

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

Кафедра цифрової економіки та бізнес-аналітики

КУРСОВА РОБОТА

з дисципліни «Проектування та адміністрування БД і СД»

на тему:

«Інформаційна система магазину одягу»

спеціальність: 051 «Економіка»

(код та найменування спеціальності)

спеціалізація: «Інформаційні технології в бізнесі»

(найменування спеціалізації)

освітній ступінь: бакалавр

(бакалавр/магістр)

Науковий керівник:

К.е.н., доц. Ярема О.Р.

(науковий ступінь, посада, прізвище, ініціали)

“ ” 2020 р.
(підпис)

Виконавець:

студентка групи УФЕ-31с

Оссовська Т.О.

(прізвище, ініціали)

“ ” 2020 р.
(підпис)

Загальна кількість балів _____

(підпис, ППП членів комісії)

ЛЬВІВ 2020

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ	5
1.1 Постановка завдання.....	5
1.2 Розробка моделі варіантів використання веб-сайту	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ	8
2.1 Опис моделі даних	8
2.2 Нормалізація реляційних відношень	11
2.3 Визначення типів даних	13
2.4 Обмеження цілісності даних.....	14
2.5 Реалізація SQL-скрипту	15
РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ	17
3.1 Структура веб-сайту	17
3.2 Макети сторінок веб-сайту.....	17
Зовнішній вигляд сайту написаний мовами HTML та CSS.	17
3.3 Програмування серверної частини	21
3.4 Програмування клієнтської частини	23
3.5 Розміщення веб-сайту на локальному сервері	24
ВИСНОВКИ.....	25
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	26

ВСТУП

На сьогодні створення веб-сайтів за допомогою новітніх технологій дозволяє не тільки рекламувати свій бізнес в Інтернеті, а й ефективно просувати його через всесвітню павутину, знаходити нових клієнтів і вести співпрацю по всьому світу. Слід зауважити, що створення сайту досить складний і трудомісткий процес, який вимагає структурування підходу, чіткого планування, систематизації даних і багатьох інших складових. Особливо актуальним є створення веб-сайтів в нинішньому столітті, коли життя плавно переноситься в павутину Інтернет. Звідси і актуальність даної теми, так як створення інтернет-магазинів на сьогодні займає передову позицію в стратегії розвитку будь-якої компанії.

Метою даної роботи є реалізація web-орієнтованої інформаційної системи інтернет-магазину.

Виходячи з поставленої мети сформовано наступні завдання:

- Проектування бази даних на основі системи управління базами даних MySQL
- Забезпечення редагування даних таблиць БД та взаємозв'язків між різними таблицями;
- Розробка інтерфейсу web-сторінки
- Забезпечення взаємодії web-сторінки із створеною базою даних.

Об'єктом даної роботи є дослідження та використання засобів розробки PHP, MySQL, HTML та CSS.

Предмет даної роботи – створена база даних, web-сайт та принципи їх взаємодії.

Інформаційною базою для написання курсової роботи по даній темі стали: підручники і навчальні посібники, інформаційні матеріали з Інтернету.

Використане програмне забезпечення. Локальний LAMP Linux сервер запущений на віртуальному середовищі VirtualBox, програмне середовище phpMyAdmin, спеціальний текстовий редактор Sublime Text 3, мови веб-розробки PHP, JS, HTML5, CSS, система керування реляційними базами даних MySQL, мова програмування для взаємодії з базами даних SQL

Структура роботи. Курсова робота складається зі вступу, трьох розділів, поділених в свою чергу на підрозділи, висновку та списку використаної літератури.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

Бізнес потреба. Інформаційна система для магазину одягу

Постійне зростання числа користувачів мережі Інтернет робить Web-сайт важливим інструментом для досягнення даних цілей, а також значно розширює можливості маркетингу в глобальній мережі. Виходячи з того, що web-сайт – це свого роду віртуальний офіс, який, як і справжній офіс приймає відвідувачів, в даному випадку – користувачів мережі Інтернет. Web-сайт інтернет-магазину є основним елементом, який пов'язує відвідувачів сайту та адміністрацію магазину. Через web-сайт відвідувач, тобто клієнт, може ознайомитися з усією потрібною йому інформацією, а адміністрація, в свою чергу, може розмістити на сайті усі деталі про товари, яка приведе клієнта до покупки.

Для покупця інтернет-магазин насамперед зручний:

- доступністю, так як для входу в інтернет-магазин не потрібно виходити з дому, потрібен лише комп'ютер або смартфон, підключений до Інтернету;
- наочністю, вся інформація супроводжується ілюстраціями;
- інформативністю, на сайті є докладний опис всіх запропонованих товарів;
- широким асортиментом продукції.

Для продавців містити інтернет-магазин зручно:

- можливість розширити ринок збуту;
- немає необхідності орендувати площу для продажу і вибору товарів клієнтами;
- немає необхідності утримувати менеджерів для роботи в торговому залі, досить мінімум одного менеджера;
- інтернет-магазин допомагає оптимізувати процес замовлення товарів.

Але web-сайт потребує структури, для того щоб розмістити в собі всю вичерпну інформацію щодо діяльності магазину, для того щоб покупець був задоволений візитом і обов'язково здійснив покупку без зайвих проблем.

Щоб зробити можливим виконання цієї бізнес-ідеї постають такі технічні завдання:

- забезпечити користувачам можливість перегляду товарів;
- забезпечити вибір способу оплати та доставки;
- забезпечити підтвердження замовлення користувачем;
- створити можливість адміністратору перегляд замовлень;
- забезпечити редагування вмісту замовлення.

1.2 Розробка моделі варіантів використання веб-сайту

Головним користувачами системи виступають клієнт та адміністратор.

Варіанти використання ілюструються за допомогою однойменної діаграми, зображеної на рисунку 1.1. Діаграма варіантів використання складається з акторів «Клієнт», «Адміністратор» та доступних їм варіантів використання сайту, які зображені овалом.

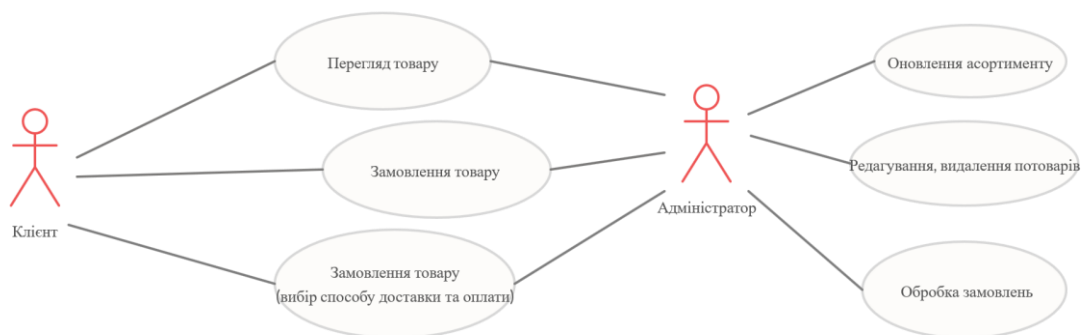


Рисунок 1.1 – Діаграма варіантів використання сайту

Отже, метою інформаційної системи для інтернет-магазину є формування замовлення зовнішнім користувачем, тобто клієнтом, та перегляд замовлення внутрішнім користувачем – адміністратором.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Опис моделі даних

Ядром будь-якої бази даних є модель даних. Модель становить безліч структур даних, обмежень цілісності й операцій маніпулювання даними. За допомогою моделі даних можуть бути представлені об'єкти предметної області й взаємозв'язку між ними.

Модель даних – це сукупність взаємозалежних структур даних і операцій над цими структурами. Вид моделі і використовувані в ній типи структур даних відбивають концепцію організації та обробки даних, використовувану в СУБД, що підтримує модель, чи в мові системи програмування, на якій створюється прикладна програма обробки даних.

СУБД (Системи управління базами даних) — це програмні засоби, які виступають посередником між БД та її користувачами. Завдяки сукупності мовних та програмних засобів, СУБД сприяють створенню, ведення та спільного використання БД різними користувачами.

Для реалізації серверної частини було обрано СУБД MySQL. Для управління базами даних використовується програма PHPMyAdmin — веб-додаток з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL. Важливим етапом створення таблиць є вибір кодування, яке дозволяє зберігати в таблицях даних символи кирилиці - utf8_general_ci.

В програмному середовищі phpMyAdmin створено базу даних «bebe», яка міститиме таблиці з даними інтернет-магазину.

Таблицями бази даних є:

- «orders» — таблиця, у якій міститься інформація про список замовлень;

- «products» – таблиця, у якій міститься інформація про продукти інтернет-магазину;

- «products_categories» – таблиця, у якій міститься інформація про категорії товарів інтернет-магазину.

- «users» – таблиця, у якій знаходяться дані для авторизації працівників інтернет-магазину.

Поля, що входять у таблиці бази даних:

Для таблиці «orders»:

- id;
- sum;
- email;
- name;
- phone;
- comment;
- delivery_address;
- cart.

Для таблиці «products»:

- id;
- rating;
- hit_sales;
- availability;
- url;

- name;
- price;
- img;
- description;
- qualification;
- category.

Для таблиці «products_categories»:

- id;
- name;
- url;
- description.

Для таблиці «users»:

- id;
- login;
- password;
- fullname.

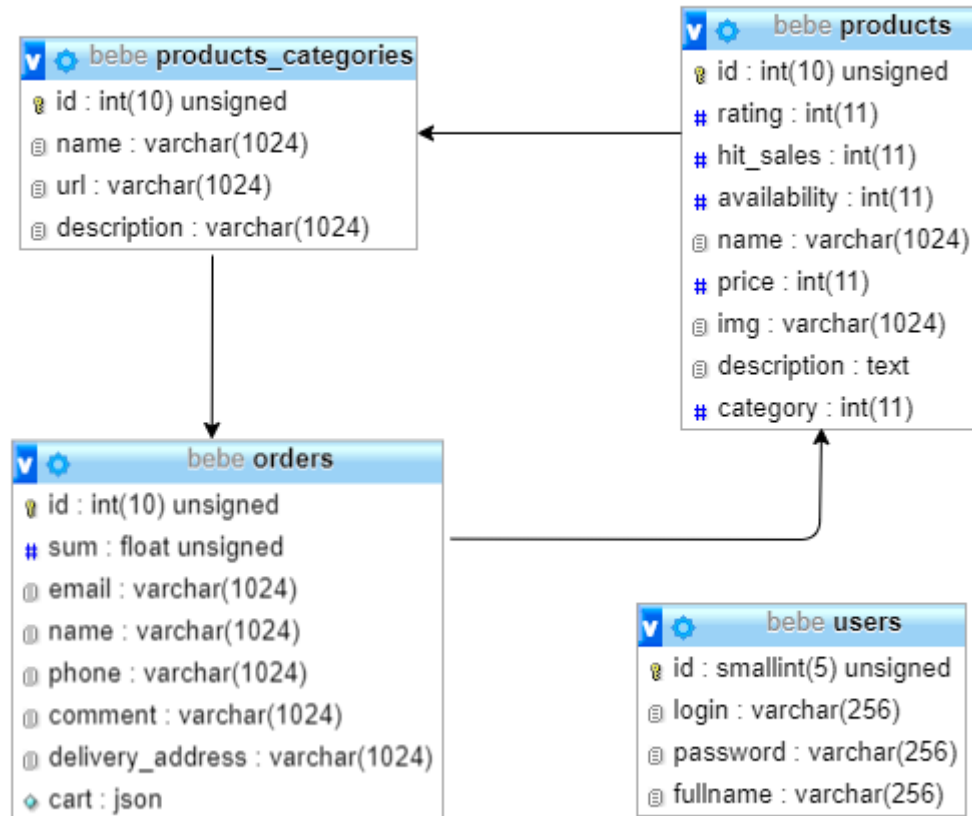


Рис.2.1 діаграма сутностей та зв'язків

На рисунку 2.1 наведено діаграму сутностей та зв'язків бази даних. Для всіх сутностей визначено первинні ключі (ПК): `id` для `product_categories`, `id` для `products`, `id` для `orders`, `id` для `users`. Первинні ключі на діаграмі позначені зображенням ключа.

2.2 Нормалізація реляційних відношень

Нормалізація - це процес організації даних в базі даних. Вона включає в себе створення таблиць та встановлення відносин між цими таблицями відповідно до правил, призначеними для захисту даних і забезпечення більшої гнучкості бази даних за рахунок виключення надмірності і неузгодженості залежності. Надлишкові дані віднімають місце на диску і створюють проблеми з обслуговуванням.

Існує кілька правил нормалізації баз даних. Кожне правило називається "звичайною формою". Якщо виконується перше правило, база даних

називається "Перша нормальна форма". Якщо виконуються перші три правила, база даних вважається "третя нормальна форма". Хоча можливі й інші рівні нормалізації. Третя нормальна форма вважається найвищим рівнем, необхідним для більшості додатків.

Для дотримання першої нормальної форми

Виключіть повторювані групи в окремих таблицях. Створіть окрему таблицю для кожного набору пов'язаних даних. Визначте кожен набір пов'язаних даних за допомогою первинного ключа. Не використовуйте кілька полів в одній таблиці для зберігання схожих даних.

Для дотримання другої нормальної форми

Створіть окремі таблиці для наборів значень, які застосовуються до кількох записів. Об'єднайте ці таблиці за допомогою зовнішнього ключа. При необхідності записи не повинні залежати від первинного ключа таблиці (складеного ключа).

Для дотримання третьої нормальної форми

Виключіть поля, що не залежать від ключа. Значення в записі, які не є частиною ключа записи, не належать таблиці. У загальному випадку, коли вміст групи полів може бути застосований до більш ніж одного запису в таблиці, рекомендується розміщувати ці поля в окремій таблиці.

Інші форми нормалізації

Четверта звичайна форма, також має назву звичайної форми бійця Кодд (БКНФ) і п'ята звичайна форма - існує, але зрідка розглядається на практиці. Відсутність обліку цих правил може призвести до меншого створення структури бази даних, але не впливає на функціональність.

2.3 Визначення типів даних

База даних – це структурований обсяг інформації, що містить таблиці, які у свою чергу складаються із записів, а записи – з полів. Поле має два атрибути – ім'я і тип даних. Нижче наведено типи даних, які використовуються у створеній базі даних – «bebe»:

- TEXT – текстовий тип даних, може зберігати не більше 65 535 символів:
- INT – тип даних, що представляє ціле число, діапазон від -2 147 483 648 до 2 147 483 647;
- VARCHAR – набір символьних даних, може зберігати не більше 255 символів.
- JSON – тип даних, що представляє об'єкти.

Таблиця 2.1. Типи даних таблиці «orders»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	id (PK)	INT	11
2.	sum	INT	11
3.	email	VARCHAR	1024
4.	name	VARCHAR	1024
5.	phone	VARCHAR	1024
6.	comment	VARCHAR	1024
7.	delivery_address	VARCHAR	1024
8.	cart	JSON	-

Таблиця 2.2. Типи даних таблиці «products»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	id (PK)	INT	11
2.	rating	INT	11
3.	hit_sales	INT	1024
4.	availability	INT	1024
5.	name	VARCHAR	1024
6.	price	INT	1024
7.	img	TEXT	1024
8.	description	VARCHAR	-
9.	category	INT	11

Таблиця 2.3. Типи даних таблиці «products_categories»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	id (PK)	INT	11
2.	name	VARCHAR	1024
3.	url	VARCHAR	1024
4.	description	VARCHAR	1024

Таблиця 2.4. Типи даних таблиці «users»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	id (PK)	INT	11
2.	login	VARCHAR	1024
3.	password	VARCHAR	1024
4.	fullname	VARCHAR	1024

2.4 Обмеження цілісності даних

Цілісність бази даних - властивість бази даних, що означає, що БД містить повну і несуперечливу інформацію, необхідну для коректного функціонування додатків. Для забезпечення цілісності накладають обмеження цілісності.

Правила цілісності бази даних – це правила, які забезпечують уникнення введення некоректних даних та дають можливість об'єднання декількох таблиць.

У проєктованій базі даних були використанні наступні типи правил цілісності:

- NOT NULL/NULL - Заборона/ дозвіл на використання не заданих або не визначених значень.
- UNIQUE - Контроль унікальності значень атрибутів.
- PRIMARY KEY - Первинний ключ.
- FOREIGN KEY - Зовнішній ключ.

За допомогою первинних та зовнішніх ключів було встановлено наступні зв'язки, зображені у діаграмі сутностей і зв'язків із зовнішніми ключами (рис.2.2).

Зовнішні ключі (FK): category в таблиці products.

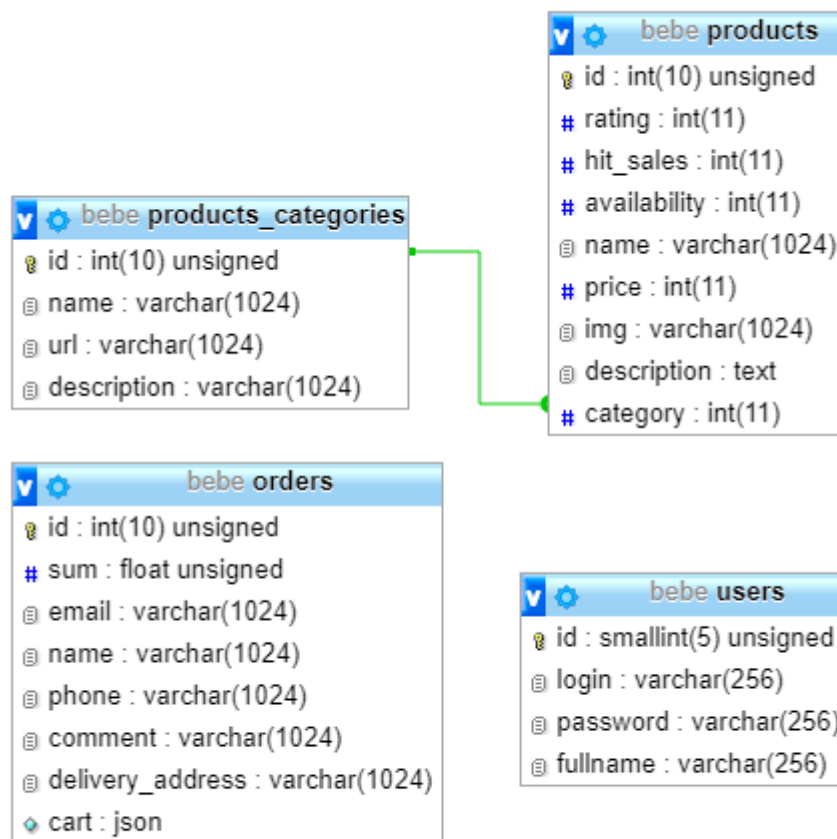


Рисунок 2.2 – Діаграма сутностей та зв'язків із зовнішніми ключами

2.5 Реалізація SQL-скрипту

Web-сайт реалізовує наступні запити до бази даних:

1. Запити типу SELECT:

- SELECT * FROM orders ORDER BY id DESC – вибір усіх замовлень з сортуванням по id.
- SELECT * FROM products WHERE id='\$id' – вибір продукту по id
- SELECT * FROM products WHERE hit_sales='1' AND availability='1' ORDER BY rating DESC – вибір наявних товарів з хіту продажів

- d. `SELECT * FROM products_categories ORDER BY id` – вивід категорій товарів для меню
- e. `SELECT * FROM products_categories WHERE id = '$category_id'` – вибір усіх товарів з обраної категорії

2. Запити типу INSERT:

- a. `INSERT INTO orders(sum, email, name, phone, comment, delivery_address, cart) VALUES ('$sum', '$email', '$name', '$phone', '$comment', '$delivery_address', '$cart_json')` – створення нового замовлення на сайті.

РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ

3.1 Структура веб-сайту

Дана програма реалізована у вигляді веб-сайту , який працює через будь-який доступний браузер.

Веб-сайт складається з наступних сторінок:

- Web-сторінка «<http://localhost/index.php>» - перше, що побачить клієнт. Це головна сторінка, на якій користувач може переглянути хіти продажу та має можливість перейти на будь-яку іншу сторінку сайту;
- Web-сторінка «<http://localhost/about.php>» - містить інформацію про компанію;
- Web-сторінки-«<http://localhost/shirts.php>», «<http://localhost/jackets.php>», «<http://localhost/overalls.php>», «<http://localhost/dresses.php>» містить інформацію про перелік наявних товарів та має можливість перейти на сторінку детального опису вибраного товару.
- Web-сторінка «<http://localhost/cart.php>» - містить корзину, де можна переглянути замовлення та подати інформацію про замовника.
- Web-сторінка «<http://localhost/product.php?id=43>» - містить детальний опис товару, можливість вибрати розмір та додати товар до кошику. (id – номер товару з бази даних)
- Web-сторінка «<http://localhost/order.php>» - підтверджує, що замовлення пройшло успішно.
- Web-сторінка «<http://localhost/admin.php>» - дозволяє адміністратору переглядати замовлення користувачів.

3.2 Макети сторінок веб-сайту

Зовнішній вигляд сайту написаний мовами HTML та CSS.

Сторінки веб-сайту мають спільне меню, яке розміщене в верху сторінки, основними елементами якого є логотип, перелік категорій та корзина замовлення.

Вигляд меню показано на рис.3.1

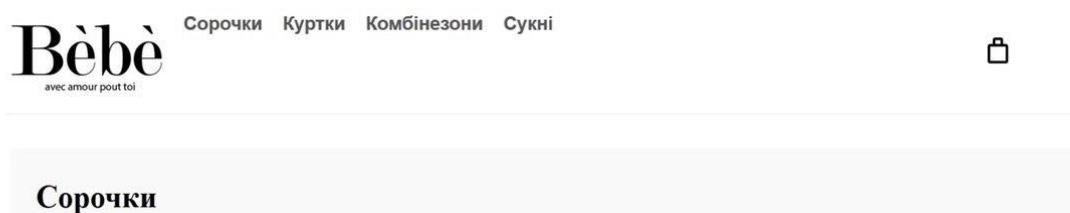


Рисунок 3.1 Вигляд меню

Головна сторінка (рис 3.2) та сторінка обраної категорії (рис 3.3) відображають перелік наявних товарів. Окрім тіла сторінки містять також блоки header.php, menu.php, footer.php.

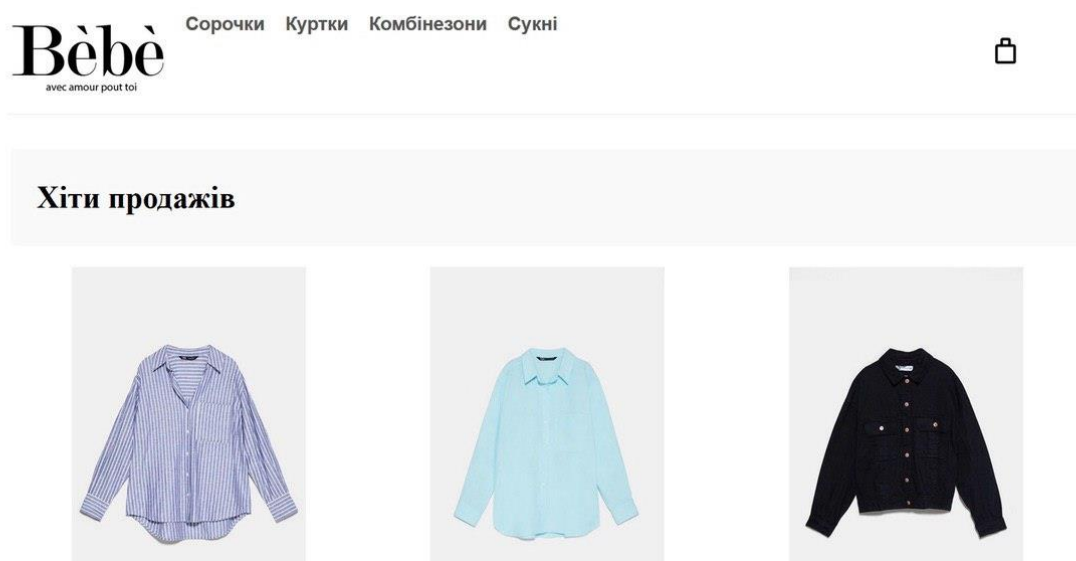


Рисунок 3.2-Головна сторінка



Сорочки



Рисунок 3.2 - перелік товарів обраної категорії

Сторінка `product.php?id=43` (рис 3.4) показує збільшене зображення вибраного товару його детальний опис.



Сукня на одне плече

900 UAH

Хлопок- 100% Колір- фуксія

розмір:

колір:

S



56

⌚ Термін доставки 2-3 робочі дні.

Додати в кошик

Рисунок 3.4 Опис товару

На рис 3.5 показана сторінка корзини. Зліва покупцеві необхідно ввести дані для замовлення, а справа перелік товарів, які є в корзині.

Рисунок 3.5 Вигляд корзини товарів

Рисунок 3.6 Підтвердження

Сторінка адміністраторської частини сайту «Admin» відображає форму авторизації (рис 3.7) містить блок header.php

Bèbè Admin
avec amour pour toi

Username:

Password:

Login

про нас
© 2020 bèbe

Рис 3.7 Авторизація адміну

3.3 Програмування серверної частини

Серверна частина сайту написана скрипковою мовою програмування PHP, в спеціалізованому текстовому редакторі Sublime Text 3. Зовнішній вигляд сайту, видимий для користувача, написаний мовами HTML та CSS.

Для забезпечення повноцінної роботи сайту створено наступні серверні скрипти:

Лістинг 3.1

Вивід категорій товарів у меню

```
<?php
$select = mysqli_query($connect,"SELECT * FROM products_categories
ORDER BY id");
for($i = 0;$i < mysqli_num_rows($select);$i++){
$row_menu = mysqli_fetch_array($select);

echo '<a style="vertical-align: bottom;" href="/'.$row_menu["url"].'"><div
style="font-weight: bold; font-size: 22px; line-height: 20px;margin-bottom:
5%; margin-top: 4px; margin-left: 20px; color: #555; display: inline-block;
vertical-align: top;"
class="menu_infotext">'.$row_menu["name"].'</div></a>';

?>
```

Лістинг 3.2

Створення нового замовлення

```

<?php
$status = 0;
$sum = 0;
$email = addslashes($_POST["email"]);
$name = addslashes($_POST["name"]);
$phone = addslashes($_POST["phone"]);
$comment = addslashes($_POST["comment"]);
$delivery_address = addslashes($_POST["address"]);
$cart_json = $_COOKIE['cart'];
$cart = json_decode($_COOKIE['cart']);

foreach ($cart as $key => $value) {
    $id = $value[0];
    $size = $value[1];
    $select = mysqli_query($connect,"SELECT * FROM products WHERE
id='$id'");
    $row = mysqli_fetch_array($select);

    $sum += (int)$row["price"];
}

$insert = mysqli_query($connect,"INSERT INTO
orders(sum,email,name,phone,comment,delivery_address,card) VALUES
('$sum','$email','$name','$phone','$comment','$delivery_address','$cart_json')");

```

Лістинг 3.3

Авторизація адміністратора інтернет магазину

```

session_start();

if(isset($_POST["username"]) and isset($_POST["password"])){
    $login = $_POST["username"]; $password = md5($_POST["password"]);
    $select = mysqli_query($connect,"SELECT * FROM users WHERE login =
'$login' AND password = '$password'");
    if($select->num_rows){
        $row_login = mysqli_fetch_array($select);
        $_SESSION['login'] = $row_login["login"];
    }
}

```

3.4 Програмування клієнтської частини

Клієнтська частина веб-сайту, має такі основні сторінки:

1. Головна – містить інформацію про хіти продажу
2. Сторінки: Сорочки, Куртки, Комбінезони, Сукні – дозволяє переглянути всі наявні товари даних категорії.
3. Про нас – опис магазину.

Адміністраторська частина дозволяє проводити операції із зміни інформації в базі даних безпосередньо на веб-сайті та містить наступні сторінки:

4. Кабінет адміністратора – головна сторінка адміністраторської частини сайту, містить меню з посиланнями на сторінки, з яких можна здійснити редагування таблиць бази даних та кнопку «Вийти».
5. Головна – головна сторінка сайту, доступна для перегляду адміністратором.
6. Перегляд і редагування послуг – сторінка для перегляду, додавання, редагування та видалення послуг.
7. Перегляд замовлень

Для розробки використаний JS, він додає сайту динамічності. Приклад використання в лістингах 3.4 та 3.5

Лістинг 3.4

Функція загрузки корзини за допомогою технології Ajax

```
function showCart() {
  if (!isEmpty($.cookie('cart')) || count($.cookie('cart')) == 2) {
    $('main-cart').html('<table><tbody><tr><td>Кошик
    порожній</div><div>Ваш кошик наразі порожній, додайте один з наших
    товарів щоб зробити замовлення.</div></td></tr></tbody></table>');
  }
}
```

```

else {
$.ajax({
url: "/ajaxData.php",
type: "POST",
data: "data="+$.cookie('cart'),
beforeSend: function() {
$(".main-cart").html('<div></div>');
},
success: function(html){
$(".main-cart").html(html);
$('.del-goods').on('click', delGoods);
}
});
}

```

Лістинг 3.5

Функція додавання товару до кошика

```

function addToCart() {
var id = $(this).attr('data-id');
var size = $(this).attr('data-size');
var new_product = count(cart);
cart.push([id,size]);

saveCart();
showCart();
document.location.href="/cart.php";
}

```

3.5 Розміщення веб-сайту на локальному сервері

Виконання php-сценаріїв виконується на стороні веб-сервера, тому для виконання розроблених програм мовою php необхідно встановити веб-сервер.

При створенні веб-сайтів перед публікацією їх у мережі Internet необхідно налаштувати проект на локальному веб-сервері. Це допоможе переглянути попередній вигляд сайту, та зекономити на сервері.

Для розміщення веб-сервера, обрано середовище віртуалізації VirtualBox, та конфігурацію Linux LAMP, яка містить у собі всі необхідні компоненти програмного забезпечення. Базовий пакет LAMP включає у себе:

PHP з підтримкою GD графіки і СУБД MySQL;

СУБД MySQL з підтримкою транзакцій (mysqld-max);

Apache з підтримкою SSI, mod_rewrite, mod_php - сервер, що приймає HTTP-запити від клієнтів, і повертає їм HTTP-відповіді, разом з усіма потрібними компонентами веб-сторінки;

phpMyAdmin – система керування MySQL через Web-інтерфейс;

Таким чином, встановивши Linux LAMP, можна створювати та тестувати програми, написані мовою PHP, на локальному веб-сервері.

ВИСНОВКИ

У результаті написання курсової роботи досліджено використання засобів розробки PHP, MySQL, JS, HTML та CSS. З метою закріплення отриманих

знань створено web-орієнтовану інформаційну систему магазину одягу, визначено потреби та можливості використання створеного сайту, спроектовано базу даних, а також запити до неї, налаштовано взаємодію сайту та бази даних.

У теоретичній частині курсової роботи у Розділі 1 описано бізнес потребу сайту та технічні завдання, що перед нами стоять.

У Розділі 2 встановлено потребу, яку виконує база даних, проаналізовано нормалізацію реляційних відношень, зазначено типи даних, сформовано правила для цілісності даних та складено запити до бази даних, які забезпечують повноцінну роботу сайту. Проектування бази даних реалізовано у програмному середовищі phpMyAdmin. База даних складається із чотирьох таблиць, має функціональні залежності між атрибутами таблиць та встановлені зв'язки між елементами бази даних. Також створено діаграми сутностей та зв'язків для демонстрації моделі даних.

Розділі 3 описує структуру створеного веб-сайту та макети його сторінок. Надано серверні скрипти на яких базується сайт, а саме серверна частина. Також описано структуру клієнтської частини.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Науковий блог [Електронний ресурс] – Режим доступу до ресурсу:
<https://naub.oa.edu.ua/2015/web-%D1%81%D0%B0%D0%B9%D1%82-%D1%8F%D0%BA-%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9->

%D0%B5%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82-
%D0%BC%D0%B0%D1%80%D0%BA%D0%B5%D1%82%D0%B8%D0%BD%D
0%B3%D1%83/.

2. Microsoft [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/office/troubleshoot/access/database-normalization-description>.

3. <https://rdb.dp.ua/uk/mnp> [Електронний ресурс] – Режим доступу до ресурсу: <https://rdb.dp.ua/uk/mnp>.

4. studfile [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/6354061/page:55/>.

5. МОДЕЛЮВАННЯ РЕЛЯЦІЙНОЇ СТРУКТУРИ БД [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5391756/>

6. НОРМАЛІЗАЦІЯ БАЗ ДАНИХ — ВІКІПЕДІЯ [Електронний ресурс]. – Режим доступу: <https://cutt.ly/tyTCqT8>

7. Пасічник В. В. Організація баз даних та знань / В.В. Пасічник, В.А. Резніченко. – К.: Видавнича група BVH, 2006. – 384 с.

8. Романюк О.Н. Організація баз даних та знань / О.Н. Романюк, Т.О. Савчук. – Вінниця: ВДТУ, 2001.

9. Ситник Н.В. Проектування баз та сховищ даних . – К.: КНЕУ, 2004.

10. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – СПб.: Питер, 2014. – 304 с.