

**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 навчальної дисципліни „ Технології проектування та  
адміністрування БД і СД ”**

**на тему:**

**Інформаційна система для магазину одягу**

**Галузь знань:** \_\_\_\_\_ 05 «Соціальні та поведінкові науки» \_\_\_\_\_

**Спеціальність:** \_\_\_\_\_ 051 «Економіка» \_\_\_\_\_

**Спеціалізація:** \_\_\_\_\_ «Інформаційні технології в бізнесі» \_\_\_\_\_

**Освітній ступінь:** \_\_\_\_\_ бакалавр \_\_\_\_\_

**Науковий керівник:**

к.е.н., Старух А. І.

(науковий ступінь, посада, прізвище, ініціали)

\_\_\_\_\_ “ \_\_\_\_\_ ” травня 2021 р.

(підпис)

**Виконавець:**

студент(ка) групи УФЕ-31 с

Гоц А. Р.

\_\_\_\_\_ “ \_\_\_\_\_ ” травня 2021 р.

(підпис)

**Загальна кількість балів** \_\_\_\_\_

\_\_\_\_\_  
(підпис, ППП членів комісії)

**ЛЬВІВ 2021**

## ЗМІСТ

ВСТУП.....	2
РОЗДІЛ 1. АНАЛІЗ ВИМОГ .....	4
1.1 Постановка завдання .....	4
1.2 Розробка моделі варіантів використання веб-сайту .....	5
1.3 Аналіз засобів реалізації.....	7
РОЗДІЛ 2. СТРУКТУРА БАЗИ ДАНИХ .....	10
2.1 Опис моделі даних .....	10
2.2. Нормалізація відношень .....	12
2.3 Визначення типів даних .....	14
2.4. Обмеження цілісності даних.....	18
2.5. Реалізація SQL-скрипту.....	23
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ .....	25
3.1 Структура веб-сайту .....	25
3.2 Макет сторінок веб сайту.....	28
3.3. Програмування серверної частини.....	29
3.4. Програмування клієнтської частини .....	31
3.5. Розміщення веб-сайту на локальному віртуальному середовищі або в Інтернеті .....	33
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	37
ДОДАТКИ.....	39

## ВСТУП

**Мотивація вибору теми та її актуальність.** Сьогодні інтернет став невід’ємною частиною життя людини. Тут люди відпочивають, спілкуються, вчаться, розвиваються, розважаються та роблять покупки. Існує безліч сайтів на різні теми, де можна знайти будь яку потрібну інформацію.

Онлайн бізнес – це не тільки актуально, але й дуже вигідно. Можна торгувати не лише в одній країні, а й по всій Землі. З кожним днем зростає кількість інтернет – магазинів та кількість клієнтів. Адже в інтернеті набагато легше та швидше можна знайти потрібний товар, при цьому навіть не виходячи з дому.

В 2021 році все більше компаній, фірм та підприємців виводять свій бізнес в онлайн – формат. Для успішного запуску інтернет – магазину потрібен сайт, що б відповідав усім критеріям та приваблював більше унікальних відвідувачів, які стануть клієнтами. А також база даних, де зберігатиметься вся інформація як про товар, так і про клієнтів та їх замовлення. Отож, дана тема є актуальною, адже веб-додатки та інформаційні системи зараз потрібні всім.

**Мета і завдання дослідження.** *Метою курсової роботи є обґрунтування теоретичних основ та реалізація Web - орієнтованої системи магазину одягу.*

Для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні завдання:

- проаналізувати предметну область інформаційної системи
- описати та розробити базу даних
- описати та розробити веб-сайт
- відобразити інформаційну систему за допомогою веб-додатку

*Об’єктом дослідження виступають база даних та веб-сайт.*

*Предметом дослідження є інформаційна система для магазину одягу.*

**Практичне значення отриманих результатів.** Інформаційна система магазину одягу, що містить базу даних та веб-сайт – зацікавить власників офлайн магазинів, що хотіли б просувати бізнес онлайн.

**Використане програмне забезпечення.** Для досягнення мети було використано мови програмування: HTML, CSS, JS, PHP, SQL, середовище розробки Visual Studio Code, MySQL та локальний сервер Open Server.

**Структура роботи.** Курсова робота складається зі вступу, трьох розділів («Аналіз вимог», «Розробка бази даних», «Розробка веб-додатку»), висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 48 сторінок.

## РОЗДІЛ 1. АНАЛІЗ ВИМОГ

### 1.1 Постановка завдання

Інформаційна система призначена для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів веб-додатку.

Інтернет-магазин повинен бути реалізований як веб-сайт, доступний в Інтернеті . Сайт повинен складатися з взаємозалежних частин, а функції кожної частини повинні бути чітко розділені. Даний додаток повинен мати наступні можливості:

- Кошик;
- Оформлення замовлення;
- Методи зворотнього зв'язку;
- Авторизація;
- Адміністративна панель;
- Можливість перегляду наповнення основних таблиць та результати запитів.

Сайт повинен складатися з двох частин, а саме адміністративної та користувацької.

Адміністративна частина сайту надає можливість додавання, редагування та видалення вмісту статичних, динамічних сторінок, розділів і блоків сайту.

Користувацький інтерфейс повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації та швидко та логічно переходити до розділів та сторінок.

Наявність навігації на всіх сторінках сайту. Правильна структура інформації дозволяє користувачам без проблем продовжувати дослідження сайту, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

Також має бути розроблена та під'єднана реляційна база даних.

При створенні сторінки, повинні використовуватись не сталі дані, а дані з бази даних. При оновленні даних у базі даних (БД) необхідно щоб вони автоматично оновлювались на сторінці сайту.

Для досягнення мети курсової роботи потрібно вирішити такі задачі:.

- Розробити моделі варіантів використання веб-сайту;
- Обрати та налаштувати інструменти реалізації;
- Проаналізувати та розробити базу даних;
- Реалізувати SQL- скрипт;
- Визначити структуру даних та спроектувати веб-сайт;
- Запрограмувати серверну та клієнтську частини;
- Розмістити веб-сайт на локальному віртуальному середовищі.

## **1.2 Розробка моделі варіантів використання веб-сайту**

Приклад діаграми використання може бути виражений як низхідний процес певного рівня від найбільш загальної та абстрактної концептуальної моделі вихідної системи до логіки відповідної програмної системи, а потім фізичної моделі.

Суть діаграми: розроблена система представлена у вигляді сутностей або суб'єктів, які взаємодіють із системою, використовуючи так звані випадки використання. У цьому випадку учасником є будь-яка організація, яка взаємодіє із системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система і може виступати джерелом впливу на систему моделювання, визначену розробником. У свою чергу, випадки використання використовуються для опису послуг, які система надає учасникам. Іншими словами, кожен випадок використання визначає певний набір операцій, які система виконує після розмови з учасниками. Приклад діаграми інтернет магазину представлено на рис. 1.1.

Інформація про акторів:

- Користувач – користувач інтернет-магазину одягу;
- Адміністратор – користувач, що виконує функції «адміністратора»;
- База даних – база даних, що зберігає інформацію.

Опис варіантів використання:

- Авторизація – функція входу в систему;
- Додавання відгуків – функція, щоб надати можливість користувачу залишити відгуки про товар;
- Перегляд товару – користувач має можливість переглянути товар на сайті;
- Пошук товару – можливість виконувати пошук бажаного товару на сайті;
- Додавання товару в кошик – функція для додавання обраного товару в кошик для оформлення замовлення;
- Редагування даних – адміністратор може редагувати дані на сайті;
- Додавання товару – адміністратор може додати новий товар на сайт;
- Перегляд замовлень – адміністратор може переглядати замовлення відвідувачів сайту;
- Оформлення замовлень – створення заявки замовлення обраного товару на сайті;

Було використано такі види взаємодії:

- Проста асоціація - відображається лінією між актором і варіантом використання (без стрілки). Відображає зв'язок актора і варіанта використання;
- Спрямована асоціація - те ж що і проста асоціація, але показує, що варіант використання ініціалізується актором. Позначається стрілкою;
- Включення (include) - показує, що варіант використання включається в базову послідовність і виконується завжди.

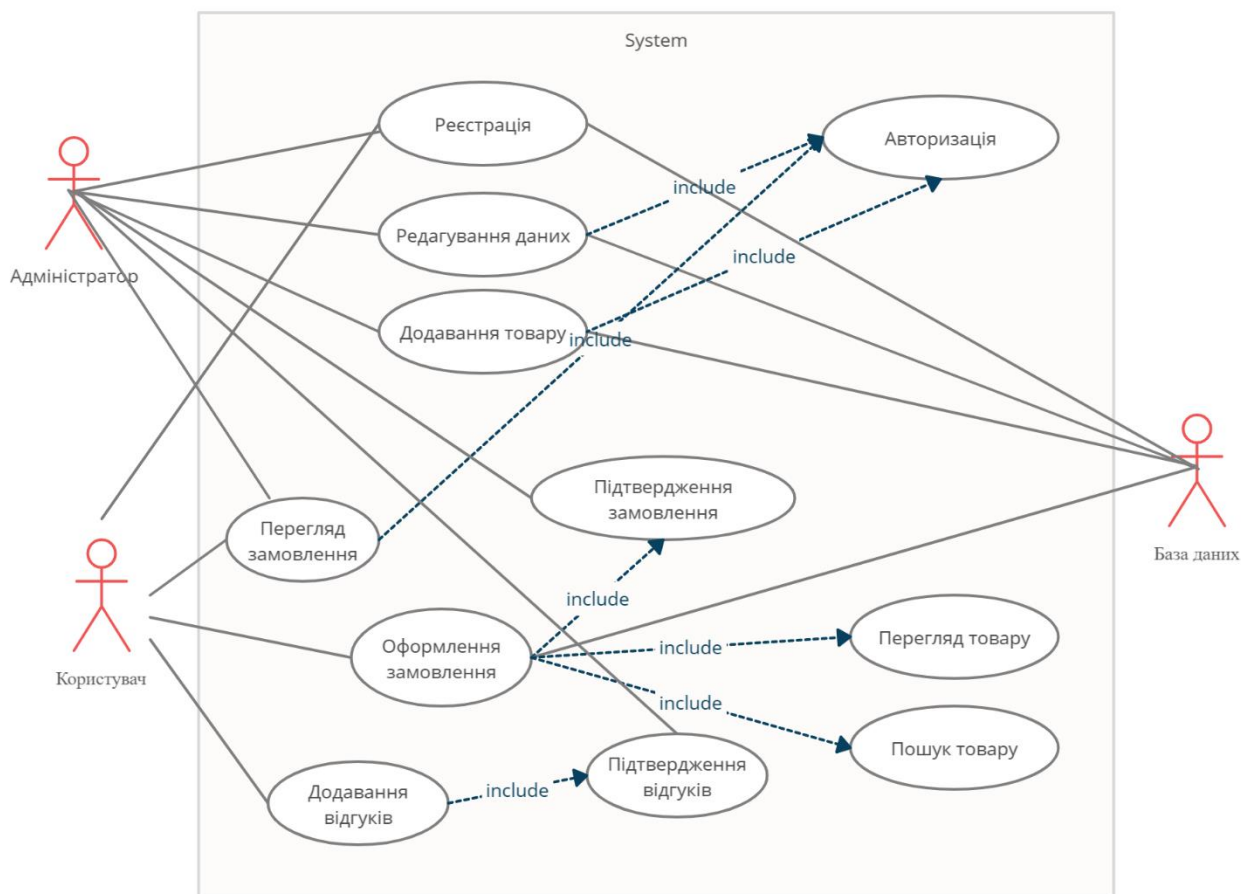


Рис. 1.1. Діаграма варіантів використання

### 1.3 Аналіз засобів реалізації

Для реалізації поставленого завдання потрібні правильно підібрані інструменти – мови програмування, середовища розробки, фреймворки.

Мова програмування – це формальна система знаків, призначена для розробки комп'ютерних програм. Мова програмування визначає набір команд, синтаксису та семантичних правил, що визначають зовнішній вигляд програми та дії, які виконує виконавець (комп'ютер) під її контролем.

Для розробки клієнтської частини сайту було обрано такі мови програмування:

- HTML - це мова розмітки гіпертексту тобто це мова веб-браузера, яка складає веб-сайт;



- CSS – це формальна мова, яка використовується для опису зовнішнього вигляду документів, написаних мовою розмітки. Розробники веб-сайтів використовують CSS для встановлення шрифтів, їх макета на сторінці та інших принципів того, як виглядають блоки веб-сторінок;

- JavaScript – що є «клієнтською» мовою програмування, це мова програмування для управління сценаріями перегляду веб-сторінок. Вона допомагає зробити сторінки сайту більш інтерактивними, обробляє дії користувачів сайту. Основна особливість цієї мови полягає в тому, що при використанні цієї мови елементи середовища відображення можуть бути змінені під час перегляду веб-сайту, тому веб-сторінку не буде перезавантажено. Наприклад, за допомогою JavaScript можна змінити колір фону веб-сторінки, замінити зображення, інтегровані у веб-сторінку, створити нове вікно відображення, або відобразити повідомлення.

Для розробки серверної частини сайту було обрано такі мови програмування:

- SQL – дозволяє отримувати конкретні дані з великих і складних баз даних. Переваги: стандартність, незалежність від конкретних СУБД, реляційна основа мови, можливість створення інтерактивних запитів, можливість динамічної зміни й розширення структури БД;

- PHP - є загальноприйнятою скриптовою мовою, яка широко використовується в розробці веб-додатків. Основними його перевагами є: безкоштовне програмне забезпечення, простий синтаксис, висока швидкість та велике співтовариство розробників. Аббревіатура PHP означає Preprocessor for Homepage. Це мова скрипту, яка працює на сервері, вбудованому в сторінку HTML. Його мета – дозволити якнайшвидше створювати динамічні веб-сторінки. Можна з упевненістю сказати, що дослідження та використання PHP корисні як для початківців, так і для професійних програмістів. PHP – це динамічна мова програмування, в якій немає необхідності вказувати тип даних під час декларування змінних та декларування самих змінних. PHP підтримує декілька об'єктно-орієнтованих функцій і повністю підтримується у п'ятій версії

мови. PHP підтримує всі три основні механізми ООР – інкапсуляція, поліморфізм та успадкування тобто батьківський клас задається за допомогою ключового слова `extends` після назви класу.

Для написання коду обрано середовище розробки Visual Studio Code.

Переваги:

- Багато налаштувань (як для всієї програми, так і для інтерфейсу);
- Розширювана бібліотека доповнень і готових рішень;
- Мультифункціональність (редактор підтримує майже всі мови, які використовують для створення додатків);

- Простота та гнучкість;

Для розробки бази даних було обрано середовище MySQL. Переваги:

- Гарна продуктивність;
- Простота у використанні;
- Надійність та безпека.

Також було використано Open Server – перспективний локальний веб-сервер, що поєднує в собі максимальну простоту з високою функціональністю.

Переваги:

- Швидко запускається і встановлюється;
- Сервер автоматично починає роботу при запуску програми;
- Можна працювати через командний рядок;
- Можна перемикати модулі HTTP, MySQL, PHP;
- Інтерфейс підтримує декілька мов.

## РОЗДІЛ 2. СТРУКТУРА БАЗИ ДАНИХ

### 2.1 Опис моделі даних

База даних – це впорядкований за певними правилами набір взаємопов’язаних даних. Основним призначенням бази даних є гарантоване систематизоване збереження даних і надання доступу до них користувачеві або комп’ютерній програмі, тобто забезпечення швидкого пошуку потрібних даних.

Основою бази даних є модель даних. Модель даних – це фіксована система понять і правил для представлення даних структури, стану та динаміки проблемної області в базі даних. Під час створення моделі бази даних використовують ту чи іншу модель даних. Найпоширенішими є такі види моделей даних: ієрархічна, мережева, реляційна. В курсовій роботі основою бази даних є реляційна модель.

Реляційна модель – модель, де дані та взаємозв’язки між ними подаються у формі прямокутних таблиць. Кожна таблиця відповідає за певний конкретний об’єкт і містить всі його атрибути. Назва «реляційна» (relational) пов’язана з тим, що кожен запис у таблиці даних містить інформацію, яка стосується (related) якогось конкретного об’єкта. Крім того, зв’язані між собою (тобто такі, що знаходяться в певних відношеннях — relations) дані навіть різних типів в моделі можуть розглядатися як одне ціле.

Для роботи з реляційною базою даних було обрано Систему управління базами даних – MySQL. Для управління базою даних використано веб-додаток – phpMyAdmin, що написаний мовою PHP і має зручний графічний веб-інтерфейс для адміністрування бази даних MySQL.

Важливим етапом створення таблиць є вибір кодування, від якого залежить відображення даних на веб-сторінці. utf8\_general\_ci – дозволяє використовувати та зберігати в таблицях бази даних символи кирилиці.

В phpMyAdmin створено базу даних «stimma\_shop», що містить інформацію про інтернет-магазин одягу. Створено декілька таблиць, що містять різні (атрибути) поля. Для полів визначено первинні (Product key). Первинні ключі призначені для атрибутів, що мають унікальні значення. Кожна таблиця може мати тільки 1 РК.

На рисунку 2.1 показано діаграму сутностей та зв'язків бази даних. Первинні ключі виділені жирним шрифтом.

В таблиці Orders, що містить інформацію про товари, є такі поля:

- **id** - РК
- name
- phone
- comment
- address
- product

В таблиці Products, що містить інформацію про клієнтів магазину, є такі поля:

- **id** - РК
- name
- price
- img
- category

В таблиці Products\_category, що містить інформацію про замовлення клієнтів, є такі поля:

- **id** - РК
- name

В таблиці Users, що містить інформацію про адміністраторів сайту, є такі поля:

- **id** - РК
- username

- email
- password

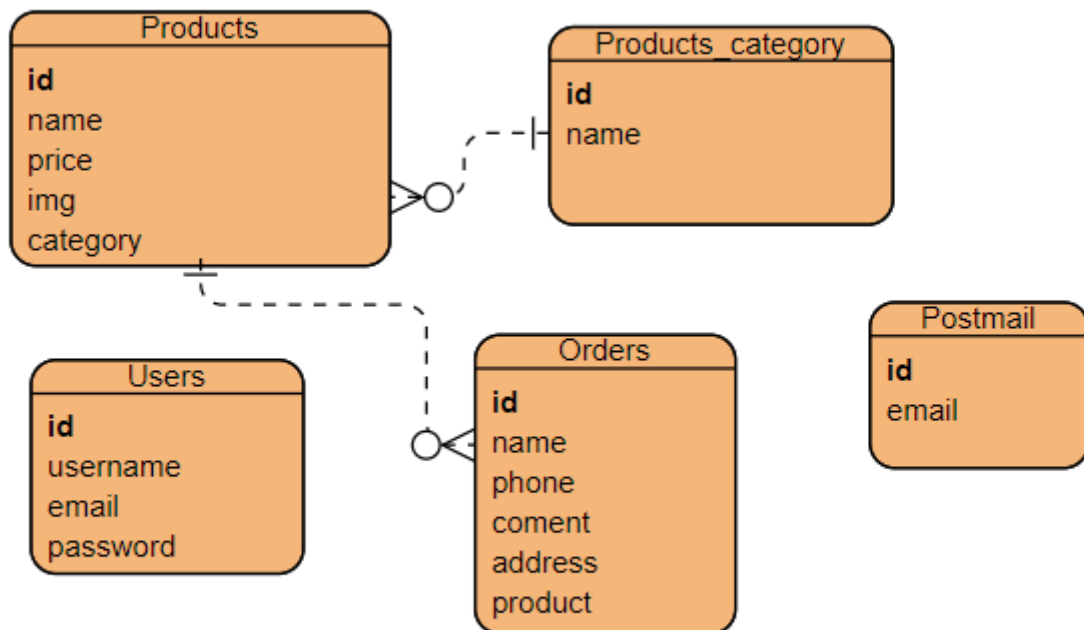


Рис. 2.1. Діаграма сутностей і зв'язків

## 2.2. Нормалізація відношень

Нормалізація – це процес, у результаті якого можна позбутися недоліків проектування бази даних. У процесі нормалізації одержують ряд нормальних форм, використовуючи набір правил, що описують те, що слід і що не слід робити із структурою таблиць. Процес нормалізації складається з розбиття таблиць на менші, внаслідок чого формується краща структура.

Нормалізація бази даних — покроковий процес розбиття одного відношення відповідно до алгоритму нормалізації на декілька відношень на базі функціональних залежностей.

Нормалізація таблиць бази даних - перший крок на шляху проектування структури реляційної бази даних.

Виділяються шість нормальних форм:

- (1NF) перша нормальна форма;
- (2NF) друга нормальна форма;
- (3NF) третя нормальна форма;
- (BCNF) нормальна форма Бойса-Кодда;
- (4NF) четверта нормальна форма;
- (5NF) п'ята нормальна форма;

В нормалізованій базі даних зменшується ймовірність виникнення помилки, і вона займає менше місця на жорсткому диску.

Переваги нормалізації:

- краща загальна організація бази даних;
- скорочення кількості непотрібного повторювання даних;
- узгоджування даних в базі даних;
- більш гнучка структура бази даних;

ефективні можливості забезпечення безпеки та надійності бази даних.

Нормалізація - це послідовне перетворення початкової бази даних до NF, при цьому кожна наступна NF обов'язково містить у собі попередню. На практиці обмежуються 3NF, її достатньо для створення надійної схеми бази даних.

Перша нормальна форма (1NF). Відношення знаходиться в 1NF, якщо на перетині кожного стовпця з кожним рядком міститься лише атомарне (нерозподілене) значення. Тобто, кожен атрибут відношення повинен зберігати одне єдине значення та не бути ні списком, ні множиною значень. Аби перейти до 1NF треба проглянути відношення та розподілити складові атрибути на різні рядки/стовпці. Можливо, цю ситуацію доведеться повторити декілька разів до тих пір, доки кожен із атрибутів не стане атомарним.

Друга нормальна форма (2NF). Відношення знаходиться в 2NF, якщо воно знаходиться в першій нормальній формі, і кожен неключовий атрибут (тобто що не є складовою частиною первинного ключа) функціонально повно залежить від первинного ключа.

Третя нормальна форма (3NF). Відношення знаходиться в 3NF, якщо воно знаходиться в другій нормальній формі та кожен його неключовий атрибут безпосередньо залежить від первинного ключа.

В ідеалі при нормалізації треба домогтися, щоб будь-яке значення зберігалось в базі в одному примірнику, причому значення це не має бути отримано розрахунковим шляхом з інших даних, що зберігаються в базі. Нормалізована база повинна містити декілька менших таблиць, а не одну велику.

База даних «shop\_stimma» містить декілька таблиць, що відповідають за різні суб'єкти та мають різні поля. Загалом розроблена база даних відповідає трьом формам нормалізації, адже на перетині кожного стовпця з кожним рядком містяться лише атомарні значення, кожен неключовий атрибут функціонально та безпосередньо залежить від первинного ключа.

### 2.3 Визначення типів даних

Нормалізовані реляційні відношення отримані в попередній главі не містять атрибутів, які є зовнішніми ключами, тобто реалізують зв'язки між відношеннями. Полям призначено зовнішні ключі і це показано на Рис. 2.2, FK виділені курсивом, а РК – жирним шрифтом.

Зовнішні (Foreign key) ключі - призначені для одного стовпця, або для комбінації стовпців, аби забезпечити та встановити зв'язок між даними у двох таблицях. FK посилається на первинний ключ в іншій таблиці. Тип даних зовнішнього ключа має співпадати з типом даних первинного ключа пов'язаного відношення. Назви зовнішніх ключів повинні бути у відповідності до назв пов'язаних відношень.

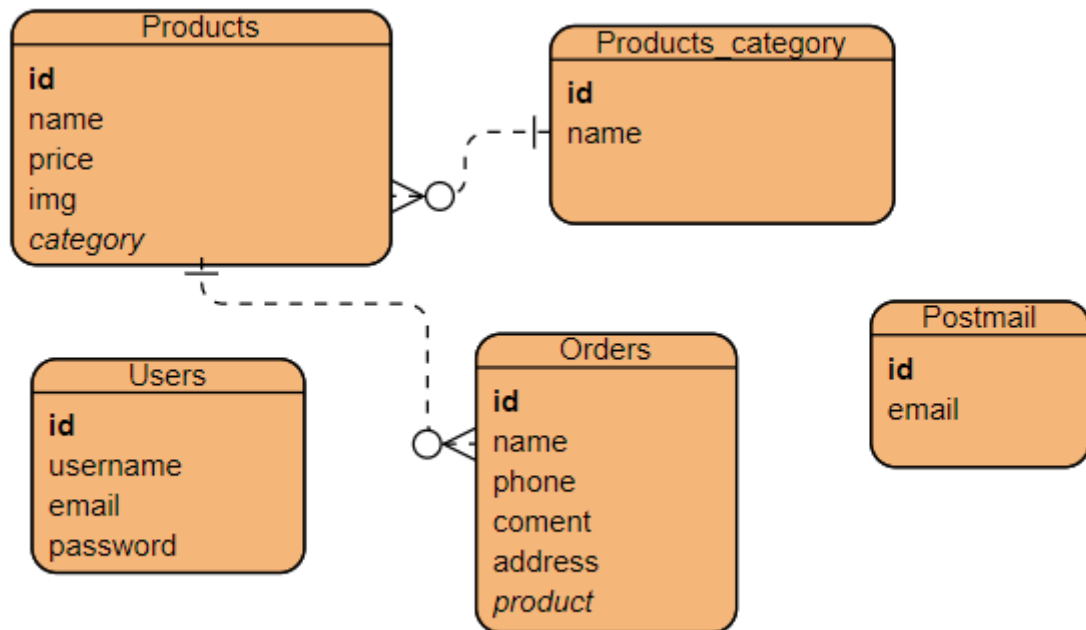


Рис. 2.2. Діаграма сутностей і зв'язків з РК та FK

MySQL підтримує кілька типів стовпців, які можна розділити на три категорії:

1. Числові типи даних;
2. Типи даних для зберігання дати та часу;
3. Символьні (строкові) типи даних.

Короткий список типів даних:

Числові типи даних:

- TINYINT [(M)] [UNSIGNED] [ZEROFILL] - Дуже мале ціле число.

Діапазон зі знаком від -128 до 127. Діапазон без знаку від 0 до 255;

- BIT, BOOL є синонімами для TINYINT (1);
- SMALLINT [(M)] [UNSIGNED] [ZEROFILL] - Мале ціле число.

Діапазон зі знаком від -32768 до 32767. Діапазон без знаку від 0 до 65535;

- MEDIUMINT [(M)] [UNSIGNED] [ZEROFILL] - Ціле число середнього розміру. Діапазон зі знаком від -8388608 до 8388607. Діапазон без знаку від 0 до 16777215;



- `INT [(M)] [UNSIGNED] [ZEROFILL]` - Ціле число нормального розміру. Діапазон зі знаком від -2147483648 до 2147483647. Діапазон без знаку від 0 до 4294967295;

- `INTEGER [(M)] [UNSIGNED] [ZEROFILL]` - синонім для `INT`;

- `BIGINT [(M)] [UNSIGNED] [ZEROFILL]` Велике ціле число. Діапазон зі знаком від -9223372036854775808 до 9223372036854775807. Діапазон без знаку від 0 до 18446744073709551615.

Типи даних для зберігання дати та часу:

- `DATE` - Дата. Підтримується інтервал від '1000-01-01' до '9999-12-31'. MySQL виводить значення `DATE` у форматі 'YYYY-MM-DD', але можна встановити значення в стовпець `DATE`, використовуючи як рядки, так і числа.

Символьні (строкові) типи даних:

- `[NATIONAL] CHAR (M) [BINARY]` - Рядок фіксованої довжини, при зберіганні завжди доповнюється пробілами в кінці рядка до заданого розміру. діапазон аргументу `M` становить від 0 до 255 символів (від 1 до 255 в версіях, що передують MySQL 3.23). Кінцеві прогалини видаляються при виведенні значення. Якщо не заданий атрибут чутливості до регістру `BINARY`, то величини `CHAR` упорядковано і порівнюються як незалежні від регістру відповідно до встановленого за замовчуванням алфавітом. Атрибут `NATIONAL CHAR` (Або його еквівалентна коротка форма `NCHAR`) Являє собою прийнятий в ANSI SQL спосіб вказівки, що в стовпці `CHAR` повинен використовуватися встановлений за замовчуванням набір символів (`CHARACTER`). В MySQL це прийнято за замовчуванням. `CHAR` є скороченням від `CHARACTER`. MySQL дозволяє створювати стовпець типу `CHAR (0)`;

- `CHAR` - Це синонім для `CHAR (1)`;

- `VARCHAR (M) [BINARY]` - Рядок змінної довжини. Кінцеві прогалини видаляються при збереженні значення (в цьому полягає відмінність від специфікації ANSI SQL). діапазон аргументу `M` становить від 0 до 255 символів (від 1 до 255 у версіях, що передують MySQL Version 4.0.2). Якщо не

заданий атрибут чутливості до регістру BINARY, то величини VARCHAR упорядковано і порівнюються як незалежні від регістру. термін VARCHAR є скороченням від CHARACTER VARYING;

- TINYBLOB, TINYTEXT - Стовбець BLOB або TEXT з максимальною довжиною  $2^8 - 1$  символів;
- BLOB, TEXT -Стовбець BLOB або TEXT з максимальною довжиною  $2^{16} - 1$  символів;
- MEDIUMBLOB, MEDIUMTEXT - Стовбець BLOB або TEXT з максимальною довжиною  $2^{24} - 1$  символів;
- LONGBLOB, LONGTEXT - Стовбець BLOB або TEXT з максимальною довжиною  $2^{32} - 1$  символів.

*Для атрибутів реляційних відношень визначено:*

- Типи даних та їх розміри
- Чи може атрибут дублюватись
- Чи є обов'язковим значення атрибута
- Можливі значення за замовчуванням

*Таблиця 2.3.1*

#### **Тип даних реляційного відношення «Products»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
name	VARCHAR	100	-
price	INT	11	-
img	VARCHAR	100	-
category	INT	11	Зовнішній ключ

*Таблиця 2.3.2*

#### **Тип даних реляційного відношення «Products\_categories»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
name	VARCHAR	100	-

*Таблиця 2.3.3*

#### **Тип даних реляційного відношення «Orders»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
email	VARCHAR	100	-
name	VARCHAR	100	-
phone	VARCHAR	50	-
comment	VARCHAR	255	-
address	VARCHAR	100	-
product	VARCHAR	100	Зовнішній ключ

Таблиця 2.3.4

**Тип даних реляційного відношення «Users»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
username	VARCHAR	100	-
email	VARCHAR	100	-
password	VARCHAR	100	-

Таблиця 2.3.5

**Тип даних реляційного відношення «Postmail»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
email	VARCHAR	100	-

**2.4. Обмеження цілісності даних**

Цілісність – це достовірність і точність інформації, що зберігається в базі. Цілісність досягається забезпеченням відповідності даних певним додатковим обмеженням, крім тих, які накладаються схемою бази на структуру даних та їхні типи.

Обмеження цілісності — це правила, які обмежують усі можливі стани бази даних, а також переходи з одного стану в інший. Таким чином, обмеження цілісності визначають множину «допустимих» станів і переходів між ними. База даних перебуває в цілісному стані, якщо вона відповідає всім визначеним для неї вимогам цілісності.

При створенні баз даних велика увага має бути приділена засобам підтримки даних в цілісному стані.

Засоби завдання обмежень вводяться з метою захисту бази даних від порушення узгодженості даних, що зберігаються в ній. До таких типів підтримки цілісності даних відносяться:

- обов'язкові дані;
- обмеження для доменів полів;
- цілісність сутностей;
- посилавна цілісність ;
- вимоги конкретного підприємства.

У SQL стандарту ANSI є 4 основних обмеження: PRIMARY KEY, CHECK, UNIQUE і FOREIGN KEY.

Велика частина перерахованих обмежень задається в операторах CREATE TABLE і ALTER TABLE.

#### **Створення таблиці:**

```
CREATE TABLE ім'я_таблиці
{(ім'я_стовпця тип_даних [ NOT NULL ][ UNIQUE]
[DEFAULT <значення>]
[ CHECK (<умова_вибору>)][,..n]}
[CONSTRAINT ім'я_обмеження]
[PRIMARY KEY (ім'я_стовпця [,..n])
{[UNIQUE (ім'я_стовпця [,..n])}]
[FOREIGN KEY (ім'я_стовпця_зовнішнього_ключа
[,..n])
REFERENCES ім'я_рід_таблиці
[(ім'я_стовпця_рід_таблиці [,..n])],
[MATCH {PARTIAL | FULL}]
[ON UPDATE {CASCADE| SET NULL |SET DEFAULT
|NO ACTION}]
[ON DELETE {CASCADE| SET NULL |SET DEFAULT
```

[NO ACTION}]

{[CHECK(<умова\_вибору>)] [...n]})

Обов'язкові дані:

Для деяких стовпців потрібно наявність в кожному рядку таблиці конкретного і допустимого значення, відмінного від опущеного значення або значення NULL. Для завдань обмежень подібного типу стандарт SQL передбачає використання специфікації NOT NULL.

Вимоги конкретного підприємства:

Оновлення даних в таблицях можуть бути обмежені існуючими в організації вимогами (бізнес-правилами). Стандарт SQL дозволяє реалізувати бізнес-правила підприємств за допомогою пропозиції CHECK і ключового слова UNIQUE.

Обмеження для доменів полів:

Кожен стовпець має власний домен - деякий набір допустимих значень. Стандарт SQL передбачає два різні механізми визначення доменів. Перший полягає у використанні пропозиції CHECK, що дозволяє задати необхідні обмеження для стовпця або таблиці в цілому, а другою припускає застосування оператора CREATE DOMAIN.

Цілісність сутностей:

Первинний ключ таблиці повинен мати унікальне не порожнє значення в кожному рядку. Стандарт SQL дозволяє задавати подібні вимоги підтримки цілісності даних за допомогою фрази PRIMARY KEY. В межах таблиці вона може вказуватися тільки один раз. Проте існує можливість гарантувати унікальність значень і для будь-яких альтернативних ключів таблиці, що забезпечує ключове слово UNIQUE. Крім того, при визначенні альтернативних ключів рекомендується використовувати і специфікатори NOT NULL.

Посилальна цілісність:

Зовнішні ключі є стовпцями або наборами стовпців, призначеними для зв'язування кожної з рядків дочірньої таблиці, що містить цей зовнішній ключ, з рядком батьківської таблиці, що містить відповідне значення потенційного

ключа. Стандарт SQL передбачає механізм визначення зовнішніх ключів за допомогою пропозиції FOREIGN KEY, а фраза REFERENCES визначає ім'я батьківської таблиці, тобто таблиці, де знаходиться відповідний потенційний ключ. При використанні цієї пропозиції система відхилить виконання будь-яких операторів INSERT або UPDATE, за допомогою яких буде зроблена спроба створити в дочірній таблиці значення зовнішнього ключа, що не відповідає одному із вже існуючих значень потенційного ключа батьківської таблиці. Коли дії системи виконуються при вступі операторів UPDATE і DELETE, що містять спробу відновити або видалити значення потенційного ключа у батьківській таблиці, якому відповідає одна або більше за рядки дочірньої таблиці, то вони залежать від правил підтримки посилювальної цілісності, вказаних у фразах ON UPDATE і ON DELETE пропозиції FOREIGN KEY. Якщо користувач робить спробу видалити з батьківської таблиці рядок, на який посилюється одна або більше за рядки дочірньої таблиці, мова SQL надає наступні можливості:

- CASCADE - виконується видалення рядка з батьківської таблиці, що супроводжується автоматичним видаленням усіх рядків дочірньої таблиці, що посилюються на неї;
- SET NULL - виконується видалення рядка з батьківської таблиці, а в зовнішні ключі усіх рядків дочірньої таблиці, що посилюються на неї, записується значення NULL ;
- SET DEFAULT - виконується видалення рядка з батьківської таблиці, а в зовнішні ключі усіх рядків дочірньої таблиці, що посилюються на неї, заноситься значення, що приймається за умовчанням;
- NO ACTION - операція видалення рядка з батьківської таблиці відміняється. Саме це значення використовується за умовчанням в тих випадках, коли в описі зовнішнього ключа фраза ON DELETE опущена.

Ті ж самі правила застосовуються в мові SQL і тоді, коли значення потенційного ключа батьківської таблиці оновлюється.

Визначник MATCH дозволяє уточнити спосіб обробки значення NULL в зовнішньому ключі.

При визначенні таблиці пропозиція FOREIGN KEY може вказуватися довільна кількість разів.

У операторі CREATE TABLE використовується необов'язкова фраза DEFAULT, яка призначена для задання по замовчуванню значення, коли в операторі INSERT значення в цьому стовпці буде відсутнє.

Фраза CONSTRAINT дозволяє задати ім'я обмеженню, що дозволить згодом відмінити те або інше обмеження за допомогою оператора ALTER TABLE.

### **Зміна і видалення таблиці:**

```
ALTER TABLE ім'я_таблиці
[ADD [COLUMN]ім'я_стовпця тип_даних
[ NOT NULL ][UNIQUE]
[DEFAULT <значення>][ CHECK (<умова_вибору>)]]
[DROP [COLUMN] ім'я_стовпця [RESTRICT | CASCADE ]]
[ADD [CONSTRAINT [ім'я_обмеження]]
[{PRIMARY KEY (ім'я_стовпця [,..n])
[UNIQUE (ім'я_стовпця [,..n])}]
[FOREIGN KEY (ім'я_стовпця_зовнішнього_ключа [,..n])
REFERENCES ім'я_рід_таблиці
(ім'я_стовпця_рід_таблиці [,..n])],
[ MATCH {PARTIAL | FULL}
[ON UPDATE {CASCADE| SET NULL |
SET DEFAULT | NO ACTION}]
[ON DELETE {CASCADE| SET NULL |
SET DEFAULT | NO ACTION}]
[CHECK(<умова_вибору>)][,..n]}}]
[DROP CONSTRAINT ім'я_обмеження
[RESTRICT | CASCADE]]
[ALTER [COLUMN] SET DEFAULT <значення>]
[ALTER [COLUMN] DROP DEFAULT]
```

Тут параметри мають те ж саме призначення, що і у визначенні оператора CREATE TABLE.

Оператор ALTER TABLE реалізований не в усіх діалектах мови SQL. У деяких діалектах він підтримується, проте не дозволяє видаляти з таблиці вже існуючі стовпці.

Для видалення таблиці використовується команда DROP TABLE.

## 2.5. Реалізація SQL-скрипту

Sql- скрипт – це програмний код, який реалізує запити, оновлення, видалення, тощо в базу даних. Не завжди в результаті успішного компілювання скрипту міняється структура бази даних.

Так наприклад, якщо взяти скрипт запиту, то він лише добуде певну інформацію з бази даних, яка може бути оброблена та візуалізована користувачу.

Щоб реалізувати sql скрипт необхідно для початку під'єднатись до бази даних. Це можна зробити наступним методом.

```
<?php $db = mysqli_connect('localhost', 'root', 'root', 'shop_stimma'); ?>
```

Реєстрування користувачів відбувається через запит:

```
$query = "INSERT INTO users (username, email, password)
VALUES('$username', '$email', '$password');"
```

Також використовується запит на існування користувачів в базі даних при вході, а саме перевірка логіну та паролю:

```
"SELECT*FROM users WHERE username='$username' AND password='$password'";
```

Користувачі не можуть використовувати для реєстрації один і той самий логін та пошту:

```
"SELECT * FROM users WHERE username='$username' OR email='$email'
LIMIT 1";
```

Також можна додати товар на сайт в базу даних:



```
$query="insert into products(name,price,img,category) VALUES ('$name','$price','$filename','$category')";
```

```
}else{
```

```
$query="insert into products(name,price,category) VALUES ('$name','$price','$category')";
```

Для виводу всіх товарів на сайт з бази даних:

```
$query="select name,img,price from products"
```

Для виводу товарів, що відносяться до категорії «Для дівчат»:

```
$query="select name,img,price from products where category=1"
```

«Для хлопців»:

```
$query="select name,img,price from products where category=2"
```

Запит, що вносить дані в таблицю «orders», при оформленні замовлення на сайті через форму:

```
$query="insert into orders(email,name,phone,comment,address,product) VALUES ('$email','$name','$comment','$phone','$address', $product)";
```

Запит, що вносить дані в таблицю «postmail», при оформленні підписки на сайті через форму:

```
$query="insert into orders(email,name,phone,comment,address,product) VALUES ('$email','$name','$comment','$phone','$address','$product)";
```

## РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ

### 3.1 Структура веб-сайту

Веб-сторінка— інформаційний ресурс, доступний у мережі World Wide Web, який можна переглянути у веб-браузері. Зазвичай ця інформація записана у форматі HTML або XHTML і може містити гіпертекст із навігаційними гіперпосиланнями на інші веб-сторінки.

Веб-сайт— сукупність веб-сторінок, доступних в Інтернеті, які об'єднані як за змістом, так і навігаційно.

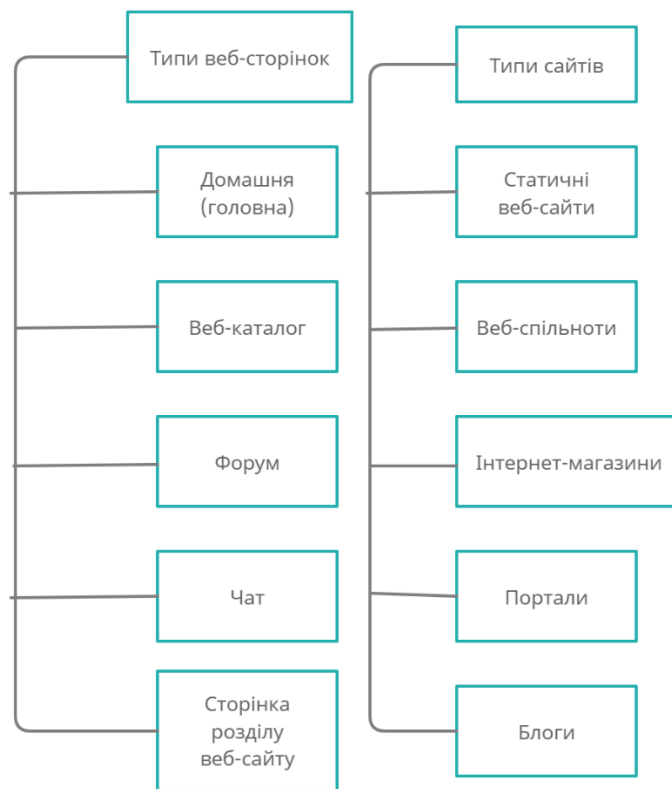


Рис. 3.1. Типи веб-сторінок та веб-сайтів

Структура веб-сайтів:

1. Головна сторінка
2. Меню сайту
3. Гіперпосилання на інші сторінки або сайти

На будь-якому сайті першою відкривається головна сторінка. У верхній частині головної сторінки зазвичай розташована так звана шапка, яку дублюють

на інших сторінках сайту. Це роблять спеціально, адже ця частина відображається у вікні браузера першою і відвідувач, насамперед, звертає увагу на неї. Щоб забезпечити швидкий перехід до основних тематичних розділів сайту, створюють меню сайту — список гіперпосилань на його розділи. Горизонтальне меню зазвичай розташовують у шапці, іноді дублюючи його в нижній частині сторінки, а вертикальне — переважно в лівій частині сторінки, у місці, звідки відвідувач починає її переглядати. Меню є одним із найважливіших компонентів сайту, користувач постійно звертає на нього увагу, і тому вимоги до нього високі. Меню має бути зручним, помітним і зрозумілим, інакше користувач не знатиме, як перейти до потрібного розділу, і покине сайт. Пункти меню мають бути чітко відділені один від одного. Гіперпосилання, розміщені в тексті чи у вигляді графічних об'єктів, дозволяють переходити на різні сторінки сайту або навіть на інші сайти. На сайтах із дуже великим обсягом інформації є сторінки третього рівня, а якщо необхідно — то й четвертого, п'ятого тощо.

Виділяють три типи структур веб-сайтів — лінійну, деревоподібну та довільну. Подорожуючи сайтом із *лінійною* структурою, з головної сторінки ви перейдете на другу сторінку, з неї — на третю тощо. На сайті з *деревоподібною* структурою з головної сторінки можна потрапити на одну зі сторінок другого рівня, звідти — на одну зі сторінок третього рівня тощо. Сайт із *довільною* структурою видається зовсім неорганізованим, але саме в цьому й полягає принцип його створення. Подорожуючи таким сайтом, можна переходити з однієї його сторінки на інші в різні способи, і шлях назад не обов'язково має бути таким самим. Вибір структури визначається особливостями завдань, що розв'язуються за допомогою веб-сайту.

Структура сайту, розробленого для курсової роботи:

Довільна або павутина. У цьому випадку всі сторінки веб-сайту містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу. Ця схема може перетворитися на лабіринт, якщо вийде з-під контролю, але вона популярна в тих випадках, коли посиланнями на документи користуються не надто часто.

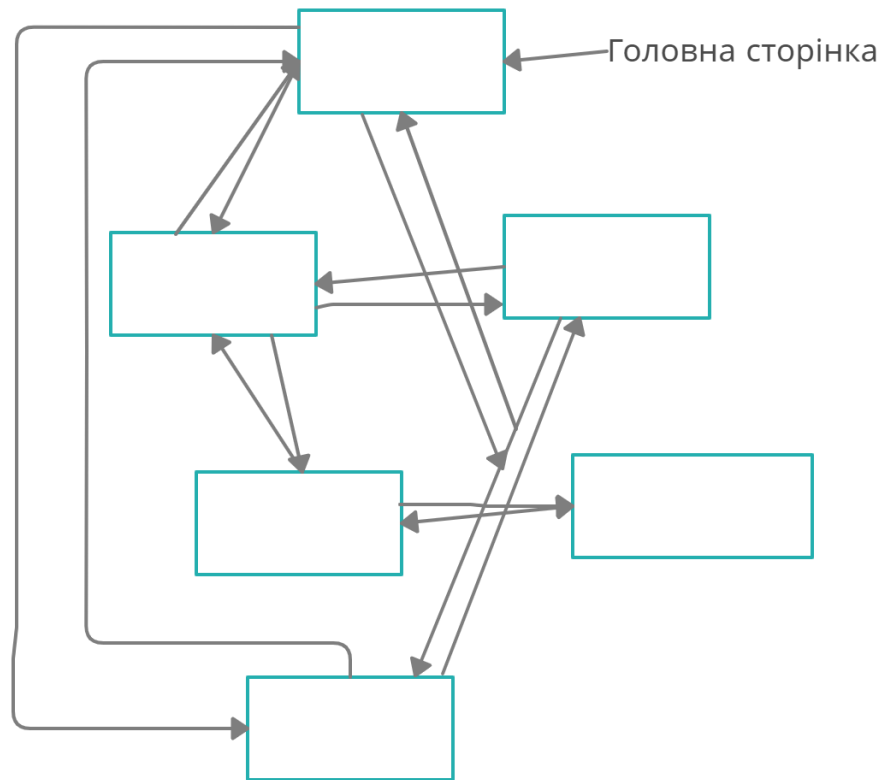


Рис. 3.2. Схема структури сайту

Основне призначення веб-сайту: сайт, де користувач з легкістю може зробити замовлення.

Можливості, що повинен реалізовувати веб-сайт:

- перегляд основної інформації про магазин;
- перегляд товарів, їх цін та наявності;
- реєстрація на сайті;
- підписка на оновлення магазину;
- перегляд розмірів одягу;
- перегляд контактів магазину;
- посилання на соцмережі;
- перегляд адреси магазину, через гугл карти;
- оформлення замовлення.

Веб-сайт складається з таких сторінок:

- Головна сторінка, де можна побачити меню, гасло сайту, переглянути декілька товарів, що найбільше купують, підвал сайту;

- Для дівчат – сторінка, де можна побачити перелік жіночого одягу, а також меню та підвал сайту;
- Для хлопців - сторінка, де можна побачити перелік чоловічого одягу, а також меню та підвал сайту;
- Товари - сторінка, де можна побачити перелік всього одягу, а також меню та підвал сайту;
- Вхід – сторінка, зареєстрований користувач може зайти на сайт;
- Реєстрація – сторінка, де користувач може зареєструватись;
- Розміри – сторінка, де можна побачити розміри одягу, а також меню та підвал сайту;
- Кошик – сторінка, де можна зробити замовлення;

### 3.2 Макет сторінок веб сайту

Макет сайту – це візуальний образ майбутнього проекту, розроблений з урахуванням технічних можливостей HTML верстки. Такий макет показує, як візуально буде виглядати сайт після верстки та наповнення.

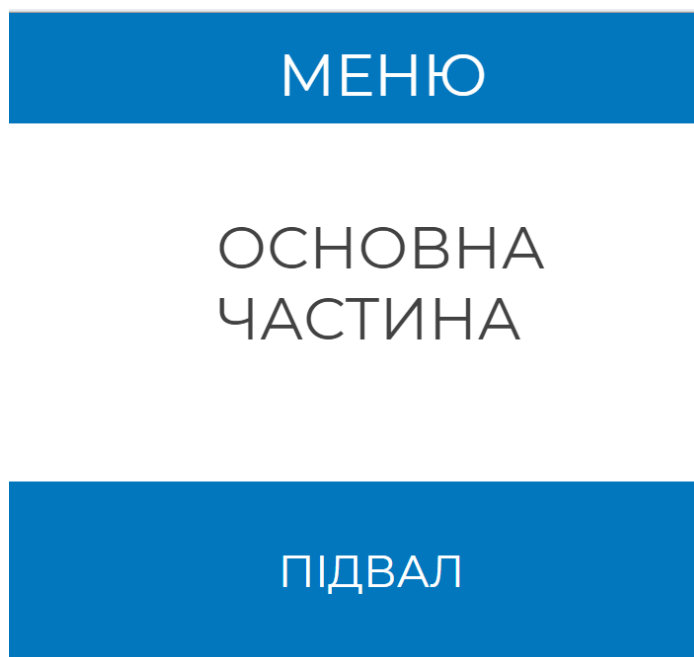


Рис. 3.3. Макет сайту

В частині «Меню» - буде розміщено логотип, посилання на сторінку кошик, номер телефону, пошук на сайті та навігаційна панель, що складається з переліку основних сторінок і на які можна перейти. Для кожної сторінки сайту ця частина однакова, окрім сторінки «Вхід» та «Реєстрація».

В частині «Основна частина» - буде розміщено основну інформацію, на кожній з основних сторінок різне наповнення, на головній – гасло магазину, декілька товарів, на сторінці «Для дівчат» - жіночий одяг, на сторінці «Для хлопців» - чоловічий одяг, на сторінці «Товари» - весь одяг, на сторінці «Вхід» - форма для входу, на сторінці «Реєстрація» - форма для реєстрації, на сторінці «Розміри» - розміри одягу, на сторінці «Кошик» - форма для замовлення.

В частині «Підвал» - буде розміщено блоки, перший – посилання на соц мережі, другий – підписка на оновлення магазину, третій – перелік категорій одягу, четвертий – перелік сторінок з одягом, п'ятий – перелік доставки товарів, шостий – контакти магазину та гугл карта, де відразу можна побачити розташування магазину.

Вигляд сайту після написання коду та розміщення на сервері можна побачити в Додатку А.

### **3.3. Програмування серверної частини**

Розробка динамічного веб-сайту складається з двох частин, а саме серверної (Backend) та клієнтської (Frontend).

Backend – це та «задня» частина сайту, яка невидима відвідувачеві, але яка безпосередньо відповідає за роботу сайту. Якщо порівняти сайт з кораблем, то бекенд - це машинне відділення. Для простоти можна вважати, що бекенд - це CMS або движок сайту з усіма його зв'язками з механізмами хостинга - базою даних, веб-сервером, поштовими службами.

Backend - це ніби «невидимий двигун» сайту. Для написання серверної частини використовують такі популярні мови програмування, як PHP, Ruby on

Rails, Python, .NET та інші. Коли потрібно виконати якусь операцію, Backend-код взаємодіє з базою даних (використовуючи MySQL, SQL, Microsoft Access і т.д.). Після цього необхідна інформація повертається користувачу в формі Frontend -коду.

Для написання Backend частини веб-сайту було обрано мову програмування PHP.

PHP – (англ. PHP: Hypertext Preprocessor - «PHP: препроцесор гіпертексту») - скриптова мова програмування, застосовується для розробки додатків і сайтів. В даний час вона підтримується багатьма хостинг-компаніями і є одним з лідерів серед мов програмування, які застосовуються для написання динамічних веб-сторінок.

За допомогою скриптів PHP розроблено:

- Форму для замовлення, що вносить дані в базу даних, в таблицю «Orders»;
- Форму для реєстрації користувачі, що вносить дані в базу даних, в таблицю «Users»;
- Форму для входу на сайт, що перевіряє дані в базі даних, в таблиці «Users»;
- Форму для додавання товарів, що вносить дані в базу даних, в таблицю «Products»;
- Форму для введення пошти, аби підписатись на оновлення товарів, що вносить дані в базу даних, в таблицю «Postmail»;
- На сторінці «Для дівчат» виведено перелік жіночого одягу, тобто виведено інформацію з бази даних, а саме з таблиці «Products»;
- На сторінці «Для хлопців» виведено перелік чоловічого одягу, тобто виведено інформацію з бази даних, а саме з таблиці «Products»;
- На сторінці «Товари» виведено перелік всього одягу, тобто виведено інформацію з бази даних, а саме з таблиці «Products»;

### 3.4. Програмування клієнтської частини

Frontend – це публічна частина веб-додатків (веб-сайтів), з якою користувач може взаємодіяти і контактувати напрямку. У Frontend входить відображення функціональних завдань призначеного для користувача інтерфейсу, що виконуються на стороні клієнта, а також обробка запитів користувачів. По суті, фронтенд – це все те, що бачить користувач при відкритті веб-сторінки.

Компоненти фронтенд розробки:

- HTML (HyperText Markup Language)– це мова розмітки всіх елементів і документів на сторінці, і їх взаємодія в структурі сторінки;
- CSS (Cascading Style Sheets) – це мова характеристики і стилізації зовнішнього вигляду документа. За допомогою CSS-коду браузер розуміє, як саме необхідно відображати елементи. CSS створює шрифти, кольори, визначає розташування блоків сайту, та інше. Також адаптує один і той же документ в різних стилях, виводить передачу на екран або для читання голосом;
- JavaScript– мова, створена оживляти веб-сторінки. Завдання JavaScript – відгукуватися на дії користувача, обробляти натискання клавіш, переміщення курсора, кліки мишкою. JavaScript також дає можливість вводити повідомлення, посилати запити на сервер, а також завантажує дані без перезавантаження сторінки, і так далі.

Для розробки веб-сайту було використано HTML, CSS, JavaScript.

Було розроблено основні сторінки та різні функції:

- Меню, що є для всіх сторінок однаковим. Тут можна перейти на інші сторінки;

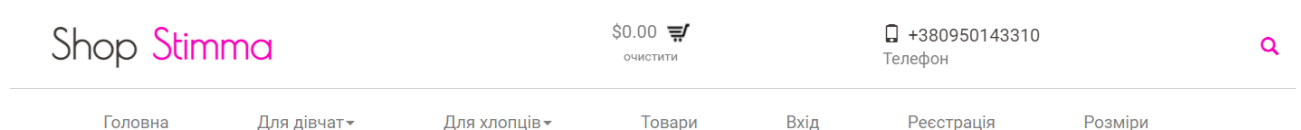


Рис. 3.4. Меню



- Підвал сайту, що є для всіх сторінок однаковим. Тут можна перейти на сторінки з жіночим одягом, чоловічим та з всім переліком одягу, натиснувши або на одну з категорій, або на назви «Для всіх», «Для нього», «Для неї»; натиснувши на один з значків соцмережей перейти на соцмережі; натиснувши на гугл карту побачити адресу магазину одягу; Натиснувши на «Anastasiya Gots» перейти до форми, за допомогою якої можна внести товар в базу даних та на сайт.

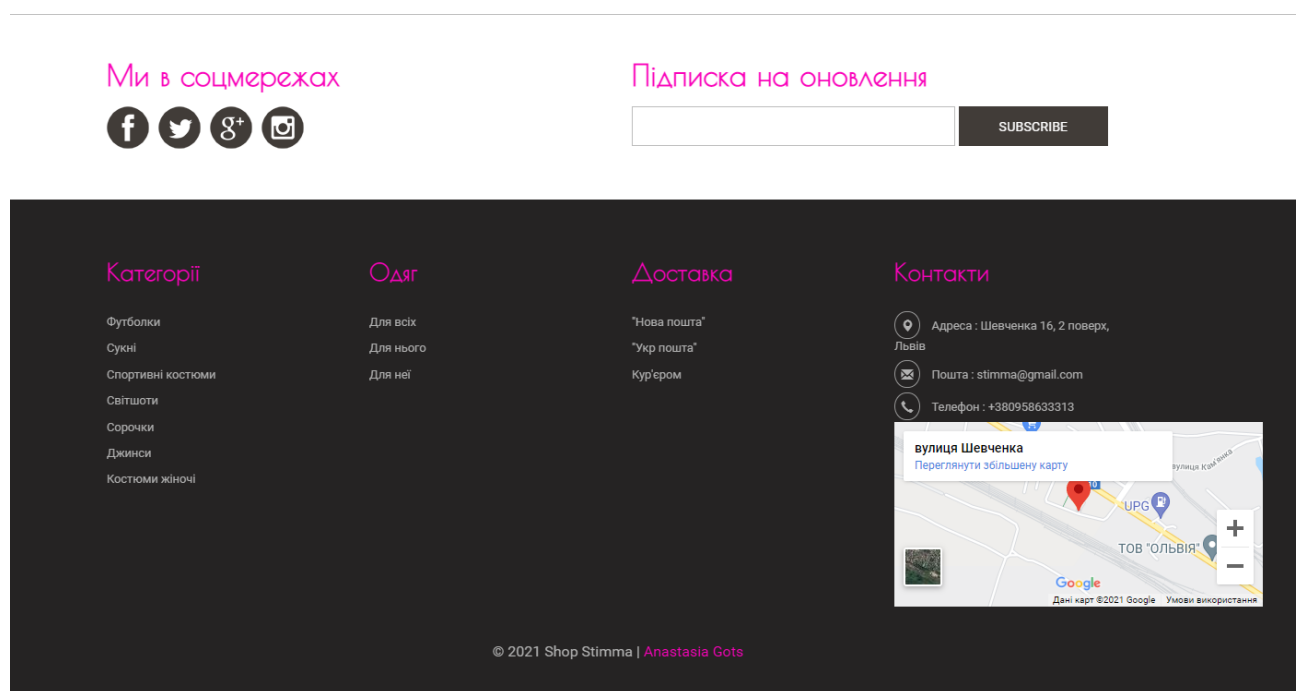


Рис. 3.5. Підвал сайту

- **Головна сторінка**  
Тут можна; перейти на сторінку з жіночим або чоловічим одягом, натиснувши на зображення; натиснувши на «Додати в кошик» можна додати суму до замовлення, що буде відображатись у верхній частині сторінці, біля значка кошик; натиснувши на картинку з сантиметром перейти на сторінку з розмірами одягу;
- **«Для дівчат»**  
Тут можна побачити перелік жіночого одягу, що є в базі даних; натиснувши на «Додати в кошик» можна додати суму до замовлення, що буде відображатись у верхній частині сторінці, біля значка кошик;
- **«Для хлопців»**

Тут можна побачити перелік чоловічого одягу, що є в базі даних; натиснувши на «Додати в кошик» можна додати суму до замовлення, що буде відображатись у верхній частині сторінці, біля значка кошик;

- «Товари»

Тут можна побачити перелік всього одягу, що є в базі даних; натиснувши на «Додати в кошик» можна додати суму до замовлення, що буде відображатись у верхній частині сторінці, біля значка кошик;

- «Вхід»

Тут розміщена форма для входу на сайт. Коли користувач введе свої дані, то буде надіслано запит до бази даних, а саме в таблицю «Users» і так буде перевірено чи користувач вже зареєстрований;

- «Реєстрація»

Тут розміщена форма для реєстрації на сайті. Коли користувач введе свої дані, то буде надіслано запит до бази даних і буде внесено дані в таблицю «Users»;

- «Розміри»

Тут можна побачити перелік розмірів одягу;

- Кошик

Тут розміщена форма для замовлення на сайті. Коли користувач введе дані замовлення, то буде надіслано запит до бази даних і буде внесено дані в таблицю «Orders».

Вигляд веб-сайту можна побачити в Додатку А.

### **3.5. Розміщення веб-сайту на локальному віртуальному середовищі або в Інтернеті**

Для розміщення сайту магазину одягу використано Open Server.

Open Server – це портативний локальний WAMP/WNMP сервер, що має багатофункціональну управлінську програму і великий вибір компонентів, що підключаються.

### Основні компоненти :

- Apache 2.2.31 / 2.4.38 / 2.4.41 / 2.4.43
- Bind 9.16.1;
- ConEmu 19.10.12;
- FTP FileZilla 0.9.60;
- Ghostscript 9.52;
- HeidiSQL 11.0.0.5944;
- Nginx 1.17.10;
- NNCron Lite 1.17;
- Opera 67.0.3575.137;
- Sendmail 32; ☐ Sublime 3.2.2.3211;
- Wget 1.20.3.

### Системи керування базами даних:

- MariaDB 5.5.67 / 10.1.44 / 10.2.31 / 10.3.22 / 10.3.22 / 10.4.12;
- Memcached 1.2.6 / 1.4.5 / 1.5.10;
- MongoDB 2.4.14 / 2.6.12 / 3.0.15 / 3.2.22 / 3.4.24 / 3.6.17 / 4.0.17 / 4.2.5;
- MySQL 5.1.73 / 5.5.62 / 5.6.47 / 5.7.29 / 8.0.19;
- PostgreSQL 9.2.24 / 9.3.25 / 9.4.26 / 9.5.21 / 9.6.17 / 10.12 / 11.7 / 12.2;

Redis 2.8.2402 / 3.0.504 / 3.2.100 / 4.0.14.2 / 5.0.6-dev.

PHP моделі: 5.2.17-7.4.05.

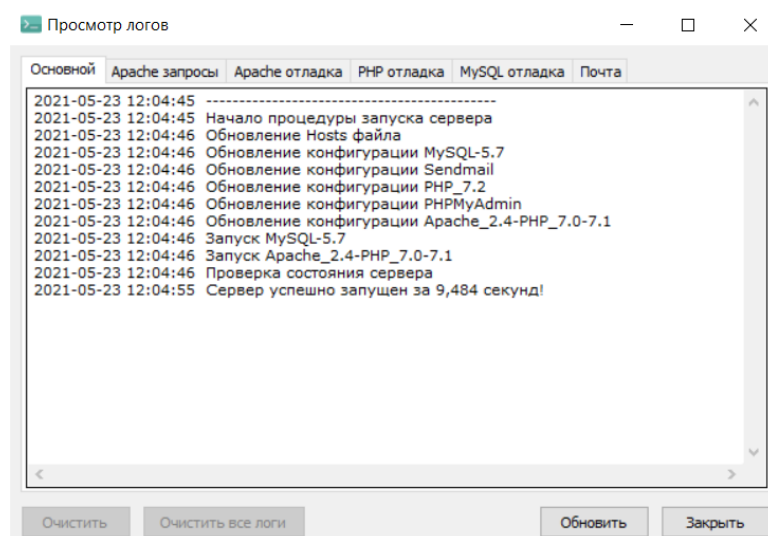


Рис. 3.6. Приклад роботи Open Server

## ВИСНОВКИ

У процесі виконання дослідження мета була досягнута, досягнуто всіх цілей проекту та виконано завдання, а саме створення інформаційної системи для магазину одягу.

Виконано основні завдання курсової роботи, а саме проведено аналіз предметної області інформаційної системи, описано та розроблено реляційну базу даних «shop\_stimma» в MyPHPAdmin на основі MySQL, описано та розроблено веб-сайт за допомогою мов програмування HTML, CSS, JavaScript, PHP, SQL в програмному середовищі Visual Studio Code, відображено інформаційну систему за допомогою веб-додатку і розміщено сайт на локальному сервері.

В базі даних «shop\_stimma» створено такі таблиці:

- Orders – містить інформацію про замовлення, що зробить користувач на сайту на сторінці «Кошик» за допомогою спеціальної форми;
- Products – містить інформацію про товари, що є в базі даних і розміщені на сайті, також сюди буде додаватись товар, коли адміністратор сайту додасть його за допомогою форми, що відкривається при натисканні на прізвище та ім'я в підвалі сайту;
- Products\_categories – містить інформацію про категорії товарів, а саме «Для дівчат» та «Для хлопців»;
- Users – містить інформацію про користувачів, що зареєстровані на сайті. Зареєструватись можна за допомогою форми, що відкривається при натисканні на «Реєстрація» в меню сайту. Також до цієї таблиці сайт звертається, коли користувач хоче зайти на сайт, що реалізується за допомогою форми, що знаходиться на сторінці «Вхід»;
- Postmail – містить інформацію про користувачів, що підписались на оновлення товару.

Веб-сайт складається з таких сторінок:

- Головна;

- Для дівчат;
- Для хлопців;
- Товари;
- Вхід;
- Реєстрація;
- Розміри;
- Кошик.

Для вдосконалення веб-додатку в перспективі можна:

- створити функціонал для адміністратора сайту;
- створити кошик для замовлень;
- створити список вподобань;
- додати можливість вибору розміру;
- додати можливість вибору та здійснення оплати на сайті;
- додати можливість вибору доставки замовлення;
- додати пошук на сайті;
- розділити товар на категорії;
- додати хіти продажу;
- додати можливість писати відгуки для користувачів.

Так як інтернет-торгівля розвивається з кожним роком, збільшується кількість інтернет-магазинів. Люди все частіше роблять покупки на просторах інтернету. Власники магазинів переводять свій бізнес в онлайн формат. Тому розроблений проект є актуальним та може бути корисним для людей, що хочуть створити власний інтернет магазин.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пасічник В. В. Організація баз даних та знань / В.В. Пасічник, В.А. Резніченко. – К.: Видавнича група BHV, 2017. – 384 с.
2. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования, 3-е издание = PHP Objects, Patterns and Practice, Third Edition. — М.: «Вильямс», 2015. — С. 560
3. Кузнецов Максим, Симдянов Игорь. PHP. Практика создания Webсайтов. — 2-е изд. перераб. и доп. — СПб.: «БХВ-Петербург», 2018. — С. 1264.
4. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – СПб.: Питер, 2014. – 304 с.
5. Дакетт Джон. HTML и CSS. Разработка и создание веб-сайтов. – 550 с.
6. Андрій Кириченко, Олена Дубовик, Олександр Нікольський. Web на практиці. CSS, HTML, JavaScript, MySQL, PHP для fullstack-розробників. - Наука и техника.- 360 с.
7. Фрімен Ерік, Вивчаємо HTML, XHTML и CSS. - Питер Пресс, 2019. – 600 с.
8. Нейт Купер, Как создать сайт. Комикс-путеводитель по HTML, CSS и WordPress. - Манн, Иванов і Фербер. – 560 с.
9. Современный учебник JavaScript [Електронний ресурс]. – Режим доступу: <http://learn.javascript.ru/>
10. Веб-дизайн, структура сайту [Електронний ресурс]. – Режим доступу: <https://webstudio2u.net/ua/design-web/403-structure-models.html>
11. Bootstrap [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap>
12. Переваги MySQL [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5607354/page:3/>
13. Переваги PHP [Електронний ресурс]. – Режим доступу: <https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movy-programuvannia-php>

14. Основне про PHP [Електронний ресурс] – Режим доступу:  
<https://metanit.com/php/tutorial/7.4.php>
15. CSS styles [Електронний ресурс] – Режим доступу: <http://htmlbook.ru/css>
16. Моделі даних [Електронний ресурс]. – Режим доступу:  
[https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%B%D1%96\\_%D0%B1%D0%B0%D0%B7\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%B%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)
17. JQuery [Електронний ресурс]. – Режим доступу:  
<https://uk.wikipedia.org/wiki/JQuery>
18. Типи даних MySQL [Електронний ресурс]. – Режим доступу:  
[https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85\\_MySQL](https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_MySQL)
19. Нормалізація баз даних [Електронний ресурс]. – Режим доступу:  
[http://wiki.fizmat.tnpu.edu.ua/index.php/Нормалізація\\_баз\\_даних](http://wiki.fizmat.tnpu.edu.ua/index.php/Нормалізація_баз_даних)
20. Open Server [Електронний ресурс]. – Режим доступу: <https://ospanel.io>

## ДОДАТКИ

Додаток А


Shop **Stimma** \$0.00  +380950143310  
Телефон 

Головна Для дівчат Для хлопців Товари Вхід Реєстрація Розміри

**MEN & WOMEN**  
for all people


**Special**  
for you

**ТРЦ "Форум"**




Сукня блакитна  
1000 грн

[Додати в кошик](#)




Сукня з кіткою  
1200 грн

[Додати в кошик](#)




Сорочка чорна  
750 грн

[Додати в кошик](#)




Світшот синій  
530 грн

[Додати в кошик](#)



Футболка з полою  
350 грн

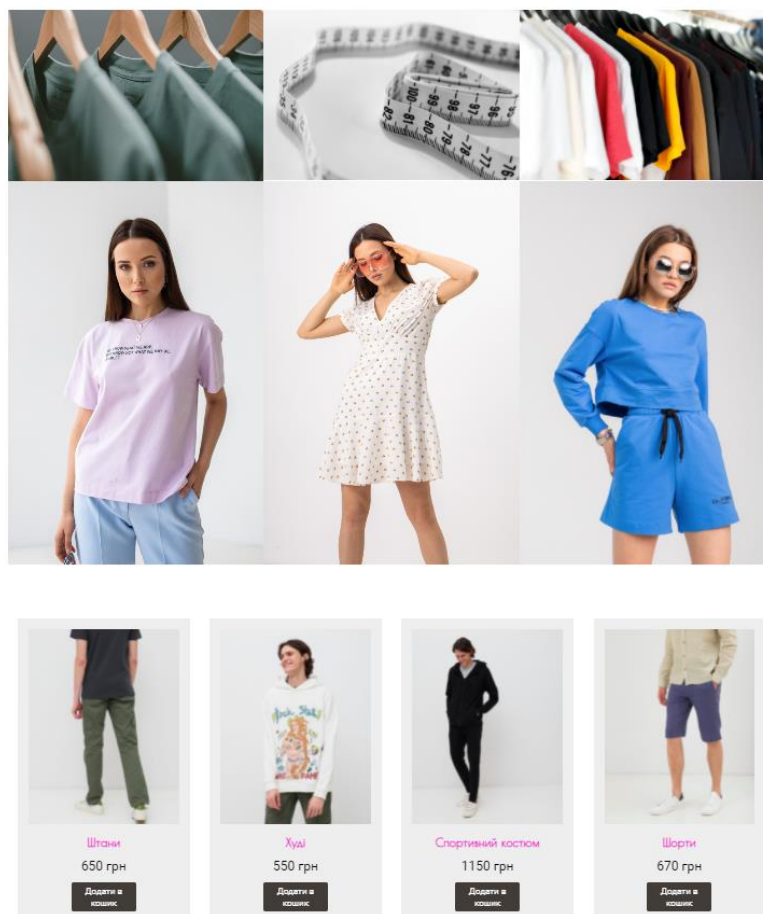
[Додати в кошик](#)



Костюм фіолетовий  
1700 грн

[Додати в кошик](#)





### Ми в соцмережах



### Підписка на оновлення

SUBSCRIBE

### Категорії

Футболки  
Сукні  
Спортивні костюми  
Світшоти  
Сорочки  
Джинси  
Костюми жіночі

### Одяг

Для всіх  
Для нього  
Для неї

### Доставка

"Нова пошта"  
"Укр пошта"  
Кур'єром

### Контакти

Адреса : Шевченка 16, 2 поверх,  
Львів  
Пошта : stimma@gmail.com  
Телефон : +380958633313

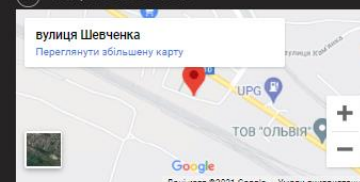


Рис. А.1. Головна сторінка сайту

## Для неї



Сукня блакитна

1000

[Додати в кошик](#)

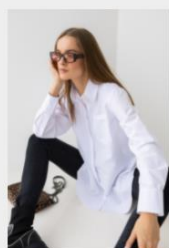
Сукня в квіточку

1200

[Додати в кошик](#)

Сукня в горошок

1000

[Додати в кошик](#)

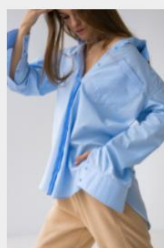
Сорочка біла

700

[Додати в кошик](#)

Сорочка чорна

750

[Додати в кошик](#)

Сорочка блакитна

780

[Додати в кошик](#)

Футболка біла

300

[Додати в кошик](#)

Футболка рожева













350

[Додати в кошик](#)

Футболка в полоську

350

[Додати в кошик](#)

 <p>Футболка фіолетова</p> <p>300</p> <p>Додати в кошик</p>	 <p>Світшот зелений</p> <p>530</p> <p>Додати в кошик</p>	 <p>Світшот жовтий</p> <p>550</p> <p>Додати в кошик</p>
 <p>Світшот синій</p> <p>530</p> <p>Додати в кошик</p>	 <p>Костюм молочний</p> <p>1500</p> <p>Додати в кошик</p>	 <p>Костюм фіолетовий</p> <p>1700</p> <p>Додати в кошик</p>
 <p>Костюм темно-синій</p> <p>1700</p> <p>Додати в кошик</p>	 <p>Костюм світло-зелений</p> <p>1550</p> <p>Додати в кошик</p>	 <p>Джинси темні</p> <p>999</p> <p>Додати в кошик</p>
 <p>Джинси світлі</p> <p>1050</p> <p>Додати в кошик</p>	 <p>Джинси сірі</p> <p>899</p> <p>Додати в кошик</p>	 <p>Джинси чорні</p> <p>600</p> <p>Додати в кошик</p>

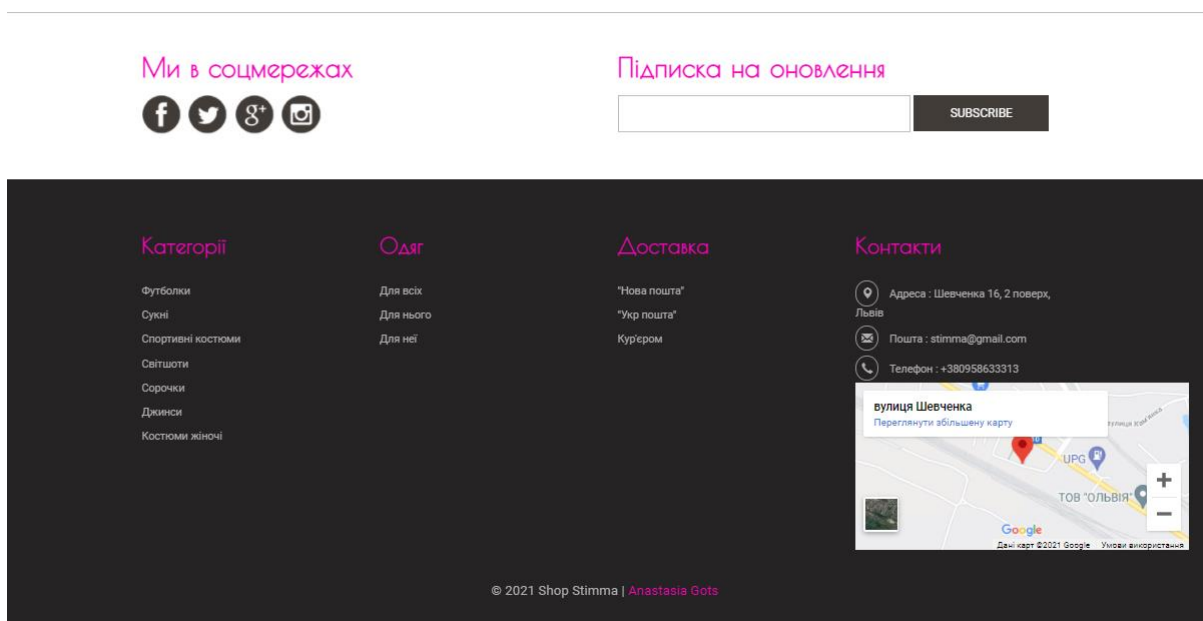
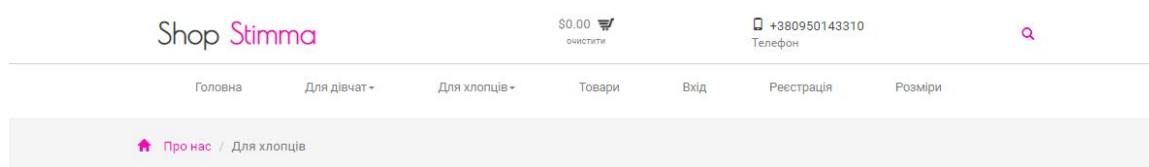
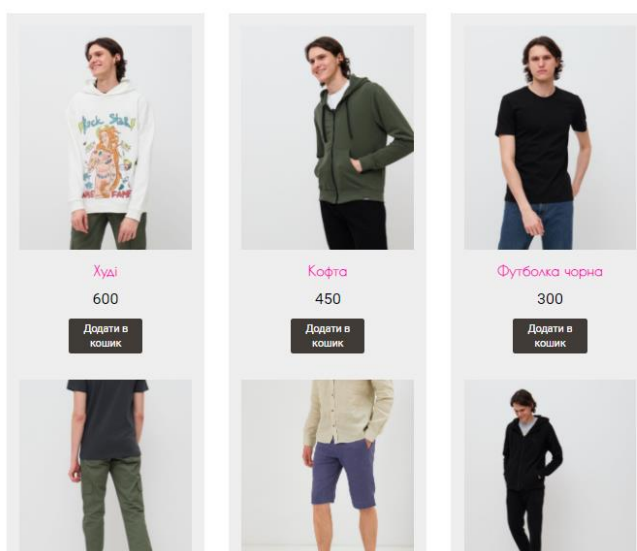
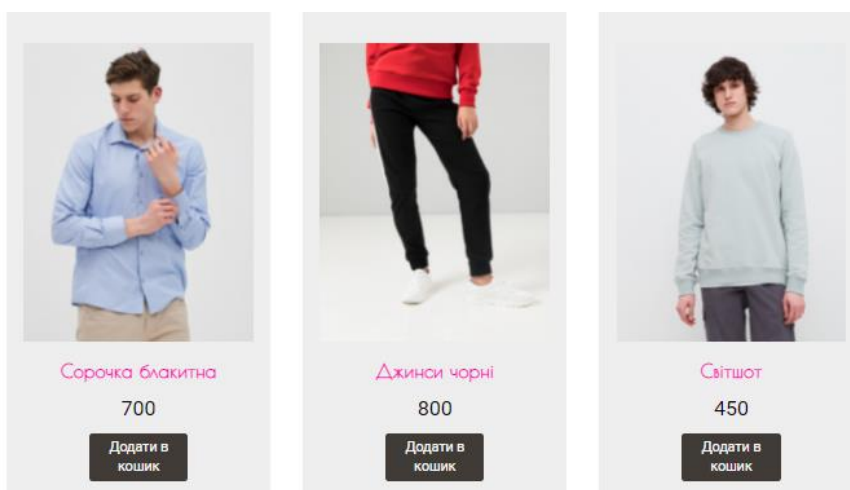


Рис. А. 2. Сторінка «Для дівчат»



## Для нього





### Ми в соцмережах



### Підписка на оновлення

SUBSCRIBE

### Категорії

Футболки  
Оукі  
Спортивні костюми  
Світшоти  
Сорочки  
Джинси  
Костюми жіночі

### Одяг

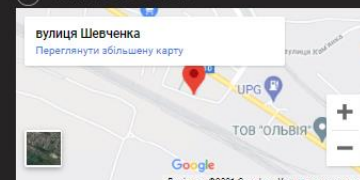
Для всіх  
Для нього  
Для неї

### Доставка

"Нова пошта"  
"Укр пошта"  
Кур'єром

### Контакти

Адреса : Шевченка 16, 2 поверх,  
Львів  
Пошта : stimma@gmail.com  
Телефон : +380958633313



© 2021 Shop Stimma | Anastasia Gots

Рис. А. 3. Сторінка «Для хлопців»

## Товари



Сукня блакитна

1000

Додати в  
кошик

Сукня в квіточку

1200

Додати в  
кошик

Сукня в горошок

1000

Додати в  
кошик

Джинси світлі

1050

Додати в  
кошик

Джинси сірі

899

Додати в  
кошик

Джинси чорні

600

Додати в  
кошик

Худі

600

Додати в  
кошик

Кофта

450

Додати в  
кошик

Футболка чорна

300

Додати в  
кошик

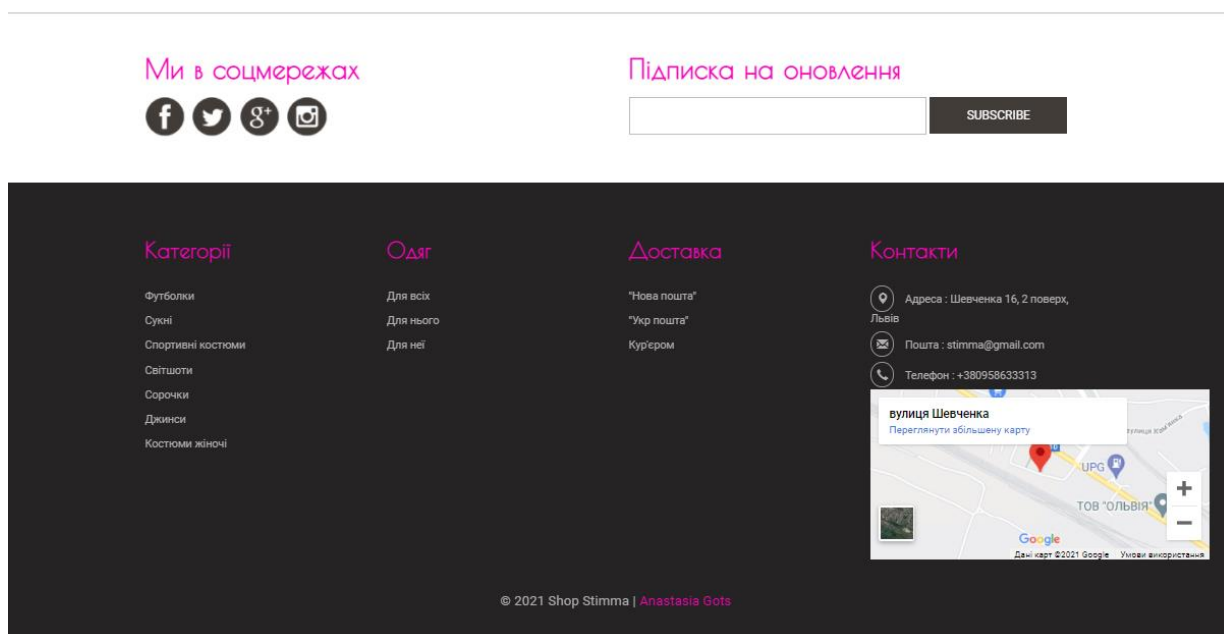


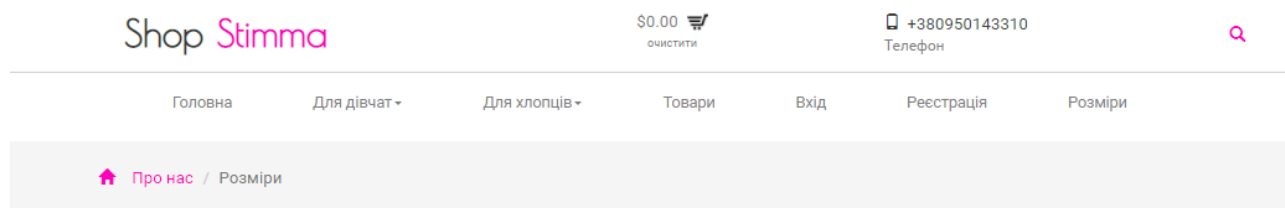
Рис. А. 3. Сторінка «Товари»

The screenshot shows the 'Login' form, which has a pink header with the title 'Login'. It contains two input fields: 'Username' and 'Password'. Below the password field is a pink 'Login' button. At the bottom, there's a link: 'Not yet a member? [Sign up](#)'.

Рис. А. 4. Сторінка «Вхід»

The screenshot shows the 'Register' form, which has a pink header with the title 'Register'. It contains four input fields: 'Username', 'Email', 'Password', and 'Confirm password'. Below the 'Confirm password' field is a pink 'Register' button. At the bottom, there's a link: 'Already a member? [Sign in](#)'.

Рис. А. 5. Сторінка «Реєстрація»



Розміри				
XS	S	M	L	XL
38	40	42	44	46

Рис. А. 6. Сторінка «Розміри»