщо оплачувати зразу, а не на місці, згодом замовлення відправляється до покупця після чого він приймає товар і тоді вже товар вважається проданим, функціональна схема роботи інтернет-магазину зображена бачити на рисунку 2.2. Функціональна схема роботи інтернет-магазину.



Рисунок 2.2 – Функціональна схема роботи інтернет-магазину

2.6 Висновки за розділом

В данному розділі було обрано середовище розробки, що відповідає критеріям, зазначеним при постановці завдання, та сформований необхідний набір інструментів для вирішення проєктного завдання.

Було проведено обґрунтування вибору мови програмування та середовища функціонування системи.

Для розробки проєкту було обрано мову PHP для розробки серверної частини сайту, HTML та CSS було обрано для клієнтської частини інтернет-магазину.

В якості бази даних було обрано СКБД MySQL.

Вибрано найбільш відповідну базу даних для розв'язання задачі проєктування.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Основні відомості про розробку Інтернет-магазину

Інтернет-магазин футбольних товарів дає можливість покупцям здійснювати замовлення товарів у режимі онлайн, а адміністраторам інтернет-магазину додавати та редагувати товар теж онлайн.

У ході розробки інтернет-магазину було використано такі мови програмування як Php, та мову розмітки як Html з таблицями каскадних стилів CSS, стилі сторінки генеруються у динмічному режимі, також використовувався javascript, для прискорення роботи.

Також оскільки для будь-якого інтернет-магазину необхідна база даних, то використовувалася база даних MySQL, як мова запитів до бази використовувався SQL . Приклади використаних SQL запитів було наведено в таблиці 3.1.

Таблиця 3.1 – Приклади використаних при розробці SQL запитів

Текст запиту	Опис
SELECT tovar.naimenovanie, categoria.[id kategorii], '+ ' categoria.nazvanie, tovar.[id tovara], tovar.naimenovanie, tovar.cena, tovar.opisanie, tovar.[id kategorii]'+ 'FROM categoria, tovar '+ 'WHERE categoria.[id kategorii] = tovar.[id kategorii] and tovar.naimenovanie LIKE "%Getri nike%";	Пошук товару за назвою
SELECT pokupatel.FIO, pokupatel.Adress, pokupatel.[Date] '+ 'FROM pokupatel WHERE pokupatel.[FIO] Like "%Borovik Danil Igorevich%";	Пошук покупця за ФІО
'DELETE * FROM tovar WHERE tovar.[id товара]='	Видалення товару за id товару
'INSERT INTO prodaja ([kst], [id check], [id tovara], [stoimost]) '+ ' Values ('+kst+', '+idCheck+', '+ChekNom+', ' +Cenatov+');	Додавання інформації товару в чек
'INSERT INTO pokupatel (FIO,Adress,[DATE]) values ('"Atonenko Anton Antonovich"', "savhoznaja 3" , "13.01.1999");	Додавання нового покупця

3.2 Прицип роботи кошика товарів у інтернет-магазині

Кошик товарів формується разом із головною сторінкою. Кошик у інтернет-магазині потрібен для того, щоб користувач міг зарезервувати товари, що його цікавлять, і продовжити процес оформлення замовлення у магазині.

Якщо кошик товарів порожній, тоді користувач може продовжити вибір товарів. Якщо товари вже зареєстровані у кошику, користувач може оформити замовлення.

Процес замовлення був реалізований таким чином.

```
$connect = mysqli_connect("localhost", "root", "", "cart");
if(isset($_POST["add_to_cart"]))
{
    if(isset($_SESSION["shopping_cart"]))
    {
        $item_array_id = array_column($_SESSION["shopping_cart"], "item_id");
        if(!in_array($_GET["id"], $item_array_id))
        {
            $count = count($_SESSION["shopping_cart"]);
            $item_array = array(
                'item_id' => $_GET["id"],
                'item_name' => $_POST["hidden_name"],
                'item_price' => $_POST["hidden_price"],
                'item_quantity' => $_POST["quantity"]
            );
            $_SESSION["shopping_cart"][$count] = $item_array;
        }
        else
        {
            echo '<script>alert("Item Already Added")</script>';
        }
    }
    else
    {
        $item_array = array(
            'item_id' => $_GET["id"],
            'item_name' => $_POST["hidden_name"],
```

```

'item_price' => $_POST["hidden_price"],
'item_quantity' => $_POST["quantity"]
);
$_SESSION["shopping_cart"][0] = $item_array;
}
}
if(isset($_GET["action"]))
{
if($_GET["action"] == "delete")
{
foreach($_SESSION["shopping_cart"] as $keys => $values)
{
if($values["item_id"] == $_GET["id"])
{
unset($_SESSION["shopping_cart"][$keys]);
echo '<script>alert("Item Removed")</script>';
echo '<script>>window.location="index.php"</script>';
}
}
}
}

?>
<!DOCTYPE html>
<html>
<head>
<title>Shopping Cart In PHP and MySQL | Webdevtrick.com</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
<br />
<div class="container">
<br />
<br />
<br />
<h3 align="center">Shoping Cart With PHP And MySQL | Source Code By <a
href="https://webdevtrick.com">Webdevtrick.com</a></h3><br />
<br /><br />
<?php
$query = "SELECT * FROM tbl_product ORDER BY id ASC";

```

```

$result = mysqli_query($connect, $query);
if(mysqli_num_rows($result) > 0)
{
while($row = mysqli_fetch_array($result))
{
?>
<div class="col-md-4">
<form method="post" action="index.php?action=add&id=?php echo $row["id"]; ?>">
<div style="border:3px solid #5cb85c; background-color:whitesmoke; border-radius:5px; padding:16px;"
align="center">
" class="img-responsive" /><br />
<h4 class="text-info"><?php echo $row["name"]; ?></h4>
<h4 class="text-danger">$ <?php echo $row["price"]; ?></h4>
<input type="text" name="quantity" value="1" class="form-control" />
<input type="hidden" name="hidden_name" value="<?php echo $row["name"]; ?>" />
<input type="hidden" name="hidden_price" value="<?php echo $row["price"]; ?>" />
<input type="submit" name="add_to_cart" style="margin-top:5px;" class="btn btn-success" value="Add to Cart"
/>
</div>
</form>
</div>
<?php
} }
?>
<div style="clear:both"></div>
<br />
<h3>Order Details</h3>
<div class="table-responsive">
<table class="table table-bordered">
<tr>
<th width="40%">Item Name</th>
<th width="10%">Quantity</th>
<th width="20%">Price</th>
<th width="15%">Total</th>
<th width="5%">Action</th>
</tr>
<?php
if(!empty($_SESSION["shopping_cart"]))
{
$total = 0;
foreach($_SESSION["shopping_cart"] as $keys => $values)
{

```

```

?>
<tr>
<td><?php echo $values["item_name"]; ?></td>
<td><?php echo $values["item_quantity"]; ?></td>
<td>$ <?php echo $values["item_price"]; ?></td>
<td>$ <?php echo number_format($values["item_quantity"] * $values["item_price"], 2);?></td>
<td><a href="index.php?action=delete&id=<?php echo $values["item_id"]; ?>"><span class="text-
danger">Remove</span></a></td>
</tr>
<?php
$total = $total + ($values["item_quantity"] * $values["item_price"]);
}
?>
<tr>
<td colspan="3" align="right">Total</td>
<td align="right">$ <?php echo number_format($total, 2); ?></td>
<td></td>
</tr>
<?php
}
?>
</table>
</div>
</div>
</div>
<br />
</body>
</html>

```

В першу чергу оформлення замовлення в магазині починається з формування сторінки кошика, сам вміст сторінки кошика складається з товарів, які були зарезервовані користувачем.

Продовження замовлення в магазині відбувається при переході на наступну сторінку, де користувач повинен погодитися зі змістом замовлення, а потім ввести контактну інформацію, щоб з ним мали змогу зв'язатися співробітники або адміністратори.

Потім замовлення завершується відображенням сторінки з повідомленням про успішне замовлення та занесенням замовлення до бази даних.

У разі непередбачених ситуацій при оформленні замовлення є перевірка, яка виводить повідомлення.

Блок-схему роботи кошика зображено нижче на рисунку 3.1.

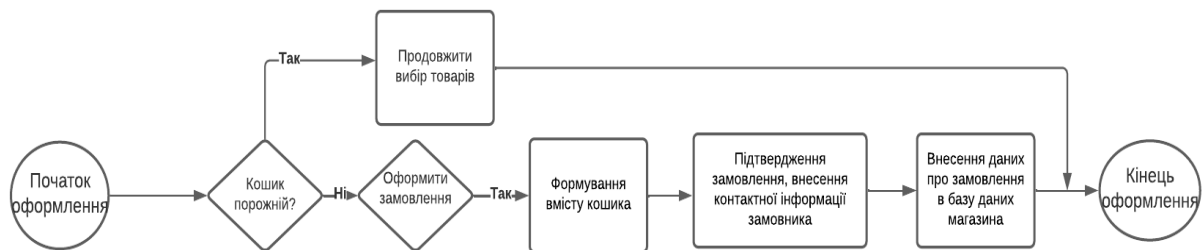


Рисунок 3.1 – Блок-схема роботи кошика замовлень

3.3 Алгоритм роботи головної сторінки інтернет-магазину

Після запуску сайту у його корневій папці шукається файл mainpage.php. Цей файл є головною сторінкою інтернет-магазину. Інтернет-магазин виконує код цього файлу та передає результати замовнику у вигляді скомпільованої сторінки.

На початку виконання mainpage.php проводиться певний аналіз параметрів адресного рядка, робить запит до бази даних інтернет-магазину та формується сторінка на основі інформації з бази даних. Схематичний алгоритм генерації головної сторінки зображено на рисунку 3.2.

Функція підключення до бази даних.

```

function connect($database)
$connect = mysqli_connect("localhost", "root", "", "cart");
function buildQuery( $type='INSERT', $table, $values, $whereClause="", $doNotEnclose=array() ) {
    global $vmLogger;
    if( empty($table) || empty($values) ) {
        return;
    }
  
```

```

$table = trim( $table );
$type = trim( $type );
$type = strtoupper($type);
switch( $type ) {
    case 'INSERT':
    case 'REPLACE':
        $q = "$type INTO `$table` (" ;
        $q .= implode( "`,\n`", array_keys($values) );
        $q .= ") VALUES (\n";
        $count = count( $values );
        $i = 1;
        foreach ( $values as $key => $value ) {
            if( in_array( $key, $doNotEnclose ) ) {
                // Important when using MySQL functions like
                "AES_ENCRYPT", "ENCODE", "REPLACE" or such
                $q .= $value;
            }
            else {
                $q .= "\" . $this->getEscaped($value). "\"\n";
            }
            if( $i++ < $count ) {
                $q .= ',';
            }
        }
        $q .= ')';
        break;

    case 'UPDATE':

        $q = "UPDATE `$table` SET ";
        $count = count( $values );
        $i = 1;
        foreach ( $values as $key => $value ) {
            $q .= "`$key` = " . $this->getEscaped($value). " ";
            if( $i++ < $count ) {
                $q .= ",\n";
            }
        }
        $q .= "\n$whereClause";
        break;

    default:

```

```

$vmLogger->debug( 'Function '.__FUNCTION__.' can\'t build a query of the
type "'.$type.'" );

return;

}

$this->setQuery( $q );

```

Нижче на рисунку 3.2 подано блок-схему. Спочатку користувач у блоці 2 робить спробу підключення до бази даних далі у блоці 3 в залежності від дій користувача і аналізу адресного рядка або підключаються стилі, формується головна сторінка магазину або формується повідомлення про неможливість відображення головної сторінки.



Рисунок 3.2 – Блок-схема генерації головної сторінки

3.4 Алгоритм функціонування адміністративної частини

У розробленому вебсайті була розроблена адміністративна панель що слугує для спрощення процесу керування наповненням вебсайту та роботи з замовленнями.

Щоб приступити до роботи з адміністративною паннелю спочатку необхідно авторизуватися. Для цього потрібно ввести облікові дані, якщо облікові дані вірні, то виконається підключення до бази даних і адміністратор перейде до головного меню адміністративної панелі, якщо облікові дані було введено не вірно, то користувачу видасть помилку.

Нижче наведений код та на рисунку 3.3 блок-схема функціонування адміністративної панелі:

```

<?php include('server.php') ?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<title>SB Admin - Start Bootstrap Template</title>
<!-- Bootstrap core CSS-->
<link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<!-- Custom fonts for this template-->
<link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
<!-- Custom styles for this template-->
<link href="css/sb-admin.css" rel="stylesheet">
</head>
<body class="bg-dark">
<div class="container">
<div class="card card-register mx-auto mt-5">
<div class="card-header">Register an Account</div>
<div class="card-body">
<form method="post" action="register.php">
<?php include('errors.php'); ?>
<div class="form-group">
<div class="form-row">
<div class="col-md-12">
<label for="exampleInputName">Username</label>
<input class="form-control" id="exampleInputName" type="text" name="username" value="<?php echo
$username; ?>" >
</div>
</div>
</div>
<div class="form-group">
<label for="exampleInputEmail1">Email address</label>
<input class="form-control" id="exampleInputEmail1" type="email" aria-describedby="emailHelp"
name="email" value="<?php echo $email; ?>" >
</div>
<div class="form-group">
<div class="form-row">
<div class="col-md-6">
<label for="exampleInputPassword1">Password</label>

```

```

    <input class="form-control" id="exampleInputPassword1" type="password" name="password_1" >
  </div>
  <div class="col-md-6">
    <label for="exampleInputPassword1">Confirm Password</label>
    <input class="form-control" id="exampleInputPassword2" type="password" name="password_2" >
  </div>
</div>
<div>
  <button type="submit" class="btn btn-primary btn-block" name="reg_user">Register</button>
</form>
<div class="text-center">
  <a class="d-block small mt-3" href="login.php">Login Page</a>
  <!-- <a class="d-block small" href="forgot-password.html">Forgot Password?</a-->
</div>
</div>
</div>
</div>
<!-- Bootstrap core JavaScript-->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- Core plugin JavaScript-->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>
</body>
</html>

```

Блок-схема функціонування адміністративної панелі наведена на рисунку 3.3.

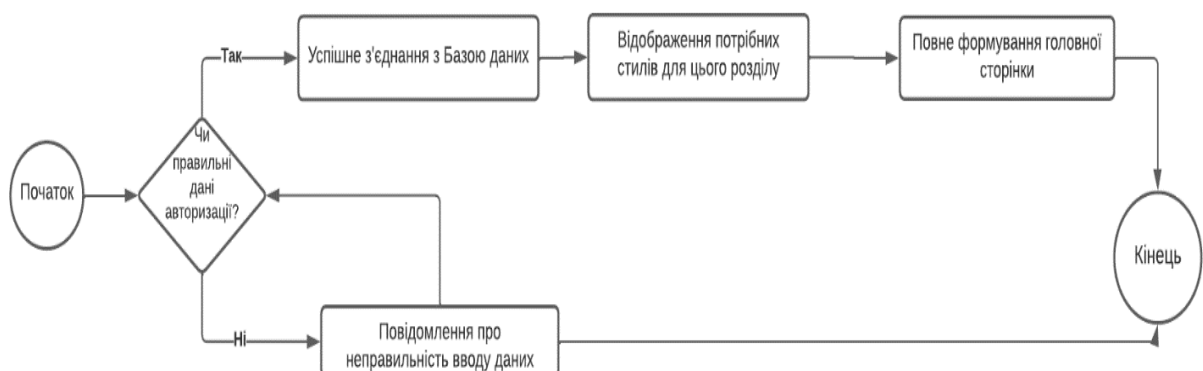


Рисунок 3.3 – Блок-схема функціонування адміністративної панелі

Описуючи нижче наведену блок-схему функціонування адміністративної панелі, потрібно сказати, що у блоці 1 проводиться перевірка параметрів авторизації користувача в залежності від введених користувачем даних і послідууючої перевірки, якщо дані були введені правильно, то проводиться успішне з'єднання з базою даних блок 3, з послідуочим відображенням потрібних стилів, блок 4, та остаточне формування головної сторінки адміністративної панелі блок 5, якщо ж дані були введені неправильно формується повідомлення про неправильність вводу даних.

3.5 Висновки за розділом

В даному розділі описана основна інформація про розробку інтернет-магазину.

Розроблено основні алгоритми, що були використані під час роботи над проєктом – а саме алгоритм роботи кошика магазину, алгоритм роботи головної сторінки магазину. Створено принцип роботи адміністративної панелі у магазині.

Зроблено загальну характеристику створеного вебдодатку та розглянуто основні блоки та процедури виконання.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови використання Інтернет-магазину з продажу футбольних товарів

Вебсайт призначений для автоматизації процесу продажу товарів.

Серверна частина сайту була розроблена за допомогою мови програмування PHP, мови гіпертекстової розмітки HTML, каскадної таблиці стилів CSS. У якості бази даних використовується СКБД MYSQL.

Умовою надійної експлуатації вебсайту є наявність надійного з'єднання з мережею інтернет, а також наявність усіх модулів сайту та бази даних на вебсервері.

Для роботи вебсайту потрібен - комп'ютер з пристроями введення інформації (клавіатура та комп'ютерна миша) або ноутбук, або інший пристрій з можливістю відображення вебсторінок, та виходом до мережі інтернет, з попередньо встановленим веббраузером: Google Chrome, Opera, Mozilla Firefox, Internet Explore або інший.

4.2 Характеристики розробленого продукту

При розробці вебдодатку були реалізовані наступні логічні блоки, які описують певний функціонал інтернет-магазину. Основними такими логічними частинами є:

- main .php - основний логічний блок з обробки списку товарів, включає функції роботи з адміністративної частини сайту;
- product_details.php - блок роботи з базою даних;
- categorie.php - блок, що відповідає за відображення категорій товарів;
- mainpage.php - блок, що формує головну сторінку та користувацькі частини інтернет-магазину та кошика.

Нижче будуть наведені певні логічні процедури наявні в блоках.

Функція отримання поточного статусу замовлення, ця функція насамперед потрібна для отримання клієнтом поточного статусу замовлення код функції представлений нижче.

```
// get the current order status
    $curr_order_status = @$d["current_order_status"];
    $notify_customer = empty($d['notify_customer']) ? "N" : $d['notify_customer'];
    if( $notify_customer=="Y" ) {
        $notify_customer=1;
    }
    else {
        $notify_customer=0;
    }
else {
    if (stristr($vmUser->usertype,"Administrator")) {
        $auth["perms"] = "admin";
    }
    elseif (stristr($vmUser->usertype,"Manager")) {
        $auth["perms"] = "storeadmin";
    }
    else {
        $auth["perms"] = "shopper"; // DEFAULT
    }
    $auth["shopper_group_id"] = $shopper_group["shopper_group_id"];
    $auth["shopper_group_discount"] =
$shopper_group["shopper_group_discount"];
    $auth["show_price_including_tax"] =
$shopper_group["show_price_including_tax"];
    $auth["default_shopper_group"] = 1;
    $auth["is_registered_customer"] = false;
}
```

Нижче на рисунку 4.1 представлена блок-схема функції отримання поточного статусу замовлення.

При отриманні поточного статусу замовлення спочатку користувач з'єднується з базою даних, якщо з'єднання було не успішне, то він не отримає поточний статус замовлення, якщо з'єднання було успішне, то

почнеться ідентифікація користувача й користувач побачить поточний статус замовлення.

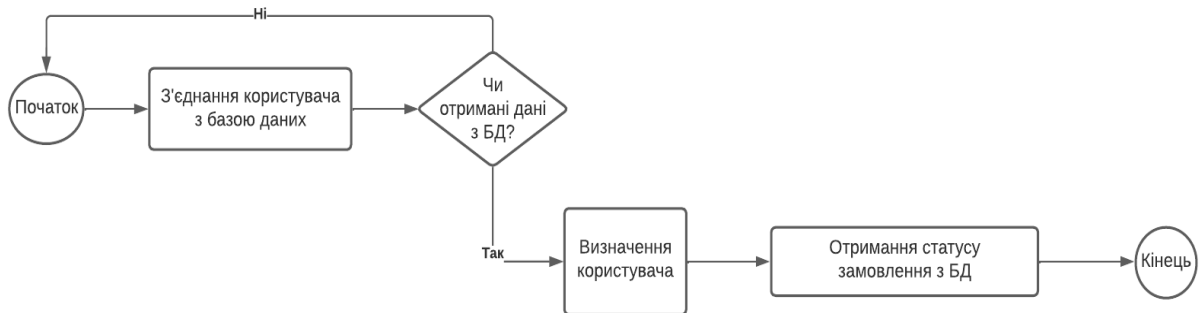


Рисунок 4.1 – Блок-схема функції отримання поточного статусу замовлення

Функція реєстрації користувача що потрібна для того щоб новий користувач магазину мав змогу зареєструватися, код функції представлений нижче.

```

$login=$_POST['login'];
$pas=$_POST['password'];
$db=mysqli_connect('localhost', 'root', '', 'mybase');
$query="INSERT INTO users (login, pas) VALUES ('$login', '$pas')";
$result=mysqli_query($db, $query);
if ($result)
header("Location: formreg.php");
mysqli_close($db);?>
  
```

Нижче на рисунку 4.2 зображена блок-схема Функція реєстрації користувача.

Спочатку користувач вводить дані авторизації після чого проводиться з'єднання з базою даних, після чого виконується перевірка даних і якщо запит успішний, то дані записуються до бази даних, а якщо ні то користувач сторінка оновиться і користувачу потрібно внести інші дані.

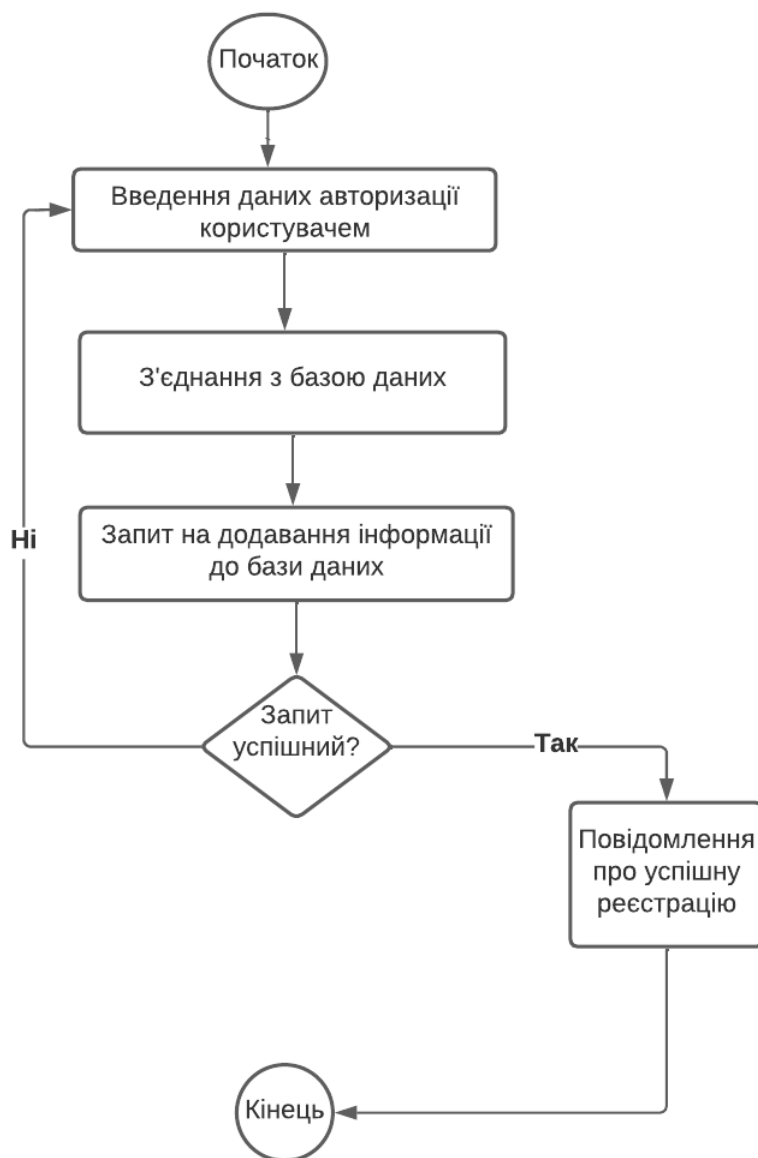


Рисунок 4.2 – Блок-схема функції реєстрації користувача

Функція що виконує запит до бази даних і встановлює ідентифікатор запиту для класу. Код функції представлений нижче.

```

function query( $q="") {
    global $mosConfig_dbprefix, $mosConfig_debug, $vmLogger;
    $prefix = "#__";
    $vm_prefix = "{vm}";

    if (empty($q) ) {
        if( empty($this->_sql)) {

```

```

        $vmLogger->debug( "".__CLASS__.'::'.__FUNCTION__." called without a
pending query.");
    }
    elseif( !$this->_query_set ) {
        $vmLogger->debug( "".__CLASS__.'::'.__FUNCTION__." A query was run
twice without having changed the SQL text.");
    }
}
else {
    $this->setQuery( $q );
}

$this->row = 0;
$this->called = false;
$this->record = null;
$this->record = Array(0);

if (strtoupper(substr( $this->_sql , 0, 6 )) == "SELECT"
    || strtoupper(substr( $this->_sql , 0, 4 ))=='SHOW'
    || strtoupper(substr( $this->_sql , 0, 7 ))=='EXPLAIN'
    || strtoupper(substr( $this->_sql , 0, 8 ))=='DESCRIBE'
) {
    $this->record = $this->_database->loadObjectList();

    if( $this->record === false ) {
        $result = false;
    }
}
else {
    $result = $this->_database->query();
}

$this->_query_set = false;

if( isset( $result )) {
    return $result;
}

```

На рисунку 4.3 зображену блок-схему встановлення ідентифікатору запиту для класу.

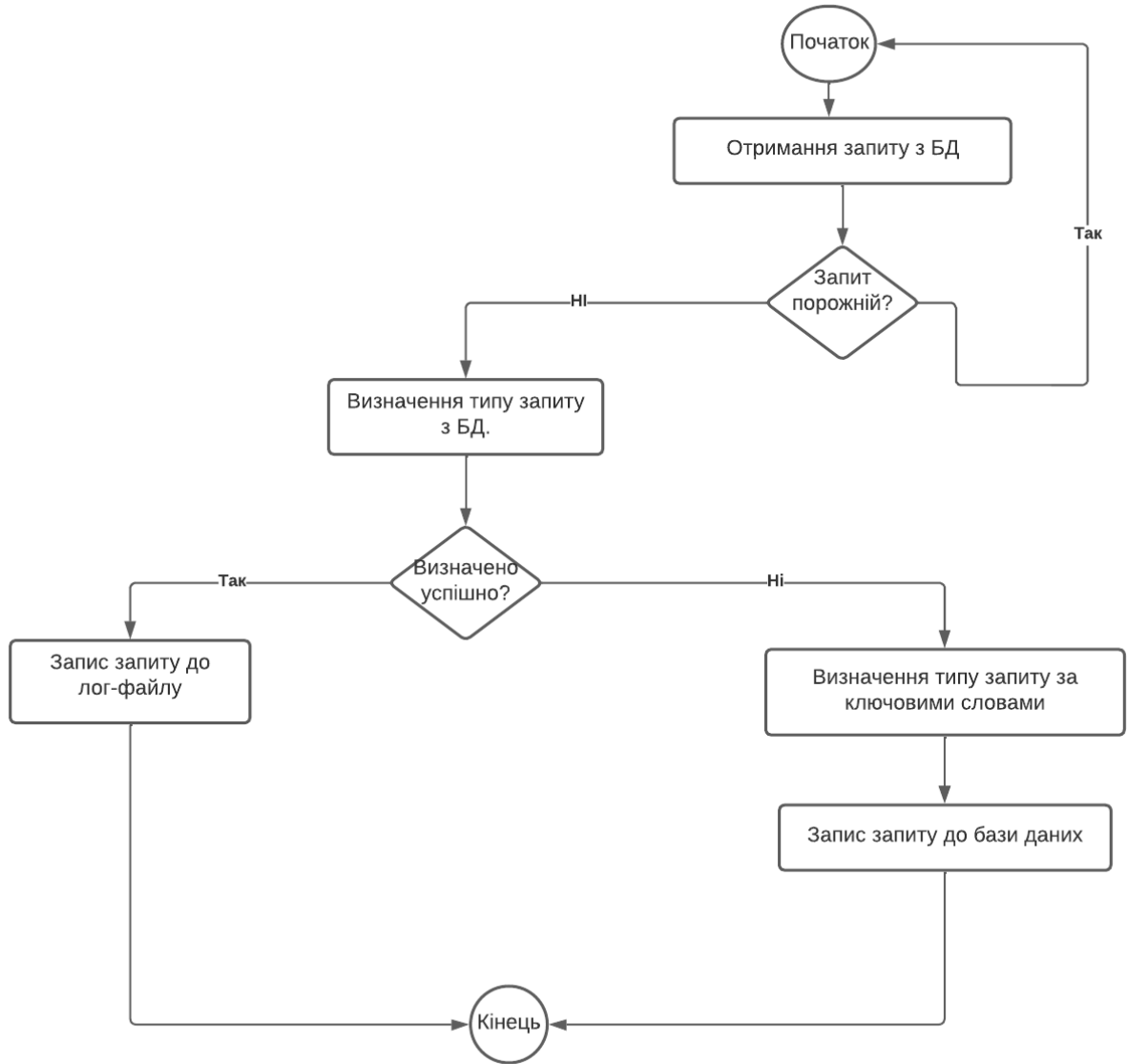


Рисунок 4.3 – Блок-схема встановлення ідентифікатору запиту для класу

Ця функція була розроблена для відстеження хибних запитів, та логування усіх виконаних запитів у процесі роботи вебсайту. Також за допомогою цієї функції визначаються однотипні запити що можуть бути наслідком Ddos-атаки і програміст за допомогою лог-файлу може це визначити.

Спочатку програміст отримує запит з сайту та перевіряє чи він порожній, якщо запит порожній, то він його ігнорує, якщо ні то виконується пошук подібних запитів в базі даних, якщо визначення запиту успішне то він записується до лог-файлу, якщо ні то запит розбивається на складові частини,

та виконується його ідентифікація за допомогою ключових слів, після чого запит додається до лог-файлу та бази даних.

Функція додавання товару до бази даних була розроблена для того щоб адміністратор сайту мав змогу додавати новий товар до інтернет-магазину, код функції представлений нижче.

```
$fields = array ( 'vendor_id' => $vendor_id,
                  'product_sku' => vmGet($d,'product_sku'),
                  'product_name' => vmGet($d,'product_name'),
                  'product_desc' => vmRequest::getVar('product_desc', "",
                  'default', "", VMREQUEST_ALLOWHTML),
                  'product_s_desc' => vmRequest::getVar('product_s_desc', "",
                  'default', "", VMREQUEST_ALLOWHTML),
                  'product_thumb_image' => vmGet($d,'product_thumb_image'),
                  'product_full_image' => vmGet($d,'product_full_image'),
                  'product_publish' => $d['product_publish'],
                  'product_weight' => vmRequest::getFloat('product_weight'),
                  'product_weight_uom' => vmGet($d,'product_weight_uom'),
                  'product_length' => vmRequest::getFloat('product_length'),
                  'product_width' => vmRequest::getFloat('product_width'),
                  'product_height' => vmRequest::getFloat('product_height'),
                  'product_lwh_uom' => vmGet($d,'product_lwh_uom'),
                  'product_unit' => vmGet($d,'product_unit'),
                  'product_packaging' => (($d["product_box"] << 16) |
                  ($d["product_packaging"]&0xFFFF)),
                  'product_url' => vmGet($d,'product_url'),
                  'product_in_stock' => vmRequest::getInt('product_in_stock'),
                  'attribute' => ps_product_attribute::formatAttributeX(),
                  'custom_attribute' => vmGet($d,'product_custom_attribute'),
                  'product_available_date' =>
                  $d['product_available_date_timestamp'],
                  'product_availability' => vmGet($d,'product_availability'),
                  'product_special' => $d['product_special'],
                  'child_options' => $d['child_options'],
                  'quantity_options' => $d['quantity_options'],
                  'product_discount_id' =>
                  vmRequest::getInt('product_discount_id'),
                  'mdate' => $timestamp,
                  'product_tax_id' => vmRequest::getInt('product_tax_id'),
```

```

'child_option_ids' => vmGet($d,'included_product_id'),
'product_order_levels' => $d['order_levels'] );

$db->buildQuery( 'UPDATE',  '#{vm}_product',  $fields,      'WHERE  product_id=' .
(int)$d["product_id"] . ' AND vendor_id=' . (int)$old_vendor_id );
$db->query();

```

На рисунку 4.4 зображену блок-схему функції додавання товару до бази даних.

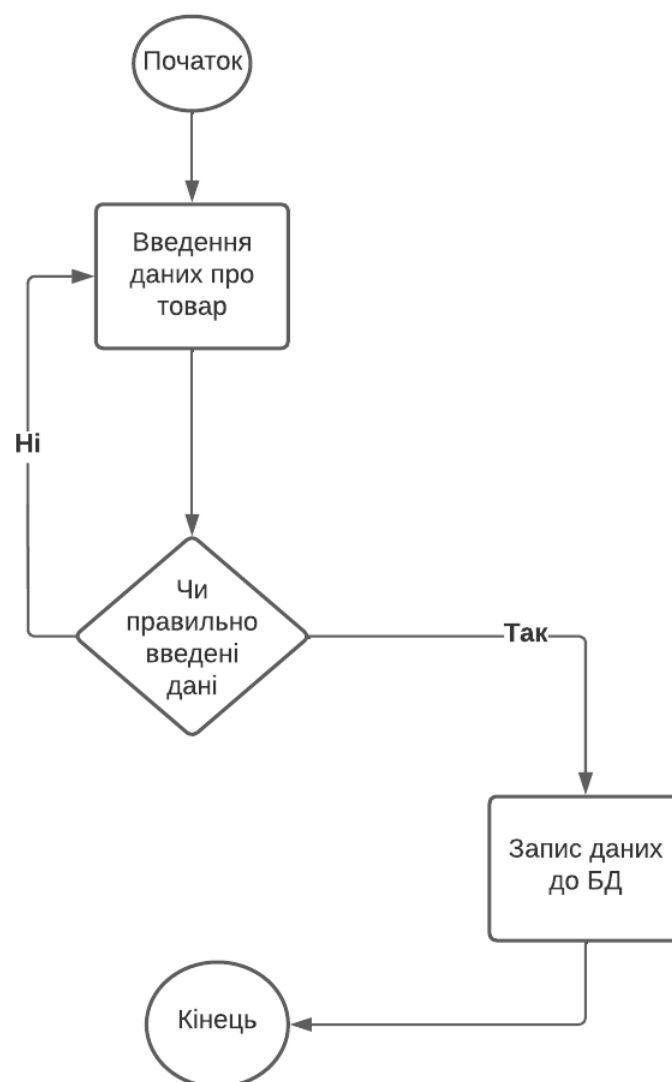


Рисунок 4.4 – Блок-схема функції додавання товару до бази даних

Ця функція потрібна для додавання товару до бази даних.

Спочатку адміністратор вводить дані про товар, якщо дані було введено не вірно, то він повинен перевірити правильність вводу даних, а якщо дані було введено вірно то дані будуть записані до бази даних.

Вхідними даними для онлайн магазину є інформація про товари, що продаються в магазині футбольних товарів.

Вихідними даними для інтернет-магазину є сформовані сторінки сайту з інформацією про продукцію магазину футбольних товарів та замовленнями, що були зроблені користувачам магазину.

4.3 Структура бази даних

В основі розробки інтернет-магазину, лежить база даних, структура якої зображена на рисунку 4.5.

База даних містить вміст даних розробленого проекту, а саме:

- категорії товарів (таблиця Категорія);
- інформація про товари, які продаються в Інтернет - магазині (таблиця Товар);
- замовлень (таблиця Продажа);
- чеків(таблиця Чек);
- покупців (Таблиця.Покупець).

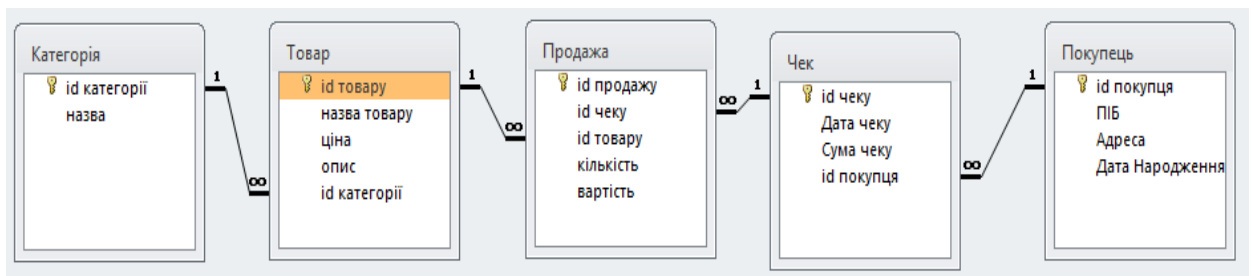


Рисунок 4.5 – Структурна схема бази даних інтернет-магазину

Всі таблиці бази даних пов'язані одна з одною зв'язком «один до багатьох». За допомогою зовнішніх та первинних ключів таблиці зв'язані одна з одною. Таблиці бази даних мають наступні зв'язки:

- таблиця «Категорія» зв'язана з таблицею «Товар» полями «id_категорії»;
- таблиця «Товар» зв'язана з таблицею «Продажа» полями «id_товару»;
- таблиця «Продажа» зв'язана з таблицею «Чек» полями «id_чеку»;
- таблиця «Чек» зв'язана з таблицею «Покупець» полями «id_покупця».

4.4 Структурна схема компонентів інтернет-магазину

Якщо розглядати інтернет-магазин як саме вебсайт то можна поділити його на дві основні логічні частини це: адміністративна частина, яка дозволяє адміністратору інтернет-магазину (після входу в систему адміністрування), редагувати базу даних та додавати до неї додаткові елементи (товари). У Клієнтській частині ця можливість редагування даних відсутня, тобто доступ до ресурсів які можна редагувати закритий для загального огляду. Та клієнтська частина дозволяє всім користувачам і відвідувачам інтернет-магазину переглядати, шукати, вибирати потрібний товар, розміщений у різних групах. Це значно спрощує роботу користувача під час пошуку конкретної інформації. Клієнт може оформити замовлення на товар, вибрати спосіб оплати та доставки товару. Структурно схему компонентів вебсайту зображено на рисунку 4.6.

4.5 Висновки за розділом

У цьому розділі було розглянуто основне призначення й умови використання розроблюваного інтернет-магазину.

Структурна схема компонентів інтернет-магазину

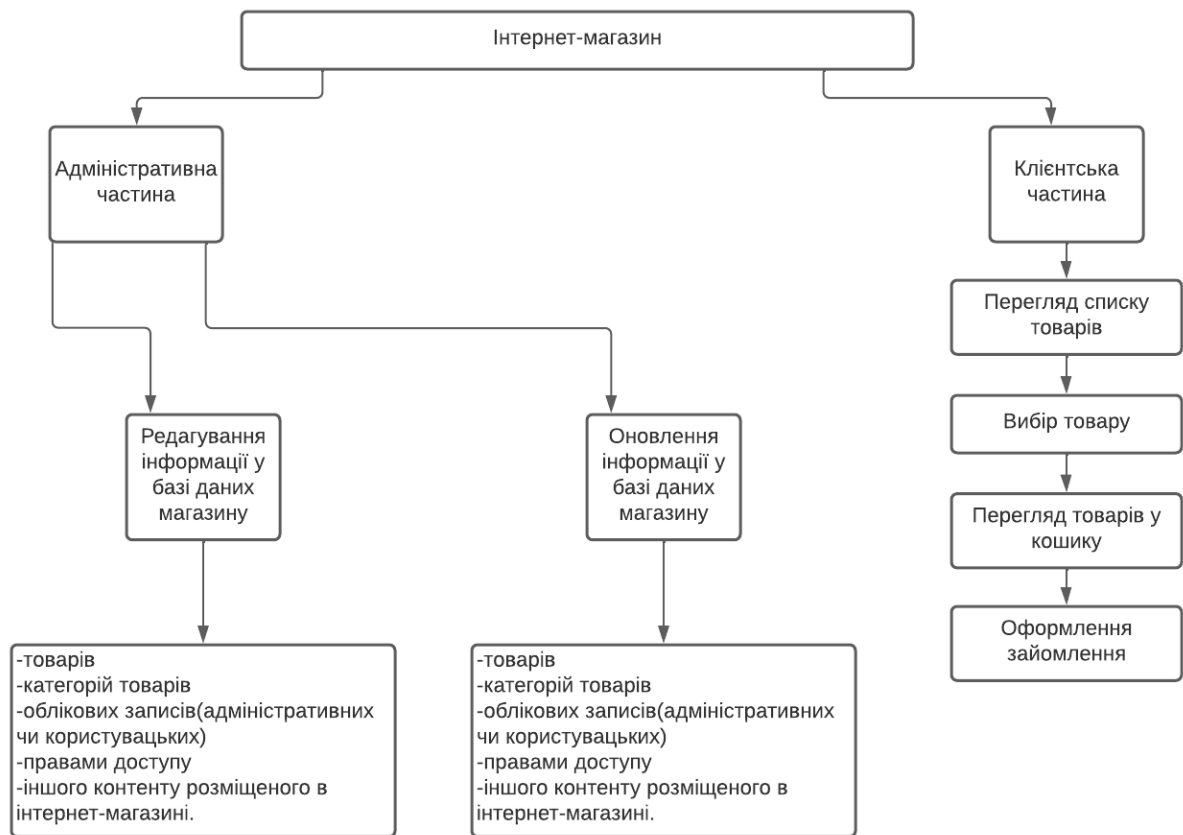


Рисунок 4.6 – Структурна схема компонентів інтернет-магазину

Описані основні логічні блоки що були розроблені у процесі розробки інтернет-магазину, та деякі логічні функції що наявні у логічних блоках.

Було розглянуто структуру бази даних первинні ключі та основні зв'язки у таблицях.

Також було розглянуто структурну схему компонентів інтернет-магазину що демонструю логічну схему взаємодії компонентів, розроблених у магазині.

Створено структуру бази даних інтернет-магазину.

Описано структурну схему розробленої системи.

5 КЕРІВНИЦТВО СИСТЕМНОГО АДМІНІСТРАТОРА

5.1 Розгортання інтернет-магазину

Інтернет-магазин знаходиться в архівному файлі під назвою «4football.zip».

Файл можна надати за допомогою флешки.

Як на локальному вебсервері за допомогою створення віртуального хоста так і в мережі інтернет.

Щоб розмістити сайт в мережі Інтернеті, необхідно виконати наступні дії:

- створити віртуальний хост на сервері;
- скопіювати вміст архіву «4football.zip» у кореневий каталог віртуального хоста;
- створити базу даних сайту за допомогою системи управління MySQL;
- перейти на головну сторінку сайту.

5.2 Створення віртуального хоста на сервері

Щоб створити віртуальний хост, вам потрібно написати директиви для контейнера <VirtualHost> у файлі httpd.conf, які визначають конфігурацію віртуального хоста:

```
<VirtualHost 192.168.1.1>  
    ServerAdmin WebMaster@my_domain.ua  
    DocumentRoot d:/www/4football  
    ServerName 4football  
    ErrorLog logs/ 4football -error.log  
    CustomLog logs/ 4football -access.log
```

Директиви віртуального хоста:

«DocumentRoot» - вказує каталог, де знаходяться файли (сторінки) цього віртуального хоста;

"ServerName" - вказує ім'я віртуального хоста, до якого можна отримати доступ;

«ErrorLog» – файл журналу помилок;

CustomLog – це звичайний файл журналу подій.

5.3 Розгортання бази даних вебсайту.

Для створення бази даних використовується вебдодаток «phpMyAdmin», який значно спрощує керування вмістом бази даних через наявність графічного інтерфейсу, без зайвих sql запитів і потреби у додатковому ПО, мною була використана версія 2.6.1. через надійність у використанні.

Отже для виклику цього додатку необхідно перейти до веббраузера і у ньому задати наступну вебадресу: «<http://localhost/Tools/phpmyadmin/index.php>», у результаті ми перейдемо до вебсторінки де з'явиться вікно, як на рисунку 5.1.

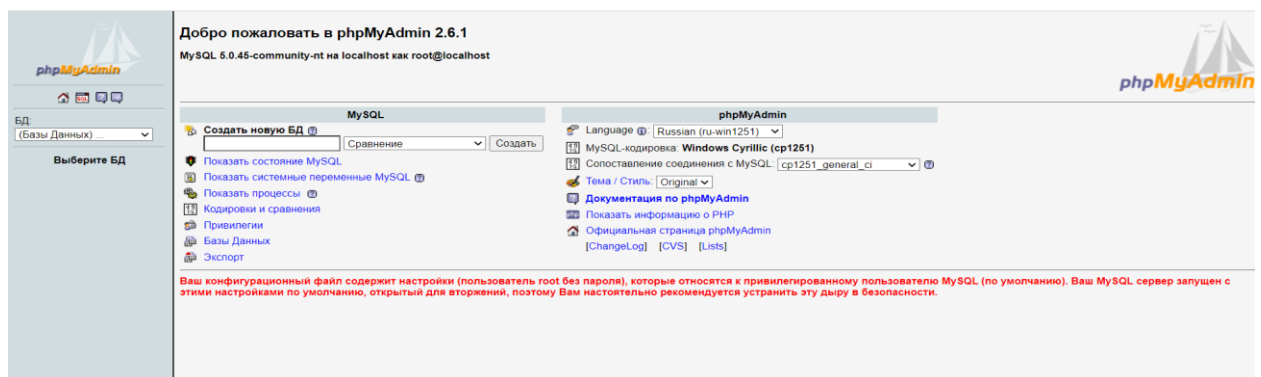


Рисунок 5.1 – Вікно вебдодатку «phpMyAdmin»

Далі наступним кроком в розгортанні бази потрібно у полі «Створити нову базу» ввести назву бази даних магазину, а саме «bd » і натиснути на

кнопку «Створити». Далі потрібно перейти до створеної бази даних, до вкладки «SQL», з файлу «BD_bd.txt» скопіюйте вміст файлу і вставте текстове поле на вкладку «SQL», потім натисніть «Перейти». Вікно виконання запитів для створення бази даних показано на рисунку 5.2.

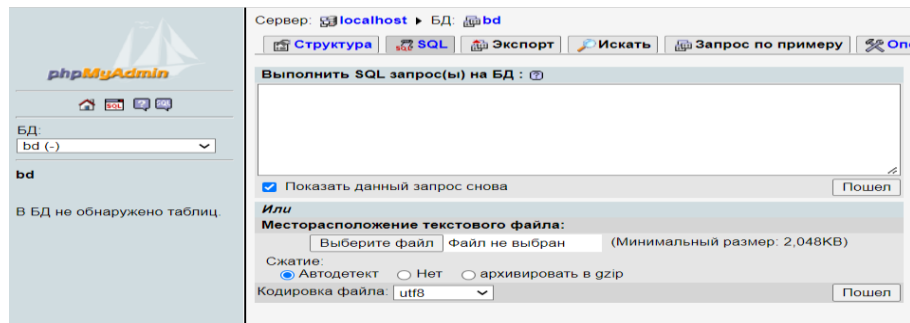


Рисунок 5.2 – Вікно виконання SQL запитів для створень бази даних

5.4 Адміністративна панель інтернет магазину

В інтернет-магазині передбачена адміністративна панель яка доступна виключно для адміністраторів вебсайту, задля автоматизації та спрощення роботи з необхідним контентом та здійсненими замовленнями у магазині. Щоб потрапити в панель керування досить перейти за адресою виду адреса_сайту / administrator / на відображеній сторінці авторизуватися (ввести зазначені під час установки движка логін і пароль) і натиснути кнопку увійти.

Вікно входу в адміністративну панель на рисунку 5.3.

Рисунок 5.3 – Вікно входу в адміністративну панель

Після вдалої авторизації можна потрапити в адміністративну панель. Тут зосереджені всі наявні інструменти для редагування і додавання товарів, та роботи з замовленнями.

Потрапивши до вікна роботи з товарами в адміністратора є можливість додавати, редагувати та видаляти будь який з товарів також пошук зі списку за категорією.

Також у кожного товару є такі поля як: назва, ціна, категорія, виробник, та ключове id поле.

Для того щоб зробити будь яку дію з товаром ми спочатку повинні його обрати у відповідному елементі checkbox, а згодом робити потрібну нам маніпуляцію. Вікно роботи з товаром відображено на рисунку 5.4.

Фільтр: 

#	☐ Название товара	Цена	Категория	Производители	Удалить	Id
1	☒ ibvv	0		Adidas		79
2	☐ Бандаж на голеностоп Select 6100 561000	450 UAN	Футбольный инвентарь	Select		65
3	☐ БУТСЫ NIKE MERCURIAL NEYMAR VICTORY 6 DYNAMIC	1 600 UAN	Бутсы	Nike		50
4	☐ Бутсы Nike Mercurial Vapor Academy	1 500 UAN	Бутсы	Nike		46
5	☐ Бутсы nike mercurial victory CR7	1 450 UAN	Бутсы	Nike		49
6	☐ Бутсы Nike Vapor Academy AH7375-081	1 500 UAN	Бутсы	Nike		47
7	☐ Бутсы с носком nike mercurial victory	1 700 UAN	Бутсы	Nike		48
8	☐ ГЕТРЫ БЕЗ НОСКА NIKE TS STIRRUP III GAME SOCK 507819-100	150 UAN	Гетры	Nike		53
9	☐ Заморозка спортивная в аэрозоле IceMix 200 ml	85 UAN	Футбольный инвентарь	Select		62
10	☐ Компрессионные кинезио гетры Select 6150 561500 (2 штуки)	800 UAN	Футбольный инвентарь	Select		66
11	☐ Компрессионный кинезио бандаж на бедро Select 6350 563500	450 UAN	Футбольный инвентарь	Select		67
12	☐ Компрессионный наколенник для стабилизации коленного сустава с к	800 UAN	Футбольный инвентарь	Select		64
13	☐ Манометр Select для футбольных мячей	450 UAN	Футбольный инвентарь	Select		61
14	☐ Мяч Adidas Finale Milano Competition AC5492 №4	1 500 UAN	Мячи	Adidas		54

Рисунку 5.4 –Вікно роботи з товаром

Також у адміністративній панелі передбачена можливість роботи з замовленнями магазину. Ця функція перш за все необхідна для значного спрощення роботи адміністрації магазину з потоком користувацьких замовлень вона дає можливість за допомогою достатнього простого і нативно зрозумілого інтерфейсу в режимі реального часу проводити роботу з замовленнями клієнтів адміністратору магазину. Для початку роботи з замовленнями необхідно перейти у вікно роботи з замовленнями, натиснувши на меню “замовлення” де ми можемо бачити список замовлень

магазину, і видаляти чи редагувати статус замовлення. Оскільки у кожного замовлення існує 4 статуси замовлення для того щоб адміністратор магазину міг розрізняти поточний статус того чи іншого замовлення, а саме існують такі статуси як на розгляді, підтверджено, скасовано та відправлено. Для того щоб зробити певну дію з тим чи іншим замовлення треба його обрати, натиснення до елементу checkbox і після того зробити ту чи іншу маніпуляцію. Вікно роботи з замовленнями представлено на рисунку 5.5.

Список замовлень

Пошук

[Pending](#) | [Confirmed](#) | [Cancelled](#) | [Shipped](#) | [Усе](#)

#	☐	Номер замовлення	Ім'я	Вид для друку	дата замовлення	Змінено	Статус	Оновити	Проміжний підсумок	видалити
1	☐	00000019	Антон Антоненко		13-квіт-22 23:05	13-квіт-22 23:05	Pending ▾	☐ Повідомити покупця? Оновити статус	530 UAN	
2	☐	00000018	Алла Смирнова		13-квіт-22 23:05	13-квіт-22 23:05	Pending ▾	☐ Повідомити покупця? Оновити статус	279 UAN	
3	☐	00000017	Ольга Сладкова		13-квіт-22 22:49	13-квіт-22 22:49	Pending ▾	☐ Повідомити покупця? Оновити статус	530 UAN	
4	☐	00000016	Антон Антоненко		13-квіт-22 22:49	13-квіт-22 22:49	Pending ▾	☐ Повідомити покупця? Оновити статус	250 UAN	
5	☐	00000015	Антон Антоненко		13-квіт-22 22:46	13-квіт-22 22:46	Pending ▾	☐ Повідомити покупця? Оновити статус	2 300 UAN	

Рисунок 5.5 –Вікно роботи з замовленнями

5.5 Висновки за розділом

У цьому розділі було надано опис призначення, умов експлуатації вебсайту.

Описаний процес розгортання електронного магазину та процес створення віртуального хоста на сервері.

Було розглянути способи керування адміністративною паннеллю магазину, додаванням нових товарів, редагуванням їх через адміністративну панель та керуванням замовлень.

Також у розділі описаний процес розгортання БД за допомогою вебдодатку «phpMyAdmin».

6 КЕРІВНИЦТВО КОРИСТУВАЧА

6.1 Призначення сайту

Вебсайт призначений для користувачів котрі хочуть придбати футбольні товари.

Серверна частина сайту була розроблена за допомогою мови програмування PHP, мови гіпертекстової розмітки HTML, каскадної таблиці стилів CSS. У якості бази даних використовується СКБД MYSQL.

Умовою надійної експлуатації вебсайту є наявність надійного з'єднання з мережею інтернет, а також наявність всіх модулів та бази даних на вебсервері.

Для роботи вебсайту потрібен - комп'ютер з пристроями введення інформації (клавіатура та комп'ютерна миша) або ноутбук, або інший пристрій з можливістю відображення вебсторінок, та виходом до мережі інтернет, з попередньо встановленим веббраузером: Google Chrome, Opera, Mozilla Firefox, Internet Explore або інший.

6.2 Вхід до вебсайту та процес авторизації й замовлення товару

Для відкриття даного вебсайту необхідно в полі пошуку веббраузера ввести вебадресу сайту.

Вікно вхід на сайт на рисунку 6.1.

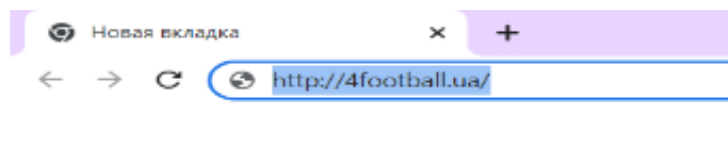


Рисунок 6.1 –Вхід до вебсайту з браузера

Далі обрати товар за допомогою пошуку чи вибору товару в певній категорії, обрати кількість товару, колір та розмір, якщо є купон на знижку то ввести у відповідне поле та натиснути кнопку “купити”. Вікно оформлення замовлення на рисунку 6.4.

Оформити замовлення

Назва	Ціна	Кількість / Оновити	Проміжний підсумок
Футбольний м'яч Select TEAM #5 - Профі	800 грн.	1	800 грн.
Проміжний підсумок:			800 грн.
Разом:			800 грн.

Рисунок 6.4 – Вікно оформлення замовлення

Після чого перейти до кошика натиснути кнопку підтвердження замовлення, обрати товар, спосіб його доставки, заповнити необхідні поля для замовлення (свої персональні дані), поставити галочку щоб погодитися на умови інтернет-магазину. Вікно підтвердження замовлення представлено на рисунку 6.5.

Та потім чекати повідомлення на вашу електронну адресу від адміністратора інтернет-магазину, який повинен підтвердити замовлення.

6.3 Реалізація зворотнього зв'язку з клієнтами інтернет-магазину

При виникненні певних запитань клієнта до адміністрації, на сайті було реалізовано можливість зворотнього зв'язку, та також можливість залишити коментарій до того чи іншого товару. Для того щоб залишити коментарій потрібно перейти до того товару який вас зацікавив, заповнити запропоновані поля, серед яких ваше ім'я, електронна адреса або вебсайт,

залишити сам коментар, та натиснути кнопку відправити.

Будь ласка, перевірте введені дані та підтвердіть замовлення!

Адреса доставки:

Тариф доставки: STD; Standard Shipping under 10000.00; 0 грн.;

Спосіб оплати: Готівкою кур'єру

За бажанням, Ви можете залишити коментар до замовлення:

Подзвонити перед доставкою

☒ Я згоден з цими умовами

Політика повернення Ви можете скасувати це замовлення не пізніше 2-х тижнів після того, як ми його отримали. Ви можете повернути нові (невідкриті товари) зі скасованого замовлення, але не пізніше 2-х тижнів після того, як вони були доставлені до вас. Товари повинні бути повернені до їх первинної упаковки. Для більш детальної інформації щодо скасування замовлень та повернення товару, див. Політика щодо повернення.

Підтвердити

Рисунок 6.5 – Вікно підтвердження замовлення

Для того щоб зв'язатися з адміністрацією магазину потрібно перейти до форми зворотнього зв'язку, заповнити необхідні дані, й відправити повідомлення до адміністрації на вказану електронну пошту прийде відповідь. Приклад форми зворотнього зв'язку зображено на рисунку 6.6.

Имя

Email

Заголовок

Сообщение

ОТПРАВИТЬ

Рисунок 6.6 – Форма зворотнього зв'язку

6.4 Висновки за розділом

В розділі було описано процес взаємодії користувача з інтернет-магазином, способи реєстрації, авторизації, пошуку, замовлення та оплати товару.

Інтерфейс розробленого вебдодатку є типовим для будь-якого користувача і не повинен викликати тих чи інших складнощів при використанні за основним призначенням.

ВИСНОВКИ

У процесі виконання дипломної кваліфікаційної роботи було розроблено інтернет-магазин із продажу футбольних товарів.

Після тестування та налагодження функціонування онлайн-магазину можна зробити наступні висновки:

- функціонал інтернет-магазину повністю відповідає вимогам технічного завдання;
- інтернет-магазин працює стабільно та автономно;
- інтернет-магазин з продажу футбольних товарів має зручний інтерфейс та діалог з користувачем;
- інтернет-магазин забезпечує коректність введення та виведення даних (обробка винятків);
- онлайн-магазин має необхідну інформацію про процеси роботи системи (допомога користувачеві) та коректно відображається у всіх сучасних браузерях.

Для розробки проєкту було обрано мову PHP для розробки серверної частини сайту, HTML та CSS для клієнтської частини інтернет-магазину.

Як базу даних було обрано базу даних MySQL. Створено структуру бази даних інтернет-магазину. Описано блок-схему розробленої системи.

Також проаналізоване середовище розробки, що відповідає зазначеним у постановці задачі критеріям, та необхідний набір інструментів для вирішення проєктної задачі.

Обґрунтовано вибір мови програмування та середовища функціонування системи.

В результаті випробувань розроблений інтернет-магазин дозволяє відображати інформацію про футбольні товари на сайті та продавати футбольні товари через Інтернет.

ПЕРЕЛІК ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Пономаренко Л. А. Електронна комерція [Текст] : підруч. для студ. екон. спец. / Л. А. Пономаренко, В. О. Філатов ; Київський національний торговельно-економічний ун-т. - К. : КНТЕУ, 2002. - 443 с.
2. IBEF / [Electronic resource]. – Access mode: <https://www.ibef.org/>
3. Краус К. М. Електронна комерція та інтернет-торгівля [Текст] : навч.-метод. посіб. / К. М. Краус, Н. М. Краус, О. В. Манжура. - Київ : Аграр Медіа Груп, 2021. - 454 с.
4. Краус К. М. Електронна комерція та інтернет-торгівля [Текст] : навч.-метод. посіб. / К. М. Краус, Н. М. Краус, О. В. Манжура. - Київ : Аграр Медіа Груп, 2021. - 454 с.
5. MySQL Documentation [Electronic resource]. – Access mode: <https://www.mysql.com>.
6. Иванова, Г. С. Основы программирования [Текст] : Учебник для вузов / Г. С. Иванова - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 416 с.
7. Бардзелл, Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFussion и PHP. Из первых рук (+CD – ROM) [Текст] : Учебник для вузов / Джеффри Бардзелл. - М.:Эком,2017. - 560 с.
- Бенкен, Е. PHP, MySQL, XML. Программирование для Интернета [Текст] : Учебник для вузов / Е. Бенкен. - М.: БХВ-Петербург, 2016. - 352 с.
- Дронов, Владимир PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов [Текст] : Учебник для вузов / Владимир Дронов. - М.: БХВ- Петербург, 2017. - 544 с.
10. Дунаев.В. Сценарии для Web-сайта. PHP и JavaScript [Текст] : Учебник для вузов / В. Дунаев. - М.: БХВ-Петербург, 2017. - 576 с.
11. Кузнецов, М. PHP. Практика создания Web-сайтов [Текст] : Учебник для вузов / М. Кузнецов, И. Симдянов. - М.: БХВ-Петербург, 2012. - 577 с.
12. Ленгисторф, Джейсон PHP и jQuery для профессионалов [Текст] :

Учебник для вузов / Джейсон Ленгсторф. - М.: Вильямс, 2017. - 362 с.

13. Hedengren T. D. Smashing WordPress [Text] : beyond the blog / Thord Daniel Hedengren. - Chichester, West Sussex : Wiley, 2010. - X, 308 p.

ДОДАТОК А
Технічне завдання

A.1 Підстава для розробки

Підставою для розробки послужило завдання на дипломну роботу на тему "Розробка інтернет-магазину з продажу футбольних товарів", затверджений 27 квітня 2022 року № 102 по Національному університету «Запорізька політехніка» .

A.2 Призначення розробки

Мета роботи - розробка програмного продукту, який реалізує функції Інтернет магазину з індивідуальним наповненням та працює в мережі Інтернет, і створення індивідуального наповнення Вебсайту відповідно до певної предметної області.

A.3 Основні вимоги до розроблювальної програми

A.3.1 Вимоги до надійності

Програма повинна забезпечувати коректну обробку помилок у наступних ситуаціях:

- при введенні користувачем неприпустимих значень;
- при невірній роботі з файлами (спробі доступу до файлу неприпустимого формату, спробі запису на захищений від запису диск та ін.).

A.3.2 Умови експлуатації

Під час роботи програми не рекомендується перезавантажувати ПК або вимикати живлення. Вихід із програми необхідно здійснювати послідовною зупинкою клієнтського й серверного сервісів.

Експлуатація програмного продукту здійснюється відповідно до експлуатаційної документації на розроблену програму, що відповідає

стандартам і містить інформацію, необхідну для освоєння й експлуатації програмного продукту.

А.3.3 Вимоги до складу й параметрів технічних засобів

Необхідними вимогами до складу й параметрів технічних засобів, які повинні бути застосовані для роботи Інтернет-магазин, є:

- Оперативна пам'ять - не менш 1Гб;
- HDD (жорсткий диск)- не менш 40Гб;
- відео адаптер: DirectX 9 або пізніша версія з драйвером WDDM 1.0;
- процесор - не менш 1 ГГц
- мережний адаптер.

А.3.4 Вимоги до маркування й пакування

Програма може бути записана як на Flash-накопичувач, так і на компакт-диску. На пакуванні повинна бути назва програми - " Розробка інтернет-магазину з продажу футбольних товарів".

А.3.5 Вимоги до транспортування й зберігання

Вимоги до транспортування й зберігання аналогічні вимогам, які пропонуються до носія, на якому записана програма.

А.3.6 Вимоги до програмної документації

Програма повинна поставлятися з "Керівництвом оператора".

ДОДАТОК Б
Текст програми

Б.1 Модуль main.php. Функція отримання поточного статусу замовлення

```
// get the current order status
$curr_order_status = @$d["current_order_status"];
$notify_customer = empty($d['notify_customer']) ? "N" : $d['notify_customer'];
if( $notify_customer=="Y" ) {
    $notify_customer=1;
}
else {
    $notify_customer=0;
}

else {
    if (strstr($vmUser->usertype,"Administrator")) {
        $auth["perms"] = "admin";
    }
    elseif (strstr($vmUser->usertype,"Manager")) {
        $auth["perms"] = "storeadmin";
    }
    else {
        $auth["perms"] = "shopper"; // DEFAULT
    }
    $auth["shopper_group_id"] = $shopper_group["shopper_group_id"];
    $auth["shopper_group_discount"] =
$shopper_group["shopper_group_discount"];
    $auth["show_price_including_tax"] =
$shopper_group["show_price_including_tax"];
    $auth["default_shopper_group"] = 1;
    $auth["is_registered_customer"] = false;
}
}
```

Б.2 Модуль mainpage.php. Функція реєстрації незареєстрованого користувача

```
else {
    if (strstr($vmUser->usertype,"Administrator")) {
        $auth["perms"] = "admin";
    }
}
```

```

elseif (strpos($vmUser->usertype,"Manager")) {
    $auth["perms"] = "storeadmin";
}
else {
    $auth["perms"] = "shopper"; // DEFAULT
}
$auth["shopper_group_id"] = $shopper_group["shopper_group_id"];
$auth["shopper_group_discount"] =
$shopper_group["shopper_group_discount"];
$auth["show_price_including_tax"] =
$shopper_group["show_price_including_tax"];
$auth["default_shopper_group"] = 1;
$auth["is_registered_customer"] = false;
}

```

Б.3 Модуль `product_details.php`. Фунція що виконує запит до бази даних і встановлює ідентифікатор запиту для класу

```

function query( $q="") {
    global $mosConfig_dbprefix, $mosConfig_debug, $vmLogger;
    $prefix = "#__";
    $vm_prefix = "{vm}";

    if (empty($q) ) {
        if( empty($this->_sql)) {
            $vmLogger->debug( ".__CLASS__.'::'.__FUNCTION__.'" called without a
pending query.');
```

```

        }
        elseif( !$this->_query_set ) {
            $vmLogger->debug( ".__CLASS__.'::'.__FUNCTION__.'" A query was run
twice without having changed the SQL text.');
```

```

        }
    }
    else {
        $this->setQuery( $q );
    }

    $this->row = 0;
    $this->called = false;
    $this->record = null;

```

```

$this->record = Array(0);

if (strtoupper(substr( $this->_sql , 0, 6 )) == "SELECT"
    || strtoupper(substr( $this->_sql , 0, 4 ))=='SHOW'
    || strtoupper(substr( $this->_sql , 0, 7 ))=='EXPLAIN'
    || strtoupper(substr( $this->_sql , 0, 8 ))=='DESCRIBE'
) {
    $this->record = $this->_database->loadObjectList();

    if( $this->record === false ) {
        $result = false;
    }
}
else {
    $result = $this->_database->query();
}

$this->_query_set = false;

if( isset( $result )) {
    return $result;
}
}

```

Б.4 Модуль product_details.php. Функція додавання товару до бази даних

```

$fields = array ( 'vendor_id' => $vendor_id,
                  'product_sku' => vmGet($d,'product_sku'),
                  'product_name' => vmGet($d,'product_name'),
                  'product_desc'  =>  vmRequest::getVar('product_desc',  "",
                  'default', "", VMREQUEST_ALLOWHTML),
                  'product_s_desc' =>  vmRequest::getVar('product_s_desc',  "",
                  'default', "", VMREQUEST_ALLOWHTML),
                  'product_thumb_image' => vmGet($d,'product_thumb_image'),
                  'product_full_image' => vmGet($d,'product_full_image'),
                  'product_publish' => $d['product_publish'],
                  'product_weight' => vmRequest::getFloat('product_weight'),
                  'product_weight_uom' => vmGet($d,'product_weight_uom'),
                  'product_length' => vmRequest::getFloat('product_length'),

```

```

'product_width' => vmRequest::getFloat('product_width'),
'product_height' => vmRequest::getFloat('product_height'),
'product_lwh_uom' => vmGet($d,'product_lwh_uom'),
'product_unit' => vmGet($d,'product_unit'),
'product_packaging' => (($d["product_box"] << 16) |
($d["product_packaging"]&0xFFFF)),

'product_url' => vmGet($d,'product_url'),
'product_in_stock' => vmRequest::getInt('product_in_stock'),
'attribute' => ps_product_attribute::formatAttributeX(),
'custom_attribute' => vmGet($d,'product_custom_attribute'),
'product_available_date' =>

$d['product_available_date_timestamp'],

'product_availability' => vmGet($d,'product_availability'),
'product_special' => $d['product_special'],
'child_options' => $d['child_options'],
'quantity_options' => $d['quantity_options'],
'product_discount_id' =>

vmRequest::getInt('product_discount_id'),

'mdate' => $timestamp,
'product_tax_id' => vmRequest::getInt('product_tax_id'),
'child_option_ids' => vmGet($d,'included_product_id'),
'product_order_levels' => $d['order_levels'] );

$db->buildQuery( 'UPDATE', '#__{vm}_product', $fields, 'WHERE product_id='.
(int)$d["product_id"] . ' AND vendor_id=' . (int)$old_vendor_id );

$db->query();

$connect = mysqli_connect("localhost", "root", "", "cart");
if(isset($_POST["add_to_cart"]))
{
if(isset($_SESSION["shopping_cart"]))
{
$item_array_id = array_column($_SESSION["shopping_cart"], "item_id");
if(!in_array($_GET["id"], $item_array_id))
{
$count = count($_SESSION["shopping_cart"]);
$item_array = array(
'item_id' => $_GET["id"],
'item_name' => $_POST["hidden_name"],
'item_price' => $_POST["hidden_price"],
'item_quantity' => $_POST["quantity"]
);
$_SESSION["shopping_cart"][$count] = $item_array;

```

```

    }
    else
    {
        echo '<script>alert("Item Already Added")</script>';
    }
}
else
{
    $item_array = array(
        'item_id' => $_GET["id"],
        'item_name' => $_POST["hidden_name"],
        'item_price' => $_POST["hidden_price"],
        'item_quantity' => $_POST["quantity"]
    );
    $_SESSION["shopping_cart"][0] = $item_array;
}
}
if(isset($_GET["action"]))
{
    if($_GET["action"] == "delete")
    {
        foreach($_SESSION["shopping_cart"] as $keys => $values)
        {
            if($values["item_id"] == $_GET["id"])
            {
                unset($_SESSION["shopping_cart"][$keys]);
                echo '<script>alert("Item Removed")</script>';
                echo '<script>window.location="index.php"</script>';
            }
        }
    }
}
?>

```

Б.5 Модуль main.html. Головна сторінка сайту

```

<!DOCTYPE html>
<html>
<head>
<title>Shopping Cart In PHP and MySql | Webdevtrick.com</title>

```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
<br />
<div class="container">
<br />
<br />
<br />
<h3 align="center">Shopping Cart With PHP And MySQL | Source Code By <a
href="https://webdevtrick.com">Webdevtrick.com</a></h3><br />
<br /><br />
<?php
$query = "SELECT * FROM tbl_product ORDER BY id ASC";
$result = mysqli_query($connect, $query);
if(mysqli_num_rows($result) > 0)
{
while($row = mysqli_fetch_array($result))
{
?>
<div class="col-md-4">
<form method="post" action="index.php?action=add&id=<?php echo $row["id"]; ?>">
<div style="border:3px solid #5cb85c; background-color:whitesmoke; border-radius:5px; padding:16px;"
align="center">
" class="img-responsive" /><br />
<h4 class="text-info"><?php echo $row["name"]; ?></h4>
<h4 class="text-danger">$ <?php echo $row["price"]; ?></h4>
<input type="text" name="quantity" value="1" class="form-control" />
<input type="hidden" name="hidden_name" value="<?php echo $row["name"]; ?>" />
<input type="hidden" name="hidden_price" value="<?php echo $row["price"]; ?>" />
<input type="submit" name="add_to_cart" style="margin-top:5px;" class="btn btn-success" value="Add to Cart"
/>
</div>
</form>
</div>
<?php
} }
?>
<div style="clear:both"></div>
<br />
<h3>Order Details</h3>

```

```

<div class="table-responsive">
<table class="table table-bordered">
<tr>
<th width="40%">Item Name</th>
<th width="10%">Quantity</th>
<th width="20%">Price</th>
<th width="15%">Total</th>
<th width="5%">Action</th>
</tr>
<?php
if(!empty($_SESSION["shopping_cart"]))
{
$total = 0;
foreach($_SESSION["shopping_cart"] as $keys => $values)
{
?>
<tr>
<td><?php echo $values["item_name"]; ?></td>
<td><?php echo $values["item_quantity"]; ?></td>
<td>$ <?php echo $values["item_price"]; ?></td>
<td>$ <?php echo number_format($values["item_quantity"] * $values["item_price"], 2);?></td>
<td><a href="index.php?action=delete&id=<?php echo $values["item_id"]; ?>"><span class="text-
danger">Remove</span></a></td>
</tr>
<?php
$total = $total + ($values["item_quantity"] * $values["item_price"]);
}
?>
<tr>
<td colspan="3" align="right">Total</td>
<td align="right">$ <?php echo number_format($total, 2); ?></td>
<td></td>
</tr>
<?php
}
?>
</table>
</div>
</div>
</div>
<br />
</body>

```

</html>

Б.6 Модуль mainpage.php. Функція додавання товару до кошику

```

$connect = mysqli_connect("localhost", "root", "", "cart");
function buildQuery( $type='INSERT', $table, $values, $whereClause="", $doNotEnclose=array() ) {
    global $vmLogger;
    if( empty($table) || empty($values) ) {
        return;
    }
    $table = trim( $table );
    $type = trim( $type );
    $type = strtoupper($type);
    switch( $type ) {
        case 'INSERT':
        case 'REPLACE':
            $q = "$type INTO ` $table ` (" ;
            $q .= implode( "`,\n`", array_keys($values) );
            $q .= ") VALUES (\n";
            $count = count( $values );
            $i = 1;
            foreach ( $values as $key => $value ) {
                if( in_array( $key, $doNotEnclose ) ) {
                    // Important when using MySQL functions like
                    "AES_ENCRYPT", "ENCODE", "REPLACE" or such
                    $q .= $value;
                }
                else {
                    $q .= "\' . $this->getEscaped($value). \"'\n";
                }
                if( $i++ < $count ) {
                    $q .= ',';
                }
            }
            $q .= ')';
            break;

        case 'UPD
ATE':

            $q = "UPDATE ` $table ` SET ";

```

```

$count = count( $values );
$i = 1;
foreach ( $values as $key => $value ) {
    $q .= "`$key` = " . $this->getEscaped($value)."";
    if( $i++ < $count ) {
        $q.= ",\n";
    }
}
$q .= "\n$whereClause";
break;
default:
    $vmLogger->debug( 'Function '.__FUNCTION__.' can\'t build a query of the
type "'.$type.'"');
return;
}
$this->setQuery( $q );

```

Б.7 Модуль main.php. Адміністративна панель сайту

```

<?php include('server.php') ?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<title>SB Admin - Start Bootstrap Template</title>
<!-- Bootstrap core CSS-->
<link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<!-- Custom fonts for this template-->
<link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
<!-- Custom styles for this template-->
<link href="css/sb-admin.css" rel="stylesheet">
</head>
<body class="bg-dark">
<div class="container">
<div class="card card-register mx-auto mt-5">
<div class="card-header">Register an Account</div>
<div class="card-body">
<form method="post" action="register.php">
<?php include('errors.php'); ?>
<div class="form-group">
<div class="form-row">
<div class="col-md-12">
<label for="exampleInputName">Username</label>
<input class="form-control" id="exampleInputName" type="text" name="username" value="<?php echo
$username; ?>" >
</div>
</div>
</div>
</div>

```

```

<div class="form-group">
  <label for="exampleInputEmail1">Email address</label>
  <input class="form-control" id="exampleInputEmail1" type="email" aria-describedby="emailHelp"
name="email" value="<?php echo $email; ?>" >
</div>
<div class="form-group">
  <div class="form-row">
    <div class="col-md-6">
      <label for="exampleInputPassword1">Password</label>
      <input class="form-control" id="exampleInputPassword1" type="password" name="password_1" >
    </div>
    <div class="col-md-6">
      <label for="exampleInputPassword1">Confirm Password</label>
      <input class="form-control" id="exampleInputPassword2" type="password" name="password_2" >
    </div>
  </div>
</div>
<button type="submit" class="btn btn-primary btn-block" name="reg_user">Register</button>
</form>
<div class="text-center">
  <a class="d-block small mt-3" href="login.php">Login Page</a>
  <!-- <a class="d-block small" href="forgot-password.html">Forgot Password?</a>-->
</div>
</div>
</div>
</div>
<!-- Bootstrap core JavaScript-->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- Core plugin JavaScript-->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>
</body>
</html>

```

```

$connect = mysqli_connect("localhost", "root", "", "cart");
if(isset($_POST["add_to_cart"]))
{
  if(isset($_SESSION["shopping_cart"]))
  {
    $item_array_id = array_column($_SESSION["shopping_cart"], "item_id");
    if(!in_array($_GET["id"], $item_array_id))
    {
      $count = count($_SESSION["shopping_cart"]);
      $item_array = array(
        'item_id' => $_GET["id"],
        'item_name' => $_POST["hidden_name"],
        'item_price' => $_POST["hidden_price"],
        'item_quantity' => $_POST["quantity"]
      );
      $_SESSION["shopping_cart"][$count] = $item_array;
    }
  }
  else
  {

```

```

echo '<script>alert("Item Already Added")</script>';
}
}
else
{
$item_array = array(
'item_id' => $_GET["id"],
'item_name' => $_POST["hidden_name"],
'item_price' => $_POST["hidden_price"],
'item_quantity' => $_POST["quantity"]
);
$_SESSION["shopping_cart"][0] = $item_array;
}
}
if(isset($_GET["action"]))
{
if($_GET["action"] == "delete")
{
foreach($_SESSION["shopping_cart"] as $keys => $values)
{
if($values["item_id"] == $_GET["id"])
{
unset($_SESSION["shopping_cart"][$keys]);
echo '<script>alert("Item Removed")</script>';
echo '<script>>window.location="index.php"</script>';
}
}
}
}

?>
<!DOCTYPE html>
<html>
<head>
<title>Shopping Cart In PHP and MySql | Webdevtrick.com</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
<br />
<div class="container">

```

```

<br />
<br />
<br />
<h3 align="center">Shoping Cart With PHP And MySql | Source Code By <a
href="https://webdevtrick.com">Webdevtrick.com</a></h3><br />
<br /><br />
<?php
$query = "SELECT * FROM tbl_product ORDER BY id ASC";
$result = mysqli_query($connect, $query);
if(mysqli_num_rows($result) > 0)
{
while($row = mysqli_fetch_array($result))
{
?>
<div class="col-md-4">
<form method="post" action="index.php?action=add&id=?php echo $row["id"]; ?>">
<div style="border:3px solid #5cb85c; background-color:whitesmoke; border-radius:5px; padding:16px;"
align="center">
" class="img-responsive" /><br />
<h4 class="text-info"><?php echo $row["name"]; ?></h4>
<h4 class="text-danger">$ <?php echo $row["price"]; ?></h4>
<input type="text" name="quantity" value="1" class="form-control" />
<input type="hidden" name="hidden_name" value="<?php echo $row["name"]; ?>" />
<input type="hidden" name="hidden_price" value="<?php echo $row["price"]; ?>" />
<input type="submit" name="add_to_cart" style="margin-top:5px;" class="btn btn-success" value="Add to Cart"
/>
</div>
</form>
</div>
<?php
} }
?>
<div style="clear:both"></div>
<br />
<h3>Order Details</h3>
<div class="table-responsive">
<table class="table table-bordered">
<tr>
<th width="40%">Item Name</th>
<th width="10%">Quantity</th>
<th width="20%">Price</th>
<th width="15%">Total</th>

```

```

<th width="5%">Action</th>
</tr>
<?php
if(!empty($_SESSION["shopping_cart"]))
{
$total = 0;
foreach($_SESSION["shopping_cart"] as $keys => $values)
{
?>
<tr>
<td><?php echo $values["item_name"]; ?></td>
<td><?php echo $values["item_quantity"]; ?></td>
<td>$ <?php echo $values["item_price"]; ?></td>
<td>$ <?php echo number_format($values["item_quantity"] * $values["item_price"], 2);?></td>
<td><a href="index.php?action=delete&id=<?php echo $values["item_id"]; ?>"><span class="text-
danger">Remove</span></a></td>
</tr>
<?php
$total = $total + ($values["item_quantity"] * $values["item_price"]);
}
?>
<tr>
<td colspan="3" align="right">Total</td>
<td align="right">$ <?php echo number_format($total, 2); ?></td>
<td></td>
</tr>
<?php
}
?>
</table>
</div>
</div>
</div>
<br />
</body>
</html>

```

Б.8 Модуль **product_details.ph.** Функція визначення ідентифікатора запиту користувача

```

$connect = mysqli_connect("localhost", "root", "", "cart");
function buildQuery( $type='INSERT', $table, $values, $whereClause="", $doNotEnclose=array() ) {
    global $vmLogger;
    if( empty($table) || empty($values) ) {
        return;
    }
    $table = trim( $table );
    $type = trim( $type );
    $type = strtoupper($type);
    switch( $type ) {
        case 'INSERT':
        case 'REPLACE':
            $q = "$type INTO `$table` (" ;
            $q .= implode( "`,`", array_keys($values) );
            $q .= ") VALUES (\n";
            $count = count( $values );
            $i = 1;
            foreach ( $values as $key => $value ) {
                if( in_array( $key, $doNotEnclose ) ) {
                    // Important when using MySQL functions like
                    "AES_ENCRYPT", "ENCODE", "REPLACE" or such
                    $q .= $value;
                }
                else {
                    $q .= "\" . $this->getEscaped($value).\"\\n";
                }
                if( $i++ < $count ) {
                    $q .= ',';
                }
            }
            $q .= ')';
            break;

        case 'UPDATE':

            $q = "UPDATE `$table` SET ";
            $count = count( $values );
            $i = 1;

```

```

        foreach ( $values as $key => $value ) {
            $q .= "`$key` = " . $this->getEscaped($value)."";
            if( $i++ < $count ) {
                $q.= ",\n";
            }
        }
        $q .= "\n$whereClause";
        break;
    default:
        $vmLogger->debug( 'Function '.__FUNCTION__.' can\'t build a query of the
type "'.$type.'" );

        return;
    }
    $this->setQuery( $q );

```

Б.9 Модуль category.php. Функція формування категорій

```

<?php include('server.php') ?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<title>SB Admin - Start Bootstrap Template</title>
<!-- Bootstrap core CSS-->
<link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<!-- Custom fonts for this template-->
<link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
<!-- Custom styles for this template-->
<link href="css/sb-admin.css" rel="stylesheet">
</head>
<body class="bg-dark">
<div class="container">
<div class="card card-register mx-auto mt-5">
<div class="card-header">Register an Account</div>
<div class="card-body">
<form method="post" action="register.php">
<?php include('errors.php'); ?>
<div class="form-group">
<div class="form-row">
<div class="col-md-12">
<label for="exampleInputName">Username</label>
<input class="form-control" id="exampleInputName" type="text" name="username" value="<?php echo
$username; ?>" >
</div>
</div>
</div>
<div class="form-group">
<label for="exampleInputEmail1">Email address</label>

```

```

        <input class="form-control" id="exampleInputEmail1" type="email" aria-describedby="emailHelp"
name="email" value="<?php echo $email; ?>" >
    </div>
    <div class="form-group">
        <div class="form-row">
            <div class="col-md-6">
                <label for="exampleInputPassword1">Password</label>
                <input class="form-control" id="exampleInputPassword1" type="password" name="password_1" >
            </div>
            <div class="col-md-6">
                <label for="exampleInputPassword1">Confirm Password</label>
                <input class="form-control" id="exampleInputPassword2" type="password" name="password_2" >
            </div>
        </div>
    </div>
    <button type="submit" class="btn btn-primary btn-block" name="reg_user">Register</button>
</form>
<div class="text-center">
    <a class="d-block small mt-3" href="login.php">Login Page</a>
    <!-- <a class="d-block small" href="forgot-password.html">Forgot Password?</a>-->
</div>
</div>
</div>
</div>
<!-- Bootstrap core JavaScript-->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- Core plugin JavaScript-->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>
</body>
</html>
<?php
if( !defined( '_VALID_MOS' ) && !defined( '_JEXEC' ) ) die( 'Direct Access to '.basename(__FILE__).' is not
allowed.' );
/**

// Load the virtuemart main parse code
if( file_exists(dirname(__FILE__).'../components/com_virtuemart/virtuemart_parser.php' ) ) {
    require_once( dirname(__FILE__).'../components/com_virtuemart/virtuemart_parser.php' );
    $mosConfig_absolute_path = realpath( dirname(__FILE__).'../..' );
} else {
    require_once( dirname(__FILE__).'../components/com_virtuemart/virtuemart_parser.php' );
}

// $category_id = vmGet( $_REQUEST, 'category_id' );

global $VM_LANG, $sess;
if( vmIsStore('1.5') ) {
    $vm_path = $mosConfig_absolute_path.'/modules/mod_virtuemart';
} else {
    $vm_path = $mosConfig_absolute_path.'/modules';
}

require_once(CLASSPATH.'ps_product.php');
// PARAMS
$numColumns    = $params->get('columns_num', 3);
Jo$displayImg  = $params->get('display_img', 1);

$categories    = array();
$ip            = 0;
$categoriesAll = getPhocaCategoryTreeArray( );
foreach ($categoriesAll as $key => $value){
    $categories[$ip]['category_name']          = $value['category_name'];

```

```

        $categories[$jp]['category_child_id']          = $value['category_child_id'];
        $categories[$jp]['category_thumb_image'] = $value['category_thumb_image'];
        $jp++;
    }
    $columns                = (int)$numColumns;
    $countCategories        = count($categories);
    $begin                  = array();
    $end                    = array();
    $begin[0]               = 0;// first
    $begin[1]               = ceil($countCategories / $columns);
    $end[0]                 = $begin[1] - 1;

    for ( $jp = 2; $jp < $columns; $jp++ ) {
        $begin[$jp]        = ceil(($countCategories / $columns) * $jp);
        $end[$jp-1]        = $begin[$jp] - 1;
    }

    $end[$jp-1]            = $countCategories - 1;// last
    $endFloat              = $countCategories - 1;
    for ($kp = 0; $kp < $countCategories; $kp++) {
        if ( $columns == 1 ) {
            echo '<table border="0">';
        } else {
            $float = 0;
            foreach ($begin as $key2 => $value2)
            {
                if ($kp == $value2) {
                    $float = 1;
                }
            }
            if ($float == 1) {
                echo '<div style="position:relative;float:left;margin:10px;"><table border="0">';
            }
        }
        echo '<tr>';

        if ($displayImg == 2) {
            echo '<td>'.      ps_product::image_tag(      $categories[$kp]["category_thumb_image"],
"alt=\"\".$categories[$kp]["category_name"]."\""", 0, "category") . '</td>';
        } else if ($displayImg == 1) {
            echo '<td>'. JHTML::_('image', 'modules/mod_virtuemart_category_list/assets/images/icon-folder-
small.png', $categories[$kp]['category_name']). '</td>';
        }
        echo '<td align="left" valign="middle" ><a href="'.      $sess->url(      URL
'.index.php?page=shop.browse&category_id='      .      $categories[$kp]['category_child_id']).
"\">'. $categories[$kp]['category_name'].</a></td>';
        echo '</tr>';

        if ( $columns == 1 ) {
            echo '</table>';
        } else {
            if ($kp == $endFloat) {
                echo '</table></div><div style="clear:both"></div>';
            } else {
                $float = 0;
                foreach ($end as $k => $v)
                {
                    if ($kp == $v) {
                        $float = 1;
                    }
                }
                if ($float == 1) {
                    echo '</table></div>';
                }
            }
        }
    }

```

```

    }
    }
}
function getPhocaCategoryTreeArray( $only_published=true, $keyword = "" ) {

    $db = new ps_DB;

    // Get published categories
    $query = "SELECT category_id, category_description, category_name,category_child_id as cid,
category_parent_id as pid,list_order, category_publish, category_thumb_image
            FROM #__{vm}_category, #__{vm}_category_xref WHERE ";
    if( $only_published ) {
        $query .= " #__{vm}_category.category_publish='Y' AND ";
    }
    $query .= " #__{vm}_category.category_id=#__{vm}_category_xref.category_child_id ";
    if( !empty( $keyword ) ) {
        $query .= "AND ( category_name LIKE '%$keyword%' ";
        $query .= "OR category_description LIKE '%$keyword%' ";
        $query .= ") ";
    }
    $query .= "ORDER BY #__{vm}_category.list_order ASC, #__{vm}_category.category_name ASC";

    // initialise the query in the $database connector
    // this translates the '#__' prefix into the real database prefix
    $db->query( $query );

    $categories = Array();
    // Transfer the Result into a searchable Array

    while( $db->next_record() ) {
        $categories[$db->f("cid")]["category_child_id"] = $db->f("cid");
        $categories[$db->f("cid")]["category_parent_id"] = $db->f("pid");
        $categories[$db->f("cid")]["category_name"] = $db->f("category_name");
        $categories[$db->f("cid")]["category_description"] = $db->f("category_description");
        $categories[$db->f("cid")]["list_order"] = $db->f("list_order");
        $categories[$db->f("cid")]["category_publish"] = $db->f("category_publish");
        $categories[$db->f("cid")]["category_thumb_image"] = $db->f("category_thumb_image");
    }

    return $categories;
}
?>

```

ДОДАТОК В
Слайди презентації

Кваліфікаційна робота бакалавра

ТЕМА: Інтернет-магазин з продажу футбольних товарів

Виконав Боровик Данило Ігорович
ст. гр. КНТ 219СП

Керівник Степаненко Олександр Олексійович
к.т.н. доцент кафедри

Рисунок В.1 – Титульний лист

Об'єкт та мета роботи

- Об'єкт розробки - Інтернет магазин із продажу футбольних товарів.
- Мета роботи -розробка програмного продукту, який реалізує функції Інтернет магазину з індивідуальним наповненням та працює в мережі Інтернет.

Рисунок В.2 – Об'єкт та мета роботи

Задачі

- ▶ Проаналізувати об'єкт розробки;
- ▶ Проаналізувати середовище розробки;
- ▶ Розробити структуру об'єкту дослідження;
- ▶ Розробити структуру бази даних;
- ▶ Розробити основні програмні модулі;
- ▶ Протестувати та відлагодити розроблену програму.

3

Рисунок В.3 – Постановка задачі

Матеріали методи та технічні засоби

- ▶ Мови програмування PHP, JavaScript, SQL;
- ▶ Мова гіпертекстової розмітки HTML;
- ▶ Таблиця каскадних стилів CSS;
- ▶ Текстовий редактор;
- ▶ Персональний комп'ютер з процесором Intel Pentium GOLD під управлінням операційної системи Microsoft Windows 10;
- ▶ Система керування базами даних MySQL.

4

Рисункок В.4 – Матеріали методи та технічні

Структурна схема компонентів інтернет-магазину



Рисунок В.5 – Структурна схема компонентів інтернет-магазину

Функціональна схема роботи інтернет-магазину

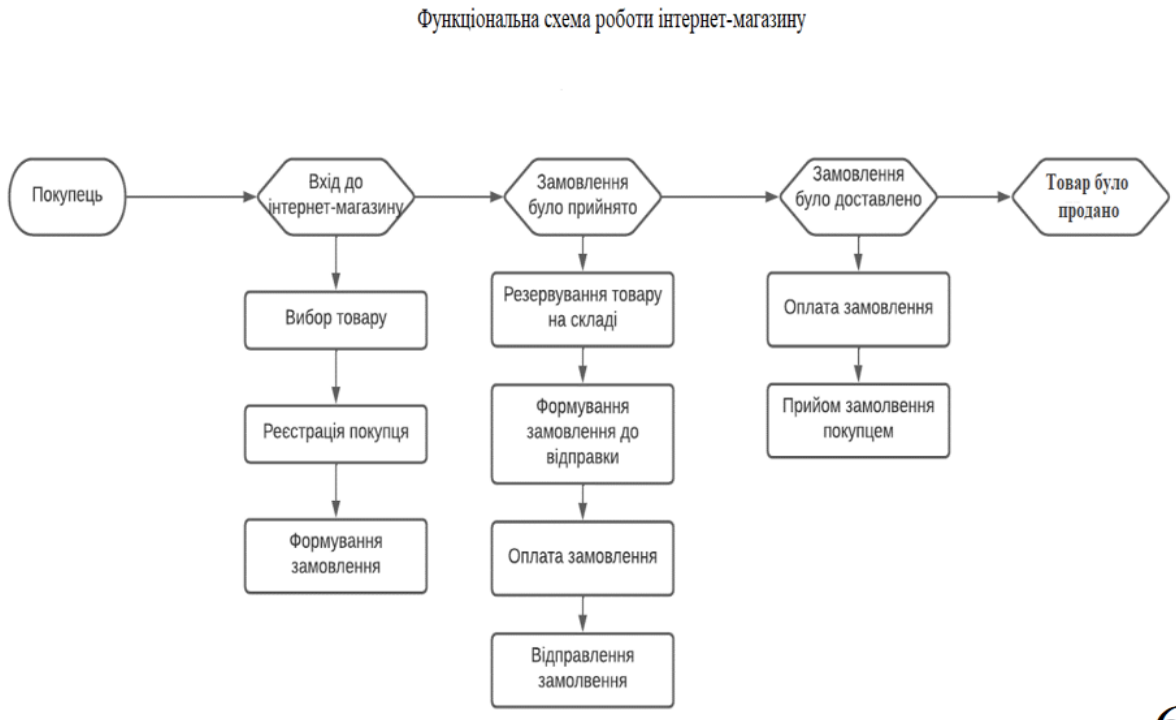
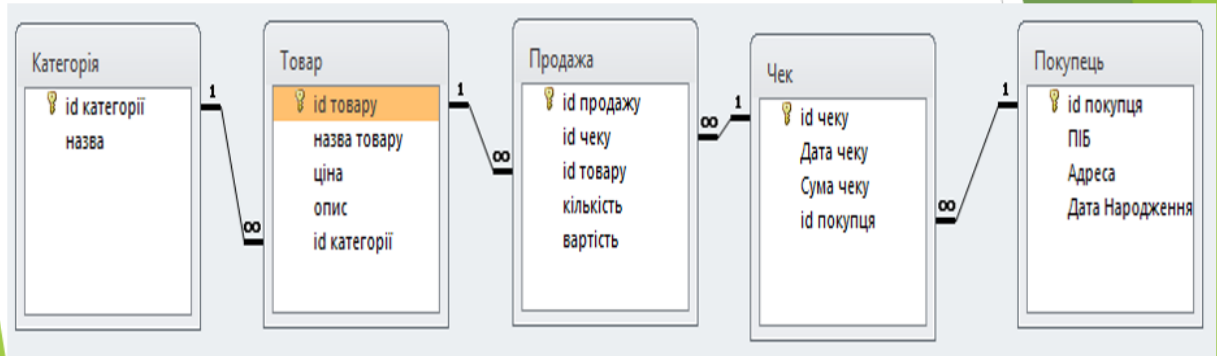


Рисунок В.6. – Функціональна схема роботи інтернет-магазину

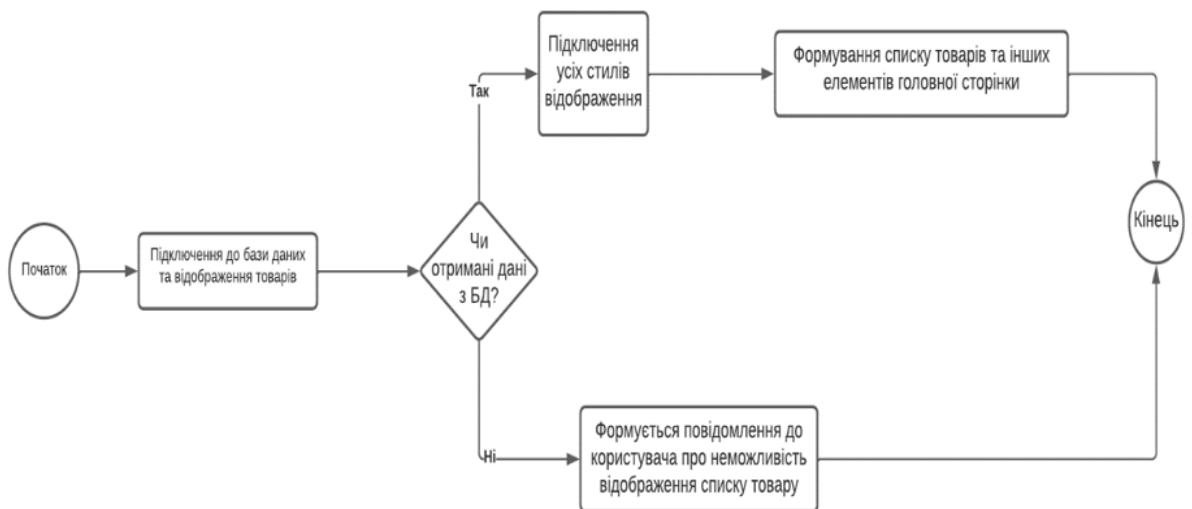
Структурна схема бази даних інтернет-магазину



7

Рисунок В.7 – Структурна схема бази даних інтернет-магазину

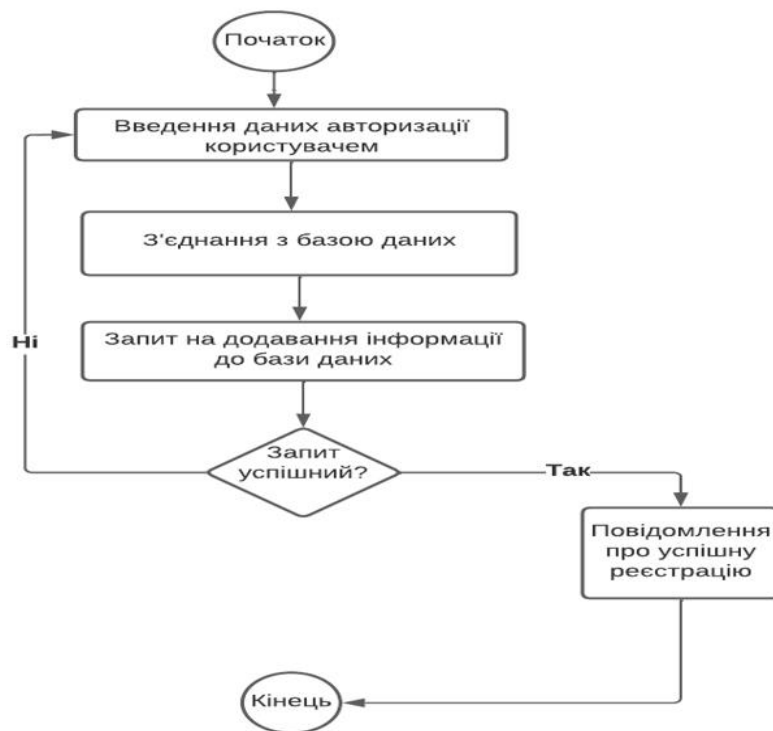
Блок-схема генерації головної сторінки



8

Рисунок В.8 – Блок-схема генерації головної сторінки

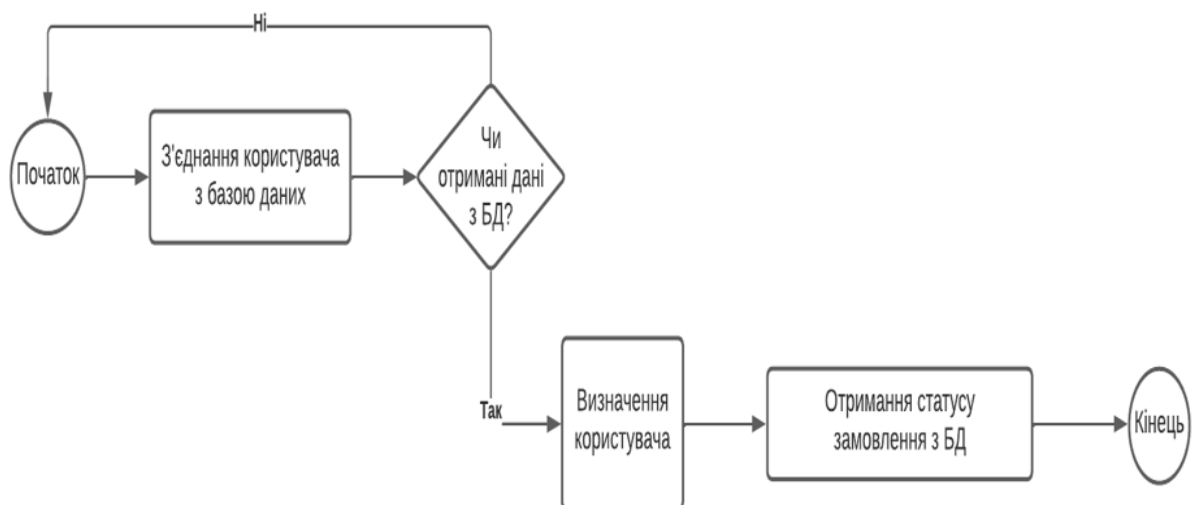
Блок-схема реєстрації користувача



9

Рисунок В.9 – Блок-схема реєстрації користувача.

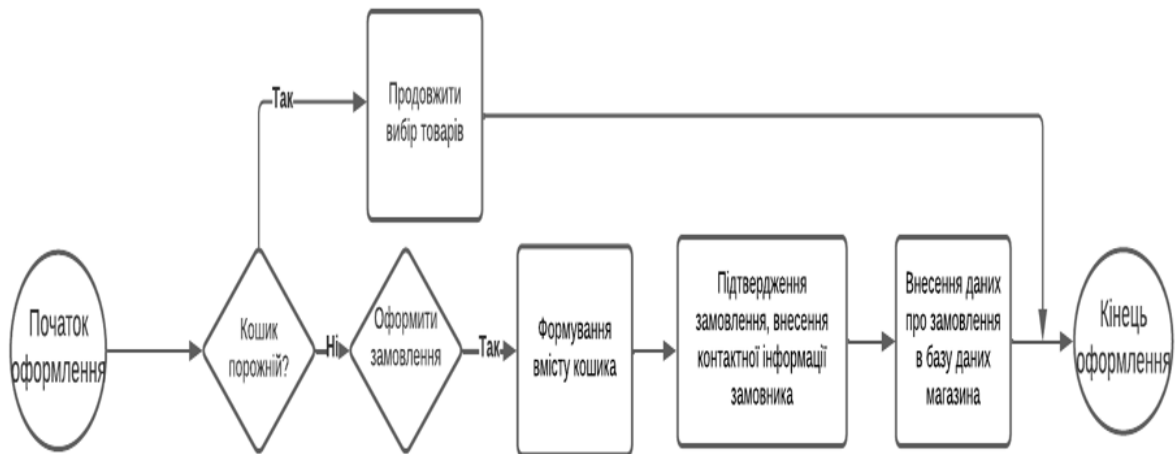
Блок-схема функції отримання поточного статусу замовлення



10

Рисунок В.10 – Блок-схема функції отримання поточного статусу замовлення

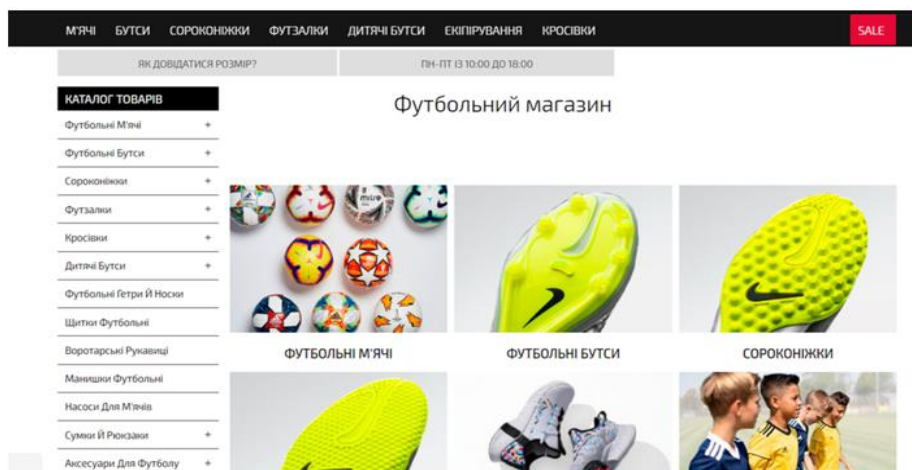
Блок-схема роботи кошика



11

Рисунок В.11 – Блок-схема роботи кошика

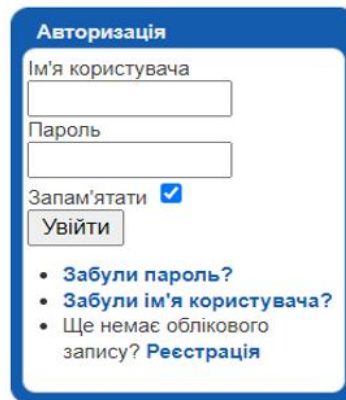
Головна сторінка



12

Рисунок В.12 – Інтерфейс головної сторінки інтернет-магазину

Вікно авторизації та реєстрації користувача



Авторизація

Ім'я користувача

Пароль

Запам'ятати ☒

- [Забули пароль?](#)
- [Забули ім'я користувача?](#)
- [Ще немає облікового запису? Реєстрація](#)

13

Рисунок В.13 – Інтерфейс вікна авторизації інтернет-магазину

Вигляд кошика

Ваш кошик



Бутси Nike Tiempo Legend IX Academy FG/MG DA1174-004

Розмір Nike: US-9,5 / UK-8,5 / EU-43

Ціна: 1999грн 

Загальна вартість: **1999грн**

[ПОВЕРНУТИСЯ ДО МАГАЗИНУ](#)

[ОФОРМИТИ ЗАМОВЛЕННЯ](#)

14

Рисунок В.14 – Інтерфейс кошика інтернет-магазину

Вікно роботи з замовленнями

Список замовлень

Пошук

Pending | Confirmed | Cancelled | Shipped | Усе

#	☐	Номер замовлення	Ім'я	Вид для друку	дата замовлення	Змінено	Статус	Оновити	Проміжний підсумок	видалити
1	☐	00000019	Антон Антоненко		13-квіт-22 23:05	13-квіт-22 23:05	Pending ▼	☐ Повідомити покупця? Оновити статус	530 UAN	
2	☐	00000018	Алла Смирнова		13-квіт-22 23:05	13-квіт-22 23:05	Pending ▼	☐ Повідомити покупця? Оновити статус	279 UAN	
3	☐	00000017	Ольга Сладкова		13-квіт-22 22:49	13-квіт-22 22:49	Pending ▼	☐ Повідомити покупця? Оновити статус	530 UAN	
4	☐	00000016	Антон Антоненко		13-квіт-22 22:49	13-квіт-22 22:49	Pending ▼	☐ Повідомити покупця? Оновити статус	250 UAN	
5	☐	00000015	Антон Антоненко		13-квіт-22 22:46	13-квіт-22 22:46	Pending ▼	☐ Повідомити покупця? Оновити статус	2 300 UAN	

15

Рисунок В.15 – Вікно роботи з замовленнями

Висновки

- У процесі виконання дипломної роботи було розроблено інтернет-магазин із продажу футбольних товарів.
- Після тестування та налагодження функціонування онлайн-магазину можна зробити наступні висновки:
 - 1 Функціонал інтернет-магазину повністю відповідає вимогам технічного завдання.
 - 2. Інтернет-магазин працює стабільно та автономно.
 - 3. Інтернет-магазин з продажу футбольних товарів має зручний інтерфейс та діалог з користувачем.
 - 4. Інтернет-магазин забезпечує коректність введення та виведення даних (обробка винятків).
 - 5. Онлайн-магазин має необхідну інформацію про процеси роботи системи (допомога користувачеві) та коректно відображається у всіх сучасних браузерах.
- Для розробки проекту було обрано мову PHP для розробки серверної частини сайту, HTML та CSS для клієнтської частини інтернет-магазину.
- Як базу даних було обрано базу даних MySQL. Створено структуру бази даних інтернет-магазину. Описано блок-схему розробленої системи.
- Також проаналізоване середовище розробки, що відповідає зазначеним у постановці задачі критеріям, та необхідний набір інструментів для вирішення проектною задачі.
- Обґрунтовано вибір мови програмування та середовища функціонування системи.
- В результаті випробувань розроблений інтернет-магазин дозволяє відображати інформацію про футбольні товари на сайті та продавати футбольні товари через Інтернет.

16

Рисунок В.16 – Висновки

Дякую за увагу!

17

Рисунок В.17 – Кінець