

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інтернет-магазину «BROX» з використанням PHP 7.5,
MySQL 8.0 та JavaScript v8

Виконав: студент IV курсу, групи СНС-42

спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Терещук В.С.

(прізвище та ініціали)

Керівник

(підпис)

Готович В.А.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Бойко І.В.

(прізвище та ініціали)

Тернопіль
2022

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О. Я. к.т.н., доцент		

7. Дата видачі завдання _____ 24 січня 2022 р. _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	24.01.2022	Виконано
2.	Підбір джерел про створення, аналіз та підтримку інтернет-магазину	04.01.2022-30.01.2022	Виконано
3.	Переклад та опрацювання джерел про розробку інтернет-магазину та теоретичний матеріал	31.01.2022-06.02.2022	Виконано
4.	Виконання дослідження щодо роботи інтернет-магазину Розроблення інтернет-магазину «BROX»	07.02.2022-13.02.2022	Виконано
5.	Оформлення розділу «Теоретична частина»	14.02.2022-06.03.2022	Виконано
6.	Оформлення розділу «Практична частина»	07.03.2022-03.04.2022	Виконано
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	04.04.2022-17.04.2022	Виконано
8.	Виконання завдання до підрозділу «Основи охорони праці»	18.04.2022-01.05.2022	Виконано
9.	Оформлення кваліфікаційної роботи	02.05.2022-15.05.2022	Виконано
10.	Нормоконтроль	16.05.2022-22.05.2022	Виконано
11.	Перевірка на плагіат	10.06.2022	Виконано
12.	Попередній захист кваліфікаційної роботи	07.06.2022	Виконано
13.	Захист кваліфікаційної роботи	24.06.2022	

Студент

_____ (підпис)

Терещук В. С.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Готович В. А.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка інтернет-магазину «BROX» з використанням PHP 7.5, MySQL 8.0 та JavaScript v8 // Кваліфікаційна робота освітнього рівня «Бакалавр» // Терещук Віктор Сергійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2022 // С. 52, рис. – 13, додат. – 4, бібліогр. – 27, табл. – 0, кресл. – .

Ключові слова: інтернет-магазин, інтернет-комерція, PHP, JavaScript, MySQL.

Кваліфікаційна робота присвячена розробці інтернет-магазину «BROX» з використанням набору технологій PHP, MySQL, JavaScript, HTML та CSS.

Метою роботи є розробка інтернет-магазину для надання послуг в сфері онлайн торгівлі.

В першому розділі кваліфікаційної роботи розглянуто:

- етапи та методи розробки інтернет-магазину;
- огляд можливих рішень;
- опис основних технологій, які були використані при розробці інтернет-магазину.

В другому розділі кваліфікаційної роботи розглянуто:

- структуру інтернет-магазину;
- описано основні можливості;
- описано створення плагіну для кошика з товарами.

В третьому розділі кваліфікаційної роботи описано:

- долікарську допомогу при кровотечах;
- вплив шуму на організм людини та розробка заходів до його зниженню

до допустимих величин.

ANNOTATION

Development of “BROX” online store using PHP 7.5, MySQL 8.0 and JavaScript v8
// Qualification work of the educational level "Bachelor" // Tereshchuk Victor
Sergiyovich // Ternopil Ivan Puluj National Technical University, Faculty of Computer
Information Systems and Software Engineering, Department of Computer Science,
Group SNs-42 // Ternopil, 2022 // P. 52, img. – 13, applications. – 4, bibliografia – 27,
tables – 0, drawings – .

Keywords: online store, e-commerce, PHP, JavaScript, MySQL.

Qualification work is devoted to the development of the online store "BROX" using the following technologies: PHP, MySQL, JavaScript, HTML, CSS.

The purpose of the work is to develop an online store to provide services in the field of online commerce.

The first section of the qualification work discusses:

- the stages and methods of developing an online store;
- an overview of possible solutions;
- a description of key technologies, which used in the development of the online store.

The second section of the qualification work considers:

- the structure of the online store;
- describes the main features;
- describes the creation of a plugin for a basket of goods.

The third section of the qualification work describes:

- pre-medical care for bleeding;
- the impact of noise on the human body and the development of measures to reduce it to acceptable levels.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ

I ТЕРМІНІВ

.NET – Програмна технологія, запропонована компанією Microsoft як платформа для створення як звичайних програм, так і веб-застосунків.

API (англ. Application Programming Interface) – Інтерфейс програмування застосунків.

CMS (англ. Content Management System) – Система керування вмістом.

CSS (англ. Cascading Style Sheets) – Каскадні таблиці стилів.

DHTML (англ. Dynamic HTML) – Концепція створення вебсайту, що розглядає HTML-документ як об'єктну структуру.

HTML (англ. HyperText Markup Language) – Мова розмітки гіпертексту.

IIS (англ. Internet Information Services) – Набір серверів для декількох служб Інтернету від компанії Майкрософт.

JS – JavaScript.

MySQL – Вільна система керування реляційними базами даних.

Perl (англ. Practical Extraction and Reporting Language) – Високорівнева, інтерпретована, динамічна мова програмування загального призначення.

PHP (англ. Hypertext Preprocessor) – Гіпертекстовий препроцесор.

PostgreSQL – Об'єктно-реляційна система керування базами даних.

SaaS (англ. Software as a service) – Модель поширення програм споживачам, при якій постачальник розробляє веб-програму, розміщує її й керує нею з метою використання її замовниками через інтернет.

SQL (англ. Structured Query Language) – Мова структурованих запитів.

SQLite – полегшена реляційна система керування базами даних.

SSL (англ. Secure Sockets Layer) – Криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером.

UML (англ. Unified Modeling Language) – Уніфікована мова моделювання.

UML (англ. Unified Modeling Language) – Уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування.

VB (англ. Visual Basic) – Найсучасніша BASIC-базована мова програмування.

WAMP/WNMP – Акронім, що позначає набір серверного програмного забезпечення, широко використовуваний з метою розробки та надання веб-сервісів.

XHTML (англ. Extensible Hypertext Markup Language) – Мова розмітки, що має таку саму виразну силу як і HTML але відповідає синтаксичним правилам XML.

XML (англ. Extensible Markup Language) – Стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками.

БД – База даних.

ЗІЗ – Засоби індивідуального захисту.

ОС – Операційна система.

ОЦК – Об'єм циркулюючої крові.

ПК – Персональний комп'ютер.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ПОСТАВЛЕНОГО ЗАВДАННЯ.....	10
1.1 Аналітичний огляд існуючих рішень.....	10
1.2 Найменування та область застосування програмного рішення	12
1.3 Етапи та методи розробки.....	15
1.4 Постановка задачі на розробку інтернет-магазину	16
1.5 Мова розмітки гіпертекстових документів HTML	18
1.6 Каскадні таблиці стилів CSS.....	19
1.7 Мова програмування PHP	20
1.8 Мова програмування JavaScript.....	21
1.9 MySQL та бібліотека RedBeanPhp	22
1.10 Локальний сервер Open Server.....	23
1.11 Висновки до першого розділу	25
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОЕКТУ ІНТЕРНЕТ-МАГАЗИНУ	26
2.1. Проектування структури та дизайну інтернет–магазину	26
2.2. Функціональні можливості	28
2.3. Написання плагіну для кошику товарів.....	30
2.4. Проектування структури каталогів інтернет–магазину.....	32
2.5. Опис програмних рішень	34
2.6. Структура інтерфейсу адміністративної частини інтернет–магазину	36
2.7. Розгортання та налаштування інтернет–магазину	36
2.8. Експлуатація інтернет-магазину	38
2.9. Висновки до другого розділу.....	41
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ.....	43
3.1. Долікарська допомога при кровотечах.....	43
3.2. Вплив шуму на організм людини та розробка заходів щодо його зниженню до допустимих величин для обладнання	46
3.3. Висновки до третього розділу	48

ВИСНОВКИ.....	8
ПЕРЕЛІК ДЖЕРЕЛ.....	49
ДОДАТКИ	50

ВСТУП

Актуальність теми. В світі, а зокрема й в Україні, величезними темпами росте кількість користувачів Інтернет і, як наслідок, кількість «електронних» покупців. Значна частина людей щодня, не виходячи з дому, купують різні товари в електронних магазинах.

Інтернет–магазин, з однієї сторони, дає можливість зменшити витрати виробника завдяки економії коштів при відкритті магазину, розширити ринки збуту, а з іншої сторони для покупців – купувати товари цілодобово, у будь-якій країні, у будь-якому місці. Саме це дає перевагу інтернет-магазинам перед звичайними магазинами. Таким чином, електронна комерція є перспективним напрямком діяльності, де вибір і замовлення продукції здійснюється через комп'ютерні мережі, а розрахунки між покупцем і постачальним здійснюються з використанням електронних засобів платежу. Отже, розробка інтернет-магазину та його використання є актуальним питанням сьогодення.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка інтернет-магазину для торгівлі товарами з використанням стеку технологій PHP + MySQL + HTML5 + JS. Для реалізації мети необхідно виконати такі **завдання**:

- проаналізувати відомі платіжні системи, які застосовуються в електронній комерції;
- розробити алгоритми функціонування програмних модулів інтернет–магазину;
- спроектувати структуру та інтерфейс користувача інтернет–магазину.

Об'єкт дослідження: інтернет-магазин онлайн торгівлі.

Предмет дослідження: методичні та практичні основи проектування програмного забезпечення.

РОЗДІЛ 1. АНАЛІЗ ПОСТАВЛЕНОГО ЗАВДАННЯ

1.1 Аналітичний огляд існуючих рішень

Інтернет-магазин є одним з важливішим методу ведення бізнесу, як для малих компаній так і для великих або тих, що тільки почали свій розвиток. Це надає власникам бізнесу безліч переваг та додаткову популярність при його наявності. Кожен клієнт має змогу не виходячи з дому та не контактуючи з продавцем отримати релевантну інформацію про наявні товари, їх комплектацію, відгуки інших покупців, тощо, щоб скласти власне враження про послуги які надає компанія з продажу. Також одною з переваг є цілодобова робота інтернет-магазину, що дозволяє людині отримати доступ до інформації та послуг в будь-який зручний для клієнта час. Це також сприяє економії самого бізнесу через непотрібність оплачувати зміни працівникам, які б працювали цілодобово в самому магазині [1].

Доступ до інтернет-магазину здійснюється через мережу Інтернет та пристрій здатний запуснути браузер для перегляду наприклад ПК або смартфон чи планшет. Тобто маючи вихід в Інтернет кожна людина з всіх куточків світу матиме змогу зайти в інтернет-магазин, що розширює аудиторію та площу на якій будуть надаватися послуги, звісно це також позитивно сприяє бізнесу.

Як правило, для досягнення намічених цілей використання сайту береться і різних підхід. Наприклад є сайти рекламні, в яких ціль направлена на окремний товар або послугу яку надають постачальники, інтернет-магазини для купівлі товару серед широкого вибору ціни, якості та самого товару [2].

Для людей купівля товару онлайн є дешевшою та простішою за пошуком, ніж у магазині. Однак є незначний недолік для тих, хто звик купувати товар на «помацки», зробити це онлайн не вийде. Однак це залежить від самої компанії оскільки більшість купленого товару можна повернути згідно чинного законодавства, що дозволить в деякій мірі оглянути товар в живу [3].

Переважно інтернет-магазини являє собою онлайн платформу в якій зображено товари та взаємодія з ними і додатковими функціями. Сучасні інтернет-магазини включають такі функції як: пошук, сортування, купівля товару та особистий кабінет. Наприклад в найпопулярнішому українському інтернет-магазині Rozetka є такий функціонал, як сортування товару за його категоріями та особистий кабінет для зміни адреси, пароля, пошти, спілкування з модераторами та інше. Подібним інтернет-магазини є Prom, в ньому також є подібний функціонал але з більш детальними параметрами сортування товару (ціна, локація, доставка та інше).

Створення інтернет-магазину може відбуватися двома шляхами: в конструкторі або створити самому (замовити). В цих двох методів є свої переваги та недоліки, наприклад при створенні в конструкторі час затрачений на це буде значно менший ніж при створенні його самому. Однак більш глибокі налаштування чи функції будуть недоступні, оскільки конструктор має лише певні можливості. У випадку створенні самому час зростає, однак і можливостей стає набагато більше.

Для створення в конструкторі особливих навичок чи знань не потрібно. При написанні самому потрібно знати технологій, мови розмітки, стилі та мови програмувань. Основною мовою для написання інтернет-магазину та будь-якого сайту є HTML. Він є основою на яку вже додають інші плагіни, та частини інтерфейсу. Також є ще декілька популярних мов розмітки: Perl, XML, DHTML, XHTML та ASP. Всі їх можна використовувати для створення основи інтернет-магазину.

Далі розробка поділяється на фронтенд та бекенд. Для фронтенда використовують CSS. Він дозволяє розміщувати та естетично змінювати елементи написані на html чи інших мовах розмітки. Окрім CSS є такі мови, як JavaScript або VBScript. Ці мови є більш професійним інструментом, оскільки крім естетичного вигляду можна писати і самі скрипти, однак слід пам'ятати, що вони будуть оброблятися на стороні клієнта.

Бекенд є тою частиною інтернет-магазину, що дозволяє записувати данні, вносити їх, тобто мати спілкування з інтернет-магазином. Найпопулярнішою мовою для написання бекенду є PHP. Він обробляється на стороні сервера та має всі необхідні функції для розробки. Однак на просторах інтернету є і значно більше мов для написання бекенду, є мови, які напряду не працюють з бекендом сайту але їх можливо використати для компілятора. Окрім PHP є також:

- Java.
- .NET.
- VB.
- Ruby.
- Python.

До цього переліку також можна внести і JavaScript, як мову, яка здатна виконувати функції бекенду. Серед перерахованих мов можна відзначити Python. Це є досить проста мова в вивченні та написання скриптів. Зараз вона почала набирати все більше та більше людей, які вивчають та використовують її не тільки для розробки бекенду інтернет-магазинів, а і для написання різного роду програм.

Для розробки великих проектів та інтернет-магазинів варто використати таку мову, яка буде оброблюватися на стороні сервера. Це дозволить захиститися від обману з сторони клієнта, оскільки скрипт написаний на JavaScript можна просто скопіювати та проаналізувати його або просто відключити виконання, що дозволяє будь який браузер. Такою мовою є PHP, на ній написано майже 80 відсотків сайтів. Вона є досить гнучкою, має приємний синтаксис та має гарні показники масштабованості при незначних втратах.

1.2 Найменування та область застосування програмного рішення

Основним завданням дипломного проекту є розробка інтернет-магазину. Інтернет-магазин повинен містити перелік товарів, з презентабельними

візуальними блоками до кожного товару, таблиці цін та опису, кошик та лічильник до нього і коментарі.

Сайт призначений для швидкого пошуку та купівлі потрібного товару за допомогою мережі Інтернет. На даний час все більше людей замовляють на різних сайтах доставку їжі, послуги та товари. Це зручно як для замовника так і для постачальника. Від замовника достатньо обрати необхідну послугу/товар, а для постачальника лише відправити її.

Розробка інтернет-магазину передбачає наявність таких функціональних елементів або розділів:

– Каталог продукції. Даний розділ є основним і містить в собі докладні описи тих чи інших товарів. У більшості випадків сторінки каталогу і картки товарів супроводжуються візуальною інформацією (фото).

– Кошик. Скориставшись цим інструментом, користувач інтернет-магазину може зібрати і оформити замовлення на придбання цікавлять його товарів. Кошик містить інформацію, яка стосується всіх покупок зареєстрованого користувача.

– Реєстрація. У функції даного розділу входить збір інформації про клієнтські контакти, а також надання постійним клієнтам додаткових послуг інтернет-магазину [4].

Сам інтернет-магазин може класифікуватися за різними видами, основними з них є:

- моделлю;
- продажами;
- доходом;
- асортиментом;
- подачею.

Модель інтернет-магазину визначає в якому напрямку буде розроблюватися інтернет-магазин. Так він може працювати тільки онлайн, наприклад зв'язуватися з робітником і методом спілкування обиратися товар чи послуга. Можливий розвиток і в напрямку аутсорсингу, тобто власник такого

інтернет-магазину не займається доставкою, купівлею чи комплектацією. Він передає зроблені замовлення тій компанії, з якої він має договір. Або ж інтернет-магазин може просто виставляти позиції товарів і при їх купівлі здійснювати пошук на інших інтернет-магазинах і купівлі на них, тобто виступати посередником між іншим магазином та клієнтом.

З цього слідує класифікація по продажу. Чи займається інтернет-магазин роздрібним продажом або оптовим. Чи надає він свої послуги певному клієнту чи цілому підприємству, як в прикладі посередника.

Дохід інтернет-магазину залежить від виду діяльності, так наприклад він може продавати товари своєї компанії або іншої за партнерською програмою. Ще один з найприбутковішим є продаж цифрової інформації, оскільки для неї не потрібно використання фізичних речей окрім самого сервера на якому і буде знаходитися та чи інша інформація.

Асортимент товару залежить від самого власника, якщо він має власний бізнес, то і буде його продавати. Також можна заключити домовленість між іншими партнерами по продажу їх послуг.

Подача грає важливу роль, бо чим привабливіший дизайн, опис та простота використання інтернет-магазину тим більше клієнтів будуть повертатися для купівлі товару саме в цій мережі.

Купівля в інтернет-магазині може відбуватися, як одразу, так і у вигляді аукціону. Одразу купити можна товар який є ходовим, а у випадку з аукціоном виставляється товар чи послуга, яка є особливою за певними факторами. В такому вигляді спочатку пропонується початкова ціна і якщо покупець хоче придбати товар чи послугу погоджується на таку ціну або підвищує її. В результаті товар чи послуга надається користувачу, який запропонував найбільшу ціна за певний проміжок часу. Такий вид продажу потребує більш детальної інформації про користувача та його можливостей (кількості валюти, якої він оперує).

1.3 Етапи та методи розробки

Розробка інтернет-магазину це складний, відповідальний та довгий процес. Існує декілька етапів завдяки яким вдасться створити інтернет-магазин [5]:

- Визначення цілей.
- Обрання домену та хостингу.
- Розробка технічного завдання.
- Написання тексту.
- Створення дизайну.
- Оптимізація.
- Верстка.
- Наповнення контенту.

Кожен з цих етапів дає можливість розробнику мінімізувати витрати та помилки, які можуть виникнути в процесі та не витратити час на непотрібну роботу при розробці. Завдяки конкретному плану розробки можна швидше та простіше отримати очікуваний результат, тобто сам інтернет-магазин.

Основні методи розробки [6]:

- Ручна розробка, базована на HTML.
- За допомогою спеціальних додатків для розробки сайтів.
- Інструментальні інструменти типу CMS.
- Застосування фреймворків.
- Платформи Saas.

Кожен з цих методів має свої переваги та недоліки. Так ручна розробка надає розробнику великий спектр можливостей, щоб втілити задуману ідею та навіть більше але потребує особливих знань та навичок для розробки. Якщо розробник тільки почав свою кар'єру можна використати Saas платформи. Вони не вимагають якихось спеціальних знань чи навичок але якщо необхідно зробити щось, що виходить за шаблон платформи це реалізувати не вдасться. Що ж стосується CMS інструментів, вони здатні до розробки сайту та підключенню окремих функцій типу нарощування різних бібліотек, зроблених під ці

інструменти. При цьому цей плюс є одночасно і його мінусом, оскільки кожна бібліотека займає досить багато місця і при фінальній версії сайт може важити 500 мб., в той час, як сайт написаний в ручну займатиме не більше 50 мб., але це залежить від його функціоналу та можливостей [7].

Серед всіх методів найкращим вибором для розробки інтернет-магазину є ручна розробка, оскільки можна втілити всі поставлені цілі та в майбутньому додати більше функціоналу або змінити існуючий.

Ручна розробка базується на написанні скриптів, оформлення та розмітки. Базова мова розмітки є HTML, вона слугує «скелетом» всього сайту. Далі створюють скрипти для отримання різних функцій, з аналогією це є «внутрішні органи та м'язи» інтернет-магазину. Можна писати плагіни в самому тілі програми або підключати свої, або вже кимось написані. Переважно для цього використовують мови програмування такі як: PHP, Python, C#, Ruby. Їх є досить багато і кожна з них має свої переваги та недоліки.

І на кінець відбувається оформлення, тобто «шкіра». Саме оформлення є окремим навиком, який може привабити широку аудиторію потенційних покупців. Воно здійснюється написанням скриптів CSS та JavaScript.

1.4 Постановка задачі на розробку інтернет-магазину

Метою кваліфікаційної роботи є розробка інтернет-магазину. Реалізація даного проекту включає 5 етапів:

- Верстка.
- Написання скриптів, бекенду.
- Створення бази даних.
- Адаптація інтернет-магазину під більшість популярних браузерів.
- Тестування.

Сам інтернет-магазин складається з декількох функціональних частин:

- Форма реєстрації/авторизації.
- Форма купівлі товару.

- Каталог товарів.
- Пошукова система.
- Корзина.
- Форма товару.
- Адмін-панель.
- Наповнення контентом.
- Переадресація на методи оплати.

Одним з важливих елементів є оплата товару, її здійснення можливо власноруч або використовуючи існуючі API. Такий API надається таким сервісом, як LIQPAY. В цьому сервісі є не тільки «кнопка», яка додається на сайт, а також платіжні боти в популярних сервісах, тобто людина може здійснити покупку, наприклад, через Telegram. При написанні власних плагінів для оплати слід приділити велику увагу захисту, оскільки це може поставити під загрозу особисті данні клієнтів, в наслідок чого їх гроші можуть бути втрачені.

Сервіс LIQPAY має зручний прийом платежів, а саме:

- з карт Mastercard / Visa;
- готівкою;
- інтернет банкінг Приват24 [8].

Ще один важливий елемент це крос-платформність інтернет-магазину. Він повинен однаково відображатися в найпопулярніших браузерах:

- Opera (76.0.4017.123).
- Mozilla (88.0.1).
- Google Chrome (90.0.4430.93).

Для розробки інтернет-магазину було обрано такі програмні засоби:

- HTML 5.
- CSS3.
- PHP 7.5.
- MySQL 8.0.
- OpenServer 5.2.7.
- JavaScript v8.

– PhpMyAdmin 5.0.2.

Всі технології є взаємо пов'язані та найкраще підходять для вирішення поставлених задач при розробці інтернет-магазину.

1.5 Мова розмітки гіпертекстових документів HTML

HTML [9] – скорочення від «HyperText Mark-up Language» – перекладається, як «Мова розмітка гіпертексту». Іншими словами HTML – це мова розмітки, або ще один спосіб зберігання інформації. Це мова розмітки, яка каже браузеру, як відобразити сторінку, яка буде відвідана. HTML може бути простим чи складним, залежно від бажання його розробника. HTML складається з серії елементів, які використовуються для того, щоб вкласти, загорнути чи розмітити частини сторінки, щоб вони мали певний вигляд. Теги можуть перетворити частину тексту на посилання на іншу сторінку, виділити її курсивом тощо. Базова будова такого документа (Див. лістинг 1.1).

Лістинг 1.1 – Базова структура тегів HTML документа

```
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>Перша сторінка</title>
</head>
<body>

</body>
</html>
```

Перший тег «html» дозволяє браузеру зрозуміти, що цей документ в якому міститься мова розмітки та атрибут «lang», який вказує на мову самою сторінки (в прикладі українська).

Тег «head» діє як контейнер для всього матеріалу, який потрібно включити в сторінку HTML, але який не є вмістом, що надається для перегляду відвідувачам сторінки. Він включає в себе такі речі, як ключові слова та опис

сторінки, які будуть відображені в результатах пошукових систем, оголошення підтримуваного набору символів («meta charset»), тощо.

В тегу «body» розміщуються всі елементи для виводу на екран користувача [10].

1.6 Каскадні таблиці стилів CSS

CSS скорочення від «Cascading Style Sheets» [11] – перекладається, як «Каскадні таблиці стилів». CSS являє собою спеціальну мову стилів для опису зовнішнього вигляду веб-сторінки. Його можна поділити на 2 частини. Перша – це селектори, вони дозволяють визначити до яких частин HTML коду потрібно буде застосовувати правила. Друга – це самі правила які і будуть вказувати на те, як саме має виглядати елемент до якого їх застосовують.

Є три основних способи використання CSS та HTML [12]:

- Всередині тегів HTML.
- Всередині тегу «head».
- Окремим файлом.

Перші два способи є подібними але їх недолік полягає в збільшенні коду, що негативно впливає на читабельність коду.

Третім способом створюється окремий файл, що дозволить розділити окремо код для зовнішнього вигляду та код для розмітки елементів. Саме тому перевагу має третій спосіб, це є одна з причин розробки CSS.

Ще одна властивість підключення CSS окремим файлом надає розробнику можливість використання одного і того самого коду повторно для декількох сторінок. Тобто маючи однакові стилі тексту, зображень та інших параметрів можна використати їх повторно для іншої сторінки або окремого блоку. Це економить час написання сайту та швидкість роботи самого сайту тому, що браузеру не потрібно буде підвантажувати «нові» стилі оскільки він вже їх має [13].

CSS є базовим інструментом для оформлення але має і складніші функції типу анімації або оформлення під різні типи браузерів. Оскільки в кожному браузері можуть виникати проблеми з зовнішнім виглядом сайту та їхніми властивостями існують спеціальні теги під такі типи браузерів:

- Chrome (webkit).
- Firefox (moz).
- Internet Explorer(ms).
- Opera (o).
- Android (webkit).
- iOS (webkit).
- Safari (webkit).

Для кожного з них можна прописати свої особливості та позбутися небажаних відмінностей в перегляду одного і того самого сайту на різних браузерах. Окрім того можна спеціально зробити відмінності між браузерами для окремого функціоналу чи просто «фішок».

1.7 Мова програмування PHP

PHP скорочення від «Hypertext Preprocessor» [14] – перекладається, як «Гіпертекстовий препроцесор». PHP – це мова спеціально розроблена для веб-додатків. Це одна з най популярніших мов для написання інтернет-магазинів та сайтів в цілому. Його популярність полягає в можливості спілкування з БД (MySQL, PostgreSQL, SQLite) готовими функціями та обробки мережеских запитів. Також одна з особливостей PHP це його гарна сумісність з HTML кодом. В самому документі він виділяється спеціальними тегами (Див. лістинг 1.2).

Лістинг 1.2 – Теги мови PHP

```
<?php  
    echo "Я Терещук Віктор!";  
?>
```

Вони дозволяють перемикатися між розміткою та логічним кодом. PHP доступний для більшості операційних систем Linux, Unix, Windows, MacOS та інших. Також є підтримка веб-серверів, як Apache та IIS. Таким чином він надає розробнику свободу вибору ОС і сервер, який можна розгорнути на локальному ПК.

1.8 Мова програмування JavaScript

JavaScript являє собою динамічну об'єктно-орієнтовану мову програмування [15]. Виконується вона на стороні клієнта та дозволяє обмінюватися інформацією з сервером і змінювати інтерфейс веб-сторінки. Його основний функціонал це створення «живого» інтерфейсу для браузера, наприклад спливаючі вікна, анімації, кнопки при натисканні на які відбувається їх зміна.

Код який написаний на JS називається сценарієм. Виконання коду може відбуватися компіляцією або інтерпретацією. Різниця між ними полягає в тому, що компілятор отримує весь код і при його успішності виконується або виводяться помилки. При інтерпретуванні машина отримує код по рядкам і одразу виконує його. В усіх сучасних браузерах, таких як Chrome, Firefox є вбудовані інтерпретатори. Саме тому браузери здатні виконувати код написаний на JS в реальному часі прямо на сторінці, завдяки чому і створюється така велика швидкість роботи. [16]

На JS можна створити будь-яку технічну ідею починаючи від чату в якому можна переписуватися в реальному часі, закінчуючи створенням роботів, які будуть думати та діяти в залежності від написаного коду чи цілими іграми.

Так для зміни заголовку тексту слід створити файл із розширенням html та стандартною структурою, після чого підключити сам JS (Див. лістинг 1.3).

Лістинг 1.3 – Підключення JS

```
<script src="main.js"></script>
```

Цей рядок потрібно підключити між відкриваючим та закриваючим тегом `body`. Після підключення файлу потрібно внести виконавчий код, який і буде взаємодіяти з `html`-сторінкою. Для цього в самому файлі `main.js` потрібно прописати наступне (Див. лістинг 1.4):

Лістинг 1.4 – Взаємодія з `html` сторінкою

```
var myHeading = document.querySelector('h1');  
myHeading.textContent = 'Tereshchuk Viktor!';
```

Це створить рівень заголовку `h1` та змінить текст на «Tereshchuk Viktor!».

Окрім викликання функцій в `JS` є стандартні математичні оператори та логічні. При використанні умовних конструкцій якраз і використовуються логічні оператори, так наприклад для перевірки холодно чи жарко можна написати конструкцію (Див. лістинг 1.5).

Лістинг 1.5 – Приклад умовної конструкції в `JS`

```
var iceCream = 'cold';  
if (iceCream === 'hot') {  
    alert('This is hot!');  
} else {  
    alert('This is cold!');  
}
```

В прикладі створюється змінна з значенням `cold` і за допомогою умовного оператора `if` порівнюється з значенням `hot` [17]. Оскільки при перевірці умова є `false` виконається 2 частина з повідомленням «This is cold!».

1.9 MySQL та бібліотека `RedBeanPhp`

`SQL` скорочення від «Structured Query Language» [18] – перекладається, як «Мова структурованих запитів». `MySQL` використовує дану мову та має власні функції для забезпечення додаткового функціоналу. Її працездатність полягає в

спілкуванні з базою даних та ці запити є основними. Працює вона за системою «клієнт-сервер». Тобто клієнт використовуючи інтерфейс може отримати відповідь з бази даних.

Бібліотека RedBeanPhp значно спрощує взаємодію з базою даних для розробника використовуючи MySQL і не тільки. Код написаний за допомогою цієї бібліотеки є більш структурований та меншим ніж написаний на чистому MySQL. Так наприклад для видалення таблиці на MySQL буде виглядати наступним чином (Див. лістинг 1.6):

Лістинг 1.6 – Видалення таблиці

```
CREATE TABLE users
( Id INT, Age INT, Name VARCHAR(30), LastName VARCHAR(30) );
TRUNCATE TABLE users;
```

Ця сама дія, написана з допомогою бібліотеки, виглядатиме так (Див. лістинг 1.7):

Лістинг 1.7 – Видалення таблиці з використанням бібліотеки

```
R::dispense(users);
R::wipe(users);
```

Однозначно код написаний з допомогою бібліотеки RedBeanPhp є більш зрозумілим та інтуїтивно зрозумілішим при розробці. Також бібліотека автоматично створює типи даних, якщо їх не було вказано. Тобто додавати інформацію можна одразу після створення пустої таблиці.

1.10 Локальний сервер Open Server

При розробці інтернет-магазину обов'язково виникне проблема з взаємодією користувача і БД. Необхідна буде мова програмування, яка буде виконуватися на стороні сервера. Отже і при розробці необхідно буде мати сервер. Існує два способи вирішення:

- Тримати окремий сервер.
- Використовувати емуляцію сервера.

Переважно при розробці використовують другий спосіб, а перший лише для готового проекту. Мінус першого способу в тому, що при написанні коду розробник хоче бачити одразу результат його виконання. Маючи віддалено сервер потрібно буде кожен раз завантажувати написаний код на хостинг. Це займає не так багато часу але виконання її кожного разу при написанні 1-2 строк коду буде суттєво збільшувати час розробки. При використанні локального сервера цього робити це не потрібно, адже він автоматично завантажує щойно написаний код та опрацьовує його.

Локальний сервер дозволяє web-розробникам реалізувати сайт на своєму комп'ютері без виходу в Інтернет. Потреба в їх використанні постійно зростає, особливо при розробці динамічних сайтів на PHP [19].

Для розробки та тестування інтернет-магазину в локальній мережі одним з кращих варіантів є Open Server. Це портативний локальний WAMP/WNMP сервер, що має багатофункціональну керуючу програму і великий вибір компонентів, що підключаються [20].

Open Server має такі особливості:

- детальний перегляд логів всіх компонентів в реальному часі;
- вибір HTTP, СУБД і PHP модулів в будь-якому поєднанні;
- підтримка SSL і кирилических доменів з коробки;
- підтримка аліасів або по іншому доменних покажчиків, а так само зручна форма їх налаштування;
- створення локального під-домена без втрати видимості основного домену в мережі інтернет;
- доступ до доменів (в один клік) і швидкий доступ до шаблонів конфігурації модулів;
- багатомовний інтерфейс (Українська, Білоруська, Англійська).

В Open Server існує три режими управління списком доменів: автопошук, ручне управління і ручне + автопошук. За замовчуванням використовується

перший режим автоматичного підключення папок з кореневої директорії зазначеної в налаштуваннях. Програма сканує задану веб-директорію на наявність папок з доменами, після чого в кожній знайденій папці проводиться пошук підпапок (кореневої папки домену), які вказані в налаштуваннях для автосканування. Якщо жодна з передбачуваних кореневих підпапок не знайдена, то коренем домену стає сама папка з доменом. При використанні суміщеного режиму управління доменами програма спочатку підключає домени створені вручну, після чого проводиться автоматичне сканування за процедурою описаною вище.

1.11 Висновки до першого розділу

Отже, після проведення аналітичного огляду існуючих рішень можна зробити висновок, що інтернет-магазин є важливим елементом для ведення бізнесу. Для потенційних покупців це є проста та швидка можливість оформити товар або просто переглянути і порівняти їх на інших сайтах чи дізнатися про їх доступність. Сам інтернет-магазин передбачає наявність таких функціональних елементів: каталог продукції, кошик, реєстрація.

Розробка інтернет-магазину повинна відбуватися згідно етапів. Це дозволить уникнути більшості проблем та спростить саму розробку поділяючи її на певні етапи.

Постановка задачі на розробку інтернет-магазину дозволить поетапно поділити написання коду та саму його реалізацію.

Для розробки та області застосування інтернет-магазину згідно етапів та методів розробки найкраще відходять технології: HTML, CSS, PHP, JavaScript, MySQL та бібліотека RedBeanPHP для роботи з БД і локальний сервер OpenServer.

РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОЕКТУ ІНТЕРНЕТ-МАГАЗИНУ

2.1. Проектування структури та дизайну інтернет-магазину

Для початку робіт над дизайном сайту, у першу чергу необхідно скласти структуру сайту, тобто визначити, які основні розділи міститиме інтернет-магазин, чи будуть додаткові меню чи підменю. Це дозволить розробнику при створенні дизайну не помилися чи забути якісь елементи інтернет-магазину.

Саме визначення структури інтернет-магазину являє логічне розбиття на певні блоки, кожен з цих блоків повинен розміщуватися згідно свого пріоритету над іншим. Використовується така схема для оцінки загальної картини функціонування сайту [21]. Найбільш оптимальним вирішенням для побудови структури інтернет-магазину є схема, представлена на рисунку 2.1. Вона враховує всю специфіку та вимоги до інтернет-магазину.

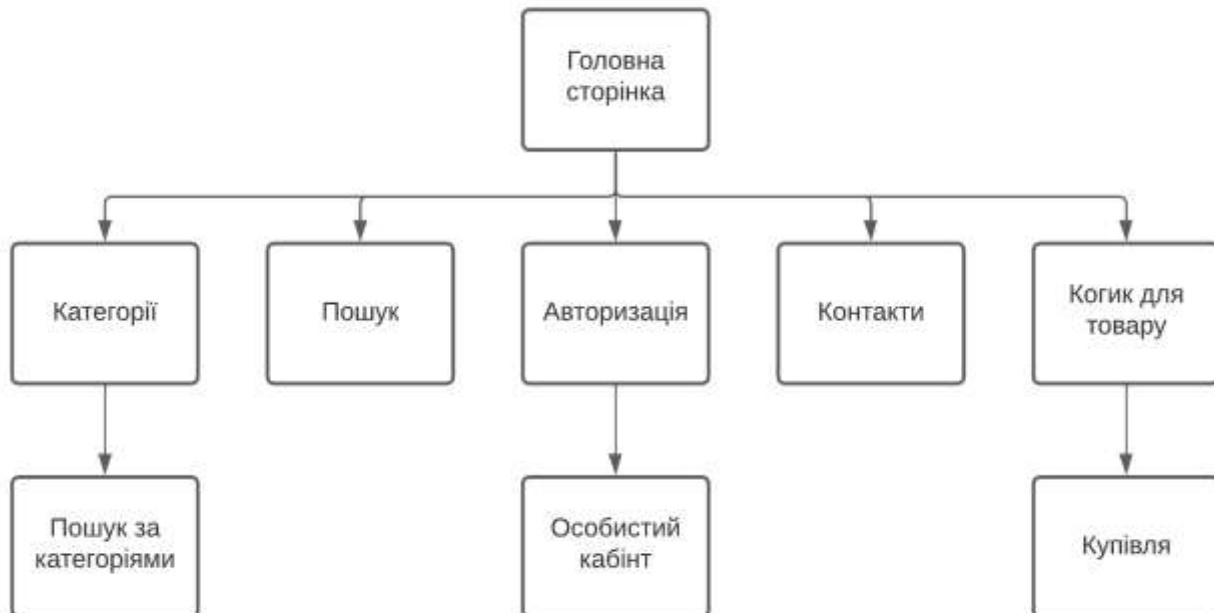


Рисунок 2.1 – Структура інтернет-магазину

Умовно сайт можна поділити на 3 частини це: шапка, тіло, та підвал, тобто кінець сайту.

Зверху інтернет-магазину розміщено «шапку» (Див. рисунок. 2.2), в ній є 6 посилань:

- Головна сторінка.
- Категорії.
- Авторизація.
- Реєстрація.
- Кошик.
- Пошук.

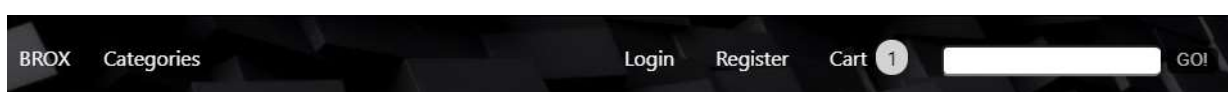


Рисунок 2.2 – Шапка інтернет-магазину

Наступною частиною інтернет є основна частина для показу наявних товарів (Див. рисунок 2.3).

Вона містить всі основні структурні елементи, перехід по яких здійснюється за допомогою гіперпосилань.

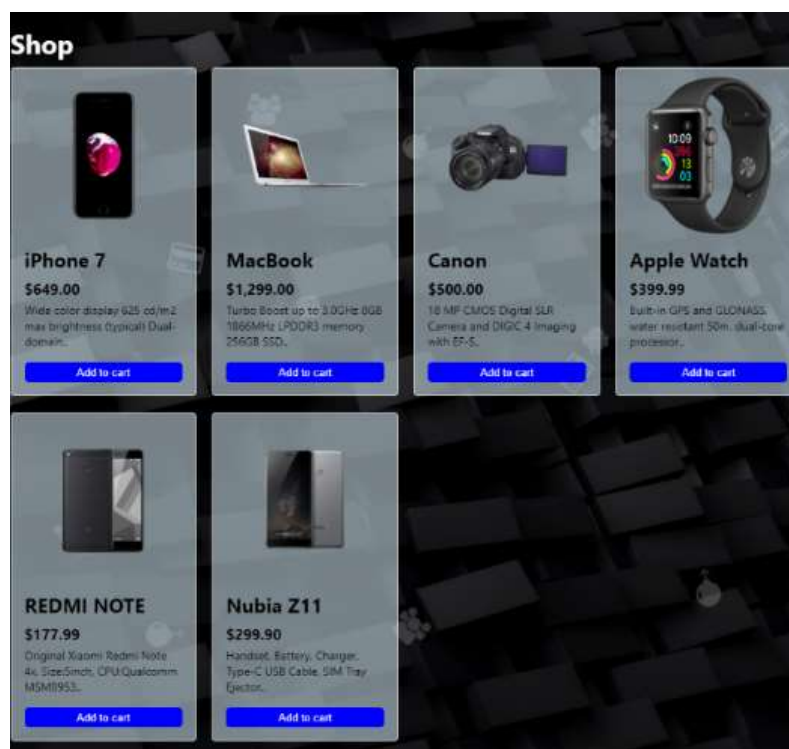


Рисунок 2.3 – Товари інтернет-магазину

І «підвал» інтернет магазину в якому знаходиться інформація про компанію (Див. рисунок 2.4).



Рисунок 2.4 – «Підвал» інтернет–магазину

В залежності чи авторизований користувач чи ні повинна змінюватися лише основна частина інтернет магазину. Інші ж частини залишаються не змінними, окрім додавання до шапки посилання на особистий кабінет авторизованого користувача.

Перші кроки створення дизайну починаються в графічному редакторі. В інтернеті існують сервіси для створення дизайну онлайн такі як: Flexbe, ULanding, LPgenerator, LPgenerator, Wix, uKit та багато інших. Також дизайн можна намалювати в Photoshop або Illustrator, в яких можна редагувати все, що потрібно.

Розробник повинен розробити один або декілька варіантів інтерфейсу, для його майбутнього втілення. Також необхідно створити візуальний вигляд головної сторінки, меню та інших сторінок згідно технічного завдання.

2.2. Функціональні можливості

Інтернет-магазин працює у деякому контексті, що визначає його зовнішнє оточення. Таке оточення формують користувачі. Кожен з користувачів взаємодіє з інтернет-магазином за своєю власною схемою та очікує від неї певної реакції.

UML діаграма можливостей користувача (Див. рисунок 2.5). Вона побудована на основі вимог до інтернет–магазину і показує можливості користувача.

Актор «Користувач» з індексної сторінки може перейти до «шапки» сторінки, де знаходиться: кошик, авторизація та реєстрація, сортування за критеріями та пошук товарів.

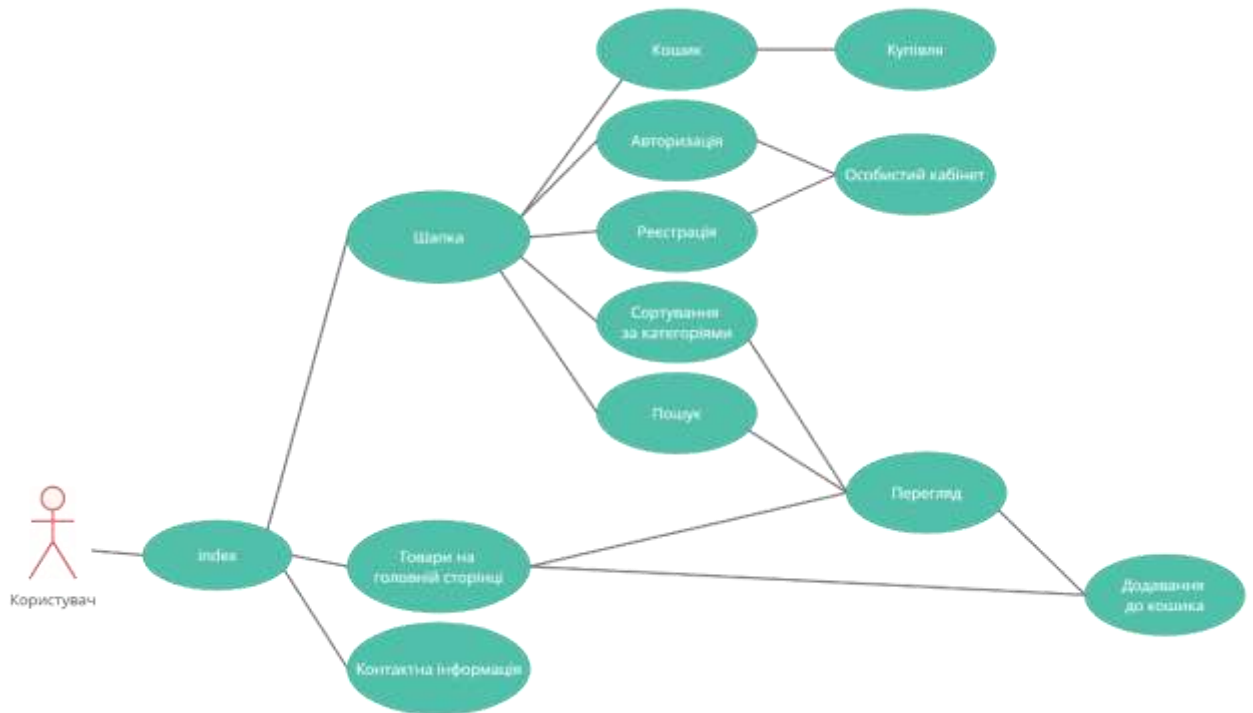


Рисунок 2.5 – UML–діаграма можливостей користувача

З кошика можна одразу перейти до купівлі, після чого користувача буде направлено на сторінку оплати, якщо користувач зареєстрований або на сторінку введення форми для доставки товару, якщо користувач не зареєстрований.

Після авторизації чи реєстрації користувача буде направлено на відповідну форму з перевітками на правильність вводу та обробкою інформації, після успішної перевірки користувачу буде доступний особистий кабінет, в якому він може змінити пароль.

Також з індексної сторінки він може перейти до товару, щоб подивитися його детальний опис або одразу додати товар до кошика. І в кінці сторінки «Користувач» може побачити контактну інформацію.

UML діаграма можливостей адміністратора представлена (Див. рисунок 2.6). Вона є майже ідентичною, як і для користувача, однак має декілька змін. Ці зміни стосуються переваг адміністратора над звичайним користувачем.

На такий діаграмі наглядно видно які дії потрібно зробити адміністратору, щоб вплинути на БД для отримання очікуваного результату.



Рисунок 2.6 – UML–діаграма можливостей адміністратора

Актор «Адміністратор» має всі ті самі можливості, що і користувач і також: можливість перегляду списку замовлень, видалення, додавання та зміна властивостей товару.

2.3. Написання плагіну для кошику товарів

Плагін – це незалежно компільований програмний модуль, що динамічно підключається до основної програми і призначений для розширення і використання її можливостей [22]. Плагін дозволить реалізувати відсутні можливості інтернет-магазину. Функції, які виконують плагіни – удосконалення базових можливостей сайту.

Плагін як модуль або доповнення може бути підключений до інтернет-магазину і відключений від нього без критичних наслідків в роботі основного функціоналу. Якщо в інтернет-магазині не вистачає будь-якої функції, немає необхідності повністю переписувати код, досить встановити необхідний плагін

для її реалізації. Модулі, як правило, не викликають перевантажень і мають досить малий обсяг.

Якщо в коді інтернет-магазину прописати абсолютно весь функціонал, то такий файл буде займати занадто багато місця, а сам ресурс навряд чи буде швидким чи динамічним. Модулі дозволяють не використовувати великі повнофункціональні програмні коди з громіздким функціоналом, а обійтися набором невеликих плагінів, необхідних проекту для виконання конкретних цілей.

Один шаблон, а відповідно і його функціонал, може використовуватися різними розробниками для різних цілей. Встановлення плагінів дозволяє проект зробити особливим і підлаштувати його під конкретні завдання.

В дипломному проекті для інтернет-магазину «BROX» було розроблено плагін для обрахування товарів та їхньої ціни. Також він дає змогу:

- одразу збільшити або зменшити кількість товару, видалити його з корзини, це також можна зробити зменшенням, якщо залишився 1 товар;
- вивести назву, ціну та скорочений опис товару;
- обрахувати загальну ціну та вивести її;

Інтерфейс даного плагіну наведено на Див. рисунок 2.7, а код в додатку А.

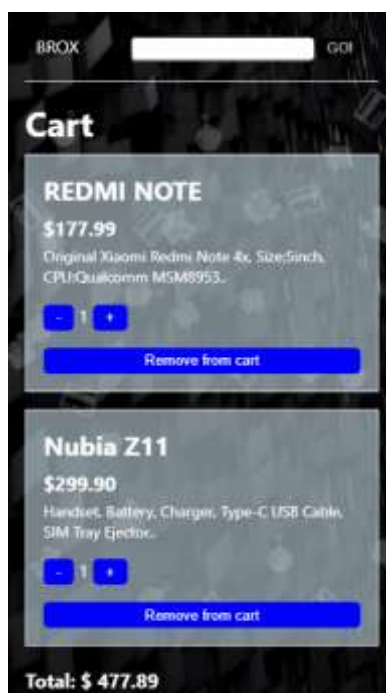


Рисунок 2.7 – Робота плагіну для корзини

Фрагмент коду, який відповідає за перевірку та видалення товару з кошика (Див. лістинг 2.1).

Лістинг 2.1 – Перевірка та видалення товару

```
if (isset($data['decrement'])) {
    if ($_SESSION['cart'][$data['product_id']] < 2) {
        unset($_SESSION['cart'][$data['product_id']]);
    } else {
        $_SESSION['cart'][$data['product_id']]--;
    }
}
```

Даний плагін перевіряє чи була натиснута кнопка «+» і якщо це сталося він додає 1 одиницю товару. Також це так само працює з відніманням, але окрім перевірки, є ще одна перевірка чи товару є менше ніж 2, тоді він видаляє товар з корзини. Також, плагін обраховує ціну товару згідно його кількості, тобто виводить інформацію про ціни, а саме ціну товару множить на їх кількість. Плагіни значно спрощують розробку інтернет-магазину та автоматизують дії при роботі з ним.

2.4. Проектування структури каталогів інтернет-магазину

Структуру каталогів інтернет-магазину (Див. рисунок 2.8). Серед усіх файлів є 4 розширення md, sql, css та php.

В файлі з розширенням css знаходяться стилі для основних елементів інтернет-магазину.

В файлі images знаходяться зображення, які будуть завантажуватися до позиції товару з інтерфейсу інтернет-магазину.

Файли мають різне функціональне призначення, одні з них заносять дані і працюють з ними, інші відображають дані, які користувач хоче побачити і приймають вхідні дані від користувача через елементи графічного інтерфейсу,

деякі файли призначені для того, щоб працювати із вхідними даними, реагувати на дії користувачів, викликати відповідні сторінки, функції, модулі, здійснювати контроль над інтернет-магазином.

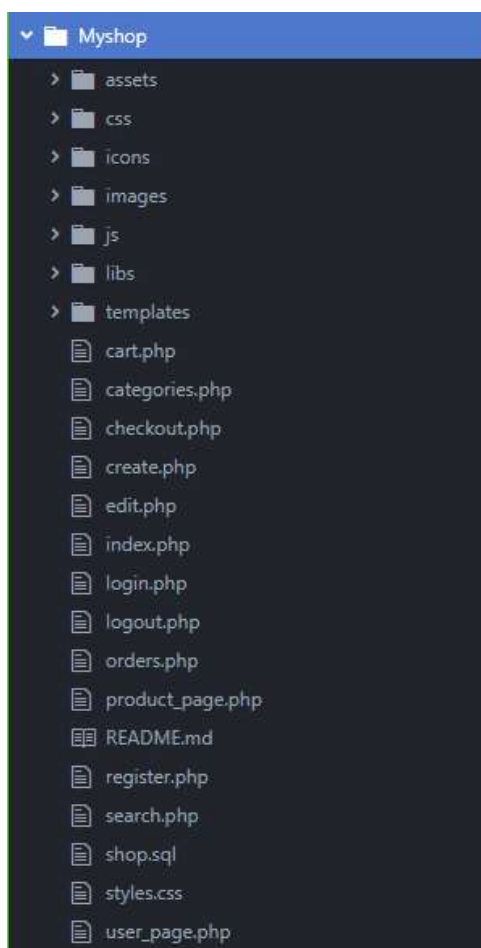


Рисунок 2.8 – Структура каталогів інтернет-магазину

До основних елементів інтернет-магазину можна віднести каталоги і файли, з допомогою яких можна виконувати дії і здійснювати необхідні операції. Кожний елемент має своє призначення [23], наприклад:

В каталозі templates файл header.php відповідає за завантаження початку сторінки і використовується для підключення до більшості сторінок інтернет-магазину. В цьому ж каталозі є файл footer.php, який відповідає за завантаження «підвалу» сторінки. В самому файлі знаходиться декілька строк коду (Див. лістинг 2.2) для виводу інформації про інтернет-магазин та закриваючі теги.

Лістинг 2.2 – Код файлу footer.php

```

</div>
<div id="footer">
<ul class="info">
<li class="info">Address:</li>
<li class="info">Tel:</li>
<li class="info">Email:</li>
<li class="info">Tel:</li>
</ul>
</div>
</div>

```

Файл register.php відповідає за вивід, заповнення та перевірки форми реєстрації. У випадку успішної реєстрації користувача буде добавлено до БД та авторизовано, тобто записано до сесії (після деавторизації ця сесія видаляється).

Файл index.php призначений для виводу головної сторінки інтернет-магазину.

В каталозі libs знаходиться 2 файли один з яких слугує файлом настройки доступу до БД, а інший для підключення бібліотеки (RedBeanPHP), яка допомагає працювати з БД. В каталозі js розміщено скрипти для анімації об'єктів на задньому фоні інтернет-магазину.

Доступ до завантаження сторінки з таких файлів як edit.php або create.php є лише в авторизованих адміністраторів, слугують для показу елементів керування товаром та зміною чи створення нового.

2.5. Опис програмних рішень

Була створена форма для реєстрації користувача. Її ціль додавання користувача до БД (Див. лістинг 2.3) після перевірки на всі помилки.

Лістинг 2.3 – Код додавання користувача до БД

```

if (empty($errors)) {
    $users = R::dispense('users');
    $users->name = $data['name'];
    $users->email = $data['email'];
    $users->address = $data['address'];
    $users->admin = 0;
}

```

```

    $users-
>password=password_hash($data['password'],PASSWORD_DEFAULT);

    R::store($users);
    $_SESSION['uname'] = $user->name;
    $_SESSION['uid'] = $user->id;
    header('Location: /index.php');
}

```

Спочатку оператором `if` перевіряється чи є відсутність помилки і, якщо умова правдива починається запис користувача до БД. Далі відбувається запис користувача до сесії, щоб можна було знаходитися в інтернет-магазині авторизованим.

Відповідно був розроблений скрипт для деавторизації користувача з інтернет-магазину. Після натискання на кнопку Logout виконується код (Див. лістинг 2.4).

Лістинг 2.4 – Код деавторизації користувача з інтернет-магазину

```

session_start();
unset($_SESSION['uname']);
unset($_SESSION['uid']);
unset($_SESSION['admincheck']);
unset($_SESSION['email']);
unset($_SESSION['address']);
session_destroy();
header('Location: /index.php');

```

В даному випадку сесія користувача видаляється, в результаті чого він стає гостем і на останній строчці відбувається перенаправлення на головну сторінку інтернет-магазину.

Скрипт для створення позиції товару (додаток Б) був розроблений для простоти експлуатації інтернет-магазину. Його роль заключається в підключенні до БД, перевірка прав доступу до форми та створення стрічки в БД з описаними даними в формі та шляху до зображення, яке завантажується на сервер в папку Images.

Для реєстрації користувачів було створену сторінку код якої зображено в додатку Г. В коді присутня стандартна перевірка та зв'язок з базою даних для

порівняння пошти та пароллю. При правильному їх введенні користувача перенаправить на головну сторінку з вже збереженою сесією, що дозволить користувачу одразу бути авторизованим без повторної авторизації.

2.6. Структура інтерфейсу адміністративної частини інтернет-магазину

Структурна схема інтерфейсу інтернет-магазину кишенькової електроніки визначає особливості взаємодії з адміністратором. Розроблена структурна схема наведена в додатку В.

На схемі можна побачити як працює частина адміністратора інтернет-магазину кишенькової електроніки. Так наприклад для редагування товару адміністратору потрібно авторизуватися в інтернет-магазині кишенькової електроніки обрати один з товарів після чого обрати дію редагування після чого ввести необхідні зміни. Щодо створення позиції нового товару необхідно перейти на головну сторінку з всіма товарами та додати позицію.

2.7. Розгортання та налаштування інтернет-магазину

Для встановлення інтернет-магазину необхідно перенести всі файли, які були розроблені в процесі дипломної роботи, в кореневу директорію хост-серверу після чого при потребі змінити налаштування версій та інших параметрів згідно інформації, яка знаходиться в файлі README.md.

Після даних маніпуляцій при заході на інтернет-магазин можна буде побачити його сторінку але з помилками зв'язаними з БД. Щоб їх усунути слід на хостинг панелі за допомогою РНРМуAdmin створити таблиці. Для цього після авторизації в РНРМуAdmin слід обрати пункт імпорт та завантажити файл shop.sql. В ньому знаходяться всі необхідні структури таблиць. Наступним кроком є налаштування доступу до БД. Це реалізовано через файл db.php, котрий знаходиться в папці libs. При його відкритті (Див. лістинг 2.5) на 3 строці, яка

починається з «R::setup», в дужках знаходиться параметри, котрі слід змінити на ті, які були вказані при авторизації до PhpMyAdmin.

Лістинг 2.5 – Код налаштування підключення до БД

```
<?php
require "rb.php";
R::setup( 'mysql:host=localhost;dbname=shop','root', 'root' );
session_start();

?>
```

Якщо назва БД була змінена також потрібно змінити її в параметрі `dbname`. Як результат на сторінці інтернет-магазину після оновлення повинні пропасти помилки і повернутися його функціональність (реєстрація, авторизація, деавторизація та інші можливості).

Також одною з основних налаштувань це є оплата товару. В файлі `checkout.php` потрібно знайти строку, яка починається з «`$json_string`» та змінити параметр «`public_key`» на той ключ який був створений в сервісі `Liqpay` та в строці «`infa1`» змінити «`private_key`» на той, що був виданий в сервісі.

Окрім цього рекомендовано створити одного користувача з адмін привілеями. Найпростіше це можна зробити знов зайшовши до `PHPMYAdmin` в таблицю `users` і створити нового користувача додавши ім'я, пошту (обов'язково потрібно ввести кінець пошти, `@gmail` тощо, адже при авторизації в інтернет-магазині форма не зможе пропустити акаунт без закінчення пошти), пароль, адресу та в полі `admin` слід замінити 0 на 1 (Див. рисунок 2.9). Також можна додавати більше адміністраторів тим самим способом.

Саме ця одиниця і дає доступ до адмін-привілеій. За замовчуванням при створенні нового користувача (реєстрації) це поле заповнюється 0.

Також, як можна побачити пароль користувача зберігається в шифрованому вигляді. Це наглядно показує, як важливо коли інформація є захищення. При відсутності такого захисту і, якщо БД була якимось чином скомпрометована, людина, яка володіє даною інформацією не зможе скористатися паролем.

Сервер: 127.0.0.1:3306 » База данных: brox » Таблица: users

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отображение строк 0 - 1 (2 всего, Запрос занял 0,0331 сек.)

SELECT * FROM `users`

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

+ Параметры

id	name	email	password	address	admin
1	123	123@gmail.com	\$2y\$10\$GAfTzskgRu4GZrUkc0aOGkU8D1m7mLrSC8IKlu051...	123	1
2	1234	1234@gmail.com	\$2y\$10\$8rAu2dHvCpJJURbC2pLHWe6UYn37wK88qppJlty5ZGc...	asd	0

Использование результатов запроса

Печать В буфер обмена Экспорт Отобразить графики Создать представление

Рисунок 2.9 – Таблица users в БД

Як результат після проектування структури інтернет-магазину було створено дизайн для кожної з сторінок. Описано основні функціональні можливості користувача та адміністратора. Детально описано створення плагіну для кошику товарів, а саме обрахування ціни, можливості змінювати кількість відповідного товару та перенаправлення користувача на сайт оплати товару з параметрами ціни. Спроектвана структура каталогів інтернет-магазину та опис деяких проблем та їх рішень, які виникали у ході розробки інтернет-магазину. Також коротко описано налаштування інтернет-магазину при його встановленні на хост.

2.8. Експлуатація інтернет-магазину

Вхід в інтернет-магазин здійснюється з ПК користувача, підключеного до мережі інтернет через веб-браузер за допомогою введення в адресному рядку адреси інтернет-магазину (Див. рисунок 2.10).

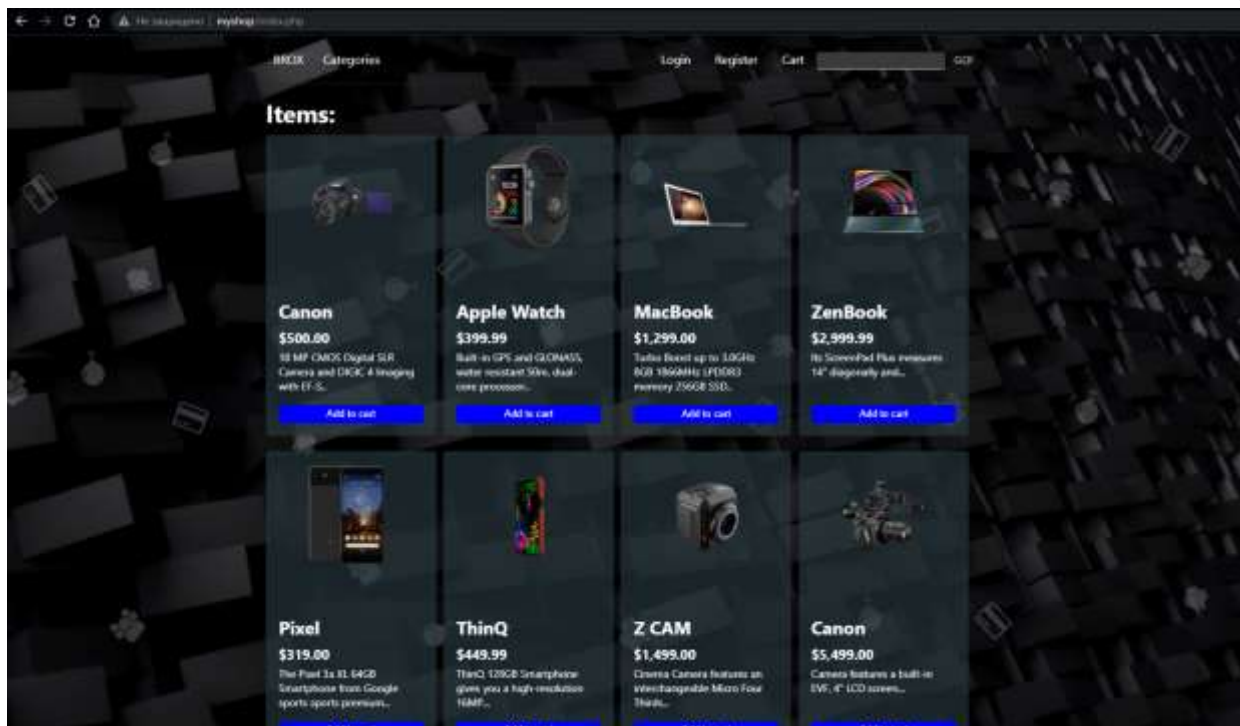


Рисунок 2.10 – Головна сторінка інтернет-магазину

Користувачеві надається можливість авторизації в інтернет-магазині через введення облікових даних – Логіна і Пароля або Реєстрації (при першому зверненні до магазину).

Для користувача, який не зареєстрований в інтернет-магазині кишенькової електроніки, після натискання кнопки реєстрація (Див. рисунок 2.11) надається можливість вказати персональну реєстраційну інформацію.

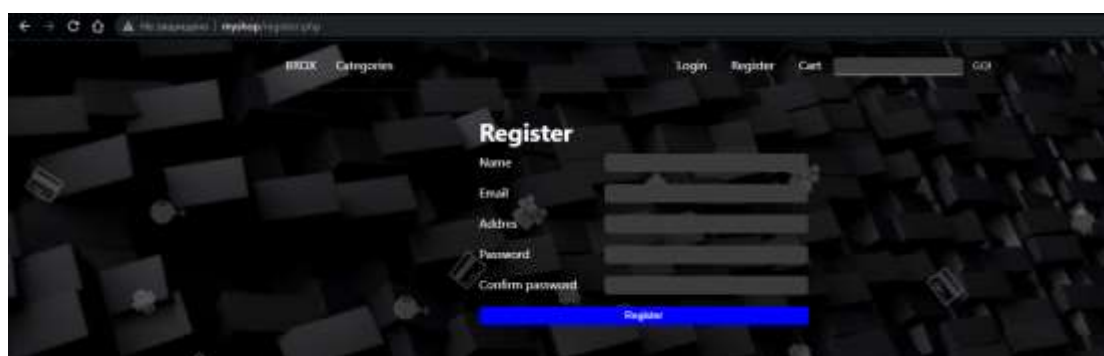


Рисунок 2.11 – Сторінка реєстрації нового користувача інтернет-магазину

Користувач має змогу переглядати товари, їх опис, ціну та фото. Якщо користувачу потрібно збільшити фото, щоб побачити його детальніше, слід

навести мишкою на зображення, після чого зображення товару в інтернет-магазині кишенькової електроніки збільшиться. При купівлі товару його перенаправляє на сторінку з оплатою товару, в якій необхідно ввести карту, переказ або оплатити готівкою з терміналу.

Стосовно адміністратора, то в нього є всі ті самі можливості як і в звичайного користувача. Для використання прав доступу та функціональних можливостей адміністратора слід авторизуватися від його імені та обрати необхідну дію.

Наприклад, щоб додати новий товар на головній сторінці є кнопка ADD (Див. рисунок 2.12), яка перенаправляє на форму створення позиції товару, після всіх внесених даних та збереження до БД вносить подана інформацію і товар з'являється в інтернет-магазині кишенькової електроніки. Щоб видалити товар слід натиснути на delete item.

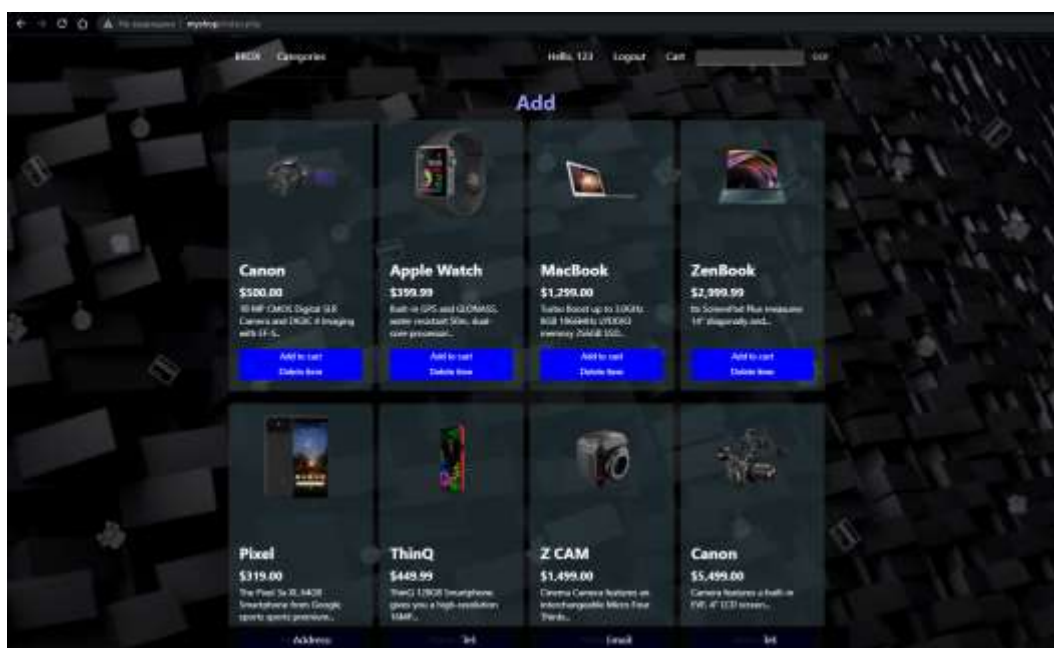


Рисунок 2.12 – Вид головної сторінки під авторизованим адміністратором інтернет-магазину

Для редагування потрібно зайти на обраний товар і натиснути кнопку edit після чого відкриється та сама форма, як і для створення але з вже введеними даними (Див. рисунок 2.13).

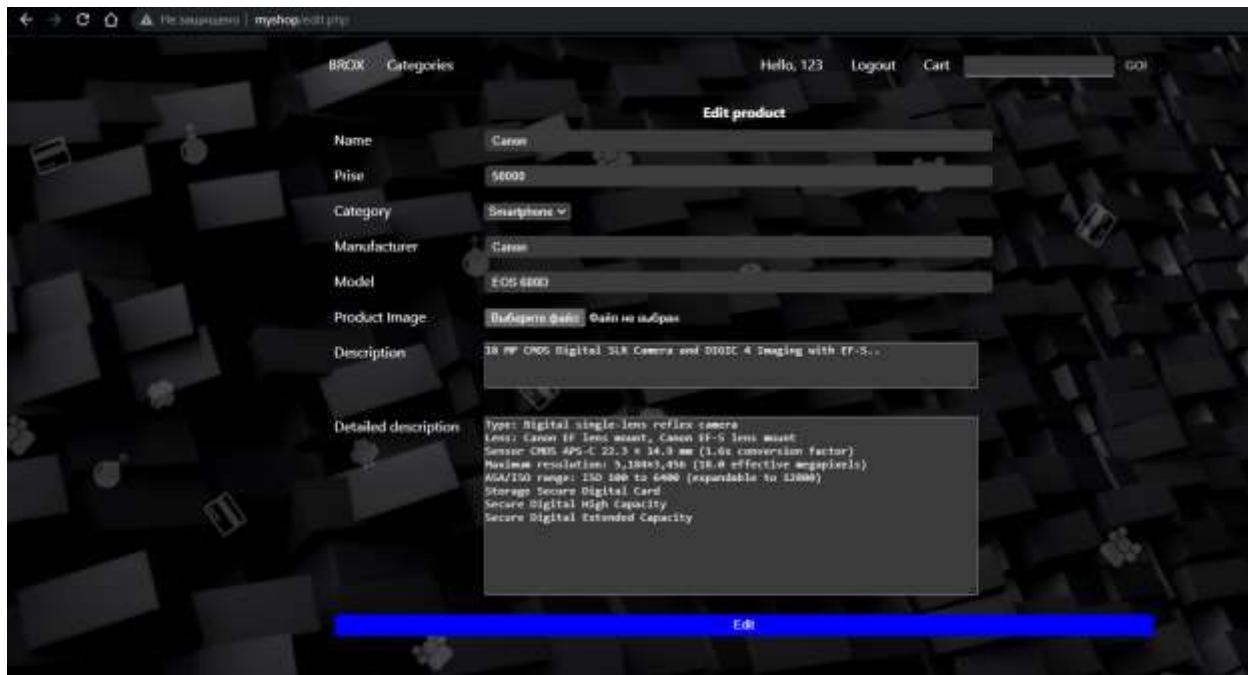


Рисунок 2.13 – Сторінка редагування товару інтернет-магазину

Товари які були оплачені користувачами також заносяться до БД і побачити їх може лише адміністратор з свого особистого кабінету. Для товару який вже був відправлений чи помилково оплачений є кнопка видалення.

2.9. Висновки до другого розділу

Проектування структури дозволяє розробнику наглядно зрозуміти, як саме будуть взаємодіяти блоки та сторінки між собою та який пріоритет вони мають між собою. Після цього розробляється дизайн, тобто створення дизайну сторінок та шаблонів для певних елементів інтернет-магазину.

UML діаграми графічно показують, які права та дії доступні для різних ролей. По ним можна зрозуміти, що може робити адміністратор або користувач і порівняти їх, щоб зрозуміти, які дії є спільні або різні для цих ролей.

Розробка плагіну спрощує розробку самого інтернет-магазину. Їх можна використовувати, як конструктор, щоб додавати різний функціонал. Так наприклад розробка плагіну для кошику товарів спрощує розробку, адже при

потребі якимось модифікувати чи змінити обробку дій потрібно буде змінювати тільки його не втручаючись в основний код інтернет-магазину.

Розгортання та налаштування інтернет-магазину є простою та чітко описана в п. 2.7. Достатньо просто перенести файли готового інтернет-магазину, налаштувати параметри самого хостингу, додати БД та змінити файл для коректної передачі ключів оплати, щоб гроші приходили на картку нового власника.

Експлуатація інтернет-магазину є простою та інтуїтивно зрозумілою, як для простого користувача так і адміністратора. Для користувачів доступна реєстрація і відповідно після неї кабінет користувача. В кабінеті адміністратора надається інформація про оформлені замовлення та редагування товарів в самому інтернет-магазині.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ

3.1. Долікарська допомога при кровотечах

Кровотеча може виникнути в будь-якої людини навіть при роботі за ПК під час виконання поставлених завдань розробникам. Зазвичай це капілярна кровотеча але може виникнути і масивніший крововилив. Необхідно знати базову допомогу при таких випадках оскільки це може врятувати чиєсь життя або своє власне.

Кровотеча виникає через порушення цілісності кровоносних судин. Загальний обсяг циркуляційної крові у дорослої людини - близько п'ять літрів. Водночас загрозою для життя є втрата понад 30% цього обсягу, особливо в короткі терміни. Кров зазвичай втрачається через рану. Раною називається пошкодження шкіри, слизової оболонки чи глибоких тканин, що супроводжується болем та кровотечею і має вигляд зяючого отвору [24].

Мікроби потрапляють до рани разом із предметом, яким заподіяно рану, а також із землею, шматками одягу, повітрям і при дотиках до рани руками. Найнебезпечнішими є мікроорганізми, що розвиваються у рані за умови відсутності повітря і спричиняють газову гангрену. Небезпечним ускладненням ран є зараження їх збудником стовбняка. Рани можуть бути поверхневими або проникаючими у порожнину черепа, грудну клітку, черевну порожнину. Проникаючі рани – найнебезпечніші. Профілактика зараження рани – найшвидше накладення на неї асептичної пов'язки, для запобігання доступу мікробів [25].

Пошкодження цілісності кровоносних судин внаслідок механічного або патологічного порушення призводять до кровотечі, які в свою чергу поділяються на:

- зовнішні - коли видно місце, звідки тече кров;
- внутрішні - коли кров виливається у внутрішні порожнини чи тканини.

В залежності від типу пошкоджених кровоносних судин кровотечі поділяються на:

- артеріальні;
- венозні;
- капілярні.

Артеріальна кровотеча характеризується яскраво-червоним кольором крові, кров б'є сильним струменем, поштовхами. Венозна кров має темно-червоне забарвлення, витікає з рани безперервно і повільно. При капілярній кровотечі кров виділяється краплями або сочиться з усієї поверхні рани. Кровотечу необхідно якнайшвидше зупинити. Для тимчасової зупинки артеріальної кровотечі здійснюють притискування артерії до кістки вище від місця поранення. Притискування здійснюють, як правило, у тих місцях, де артерія знаходиться неглибоко, декількома пальцями однієї чи обох рук, а іноді навіть кулаком [26].

Кінцівку в місці накладання гумового джгута обгортають марлею, рушником чи іншою тканиною, підіймають, джгут розтягують і роблять ним 2—3 оберти навколо кінцівки. Кінці джгута скріплюють за допомогою ланцюжка з крючком, а в разі їх відсутності - зв'язують. Якщо джгут накладено правильно, пульс нижче місця накладення зникає.

Тривалість використання джгутової пов'язки обмежується двома годинами, а взимку - однією годиною, інакше кінцівка мертва. Якщо протягом цього часу немає можливості забезпечити додаткову допомогу, то через 1,5-2 години джгут на кілька хвилин відпускають до почервоніння шкіри, кровотечу при цьому зменшують іншими методами, а потім знову затягують джгут, трохи відступивши від попереднього місця його накладання.

Капілярна кровотеча добре зупиняється стисною пов'язкою, перед чим шкіру навколо рани обробляють розчином йоду або спирту. Якщо з рани виступає сторонній предмет, його треба локалізувати і закріпити, для цього необхідно зробити у пов'язці отвір, інакше цей предмет може ще глибше проникнути всередину і викликати ускладнення.

Венозну кровотечу теж зупинити не дуже важко. При кровотечі через поранення руки чи ноги слід підняти пошкоджену кінцівку, щоб зменшити прилив крові. Утримуючи підняте положення кінцівки, слід продовжувати натискати на рану та максимально зігнути кінцівку в суглобі, обробити шкіру навколо рани розчином йоду, спиртом, горілкою, одеколоном, накласти стисну пов'язку і забинтувати, але насамперед необхідно подбати про запобігання зараженню хворобами, які передаються через кров, та вдягнути одноразові гумові рукавички.

При обробленні шкіри навколо рани не бажано тиснути на рану, якщо у ній є фрагменти кісток або сторонні предмети. У такому разі для зупинки кровотечі слід використовувати кільцеву подушечку. Зробити таку подушечку можна з вузького бинта, шматка тканини тощо. Насамперед слід зробити петлю, обмотавши тканину декілька разів навколо пальців. Потім вільний кінець тканини обмотати навколо петлі, поки не утвориться кільце [27].

У разі відсутності джгута накладають закрутку з пояса, рушника, хустки або іншого матеріалу, який не туго зав'язують навколо кінцівки. У петлю вставляють палицю і закручують. Для того, щоб не пошкодити шкіру, під закрутку необхідно підкласти бинт чи іншу тканину. Після накладання джгута чи закрутки потерпілого потрібно якомога швидше доставити до медичного закладу.

Часто кровотечі виникають внаслідок пошкодження м'яких тканин в результаті удару і є однією з ознак цього виду травми. Іншими ознаками удару м'яких тканин є болісні відчуття у місці удару, набряк, обмежена рухомість ушкодженої частини тіла.

Перша допомога у випадку пошкодження м'яких тканин за умови відсутності в цьому місці перелому чи вивиху - холод на місце удару і туга пов'язка на місце крововиливу та спокій ушкодженої частини тіла.

Серед усіх способів тимчасової зупинки кровотечі з великих артеріальних судин кінцівок найбільш надійним є накладання джгута. Його накладають поверх м'якої прокладки на стегно, плече, іноді на гомілку і передпліччя. Якщо

джгут накладений правильно, кровотеча зупиняється і периферійний пульс зникає. Якщо кінцівка синякова, це говорить про те, що передавлені тільки вени. При цьому порушується відтік крові - кровотеча посилюється.

3.2. Вплив шуму на організм людини та розробка заходів щодо його зниженню до допустимих величин для обладнання

Шум є одним з основних проблем при роботі за ПК чи ноутбуком. Він може виникати, як від самою техніки так і від інших людей. Переважно розробка інтернет-магазину відбувається працівниками, які знаходяться в офісах. Існує кілька основних чинників, які найчастіше присутні в офісах це люди та різного роду апаратура. Сам шум може спричиняти дискомфорт або сприяти неухважності працівників. Отже, шум — це чинник виробничого середовища наявний на більшості підприємств [28]. Джерелами шуму можуть бути системи вентиляції, техніка, люди та інше. Інтенсивний шум може стати причиною таких професійних захворювань, як туговухість або глухота. Крім того, у працівників, які щодня перебувають під його впливом:

- знижується продуктивність праці;
- ослаблюється увага та уповільнюється реакція, спостерігається запаморочення, дратівливість, знижується працездатність, гострота зору;
- зростає кров'яний тиск, змінюється ритм дихання та серцевої діяльності, порушується працездатність клітин кори головного мозку тощо.

Звичним для людини є шумовий фон з рівнем звукового тиску в частотах 15-35 дБ (децибел).

При збільшенні рівня звукового тиску до 40-70 дБ спостерігається деяке зниження продуктивності праці та погіршення самопочуття.

Рівень звукового тиску в межах 75-120 дБ спричиняє враження органів слуху і серцево-судинної системи. Постійний шум з рівнем звукового тиску понад 120 дБ може призвести до акустичної травми.

Це один із найбільш поширених несприятливих фізичних факторів навколишнього середовища. Наприклад, при присутності клубу біля офісу рівень шуму коливається від 120 до 140 дБ, при проїзду автомобілю — від 118 до 130 дБ, а розмова між людьми може досягати до 105 дБ.

Коли мова йде про вплив шуму, то зазвичай основну увагу приділяють стану органу слуху, так як слуховий аналізатор у першу чергу сприймає звукові коливання і подразнення його є адекватним дії шуму на організм.

Зміни, що виникають в органі слуху, деякі дослідники пояснюють травмуючою дією шуму на периферичний відділ слухового аналізатора внутрішнього вуха. Основною ознакою впливу шуму є зниження слуху по типу кохлеарного невриту. Професійне зниження слуху буває зазвичай двостороннім.

Захист від шуму повинен забезпечуватися розробкою шум безпечної техніки, застосуванням засобів і методів колективного захисту, в тому числі будівельно-акустичних, застосуванням засобів індивідуального захисту [29].

Коллективні засоби захисту поділяються на засоби, що знижують шум у джерелі його виникнення, і засоби, що знижують шум на шляху його поширення від джерела до об'єкта, що захищається.

Зниження шуму в джерелі здійснюється за рахунок поліпшення конструкції машини або зміни технологічного процесу. Методи і засоби колективного захисту, в залежності від способу реалізації, поділяються на будівельно-акустичні, архітектурно-планувальні та організаційно-технічні і включають в себе:

- зміну спрямованості випромінювання шуму;
- раціональне планування підприємств і виробничих приміщень;
- акустичну обробку приміщень;
- застосування звукоізоляції.

У низці випадків величина показника спрямованості досягає 10-15 дБ, що необхідно враховувати при використанні установок з направленим випромінюванням, орієнтуючи ці установки так, щоб максимум випромінюваного шуму був спрямований у протилежний бік від робочого місця.

Раціональне планування підприємств і виробничих приміщень дозволяє знизити рівень шуму на робочих місцях за рахунок збільшення відстані до джерел шуму.

Засоби індивідуального захисту застосовуються в тому разі, якщо іншими способами забезпечити допустимий рівень шуму на робочому місці не вдається. Застосування ЗІЗ дозволяє попередити розлад не тільки органів слуху, а й нервової системи від дії надмірного подразника.

3.3. Висновки до третього розділу

Допомога при кровотечах є важливою навичкою, яка допоможе врятувати життя, особливо в теперішній час. Для того, щоб надати першу допомогу необхідно визначити, який тип кровотечі є в потерпілого: артеріальний, венозний чи капілярний. Найбільш небезпечним є артеріальний, оскільки кров виходить швидко і необхідно терміново її зупинити. Одним з способів зупинки є накладання джгута, якщо ж його накласти неможливо потрібно зупинити кровотечу натисканням на артерію. Венозна кровотеча також є небезпечною але є більше часу для надання допомоги в порівнянні з артеріальною. Капілярна кровотеча і більшості випадків зупиняється сама але при великій рані необхідно її продезінфікувати та накласти пов'язку для запобігання зараження крові і швидшого загоєння.

Шум є негативний чинник дії на людину. Він може спричинити головні болі, знижується продуктивність та знизити уважність працівників. При шуму на високих дБ може бути спричинене сильне пошкодження барабанної перепоники, тобто слухового апарату. Для захисту від цього використовується звукоізоляція, зміну спрямованості шуму та інше. Кожен з цих методів дозволяє працівникам знаходитися в комфортних умовах для забезпечення високої продуктивності та зменшенню впливу на їх здоров'я.

ВИСНОВКИ

У даній кваліфікаційній роботі розроблено проект інтернет-магазину «BROX» з використанням PHP, MySQL та JavaScript, зі зручним, простим і зрозумілим інтерфейсом користувача. Даний інтернет-магазин призначений для продажу товарів та орієнтований для широкого застосування у сфері онлайн торгівлі. З його допомогою користувачі зможуть швидко та зручно здійснювати купівлю.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- подано інформацію про існуючі рішення поставленої задачі та область застосувань;

- розглянуто етапи та методи розробки інтернет-магазину;

- висвітлено основні технології, які найкраще підходять для розробки інтернет-магазину (HTML, CSS, PHP, JavaScript, MySQL);

- проаналізовано вимоги до розробки та поставлена задача на розробку;

В другому розділі кваліфікаційної роботи:

- розроблено структура та дизайн інтернет-магазину;

- описано основні функціональні можливості;

- спроектовано структуру каталогів інтернет-магазину;

- подано інформацію про встановлення та налаштування інтернет-магазину.

У розділі «безпека життєдіяльності та охорона праці» висвітлено долікарську допомогу при кровотечах та засоби захисту від шуму.

ПЕРЕЛІК ДЖЕРЕЛ

1. Бойчук І. В. Інтернет в маркетингу / І. В. Бойчук, О. М. Музика. – Київ: Центр учбової літератури, 2018. – 512 с. – ISBN 978-611-01-0099-1.
2. Сталий розвиток – XXI століття / [Т. Ю. Александрюк, Л. П. Артеменко, Л. М. Архипова та ін.]. – Київ: Національний університет “Києво-Могилянська академія”, 2021. – 527 с. – ISBN 978-617-7668-33-5
3. Як відкрити інтернет магазин в Україні: покроковий путівник [Електронний ресурс] // Bankchart. – 2022. – Режим доступу до ресурсу: bankchart.com.ua/biznes/biznes_start/statti/yak_vidkriti_internet_magazin_v_ukrayin_i_pokrokoviy_putivnik.
4. Пасічник О. Г. Основи веб-дизайну / О. Г. Пасічник, О. В. Пасічник, І. В. Стеценко. – Київ: Видавнича група BVH, 2019. – 337 с. – ISBN 978-966-552-228-7
5. Ключові етапи створення сайту [Електронний ресурс] / PBB design. – 2022. – Режим доступу до ресурсу: pbb.lviv.ua/statti-i-novyny/statti-shchodo-stvorennia-saitu/osnovni-etapy-stvorennia-saitu.
6. Використання HTML та програмних систем для розробки [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv.
7. Методи розробки Web-сайтів [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv.
8. Платіжний віджет. LiqPay [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: www.liqpay.ua/documentation/api/acquiring/widget.
9. Що таке html? [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: css.in.ua/article/shcho-take-css_3.
10. HTML5 Notes for Professionals book [Електронний ресурс] // GoalKicker. – 2019. – Режим доступу до ресурсу: books.goalkicker.com/HTML5Book.

11. CSS Notes for Professionals book [Електронний ресурс] // GoalKicker. – 2019. – Режим доступу до ресурсу: books.goalkicker.com/CSSBook.
12. Хрустальов О. А. HTML5 + CSS3. Основи сучасного WEB-дизайну. / О. А. Хрустальов, А. В. Кириченко. – Санкт-петербург: Наука і техніка, 2018. – 352 с. – ISBN 978-5-94387-750-6
13. Макфарланд Д. С. Нова велика книга CSS / Девід Соєр Макфарланд. – Санкт-петербург: Пітер Прес, 2018. – 720 с. – (Бестселери O'Reilly). – ISBN 978-5-496-02080-0
14. PHP Notes for Professionals book [Електронний ресурс] // GoalKicker. – 2018. – Режим доступу до ресурсу: books.goalkicker.com/PHPBook.
15. Основи JavaScript [Електронний ресурс] – 2022. – Режим доступу до ресурсу:
developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basic_s.
16. JavaScript Notes for Professionals book [Електронний ресурс]// GoalKicker. – 2019. – Режим доступу до ресурсу: books.goalkicker.com/JavaScriptBook.
17. Фрісбі М. JavaScript для професійних веб-розробників / Метт Фрісбі. – Пітер, 2022. – 1219 с. – (4). – ISBN 978-5-4461-1740-6
18. MySQL Notes for Professionals book [Електронний ресурс] // GoalKicker. – 2018. – Режим доступу до ресурсу: books.goalkicker.com/MySQLBook.
19. Види популярних локальний серверів [Електронний ресурс] // [armedsoft](https://armedsoft.com/ua/blog/vydy-populyarnyh-lokalnyy-serveriv). – 2018. – Режим доступу до ресурсу: armedsoft.com/ua/blog/vydy-populyarnyh-lokalnyy-serveriv.
20. Автоматизація роботи деканату «відомість» на open server [Електронний ресурс] // Житомирський державний технологічний університет. – 2017. – Режим доступу до ресурсу: conf.ztu.edu.ua/wp-content/uploads/2017/06/153-2.pdf.

21. Структура сайту: основні види та правила їх розробки [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: webtune.com.ua/statti/web-rozrobka/struktura-sajtu.

22. Плагін [Електронний ресурс] // Вікіпедія. – 2021. – Режим доступу до ресурсу: uk.wikipedia.org/wiki/Плагін.

23. Правила формування хорошої структури каталогу для інтернет-магазину [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: www.ekam.ru/blogs/pos/struktura-kataloga.

24. Долікарська допомога при кровотечах [Електронний ресурс] // Державна установа «Черкаський обласний центр контролю та профілактики хвороб МОЗ України». – 2022. – Режим доступу до ресурсу: www.oblses.ck.ua/index.php?option=com_content&view=article&id=2710:dolikarska-dopomoha-pri-krovotechakh&catid=41:2013-05-13-02-14-47&Itemid=57.

25. Перша долікарська допомога при кровотечі [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: dnz416.edu.kh.ua/poradi_likarya/pamyatki/persha_dolikarsjka_dopomoga_pri_krovotechich.

26. Перша допомога при кровотечі [Електронний ресурс] // Спеціальний миротворчий центр. – 2018. – Режим доступу до ресурсу: smc.naiu.kiev.ua/persha-dopomoga-pri-krovotechich.html.

27. Долікарська допомога при кровотечах [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: aybolit.ua/dolikarska-dopomoga-pri-krovotechach.

28. Засоби захисту від шуму і вібрації [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: www.sop.com.ua/article/1071-zasobi-zahistu-vd-shumu-vbrats.

29. Вплив шуму на організм людини і засоби його мінімізації на робочому місці [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: oppb.com.ua/news/vplyv-shumu-na-organizm-lyudyny-i-zasoby-yogo-minimizaciyi-na-robochomu-misci.

ДОДАТКИ

Лістинг плагіну для корзини

```

<?php
require 'libs/db.php';
$products = R::findAll('products');
$data = $_POST;
if (isset($data['item'])) {
    if (isset($_SESSION['cart'][$data['id']])) {
        $_SESSION['cart'][$data['id']]++;
    } else {
        $_SESSION['cart'][$data['id']] = 1;
    }
}
$page_title = 'Cart';

$data = $_POST;
if (isset($data['delete'])) {
    unset($_SESSION['cart'][$data['product_id']]);
}
if (isset($data['increment'])) {
    $_SESSION['cart'][$data['product_id']]++;
}
if (isset($data['decrement'])) {
    if ($_SESSION['cart'][$data['product_id']] < 2) {
        unset($_SESSION['cart'][$data['product_id']]);
    } else {
        $_SESSION['cart'][$data['product_id']]--;
    }
}
$cart = $_SESSION['cart'];
include 'templates/header.php';
?>
<h1>Cart</h1>
<? if ($cart == 0) { ?>
    <p>Your cart is empty.</p>
<? } else { ?>
    <ul class="cart">
    <? $total = 0;
    foreach ($cart as $id=>$number) { ?>
        <li class="cart__item">
            <form action="cart.php" method="post">
                <input type="hidden" name="product_id" value="<?= $id ?>">
                <? $item = R::findOne('products', 'id=?', array($id));
                $total += intval($item->price)*$number; ?>
                <h2><?= $item->title ?></h2>
                <div class="price">
                    $<?= number_format(intval($item->price) / 100, 2); ?>
                    <? if ($number > 1) { ?>
                        &times; <?= $number ?> = $<?=
number_format(intval($item->price)*$number/100, 2) ?>

```

```

        <? } ?>
    </div>
    <p class="description">
        <?= $item->description ?>
    </p>
    <input type="submit" name="decrement" value="-">
    <?= $number ?>
    <input type="submit" name="increment" value="+">
    <br>
    <input type="submit" name="delete" value="Remove from cart">
    </form>
</li>
<?
$order .= $item->title." : ".$number." | ";
} ?>
</ul>
<? if ($total == 0) { ?>
    <p>Your cart is empty.</p>
<? } else { ?>
    <div class="price">Total: $
    <?= number_format($total / 100, 2) ?>
    </div>
    <? if ($_SESSION['uid']) { ?>
        <form action="checkout.php" method="post">
            <input type="hidden" name="total" value="<?= $total ?>">
            <input type="hidden" name="$order" value="<?= $order ?>">
            <input class="input_bottom" type="submit" name="checkout"
value="Checkout" id="input_bottom">
        </form>
    <? } else { ?>
        <a id="checkout" href="register.php">Please register to
checkout</a>
    <? } ?>
<?
include 'templates/footer.php'; ?>

```


Лістинг скрипта створення товару

```
<?php
require 'libs/db.php';
include 'templates/header.php';
$page_title = 'Create';
$data = $_POST;
if ($_SESSION ['admincheck'] != "1") {
    header("Location:index.php");
    exit;
}
if (isset($data['submit'])) {
    $errors = array();
    if (trim($data['namee'])=='') {
        $errors[]='Name is required.';
    }
    if (trim($data['price'])=='') {
        $errors[]='Price is required.';
    }
    if (trim($data['manufacturer'])=='') {
        $errors[]='Manufacturer is required.';
    }
    if (trim($data['model'])=='') {
        $errors[]='Model is required.';
    }
    if ($data['description'] == '') {
        $errors[]='Description is required.';
    }
    if ($data['descriptionall'] == '') {
        $errors[]='Detailed description is required.';
    }
    if (empty($errors)) {
        move_uploaded_file($_FILES['file']['tmp_name'],
"images/".$_FILES['file']['name']);
        $products = R::dispense('products');
        $products->image = $_FILES['file']['name'];
        $products->title = $data['namee'];
        $products->price = $data['price'];
        $products->manufacturer = $data['manufacturer'];
        $products->model = $data['model'];
        $products->description = $data['description'];
        $products->all_description = $data['descriptionall'];
        $products->category = $data['category_id'];
        R::store($products);
    }
}
?>
<section>
    <div class="container">
        <div class="row">
```

```

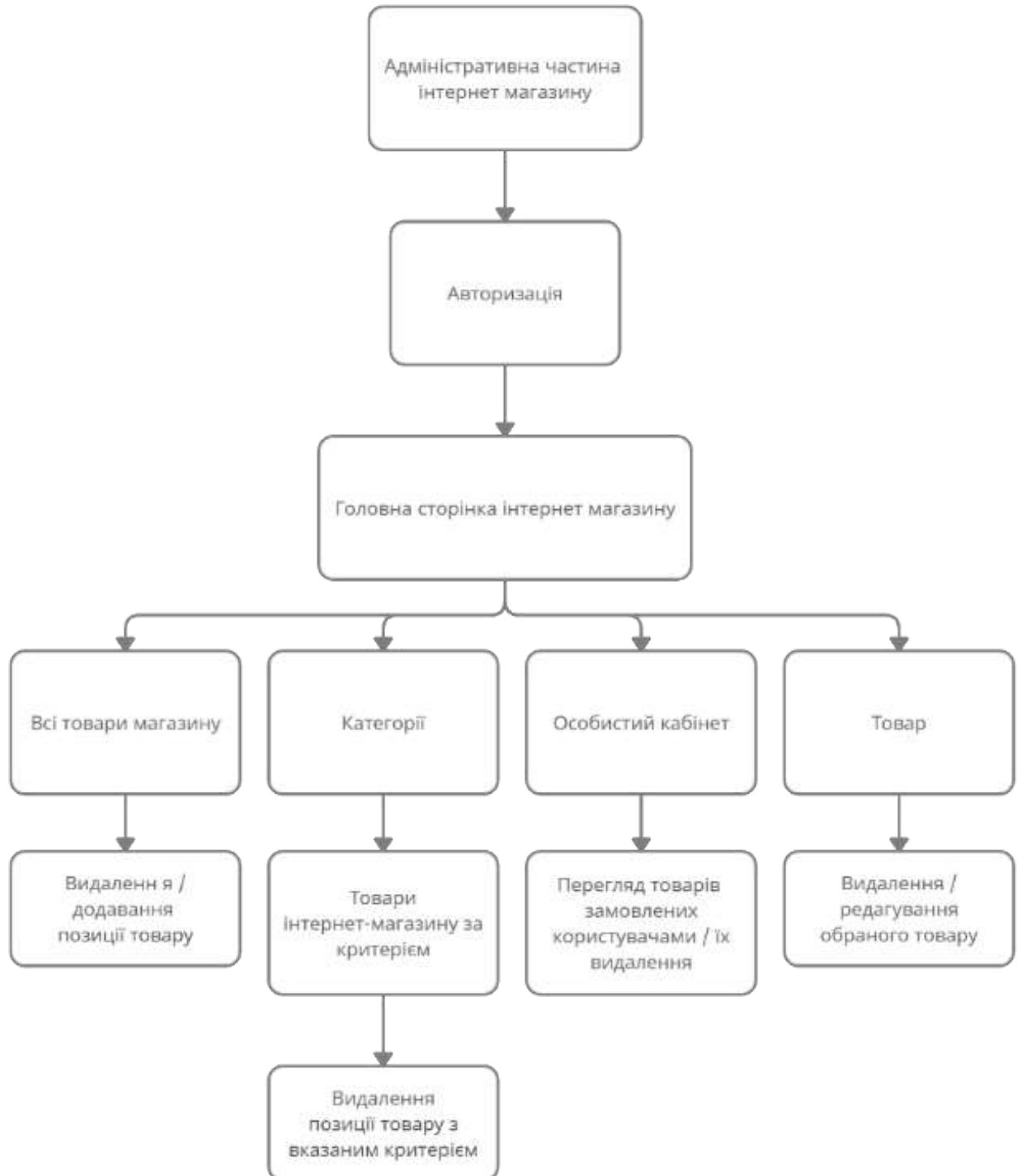
<div class="breadcrumbs">
</div>
<h4>Add new product</h4>
<? if (!empty($errors)) { ?>
  <div class="error">
    <?= implode('<br>', $errors); ?>
  </div>
<? } ?>
<div class="colt">
  <div class="create-form">
    <form      action="create.php"      method="post"
enctype="multipart/form-data">
      <div class="form__row">
        <label id="describe">Name</label>
        <input id="name" type="text" name="name"
value="">
      </div>
      <div class="form__row">
        <label id="describe">Prise</label>
        <input id="price" type="text" name="price"
value="">
      </div>
      <div class="form__row">
        <label id="describe">Category</label>
        <select          name="category_id"
id="categorycreate">
          <option value="1">Smartphone</option>
          <option value="2">Camera</option>
          <option value="3">Clock</option>
          <option value="4">Laptop</option>
        </select>
      </div>
      <div class="form__row">
        <label
id="describe">Manufacturer</label>
        <input          id="price"          type="text"
name="manufacturer" value="">
      </div>
      <div class="form__row">
        <label id="describe">Model</label>
        <input          id="price"          type="text"
name="model" value="">
      </div>
      <div class="form__row">
        <label
Image</label>
        <input type="file" name="file">
      </div>
      <div class="form__row form__row1">
        <label id="describe">Description</label>
        <textarea          name="description"
id="textareal"></textarea>

```

```

        </div>
        <div class="form__row form__row2">
            <label                id="describe">Detailed
description</label>
            <textarea                name="descriptionall"
id="textarea2"></textarea>
        </div>
        <input                type="submit"                name="submit"
value="Save">
    </form>
</div>
</div>
</div>
</div>
</section>
```

Структурна схема інтерфейсу адміністративної частини



Лістинг сторінки реєстрації користувача

```
<?php
require 'libs/db.php';
$data = $_POST;
if( isset($data['reg'])) {
    $errors = array();
    if (trim($data['name'])=='') {
        $errors[]='Name is required.';
    }
    if (trim($data['email'])=='') {
        $errors[]='Email is required.';
    }
    if (trim($data['address'])=='') {
        $errors[]='Address is required.';
    }
    if ($data['password']=='') {
        $errors[]='Password is required.';
    }
    if ($data['confirm']!=$data['password']) {
        $errors[]='Confirmation is incorrect.';
    }
    if (R::count('users',"email=?", array($data['email']))>0) {
        $errors[]='A user with this email already exists';
    }
    if (empty($errors)) {
        $users = R::dispense('users');
        $users->name = $data['name'];
        $users->email = $data['email'];
        $users->address = $data['address'];
        $users->admin = 0;
        $users->
        >password=password_hash($data['password'],PASSWORD_DEFAULT);
        R::store($users);
        $_SESSION['uname'] = $user->name;
        $_SESSION['uid'] = $user->id;
        header('Location: /index.php');
    }
}
include 'templates/header.php';
?>
<form class="form" action="register.php" method="post">
    <h1>Register</h1>
    <? if (!empty($errors)) { ?>
        <div class="error">
            <?= implode('<br>', $errors); ?>
        </div>
    <? } ?>
    <div class="form__row">
        <label for="register-name">Name</label>
```

```
<input id="register-name" type="text" name="name" value="">
</div>
<div class="form__row">
  <label for="register-email">Email</label>
  <input id="register-email" type="email" name="email" value="">
</div>
<div class="form__row">
  <label for="register-addres">Addres</label>
  <input id="register-addres" type="text" name="addres" value="">
</div>
<div class="form__row">
  <label for="register-password">Password</label>
  <input id="register-password" type="password" name="password"
value="">
</div>
<div class="form__row">
  <label for="register-confirm">Confirm password</label>
  <input id="register-confirm" type="password" name="confirm"
value="">
</div>
<div class="form__row">
  <input type="submit" name="reg" value="Register">
</div>
</form>
<?php
include 'templates/footer.php';
?>
```