

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

РЕАЛІЗАЦІЯ ОНЛАЙНОВОГО МАГАЗИНУ ПО ПРОДАЖУ ВЗУТТЯ

**Текстова частина до курсової роботи
за спеціальністю „Комп’ютерні науки”**

Керівник курсової роботи
к.ф.-м.н, ст. викладач Ющенко Ю.О.

“ ” _____ 2020 р.

Виконав студент
Мацюк В.О.
“ ” _____ 2020 р.

Київ 2020

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,

доцент, к.ф.-м.н.

_____ С. С. Гороховський
(підпис)

“ ____ ” _____ 202_ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту _____ Мацюк Вікторії _____

_____ 3 _____ курсу факультету інформатики

ТЕМА: _____ Реалізація онлайн магазину по продажу взуття _____

Вихідні дані: Онлайн магазин, розроблений на мові PHP, що взаємодіє з СКБД MySQL через веб-інтерфейс для адміністрування phpMyAdmin.

Індивідуальне завдання

Вступ

1. Аналіз предметної області. Постановка завдання курсової роботи

2. Теоретичні відомості

3. Опис реалізації програмного продукту

Висновки

Список використаних джерел

Додатки (за необхідністю)

Дата видачі „ ____ ” _____ 2020 р. Керівник _____

к.ф.-м.н, ст. викладач _____ Ющенко Ю.О. _____

Завдання отримала _____

Тема: Реалізація онлайн-магазину по продажу взуття

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи	09.10.2019	
2.	Огляд літератури за темою роботи	30.11.2019	
3.	Аналіз аналогічних онлайн-магазинів	15.01.2020	
3.	Розробка технічного завдання та алгоритму роботи онлайн-магазину	20.02.2020	
4.	Створення бази даних	01.03.2020	
5.	Програмування розробленого алгоритму	15.03.2020	
6.	Тестування розробленого онлайн-магазину	06.04.2020	
7.	Написання пояснювальної роботи	08.04.2020	
8.	Подання першої версії пояснювальної роботи науковому керівнику	10.05.2020	
9.	Корегування роботи згідно зауважень наукового керівника	10.05.2020	
10.	Створення презентації	10.05.2020	
11.	Захист курсової роботи	18.05.2020	

Студент _____ Мацюк В.О. _____

Керівник _____ Ющенко Ю.О. _____

“ _____ ” _____ 2020 р.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1: Аналіз предметної області.....	8
1.1 Обґрунтування актуальності задачі	8
1.2 Огляд існуючих веб-магазинів	8
1.3 Опис вимог до розробки веб-магазину	10
РОЗДІЛ 2: Теоретичні відомості	12
2.1. Види веб-сайтів	12
2.2 Бази даних та СКБД.....	13
2.3 Переваги обраної СКБД	14
РОЗДІЛ 3: Опис реалізації.....	16
3.1 Опис технічного завдання на розробку	16
3.2 Обґрунтування обраних засобів розробки.....	18
3.3 Опис реалізації	23
3.4 Опис реляційної моделі	23
3.5 Розробка головної програми	28
3.6 Тестування програми і результати її виконання.....	33
Висновки	37
Список використаної літератури	38

ВСТУП

Актуальність, наукове та практичне значення обраної теми

Головною тенденцією в розвитку інформаційних технологій на сьогодні є розвиток Інтернету, вдосконалення його можливостей задля задоволення потреб користувачів зокрема шляхом створення великої кількості різноманітних користувацьких веб-застосунків з використанням новітніх засобів розробки веб-застосунків. Широкого використання можливостей Інтернету набула торгівельна сфера - продажі онлайн нищівно забирають долю ринку «наземних» продаж.

Уже не потрібно бігати по місту в пошуках потрібного розміру чи годинами їхати громадським транспортом, щоб повернути товар, що не підійшов. За допомогою пари кліків ми можемо обрати, оплатити та отримати бажану річ. Але не тільки економія часу та великий вибір є плюсами онлайн-покупок. В період пандемії, що охопила наш світ, саме інтернет-продажі вийшли на перший план, і захопили навіть тих людей, що були скептично налаштовані щодо сучасних технологій.

За статистикою^[1] частка електронної комерції в Україні невідмінно зростає. Тільки в період з січня по листопад 2019 року фізичних товарів та послуг було продано на 76 млрд доларів - що на 17% більше, ніж за аналогічний період 2018 року. Серед найпопулярніших товарів – одяг та взуття, техніка та електроніка, побутові товари.

За оцінками експертів групи EVO^[2] в 2020 році очікують ріст ринку електронної комерції приблизно на 15% - до 87,2 млрд доларів.

Отож, ринок збільшується, і підприємці з кожним днем розуміють необхідність у онлайн маркетплейсі. Ідеальний онлайн-магазин не тільки розрахований на покупця, але й створює зручну систему керування для його власника – доступ до редагування онлайн вітрини (можливість додати чи видалити товар, замінити вже наявну інформацію, тощо), перегляд переліку

замовлень (можливість змінити стан замовлення, перегляд контактної інформації клієнта), або ж зворотній зв'язок з покупцями.

Мета та завдання курсової роботи

Мета: Створити зручний веб-сервіс з продажу взуття з можливістю розподілення ролей: клієнт та менеджер магазину.

Завдання: Розробити онлайн - магазин, створити систему керування онлайн-магазином для персоналу магазину, створити базу даних актуальних товарів, створити ресурс перегляду історії замовлень на особистій сторінці клієнта, та зручний засіб вибору товарів покупцями.

Об'єкт дослідження

Існуючі онлайн - магазини з продажу взуття, та інших товарів широкого вжитку

Використане програмне забезпечення

Розробка онлайн - магазину та системи керування здійснюватиметься за допомогою веб-мов програмувань на основі клієнт-серверної архітектури, створення бази даних буде впроваджено за допомогою веб-додатку phpMyAdmin для адміністрування реляційної СКБД(MySQL), обмін інформацією на сервісі здійснюватиметься за допомогою AJAX з PHP, перегляд - JQuery.

Джерела дослідження

Курсова робота зроблена за допомогою ретельного аналізу та дослідженні багатьох сторін веб-програмування. Тож, було використано у навчальних цілях й опрацьовано багато наукових інтернет ресурсів, пов'язаних з темою роботи щодо веб-магазинів, та для аналізу і порівняння наявних магазинів у мережі Інтернет.

Структура роботи

Робота складається трьох розділів.

У першому розділі детально представлено огляд теми, та обґрунтовано її актуальність. Також в розділі проведено аналіз вже існуючих в світі аналогів. Розділ завершується постановкою задач для дослідження та розробки.

У другому розділі надані теоретичні відомості щодо обраної теми, зокрема наведено класифікацію веб-сайтів. Розділ також присвячено основним засобам розробки веб-магазинів, різним типам баз даних та системам керування базами даних. Обґрунтовано обрану систему керування базами даних та надана теоретична інформація про неї.

Третій розділ присвячено обраним методам реалізації програмного продукту. Розділ надає формулювання чіткого технічного завдання, опис функцій користувачів продуктом. Також обґрунтовано обрані методи розробки. Опис певних модулів програми та фрагменти коду. В пункті 3.6 проведено тестування програмного продукту.

РОЗДІЛ 1: Аналіз предметної області

1.1 Обґрунтування актуальності задачі

На сьогоднішній день, всі підприємці великого та малого бізнесу розуміють важливість digital-сфери. За дослідженнями Дослідницького холдингу Factum Group Ukraine^[3] у III кварталі 2019 року 71% українців регулярно використовує Інтернет. Може здатись, що це не такий високий відсоток користувачів, але варто зазначити, що у I кварталі того ж року статистика надавала всього лише 64% серед всього населення. Кількість користувачів росте, і все більше людей розуміють переваги онлайн-купівлі. Рано чи пізно для кожного підприємця настане актуальне питання в створенні особистого онлайн – магазину. Тому дана робота є не тільки актуальною, а й тою, якою можна скористатись в комерційних цілях.

1.2 Огляд існуючих веб-магазинів

Intertop^[4] – одна з найвідоміших компаній з продажу взуття в Україні. Станом на жовтень 2019 року під управлінням компанії знаходяться 136 магазинів у 27 містах України. Окрім цього, компанія стрімко розвиває напрямок електронної комерції, запустивши у 2014 році онлайн - версію. Успіхи цього рішення яскраво демонструють цифри: 2,5 млн відвідувань – саме такий середньомісячний трафік магазину у 2018 році, а продажі в порівнянні з 2017 роком зросли більш, ніж у два рази. Розробкою та постійним вдосконаленням онлайн – магазину займається створений компанією IT-підрозділ.

Перше, що побачить навіть неосвідчена в інтернет-розробці людина, - це сучасний та зручний інтерфейс сайту. Користувач може обрати мову інтерфейсу – російська чи українська. Розробники одразу пропонують нам перейти до вибору товарів – або за допомогою навігаційної панелі, або через клік на блоки наявних на сайті категорій. Каталог взуття ділиться на три основні підрозділи – чоловіче, жіноче та дитяче. Кожен підрозділ додатково містить більше десятка підкатегорій вже самого товару (туфлі, черевики, кросівки, тощо). Користувач

може створити обліковий запис для перегляду історії покупок та накопичення бонусів. Великий плюс сайту – це зручна панель сортування та фільтрації товарів.

Але навіть в таких розкручених та, на перший погляд, ідеальних сайтів є недоліки. Товар можливо оформити тільки з його сторінки, що для користувачів, які вже давно обрали товар, збільшує час покупки. А як вже знаємо, що найцінніше в е-комерції – це час. Вирішенням цієї проблеми може стати додання кнопки в каталозі. Але найголовнішим недоліком є можливість покупки тільки користувачам з обліковим записом. Це не зовсім зручно для клієнтів, які розуміють, що це їх єдине замовлення в цьому магазині(наприклад, іноземці), або ж для людей, які не хочуть залишати свої дані в системі. Можливо, компанії варто переглянути свою політику заради зручності клієнта.

Ni!Legs^[5] – український бренд чоловічого та жіночого взуття, створений у 2018 році. Концепція бренду полягає в розумному споживанні та благодійним напрямком – кожен з охочих може віддати свою непотрібну пару взуття та отримати 25% знижки на нову пару від бренду. Зібране взуття реставрується та передається людям, яким воно необхідне.

Інтерфейс сайту інтуїтивно-зрозумілий та мінімалістичний. Окрім, каталогу, на панелі швидкого доступу є посилання для заповнення анкети на безкоштовну пару, а також контакти магазину. Користувач може відсортувати новинки, а також відфільтрувати моделі за кольором та стилем.

На жаль, на сайті магазину є багато недоліків. Найголовніший – це неможливість зареєструватись користувачу – це не тільки спрощує оформлення замовлення для постійних клієнтів, а й є великим плюсом для компанії, адже на наявний електронний адрес можна робити розсилки з новинками та знижками, що буде заохочувати користувача. Ще один недолік – це неможливість швидко повернутись до початку сторінки. Оскільки каталог не ділиться на сторінки, то клієнту під час перегляду може бути потрібно повернутись до меню, але, на жаль,

сайт не передбачає такої можливості. Способом вирішення є створення клавії швидкого автоповернення вгору.

Vitchiznyane-vzuttya^[6] – невеликий магазин взуття, родом з Вінниці. Є прикладом того, як малий бізнес також опановує Інтернет-простір. Сайт виконаний за допомогою безкоштовного онлайн – конструктора Webnode. На сайті є можливість придбати чоловіче, жіноче та дитяче, а також ортопедичне взуття.

Візуально онлайн-магазин програє тим, що порівнювались вище. Простий, та місцями незрозумілий функціонал сайту. Є можливість реєстрації користувача за допомогою електронної адреси, але створивши акаунт, покупець не може скористатись ані функцією швидкого замовлення, ані переглянути історію своїх замовлень. Каталог не має широкого спектру функціоналу – ані сортування, ані пошуку по назві. Користувач не може додати товар в корзину з сторінки каталогу, але й у самій корзині легко заплутатись. Розробники чомусь вирішили спочатку вивести рекомендовані товари, а вже в самому кінці - замовлення. Окрім цього, під час оформлення покупки, клієнт не може залишити коментар, що може значно спростити роботу як і менеджеру магазину, так і покупцю. Спосіб подолання цих проблем один – постійна підтримка сайту та його тестування.

Чи не найбільший недолік – відсутність програми лояльності. Маркетологами доведено^[7], що бонусна система часто позитивно впливає на прибуток – вдало проведена інформаційна кампанія може залучити нових клієнтів, та заохотити нових. Часто навіть кешбек у 3-5% може стати ключевим у рішенні покупця обрати той чи інший магазин.

1.3 Опис вимог до розробки веб-магазину

Обрано наступні функції, які мають бути реалізовані у веб-магазині:

1. Розробити онлайн-магазин, що буде відповідати сучасним вимогам: мати зручний та інтуїтивно зрозумілий інтерфейс з приємною та наочною візуалізацією товарів.
2. Впровадити гнучку систему різноманітних бонусів для постійних покупців.
3. Можливість покупки без реєстрації на сайті, а також забезпечення можливості додання коментарів до замовлення. Зручні можливості здійснення замовлень, включаючи обрання бажаного розміру та перегляду стану замовлення.
4. Можливість, за бажанням, реєстрації на сайті для перегляду історії замовлень та замовлення товарів, які зараз відсутні на складі.
5. Забезпечити якісний функціонал каталогу: можливість сортування, фільтрації, а також автопошуку товару.

РОЗДІЛ 2: Теоретичні відомості

2.1. Види веб-сайтів

Інтернет налічує мільйони, якщо навіть не мільярди сайтів. І кожного дня мільйони користувачів користуються соціальними мережами, замовляють послуги чи необхідні товари, шукають інформацію, читають, дивляться фільми чи просто грають ігри. І для того, щоб все це працювало кожного дня використовуються величезні потужності.

Сайт – це не просто статична сторінка, це динамічна анімація, набір форм, величезний функціонал клавiш, тощо. Найчастіше для розробки веб-сайтів використовують різні мови програмування. Ці програмні засоби, мають за основу клієнт - серверверну архітектуру, де клієнтом є браузер користувача, а сервером - веб-сервер. Дані зберігаються на сервері, а обмін інформацією відбувається по мережі. Головною перевагою є кросплатформенність – іншими словами, клієнти не залежать від операційної системи користувача та версії браузера.

Офіційної класифікації сайтів немає, але загалом можна класифікувати сайти за метою створення^[8]:

1. Корпоративний сайт – своєрідний довідник певної компанії, для надання про неї інформації: контакти, сфера діяльності та інтереси, перелік послуг або товарів. Є обов'язковим атрибутом ведення сучасного бізнесу.
2. Промо-сайти – організовує рекламну кампанію для певного товару/послуги/заходу. Просуває та підвищує популярність певного бренду.
3. Сайт-вітрина – сайт для демонстрації каталогу продуктів чи послуг, що надає компанія. Добре структурована, без зайвої інформації.
4. Онлайн-магазин – сайт, що працює схожим чином з сайтом-вітриною, однак, надає користувачу можливість не тільки переглянути, але й замовити, придбати чи повернути напряму через додаток.

5. Сайт-форум - зазвичай створюється на піддомені основного сайту і є інтерактивним ресурсом для спілкування людей. Форуми – це сайти з самогенерованим контентом, тобто користувачі самі створюють контент. Широко розповсюджені та є платформою для обміну інформацією між користувачами.

Отже, було переглянуто основні види сайтів, з чого робимо висновок, що для створення онлайн ресурсу для продажу взуття між сайтом-вітриною та онлайн-магазином, було обрано останнє в зв'язку з розширеним функціоналом для користувача. Оскільки, клієнт-серверна архітектура пов'язана з роботою на сервері, в наступному підрозділі буде детальніше розглянуто бази даних та проведений аналіз існуючих типів систем керування базами даних(СКБД).

2.2 Бази даних та СКБД

База даних(БД)^[9] – набір логічно взаємопов'язаних даних та їх описів, що використовується спільно та призначений для задоволення інформаційних потреб користувачів. По суті, база даних – це впорядкований набір даних, наприклад, бібліотечний формуляр чи медична картка пацієнта.

Класифікація баз даних відбувається по різним параметрам. За моделлю організації даних БД класифікують на три основні види:

1. Реляційна модель даних (relational data model) – дані представлені у вигляді таблиць(реляцій).
2. Мережна модель даних (network data model) – дані представлені в вигляді колекцій записів, а зв'язки – у вигляді наборів. Іншими словами, це дерево, що складається з об'єктів різних рівнів і зв'язки представлені, як «предок-нащадок».
3. Ієрархічна модель даних (hierarchical data model) – є підвидом мережної моделі даних, але в випадку з ієрархічною моделлю об'єкт може мати тільки одного предка.

Більшість сучасних інформаційних систем використовують реляційну модель, оскільки вона є більш незалежною від даних, на відміну двох інших.

Тому для роботи була обрана саме реляційна модель, в зв'язку з розповсюдженістю СКБД та легкістю в використанні.

Для забезпечення роботи з базами даних використовуються системи керування базами даних(СКБД). СКБД^[9] – це програмне забезпечення, за допомогою якого користувачі можуть визначати, створювати та підтримувати базу даних, а також здійснювати контрольований доступ до неї.

По суті, СКБД – це система, що приймає запит користувача (в випадку з даною роботою – SQL-запит), інтерпретує його, виконує відповідні дії і відповідає на запит. Це складна система, що забезпечує відповідні функції(реляційні бази даних):

1. Дозволяє створити базу даних (реляції, відношення, обмеження), що здійснюється зазвичай за допомогою мови визначення даних – DDL(Data Definition Language). Оскільки в даній роботі мовою DDL є SQL, то він забезпечує виконання трьома функціями – «CREATE»(створити), «ALTER»(змінити) та «DROP»(видалити).
2. Дозволяє додавання, оновлення, видалення та вибірку інформації з бази даних, що зазвичай здійснюється за допомогою мови маніпулювання даними – DML (Data Manipulation Language). Оскільки в даній роботі мовою DDL є SQL, то він забезпечує виконання чотирьома функціями – «SELECT»(вибрати), «INSERT»(вставити), «UPDATE»(оновити) та «DELETE»(видалити).
3. Надає контрольований доступ до бази даних за допомогою систем забезпечення захисту, відновлення, керування паралельною роботою та підтримки цілісності даних.

2.3 Переваги обраної СКБД

В наш час існує безліч безліч реалізацій СКБД, найвідоміші з них – Oracle, Microsoft SQL Server, PostgreSQL, Microsoft Access та MySQL. Для проекту було обрано СКБД з відкритим програмним кодом – MySQL. MySQL – це

багатопотокова система керування реляційними базами даних, заснована на клієнт-серверній архітектурі та надає доступ для великої кількості користувачів. Популярність цієї СКБД заснована на дружньому інтерфейсі, зручному користуванні в роботі з різноманітними веб-сайтами та мовами програмування, а також в великій кількості навчальних матеріалів, які допомагають легко та швидко освоїти СКБД. Серед переваг також^[10]:

1. Швидкість – тести, проведені компанією показують, що дана СКБД набагато швидше навіть лідера в цій галузі Oracle.
2. Надійність – компанія ставила за першочергову мету зробити СКБД безпечною. Тому MySQL підтримує транзакції, що знижують ризик втрати інформації, а також реплікацію та кластеризацію, які значно зменшують час простою у випадку збоїв сервера.
3. Розраховано на велику кількість користувачів – величезний плюс для розробників веб-застосунків, яким потрібно забезпечити великій кількості клієнтів одночасний доступ та використання бази даних.
4. Інтернаціоналізація – СКБД підтримує кодування Unicode та багато інших наборів символів, і при цьому дозволяє виконувати сортування, порівняння символів та збереження даних.
5. Широка підтримка застосунків – MySQL передбачує використання інтерфейсів програмування застосунків API (Application Programming Interfaces) на багатьох мовах програмування, серед них C, C++, Java, PHP, Perl, Python, Ruby, тощо.

Результатом шаленої популярності даної СКБД стала поява безлічі плагінів і розширень, що спрощують роботу, що в свою чергу додало ще більшої популярності серед розробників. Одним з таких розширень є веб-інтерфейс для адміністрування MySQL, яке було використане в даній роботі.

РОЗДІЛ 3: Опис реалізації

3.1 Опис технічного завдання на розробку

Технічне завдання має на меті розробку онлайн-магазину чоловічого та жіночого взуття. Проект розробляється під дві групи користувачів: покупців або потенційних клієнтів, та адміністратора магазину.

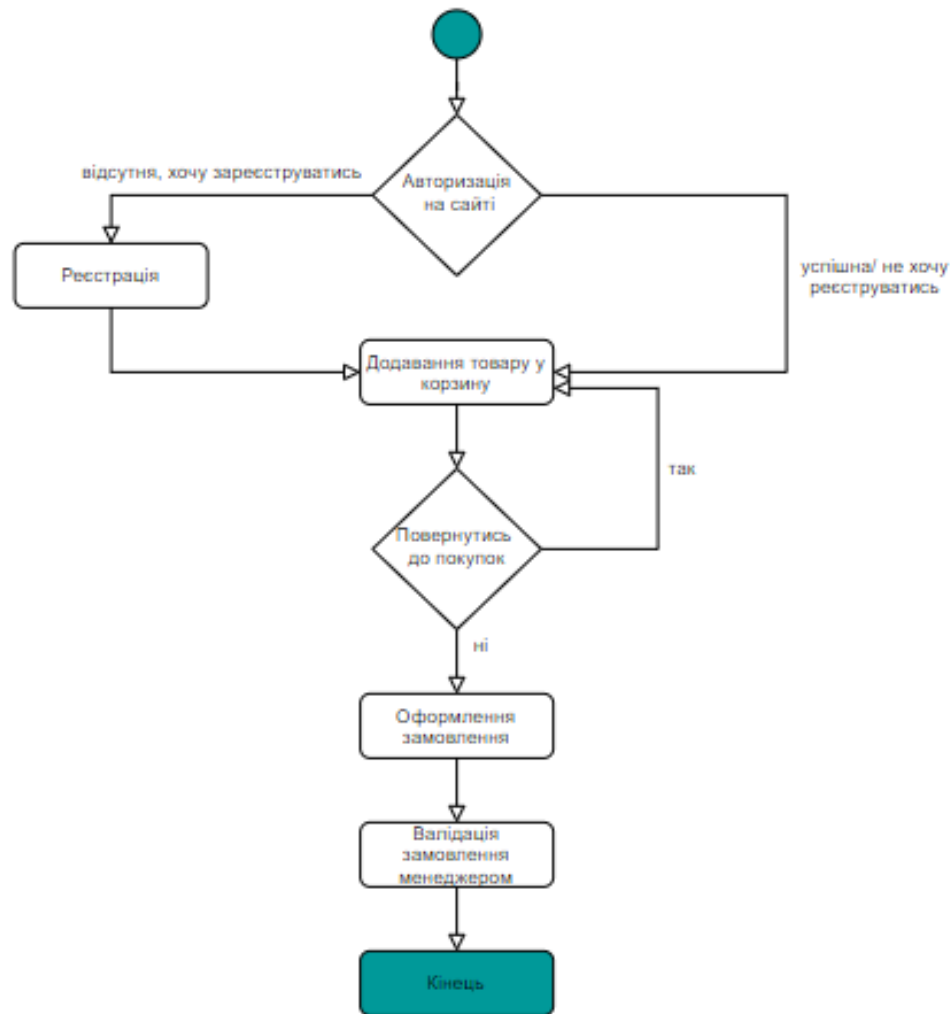
Адміністратор магазину повинен мати можливість вирішувати такі задачі:

- Управління товарами: додавати товар на сайт, редагувати інформацію по ньому та видаляти з бази.
- Управління категоріями товару: створення, редагування та видалення. В даному проекті є дві категорії – жіноче та чоловіче взуття, але в перспективі розширення магазину дана функція буде необхідною.
- Управління підкатегоріями товару: створення, редагування та видалення. Наприклад, в даному проекті – це туфлі, черевики, тощо.
- Управління замовленнями: перегляд історії замовлень магазину, обробка нових замовлень(в тому числі, редагування статусу - нове замовлення, обробляється, доставка, виконано) та видалення замовлення, якщо після обробки менеджером клієнтом замовлення більше неактуальне(відмова).

Визначення адміністратора магазину відбувається розробником в базі даних. При реєстрації кожному користувачу надається дефолтне значення user, і для визначення адміністратора необхідно змінити на admin.

Користувач сайту(покупець) має можливість вирішувати такі задачі:

- Перегляд каталогу продукції(детальний опис товару, перегляд рекомендованої продукції, пошук по товарам та категоріям та сортування).
- Оформлення замовлення за алгоритмом нижче:



- Реєстрація на сайті для створення особистого кабінету користувача з можливістю накопичення балів за замовлення та перегляду історії замовлень.
- Редагування зареєстрованому користувачу особистих даних.

В основі реалізації продукту – взаємодія онлайн-магазину з базою даних, яка зберігає дані про користувачів, їх замовлення, товари та їх категорії. Формування замовлення, реєстрація, авторизація, створення та редагування інформації відбувається до допомогою HTML form.

Для реєстрації необхідно ввести ім'я, e-mail(буде виконана валідація, якщо даний e-mail уже наявний в базі – користувачу буде показане повідомлення та запропоновано ввести інші дані) та пароль, який має бути більше шести символів.

Користувацький інтерфейс побудований інтуїтивно, тому не потребує особливих знань та навичок для користування. Спеціальні іконки, зображення,

зрозумілі назви посилань, зручне меню та функціонал, загальний дизайн та можливість пошуку та сортування в каталозі допомагають легко та швидко освоїти користування даним сайтом. Онлайн-магазин працює незалежно від операційної системи користувача або ж обраного браузера. Зауважимо, що необхідною умовою успішного користування є наявність підключення до Інтернет мережі.

3.2 Обґрунтування обраних засобів розробки

Розробка веб-сайтів залежить не тільки від знання мов веб-програмування, розуміння баз даних чи правильно обраному середовищі розробки. Хибно вважати, що тільки ці компоненти стають ключевими у розробці проекту. Дуже важливим є розуміння шаблонів проектування, знання яких значно пришвидшують написання коду та розробку бізнес-логіки завдання. Одним з найвідоміших шаблонів для створення динамічних веб-застосунків є MVC(Model, View, Controller). Це трирівневий шаблон проектування, який передбачає відділення бізнес-логіки від користувацького інтерфейсу для незалежної модифікації кожного з трьох рівнів^[11]:

1. Model – відповідальний за надання доступу для модифікації даних згідно запиту користувача. При чому, Model не контролює саму модифікацію і не має ніякої інформації про це. Зазвичай, у Model прописані методи, що звертаються до бази даних та готують дані для передачі іншим компонентам MVC.
2. View – відповідальний за представлення відповідних даних користувачеві(рівень, в якому генерується і відображається HTML-код). По суті, це взаємодія користувача з функціоналом застосунку.
3. Controller – відповідальний за обробку даних, спираючись на дії користувача, та оновлення вигляду View, завдяки сповіщенню Model. По суті, цей рівень є «мозком» програми, в якому і визначається бізнес логіка застосунку.

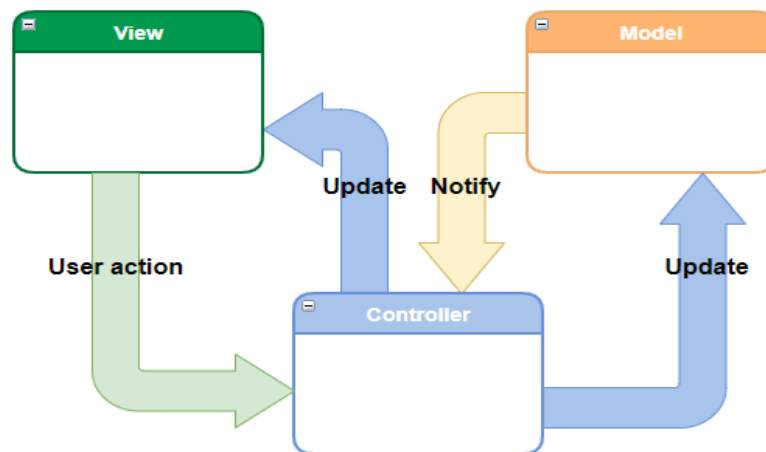


Рисунок 3.1 - Схема роботи шаблону MVC

Для цього проекту було реалізовано MVC на базі мови PHP^[12]. PHP – скриптова мова програмування з відкритим програмним кодом, випущена в 1995 році. Незважаючи на доволі немолодий вік мови та наявності на ринку безлічі сильних конкурентів(Python, JavaScript, тощо), саме PHP являється лідером серед мов серверної розробки^[13] – 78,4% усіх сайтів в Інтернеті, станом на травень 2020 року, працюють на ній.

Успіх PHP полягає в її кросплатформеності(перед відправкою клієнту сценарії PHP повністю компілюються на стороні сервера, що робить мову незалежною від браузерів; мова не розроблена під якийсь певний сервер, а отже може взаємодіяти з тим, яким захоче розробник), безлічі посібників та величезній спільноті програмістів, завдяки чому з легкістю можна знайти відповіді щодо розробки. Для простоти користування та зменшення вартості продуктів, на PHP написана незлічена кількість систем керування контентом(такі як WordPress, Opencart). Саме вони внесли в розробку інтернет-сайтів великі зміни – кожен користувач мережі може створити сайт без особливих знань в мовах програмування.

Окрім цього, синтаксис PHP інтуїтивно зрозумілий та гнучкий. PHP-код можна імплементувати безпосередньо в HTML-код сторінок, які будуть оброблюватись PHP-інтерпретатором. Це є дуже зручною функцією для веб-

розробки, і разом з перевагами, що описані вище, було прийнято рішення створювати серверну частину саме на цій мові.

Для розробки дизайну сторінки було використано HTML, CSS та JavaScript. Разом вони мають широкий спектр можливостей, щоб зробити веб-сторінку інтерактивною та максимально привабливою для користувача продукту.

HTML^[14] – це мова розмітки гіпертексту (від англ. HyperText Markup Language). Це найпоширеніший інструмент створення сайтів. Коли сторінка відкривається в браузері, він переглядає HTML-код, зчитує з нього спеціальні симовли – теги, обробляє їх, та створює відповідні елементи: таблиці, гіперпосилання, заголовки, тощо. Теги - це спеціальні маркери, які певним чином інтерпретуються браузером. По суті, кожен тег вкладає якусь інформацію для браузера. Розробники HTML постійно підтримують продукт, виробляючи нові версії, що покращують та оптимізують роботу веб-програмістів. Наприклад, оновлення HTML5, випущене у 2008 році, дало змогу покращити мультимедійну сторону продукту. Відтепер відтворення відео та музики не потребує додаткових розширень, а здійснюється за допомогою відповідних тегів.

CSS^[14] – мова програмування для опису стилізації HTML-документа (від англ. Cascading Style Sheets - каскадні таблиці стилів). Окрім HTML-документів, CSS також використовується для стилізації XML-документів будь-якого вигляду, а також для SVG-документів. По суті, всі кольори сайту, місце розташування елементів або шрифти та їх розміри – діло мови CSS. Головною ціллю розробки мови було розділення логічної структури веб-сторінки від її опису, що дозволяє покращити гнучкість керування проектом та зменшити обсяг повторюваного коду.

Стили CSS підключаються до HTML-сторінки чотирьома способами: через тег link, через директиву @import – у випадку, коли CSS-файл знаходиться окремо та має розширення .css, та за допомогою тегу style або атрибуту цього тегу, коли CSS знаходиться всередині HTML-документа.

Серед основних переваг CSS-файлів є можливість розробки проекту під різні розміри користувацьких екранів: під телефон, планшет, комп'ютер, тощо. Ця можливість відчутно полегшує та спрощує час розробки, що є найголовнішим показником в роботі.

Для роботи з динамічними елементами сайту (такі як меню-бар) були використані бібліотеки JavaScript. JavaScript^{[14][15]} – це об'єктно-орієнтована мова програмування, яка дозволяє створювати динамічно оновлюваний контент веб-сторінки без її перезавантажування. JavaScript вважається третім блоком веб-технологій, два з яких вже розглянуті HTML та CSS. Саме ці три мови дозволяють створити повноцінний та цікавий користувачу сайт. Прокрутка, слайд-меню, інтерактивні зображення, спливаючі сторінки чи повідомлення – все це з'являється на веб-сторінці саме завдяки функціоналу JavaScript.

Одна з найбільших та найкорисніших бібліотек JavaScript - є бібліотека JQuery. Серед основних переваг – можливість мінімізувати код. Додаючи вже готові блоки з бібліотеки. Окрім цього, до переваг можна віднести:

1. Підтримка маніпуляцій з DOM-елементами (інтерфейс для програм та скриптів, який допомагає їм отримати доступ до HTML-, XML-документів.
2. Велика кількість навчальних посібників та широке застосування серед програмістів.
3. Кросбраузерність та багатоплатформеність.

Для використання бібліотеки потрібно скачати її з сайту^[16] та додати до проекту.

Для полегшення розробки клієнтської частини часто використовується Bootstrap^{[17][18]}. Bootstrap – HTML/CSS/JavaScript фреймворк з відкритим програмним кодом. Він включає інструментарій найуживаніших CSS-стилів та JavaScript-скриптів для швидкого створення веб-сторінок. При цьому,

забезпечуючи необхідну гнучкість проекту. По суті, це готовий програмний розв'язок складних задач, на які було б втрачено в 3-10 разів більше часу

Bootstrap включає інструментарій для створення різних кнопок, форм, вкладок, але найголовніше, за що розробники так люблять цей фреймворк, - адаптивні сітки. До випуску Bootstrap, програмісти були змушені самостійно розташовувати блоки та панелі за допомогою CSS. Сітка Bootstrap дозволяє перейти до системи стовпців і рядків, вкладаючи блоки один в одного. Друга причина популярності закладена в тому, що дана конструкція буде відображена на будь-якій операційній системі користувача з будь-яким браузером.

AJAX^{[19][20]}(від англ. Asynchronous JavaScript and XML) – спосіб побудови взаємодії між користувачем та інтерфейсом веб-сайтів за допомогою асинхронного обміну інформацією без оновлення браузера. Цю технологію можна використати до авторизації та реєстрації на сайті, автопошуку на сайті, тощо.

Працює AJAX наступним чином:

1. Користувач ініціює подію, що призводить до ініціалізації об'єкту XMLHttpRequest за допомогою функцій JavaScript.
2. Об'єкт XMLHttpRequest виконує асинхронний запит до бази даних, з відповідними параметрами, що задав користувач.
3. На сервері виконується обробка запиту та створюється відповідь в вигляді XML-документу з відповідними даними.
4. Цей документ отримує XMLHttpRequest, виконує обробку отриманих даних та оновлює модель DOM HTML для виводу відповіді на екран користувача.

Дана технологія надає низку переваг, а саме економію трафіку користувача(дякуючи частковому оновленню ресурсів користувачу не потрібно перезавантажувати всю сторінку), зменшення кількості запитів на сервер та пришвидшення роботи сайту.

Для створення та управління базою даних було обрано веб-інтерфейс phpMyAdmin, який адмініструє СКБД MySQL. Окрім стандартних функцій цього

сервера також є можливість виконувати SQL-запити, керувати ключами, експортувати дані в CSV, XML, PDF форматах, тощо.

Цей інтерфейс є досить популярним серед розробників, адже він підтримує більшість хостів, на яких розміщуються сайти в Інтернет мережі. Також велику роль зіграв відкритий програмний код та підтримка більше, ніж 60 різних мов інтерфейсом програми.

3.3 Опис реалізації

Розробка онлайн-магазину побудована за принципом клієнт-серверної архітектури. Передбачені цим принципом сторони впливають з назви – клієнт та сервер. По суті, клієнт – це замовник різних послуг, а сервер – постачальник цих послуг. Обидві сторони представляють собою програми: в даній роботі клієнтом є браузер, а с робота на сервері реалізована за допомогою мови PHP з відправкою запитів до реляційної бази даних MySQL.

Клієнт та сервер взаємодіють між собою по Інтернет мережі з використанням протоколу HTTP через GET та POST запити. Клієнт відправляє запит на сервер, сервер його обробляє, і надсилає відповідь клієнту, який в свою чергу використовує її для оновлення даних або ж відображення вмісту на екрані.

Кількість клієнтів, що одночасно можуть взаємодіяти з сервером прямо пропорційно залежить від потужностей сервера і запитів, що він може обробити.

Робота побудована за підходом об'єктно-орієнтованого програмування, та використовує три його принципи – інкапсуляція, поліморфізм та наслідування. Код розбитий на класи, які в свою чергу розподілені по модулям.

3.4 Опис реляційної моделі

База даних складається з 5 реляцій, зв'язаних між собою.

Таблиця «Головна категорія» представлена двома атрибутами, один з яких також являється первинним ключем - «назва категорії». Другий атрибут «Статус на сайті» був доданий суто в розробницьких цілях для того, щоб можна було

приховати категорію. Оскільки магазин представляє жіноче та чоловіче взуття – на момент створення до бази внесені ці два значення.

Таблиця 3.1 – Таблиця Main Category

Main Category				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Назва категорії»
	status	Можливі значення: «0» - не відображати на сайті, «1» - відображати на сайті	NOT NULL	Атрибут простий, обов'язковий «Статус на сайті»

Таблиця «Категорія» ідентифікується первинним ключем -«Порядковий номер підкатегорії», оскільки, наприклад, підкатегорія «туфлі» може належати як і до жіночої, так і до чоловічої категорій. Атрибути «Послідовність в меню» та «Статус на сайті» були додані для того щоб, адміністратор міг редагувати вигляд слайд-меню асортимента. Таблиця має зв'язок з «Головною категорією».

Таблиця 3.2 – Таблиця «Категорія»

Category				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	AUTOINCREMENT	NOT NULL	Атрибут простий, обов'язковий «Порядковий номер підкатегорії»
	name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Назва підкатегорії»
	sort_order	INT(11)	NOT NULL	Атрибут простий, обов'язковий «Послідовність в меню»

	status	Можливі значення: «0» - не відображати на сайті, «1» - відображати на сайті	NOT NULL	Атрибут простий, обов'язковий «Статус на сайті»
FK	main_category	VARCHAR(30)	NOT NULL	Зв'язок з таблицею Main Category:name Зв'язок обов'язковий з боку Main Category

Одна з найголовніших таблиць бази – таблиця «Товар». Вона виступає сховищем товарів з каталогу та інформації про них. Розмірна сітка прописана текстовим типом, який в кодї перетворюється на масив значень. Зберігає інформацію про підкатегорію товару.

Таблиця 3.3 – Таблиця «Продукт»

Product				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	INT(11)	NOT NULL	Атрибут простий, обов'язковий «Артикул»
	name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Назва товару»
	price	FLOAT	NOT NULL	Атрибут простий, обов'язковий «Ціна моделі»
	availability	Можливі значення: «0» - не в наявності, «1» - в наявності	NOT NULL	Атрибут простий, обов'язковий «Наявність моделі»
	brand	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Бренд»
	description	VARCHAR(200)	NOT NULL	Атрибут простий, обов'язковий «Опис товару»

	size	VARCHAR(100)	NOT NULL	Атрибут простий, обов'язковий «Розмір на товару»
FK	category_id	INT(11)	NOT NULL	Зв'язок з таблицею Category:id Зв'язок обов'язковий з боку Product

Таблиця «Замовлення» зберігає інформацію про продажі магазину. Оскільки робити замовлення можуть не тільки зареєстровані користувачі, то зберігається інформація про замовника. Атрибут «Статус замовлення» визначає на якому етапі виконання він знаходиться – «1» - нове замовлення, «2» - обробляється, «3» - доставка, «4» - виконано. Для того, щоб зменшити бсяг використаної пам'яті база даних зберігає ці дані саме в числовому виді.

Таблиця має зв'язок з таблицею «Product», що допомагає підрахувати суму замовлення, а також необов'язковий зв'язок(тільки в тому випадку. Якщо користувач зареєстрований) з таблицею «User».

Таблиця 3.4 – Таблиця «Замовлення»

Product Order				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	INT(11)	NOT NULL	Атрибут простий, обов'язковий «Номер замовлення»
	user_name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Ім'я покупця»
	s_name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Прізвище клієнта»
	user_phone	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Телефон покупця»
	user_comment	VARCHAR(30)	NULL	Атрибут простий, необов'язковий «Коментар до замовлення»

	date	TIMESTAMP	NOT NULL	Атрибут простий, обов'язковий «Дата замовлення»
	status	Можливі значення: «1», «2», «3», «4»	NOT NULL	Атрибут простий, обов'язковий «Статус замовлення»
	price	FLOAT	NOT NULL	Атрибут похідний, обов'язковий «Сума замовлення»
	mail	VARCHAR(30)	NOT NULL	Атрибут похідний, обов'язковий «E-mail покупця»
	size	VARCHAR(3)	NOT NULL	Атрибут похідний, обов'язковий «Замовлений розмір»
FK	products	INT(11)	NOT NULL	Зв'язок з таблицею Product:id Зв'язок обов'язковий з боку Order
FK	user_id	AUTOINCREMENT	NULL	Зв'язок з таблицею User:id Зв'язок необов'язковий з боку Order

Таблиця «Користувач» зберігає інформацію про зареєстрованого користувача системи. Первинний ключ – автоінкремент, який кожен раз збільшується з додаванням нових користувачів. Атрибут «Роль» є допоміжним атрибутом для розділення користувачів на покупців та адміністраторів, завдяки чому буде наданий доступ до адмінпанелі.

Таблиця 3.5 – Таблиця «Користувач»

User				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	AUTOINCREMENT	NOT NULL	Атрибут простий, обов'язковий «Порядковий номер користувача»

	name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Ім'я користувача»
	email	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «E-mail користувача»
	password	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Пароль користувача»
	role	Можливі значення: « » - звичайний користувач, «admin» - доступ по адмінпанелі	NULL	Атрибут простий, необов'язковий «Роль користувача»

3.5 Розробка головної програми

Робота програми починається з створення сесії користувача:

```
?php
//Початок сесії користувача
session_start();
// Підключення головної сторінки
include_once "models/category.php";

// Підключення файлів системи
define('ROOT', dirname(__FILE__));
require_once(ROOT.'/components/Autoload.php');

// Виклик Router(маршрутер програми)
$router = new Router();
$router->run();
?>
```

Оскільки, архітектура проекту є клієнт-серверною, то серверна частина реалізується з підключення до бази даних. Відбувається це за допомогою PHP:

```
<?php
class Db
{
    //@return об'єкт класу PDO для роботи з БД
    public static function Connection()
    {
        // Отримуємо параметри підключення з файлу db_params.php'
        $paramsPath = ROOT . '/config/db_params.php';
```

```

$params = include($paramsPath);
// Встановлюємо з'єднання
$dsn = "mysql:host={$params['host']};dbname={$params['dbname']}";
$db = new PDO($dsn, $params['user'], $params['password']);
// Задаємо кодування
$db->exec("set names utf8");
return $db; }}
<?php

// Задаємо параметри входу в db_params.php :

return array(

    'host' => 'localhost',
    'dbname' => 'phpshop',
    'user' => 'root',
    'password' => '',

);

```

Для того, щоб адміністратор отримав доступ до системи керування сайтом, йому потрібно, як і новим користувачам, зареєструватись. Відбувається це завдяки методу Connection(), який надає доступ до бази даних, та SQL-запиту, що обробляє введені користувачем дані, та створює нового користувача. Функція bindParam() зв'язує PHP змінну з параметром SQL-запиту.

```

public static function register($name, $email, $password)
{
    $db = Db::Connection();

    $sql = 'INSERT INTO user (name, email, password,role) '
        . 'VALUES (:name, :email, :password,"")';

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

    $result->bindParam(':name', $name, PDO::PARAM_STR);
    $result->bindParam(':email', $email, PDO::PARAM_STR);
    $result->bindParam(':password', $password, PDO::PARAM_STR);
    return $result->execute();}

```

При реєстрації наявні перевірки на довжину ім'я користувача, довжину пароля, наявність необхідних символів в електронній адресі, а також чи вже існує даний користувач по email :

```

public static function checkEmailExists($email)
{
    $db = Db::Connection();

```

```

$sql = 'SELECT COUNT(*) FROM user WHERE email = :email';

$result = $db->prepare($sql);
$result->bindParam(':email', $email, PDO::PARAM_STR);
$sql = $result->bindParam(':email', $email, PDO::PARAM_STR);
$result->execute();

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

```

Для того, щоб зареєстрований користувач став адміністратором в базі даних йому потрібно надати роль admin. При авторизації ми отримуємо інформацію про користувача та перевіряємо його роль, якщо це адміністратор – надаємо доступ до адмінпанелі.

```

// Отримуємо інформацію про користувача в сесії
$user = User::getUserById($userId);

// Якщо роль поточного користувача "admin", пускаємо його в адмінпанель
if ($user['role'] == 'admin') {
    return true;
}

// Якщо ні – видаємо повідомлення про закритий доступ
die('Access denied');
}

```

Більша частина роботи з базою даних відбувається за допомоги інтегрованих SQL-запитів – витяг даних для подальшого використання, додавання, редагування, видалення. Наприклад, редагування особистої інформації користувача(ім'я або пароль):

```

public static function edit($id, $name, $password)
{
    $db = Db::Connection();
    $sql = "UPDATE user
        SET name = :name, password = :password
        WHERE id = :id";
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
}

```

```

$result->bindParam(':password', $password, PDO::PARAM_STR);
return $result->execute();
}

```

В проєкті використано технологію AJAX. Завдяки цьому, сторінки мають можливість підвантаження окремих своїх частин динамічно. Для відображення кількості товару в корзині, для пошуку категорії та товару за назвою, в якій більше 3х символів:

```

$('.who1').bind("keyup input click", function() {
    if(this.value.length >= 3){
        $.ajax({
            type: 'post',
            url: "/search/product", //Шлях до обробника
            data: {'referal1':this.value},
            response: 'text',
            success: function(data){
                $(".search_result1").html(data).fadeIn(); //Виводимо отримані
дані в списку
                overlay1.css('display', 'block');
            }
        })
    } else { //Якщо нічого не знайшли – залишаємо пустий список
        $(".search_result").html(null).fadeOut();
        overlay1.css('display', 'none');
    }
})

```

Для зменшення повторення коду та динамічного відображення певних частин було реалізовані header та footer для сторінок з каталогом, детального опису товару, кабінету користувача та реєстрації з авторизацією. Header прописує блок з контактною інформацією, логотипом компанії, посилання на корзину, вхід до кабінету та реєстрацію. Footer додає слайдер з рекомендованими товарами, а також кнопку з поверненням вгору. Надано частину реалізації header за допомогою HTML та PHP, при якому визначається авторизований користувач чи ні, якщо так – то замість «Вхід» буде «Вихід».

```

<div class="header-middle"><!--header-middle-->
    <div class="container">
        <div class="row">
            <div >
                <div>

```

```

        <div class="univer_logo">
            <a href="/"></a>
        </div>

    </div>
</div>
<div class="col-sm-8">
<!--
    <div class="shop-menu pull-right">-->

        <div class="menu_in_site">
            <ul class="nav navbar-nav">
<!--
                <div class="menu_in_site">-->
                    <li><a href="/cart/"><i class="fa fa-shopping-
cart"></i> Корзина (<span id="cart-
count"><?php echo Cart::countItems(); ?></span></a></li>

                        <?php if (User::isGuest()): ?>
                            <li><a href="/user/login/"><i class="fa fa-
lock"></i> Вхід</a></li>
                            <li><a href="/user/register/"><i class="fa fa-
lock"></i> Реєстрація</a></li>
                        <?php else: ?>
                            <li><a href="/cabinet/"><i class="fa fa-
user"></i> Аккаунт</a></li>
                            <li><a href="/user/logout/"><i class="fa fa-
unlock"></i> Вихід</a></li>
                        <?php endif; ?>
                    </div>-->
                </ul>

            </div>
        </div>
    </div>
</div><!--/header-middle-->

```

Витяг даних з бази даних відбувається за допомогою модуля Model, який готує дані до подальшого використання. В коді нижче представлено приклад витягу та представлення у вигляді масиву даних щодо історії замовлення користувача, що зареєстрований у базі під певною електронною адресою.

```

    public static function getUsersOrderList($email)
    {
        $db = Db::Connection();
        // Витяг даних

```



```

$sql = 'SELECT * FROM product_order WHERE mail = :mail';
$result = $db->prepare($sql);
$result->bindParam(':mail', $email, PDO::PARAM_STR);

// Вказуємо, що хочемо отримати дані в вигляді масиву
$result->setFetchMode(PDO::FETCH_ASSOC);
$result->execute();
// Повертаємо дані
$ordersList = array();
$i = 0;
    $allprod=null;
while ($row = $result->fetch()) {
    if(empty($row['id']))return null;
    $ordersList[$i]['id'] = $row['id'];
    $ordersList[$i]['date'] = $row['date'];
    $ordersList[$i]['products'] = $row['products'];
    $ordersList[$i]['status'] = $row['status'];
    $ordersList[$i]['size'] = $row['size'];
    $i++;
}

```

Усі можливі маршрути користувачів прописані в Routes. Це PHP-файл, який перенаправляє систему в заданий модуль для виконання того чи іншого посилання від користувача.

3.6 Тестування програми і результати її виконання

Головною сторінкою онлайн-магазину є сторінка каталогу. На ній показані контактні дані, слайдер з категоріями товару на сайті, всі товари даного магазину, можливість пошуку та рекомендовані товари:

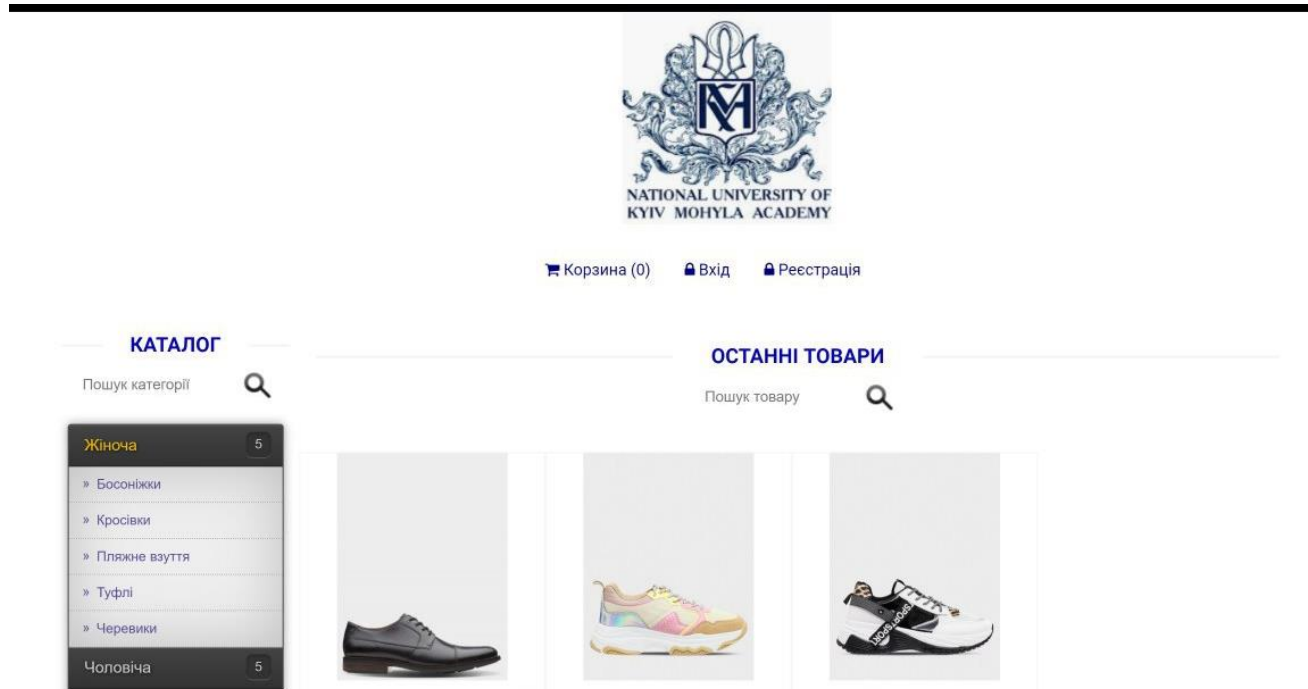


Рисунок 3.2 – Вікно каталогу продукції

Головною функцією для користувача є можливість придбання даного товару. При додаванні моделі у корзину користувач має змогу обрати підходящий розмір:

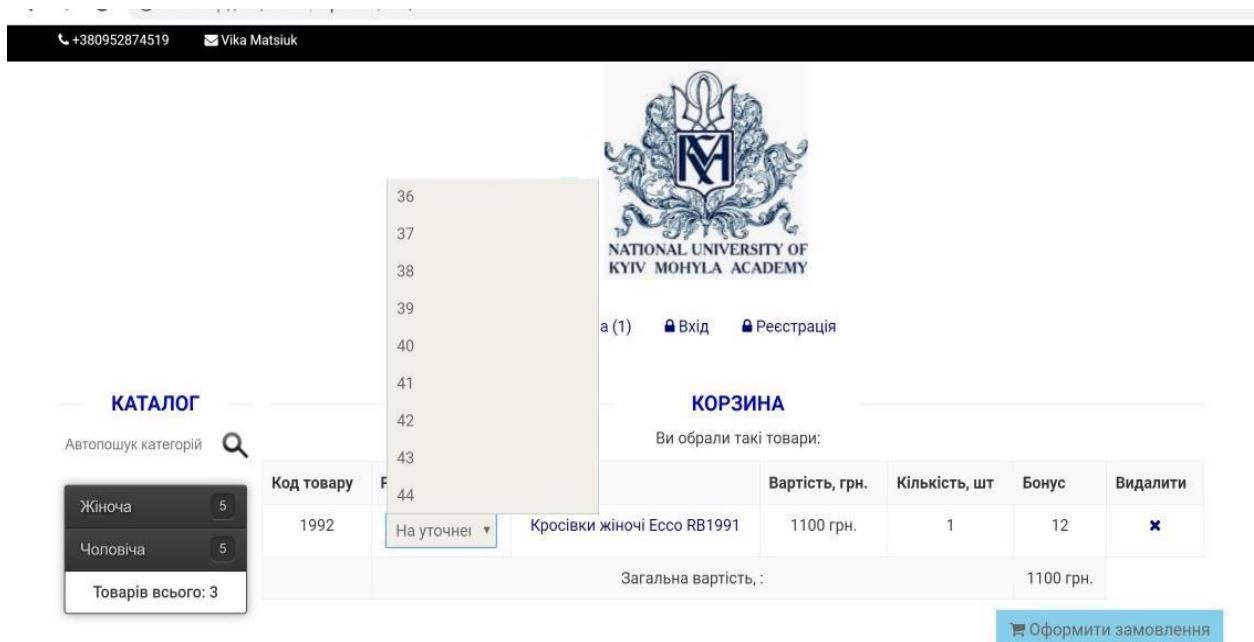


Рисунок 3.3 – Вікно корзини користувача

Для перегляду історії замовлення та накопичення бонусів користувачу пропонується створити особистий кабінет, якщо електронна адреса уже існує чи пароль не відповідає вимогам системи, буде виведено такі повідомлення:



Корзина (0) Вхід Реєстрація

- Пароль має бути довше 6ти символів
- Такий email вже існує


Реєстрація на сайті

Вікторія Мацюк
vikama@ukr.net

Реєстрація

Рисунок 3.4 – Вікно реєстрації на сайті

Створивши власний кабінет(/cabinet/#), користувач отримує ряд переваг, таких як перегляд історії замовлень, перегляд статусу його замовлення та накопичення бонусів:



Корзина (0) Аккаунт Вихід

Привіт, Вікторія Мацюк!
Редагувати дані Перелік замовлень

Кабінет користувача

Бонусів на балансі: 13!

ID замовлення	Дата придбання	Кількість	Назва товару	Розмір	Ціна товару	Сума придбання	Бонус	Статус
12	2020-05-09 12:41:18	1	Туфлі чоловічі Braska CD1203	38	1250	1250	13	Нове замовлення

Рисунок 3.5 – Вікно кабінету зареєстрованого користувача

Після перевірки на роль, адміністратор може перейти на сторінку (/admin), де будуть надані можливості на управління товарами, категоріями та підкатегоріями, а також перегляд та управління замовленнями:

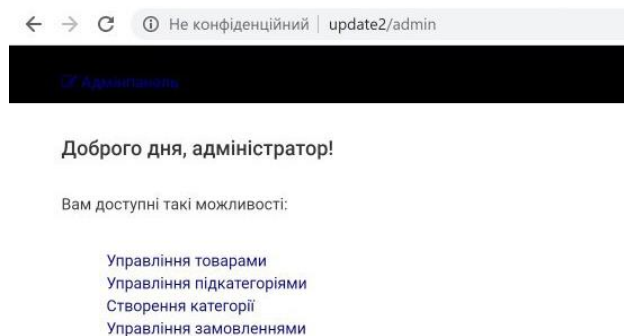


Рисунок 3.6 – Вікно адмінпанелі

Найзручніша функція для адміністратора є перегляд замовлень користувачів, він може редагувати статус замовлення, переглядати всі коментарі до замовлення та його параметри.

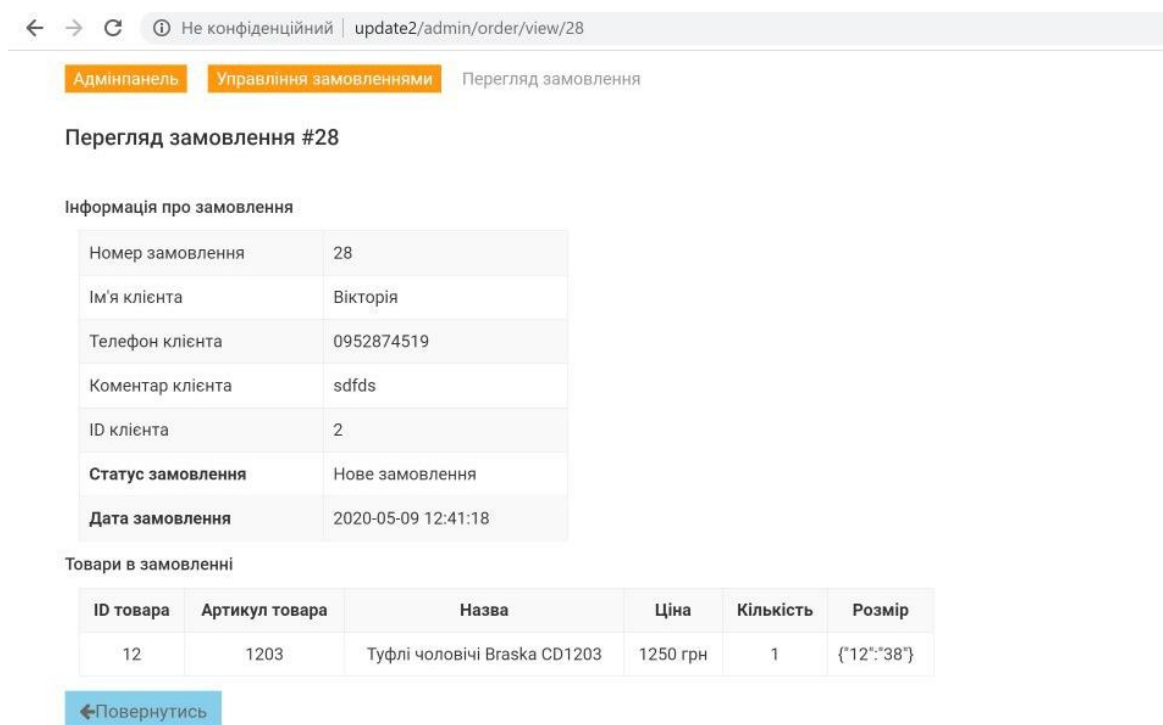


Рисунок 3.7 – Вікно перегляду замовлення для адміністратора

Висновки

По закінченню роботи всі поставлені задачі були виконані. Було ґрунтовно проаналізовано сучасний стан ринку електронної комерції та надані причини, чому підприємцям варто активізуватися в переході до онлайн продажів. Також було проаналізовано та визначено недоліки сучасних онлайн-магазинів взуття, а також представлено оптимальні рішення цих проблем. В проекті вони всі були враховані та майже всі впроваджені.

В технічному плані були освоєні та обрані сучасні мови веб-програмування, і також оглянуто обраний тип системи керування базами даних. В результаті, обрані технології дійсно гарно працюють та мають всі шанси стати конкурентами вже існуючих розрблених магазинів.

Створений онлайн-магазин можна вже зараз використовувати не тільки в навчальних цілях, але й в комерційних. Окрім цього, завдяки гнучкості програмного коду можна з легкістю продовжувати оснащувати магазин різноманітним функціоналом. Серед перспектив розвитку проекту є розширення функціоналу в вигляді додавання користувацьких відгуків на товар та створення форуму для користувачів. Також можлива автоматизація повернення товару завдяки додаванню потрібних форм. Окрім цього, варто завершити впровадження бонусної системи для постійних клієнтів(наразі можливе тільки накопичення балів). Також, варто додати можливість для менеджера магазину переглядати статистику популярних товарів, витяг інформації про продажі у вигляді Excel таблиць.

Таким чином основними результатами роботи є аналіз та дослідження баз даних, їх типів та систем керування, дослідження існуючих веб-магазинів, сучасних мов веб-програмування, виявлення їх переваг і функцій; та розробка зручного та інтуїтивно зрозумілого онлайн-магазину з врахуванням всіх набутих знань.

Список використаної літератури

1. Рынок электронной коммерции: каким он был в 2019 и каким будет в 2020 [Електронний ресурс]. 14.01.2020 Автор Еліна Редіх
https://biz.censor.net.ua/resonance/3169606/rynok_elektronnoyi_kommertsii_kakim_on_byl_v_2019_i_kakim_budet_v_2020
2. Итоги украинского e-commerce за 2019-й [Електронний ресурс]. 13.01.2020 Автор інформаційне агентство Allretail
<http://allretail.ua/topics/63438/>
3. Проникнення Інтернету в Україні [Електронний ресурс]. Жовтень, 2019 Автор статистичне агентство Factum Group
https://inau.ua/sites/default/files/file/1910/dani_ustanovchyh_doslidzhen_iii_kvartal_2019_roku.pdf
4. Інтернет-магазин взуття Intertop [Електронний ресурс]. Дата використання 23.04.2020
<https://intertop.ua/>
5. Інтернет-магазин взуття Hi!Legs [Електронний ресурс]. Дата використання 23.04.2020
<https://hilegs.com.ua/>
6. Інтернет-магазин взуття Вітчизняне взуття [Електронний ресурс]. Дата використання 23.04.2020
<https://www.vitchiznyane-vzuttya.com/>
7. Маркетинг лояльности, или Как добиться расположения клиентов? [Електронний ресурс]. 17.11.2015 Автор LPGenerator
<https://lpgenerator.ru/blog/2015/11/17/marketing-loyalnosti-ili-kak-dobitsya-raspolozheniya-klientov/>
8. ВИДИ САЙТІВ ТА ЇХНІЙ ФУНКЦІОНАЛ [Електронний ресурс]. 22.11.2019 Автор WebMaestro
<https://webmaestro.com.ua/ua/blog/vydy-saitiv/>

9. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Авторы Томас Конолли, Каролін Бегг, 2003
10. MYSQL: использование и администрирование. Автор Вікрам Васвані, 2011
11. The MVC Pattern and PHP, Part 1 [Електронний ресурс] 04.03.2013 Автор Callum Hopkins
<https://www.sitepoint.com/the-mvc-pattern-and-php-1/>
12. Преимущества PHP, портал про PHP [Електронний ресурс]. Дата використання 05.05.2020
<http://www.php.su/php/?opport>
13. Historical trends in the usage statistics of server-side programming languages for websites [Електронний ресурс]. Дата використання 05.05.2020
https://w3techs.com/technologies/history_overview/programming_language
14. HTML, CSS, PHP, JavaScript, SQL – что и зачем? [Електронний ресурс]. Дата використання 06.05.2020 Автор Дмитро Науменко
<http://codeharmony.ru/materials/125>
15. Современный учебник JavaScript [Електронний ресурс]. Дата використання 06.05.2020 Автор Ілля Кантор
<https://learn.javascript.ru/>
16. Головный сайт JQuery [Електронний ресурс]. Дата використання 06.05.2020
<https://jquery.com/download/>
17. Головный сайт Bootstrap [Електронний ресурс]. Дата використання 06.05.2020
<https://getbootstrap.com/>
18. Быстрая разработка Web-сайтов и Web-приложений с помощью Bootstrap [Електронний ресурс]. 29.11.2013 Автор Уче Огбуйі
<https://www.ibm.com/developerworks/ru/library/wa-bootstrap/>
19. Введение в Ајах для веб-приложений PHP [Електронний ресурс]. Дата використання 07.05.2020 Автор NetBeans
https://netbeans.org/kb/docs/php/ajax-quickstart_ru.html
20. AJAX [Електронний ресурс]. 23.10.2019 Автор Ingate
<https://promo.ingate.ru/seo-wikipedia/ajax/>