

**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 салону краси**

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

Перхун К.Я.

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

(підпис)

Загальна кількість балів \_\_\_\_\_

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

**ЛЬВІВ 2021**

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ .....	5
1.1 Постановка завдання .....	5
1.2 Розробка моделі варіантів використання веб-сайту; .....	6
1.2 Логічна структура сайту .....	8
1.3 Аналіз засобів реалізації (техніко-економічне обґрунтування вибору) .....	9
1.4 Інформаційний ресурс салону краси .....	12
РОЗДІЛ 2. СТРУКТУРА БАЗИ ДАНИХ .....	13
2.1 Опис моделі даних .....	13
2.2. Нормалізація відношень .....	16
2.3 Визначення типів даних .....	18
2.4 Обмеження цілісності даних.....	21
2.5 Реалізація SQL-скрипту .....	22
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ .....	25
3.1 Структура веб-сайту.....	25
3.2 Макет сторінок веб сайту.....	28
3.3 Програмування серверної частини .....	30
3.4 Програмування клієнтської частини .....	31
3.5 Розміщення веб-сайту на локальному віртуальному середовищі або в Інтернеті.....	32
ВИСНОВОК .....	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	36

## ВСТУП

**Актуальність теми дослідження.** Інформаційні системи здавна знаходять досить широке застосування в життєдіяльності людства. В останнє десятиліття завдяки бурхливому розвитку мережі Інтернет в програмуванні виділяють окремий напрямок – Web-програмування, тобто написання сайтів для Web.

У наші дні роль Web-програмування в структурі глобальної мережі відіграє набагато більшу роль, відповідно, збільшується і середня оцінка складності реалізації проблеми, яка виникла. У зв'язку з тим багато систем (наприклад, соціальні мережі) за обсягом коду набагато збільшуються порівняно з розміром вихідних кодів прикладних програм. Частка ж статичних сторінок в Web значно менша; на зміну їм приходять динамічні сторінки, де використовуються js-скрипти для збільшення інтерактивності. Одночасно з розвитком та популяризацією комп'ютерної техніки доступ до мережі Інтернет як глобальної інформаційної бази став можливим практично для всіх.

З розвитком інформаційних технологій з'являються велика кількість нових інформаційних систем, які дозволяють покращити якість обслуговування, підвищити рівень продуктивності та збільшити прибуток організацій. В наш час попит на салони краси, попит на їх послуги зростає. Деякі клієнти планують все завчасно та записуються до майстра за декілька тижнів, але є і такі, яким необхідно негайно записатися на процедуру. Іноді клієнти можуть змінювати запис або відмінити його взагалі.

Аби не втрачати тих, хто записався навімання або змінює запис коли завгодно необхідно мати розклад, що може змінюватись залежно від умов. Підвищити ефективність роботи салону можна шляхом використання автоматизованих інформаційних систем за рахунок яких є можливість підвищення якості послуг, що надаються та зменшення трудових витрат майстрів салону. Отже, при автоматизації процесу запису на будь-яку процедуру, запис клієнтів полегшиться.

**Мета і завдання дослідження.** Метою курсової роботи є дослідження та аналіз актуальності інформаційної системи для салону краси в Україні. Навчитися створювати веб-сайт і дослідити всі нюанси, які виникають під час його створенні.

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

- визначити особливості створення веб сайту для салону краси в Україні;
- розробити інформаційну систему(веб-сайт), що дає змогу переглянути послуги салону краси та онлайн записатись до салону;
- перевірити програмний продукт на наявність дефектів та знайти шляхи покращення.

*Об'єктом дослідження* виступає конкретний салон краси, для якого створюється інформаційна система та процес розробки відповідного програмного забезпечення.

*Предметом* є теоретичні, методичні та практичні аспекти розробки програмного забезпечення мовою розмітки: html5, css3; мовою програмування: Php, Js, мовою запитів MySQL для створення офіційного веб-сайту салону.

**Практичне значення отриманих результатів.** Програма може використовуватись юридичною особою, яка хоче розмістити інформацію про своє підприємство в мережі інтернет з метою залучення клієнтів.

**Використане програмне забезпечення.** Для створення програмного продукту було використано такі мови програмування:HTML, CSS, JS, PHP, SQL та середовище розробки VisualStudioCode, локальний сервер – Open Server, MySQL.

**Структура роботи.** Курсова робота складається з трьох розділів («Аналіз вимог», «Розробка бази даних» та «Розробка веб-додатку»), висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 37 сторінок.

## РОЗДІЛ 1. АНАЛІЗ ВИМОГ

### 1.1 Постановка завдання

Розвиток інформаційних технологій у сфері надання послуг поставив на новий рівень конкурентоспроможність підприємств, зокрема і салонів краси. Розроблення функціонального та структурованого інформаційного ресурсу формує позитивний образ салону у WWW, що сприяє залученню нових клієнтів та інформуванню постійних відвідувачів. Для успішної роботи і утримання високих позицій в індустрії краси необхідно не тільки якісно надавати послуги, але й проводити рекламну кампанію для розширення клієнтської бази.

Розроблення інформаційного ресурсу салону краси є комплексною задачею, яка полягає у вивченні потреб та побажань власника салону краси, щодо функціональних можливостей та дизайну веб-ресурсу. Адже, мережевий ресурс ефективно використовується як інструмент популяризації салону краси у глобальній мережі. Звертаючи увагу на вже сформовані вимоги до інформаційного ресурсу, варто адаптувати їх саме до потреб салону краси.

Створення такого інформаційного ресурсу для салону краси надасть великий спектр можливостей для популяризації салону та покращення рівня надання послуг, зокрема:

- інформування клієнтів про новинки, цікаві акції, знижки та спеціальні пропозиції, що є способом утримання постійних клієнтів та ефективним засобом для залучення нових клієнтів (наприклад, "приведи з собою подругу і отримай 50% на будь-яку послугу");
- запис на послугу до майстра у зручний для клієнта час (24/7);
- розміщення необмеженої кількості фото/відео з роботами майстра та пропозицій для клієнтів;
- автоматичний розрахунок клієнтом вартості послуги на основі заданих ним параметрів;

- актуалізація та структурування даних клієнтів, наприклад, дата народження з метою привітання, електронної адреси для інформування про зміни в салоні, а також акції та знижки;
- замовлення подарункових сертифікатів;
- оплата послуг он-лайн;
- розроблення декількох комплексів обслуговування.

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

Суть цієї діаграми полягає в наступному: 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 салону краси представлено на рис. 1.2.

*Інформація про акторів:*

- Користувач – користувач салону краси
- Адміністратор – користувач, що виконує функції «адміністратора»
- База даних – база даних, що зберігає інформацію

*Опис варіантів використання:*

- Авторизація – функція входу в систему
- Пошук послуг – користувач має можливість переглянути послуги, які надаються у салоні на сайті
- Редагування даних – адміністратор може редагувати дані на сайті
- Додавання послуг – адміністратор може додати нові послуги на сайт

- Перегляд послуг – адміністратор може переглядати послуги, які відвідувачі найбільше надають переваги

*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.

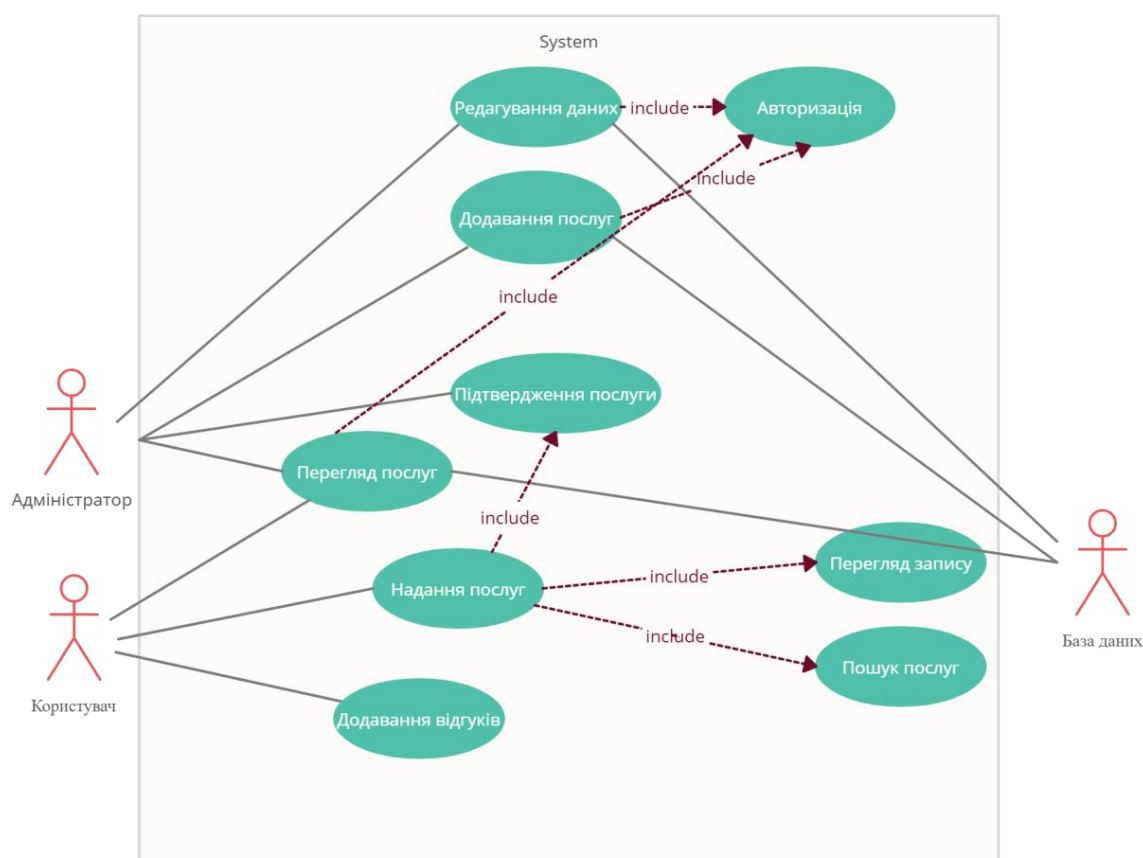


Рис. 1.2 Діаграма варіантів використання

## 1.2 Логічна структура сайту

Сайти з невдалою логікою структури не лише ускладнюють роботу відвідувачів, вони позбавляють всякого бажання на них знаходитися. Коли відвідувачі не в змозі знайти потрібну інформацію, вони схильні припускати, що на цьому сайті її взагалі немає. Розчаровані відвідувачі навряд чи запам'ятають цей ресурс, а тим більше захочуть ще раз його відвідати.

Вдалий дизайн в першу чергу має орієнтуватися на користувача, який без особливих зусиль може знайти і використати будь яку інформацію, що міститься у сайті. При цьому повинні бути абсолютно зрозумілими логічна схема сайту і логічний взаємозв'язок між окремими сторінками.

Хороша навігація сайту означає, що користувачі точно знають, де знаходяться, де розташовані елементи сайту і як використовувати ці елементи. Правильна структура інформації дозволяє користувачам без побоювання продовжувати мандрування сайтом, і бути впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

На вдало організованих сайтах відвідувачі можуть вільно переміщатися у пошуках потрібної інформації, не турбуючись про структуру сайту. Піклування про структуру сайту є завданням розробника, а не відвідувачів.

- Принципи логічного проектування

Логічне проектування передбачає організацію інформації на сайті, побудову його структури та навігацію по розділах.

При проектуванні інформаційного наповнення сайту і засобів навігації слід дотримуватися чотирьох базових принципів, які засновані на сприйнятті інформації людиною.

1. Використання позначень.
2. Доцільність.
3. Одноманітність.
4. Розділення на частини.



Застосовуйте слова і терміни, які є сталими і зрозумілими для більшості відвідувачів сайту. В цьому сенсі не дуже вдалим є застосування піктограм. Іноді, відвідувачі можуть невірно розтлумачити такі зображення. З цієї причини слід розміщувати змістовні елементи, посилання і елементи навігації з їх позначеннями, а також уникати невиразних термінів на сторінці.

- Доцільність

Розділи сайту повинні містити інформацію та елементи, які відповідають даному розділу чи фрагменту сторінки, невідповідні елементи краще перенести в інше місце або взагалі прибрати.

- Одноманітність

Використання єдиної навігації по сайту з однаковим оформленням. Однакові елементи на сторінках повинні мати один і той же розмір і знаходитися в одному і тому ж місці. Якщо сторінки, з певних причин мають відмінності, відвідувач повинен чітко розуміти, чим вони обумовлені.

- Розділення на частини

Людина не в змозі зберігати одночасно великий об'єм інформації. Як правило, люди здатні сприймати інформацію, що містить від чотирьох до шести різних елементів. Тому, відвідувачі сайту краще орієнтуються і швидше знаходять потрібні для них матеріали, коли вони візуально поділені на групи.

### **1.3. Аналіз засобів реалізації (техніко-економічне обґрунтування вибору)**

Робота з базами даних є одною з найважливіших складових програмування сайту динамічного типу. Бази даних для сайтів використаємо з метою зберігання різної інформації. Спрощено бази даних представляють собою певний набір взаємозалежних таблиць. Розміри таблиць в базах даних

різні, їх кількість довільна. Саме в базах даних буде зберігатися на сервері необхідна для роботи сайту інформація, наприклад, інформація про клієнтів, каталог товарів, статистичні дані тощо.

*Для реалізації веб-сайту було обрано наступні технології:*

- HTML5, CSS3, JS(зокрема і бібліотека JQuery), Bootstrap.(Дані технології для реалізації Front-end частини);
- PHP, MySql(Для реалізації Back-end частини).

*Для Front-end частини було обрано:*

Java Script, тому що JavaScript (Джаваскрипт)- це повноцінна динамічна мова програмування, яка застосовується до HTML документу, і може забезпечити динамічну інтерактивність на веб-сайтах. Вона допомагає зробити сторінки сайту більш інтерактивними, обробляє дії користувачів сайту.

*Для Back-end частини було обрано:*

PHP, тому що ця технологія надає такі переваги:

- Головним чинником мови PHP є практичність. PHP повинен надати програмісту інструмент для швидкого і ефективного вирішення поставлених завдань.
- Дуже важлива перевага PHP полягає в його «движку». «Движок» PHP не є ні компілятором, ні інтерпретатором. Він є транслюючим інтерпретатором. Такий «движок» PHP дозволяє обробляти сценарії з достатньо високою швидкістю.
- PHP надає в розпорядження розробників та адміністраторів гнучкі та ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня та засоби рівня програми.
- У PHP реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному налаштуванні PHP це забезпечує максимальну свободу дій і безпеку. PHP може працювати в

так званому безпечному режимі (safe mode), який обмежує можливості застосування PHP користувачами по ряду важливих показників.

- У стандартний набір функцій PHP входить ряд надійних механізмів шифрування. PHP також сумісний з багатьма додатками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції (e-commerce). Інша перевага полягає в тому, що вихідний текст сценаріїв PHP не можна переглянути у браузері, оскільки сценарій компілюється до його відправлення за запитом користувача

Sql (головні переваги):

- Незалежність від конкретної СУБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу.
- Наявність стандартів. Наявність стандартів і набору тестів для виявлення сумісності і відповідності конкретній реалізації SQL загальноприйнятому стандарту, що тільки сприяє «стабілізації» мови.
- Декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо при обробці SQL-запиту. Проте не варто думати, що це повністю універсальний принцип – програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД розбиратиме текст його запиту.
- Безпека. MySQL має систему контролю доступу до даних, забезпечує шифрування даних при передаванні;

#### 1.4. Інформаційний ресурс салону краси

Інформаційний ресурс салону краси має відповідати всім вимогам створенням інформаційного забезпечення для підприємства, таким як: уніфікація, достовірність інформаційного наповнення, контроль, цілісність системи та ін.

Особливу увагу треба приділити достовірності персональних даних працівників та інформація салону краси (освіта, стаж роботи, нагороди, відзнаки, участь у майстер-класах, стажування). Обов'язковими елементами цього інформаційного ресурсу є:

- надання контактної інформації салону краси (телефонів, адреси, карти проїзду та е-мейл);
- поширення інформації через соціальні мережі;
- мобільна версія інформаційного ресурсу;
- веб-ресурс повинен бути багатомовним (англомовний варіант є обов'язковим);
- гостьова книга для висловлення пропозицій, побажань, скарг та зауважень клієнтів.

Створення власного мережевого ресурсу салону краси виступає одним з найбільш ефективних способів залучення та інформування клієнтської аудиторії.

Запропоновано нові функціональні можливості веб-ресурсу, що сприяють покращенню бізнес процесу салону краси. Грамотне та добре продумане розроблення інформаційного ресурсу надасть вагому перевагу салону краси зпоміж конкурентів, розширить клієнтську базу, створивши позитивний імідж у глобальній мережі.

Підвищить довіру потенційних клієнтів до салону краси, слугуючи візитівкою не тільки для відвідувачів, але також і для потенційних співробітників.

## РОЗДІЛ 2. СТРУКТУРА БАЗИ ДАНИХ

### 2.1 Опис моделі даних

База даних — це певний набір даних, які пов'язані між собою спільною ознакою або властивістю, та впорядковані, наприклад, за алфавітом. Об'єднання великої кількості даних в єдину базу дає змогу для формування безлічі варіації групування інформації — особисті дані клієнта, історія замовлень, каталог товарів та будь-що інше.

Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках — це допомагає працювати з інформацією простіше і швидше.

Основою бази даних є модель даних — фіксована система понять і правил для представлення даних структури, стану і динаміки проблемної області в базі даних. У ході практичного завдання була побудована наступна модель бази даних:

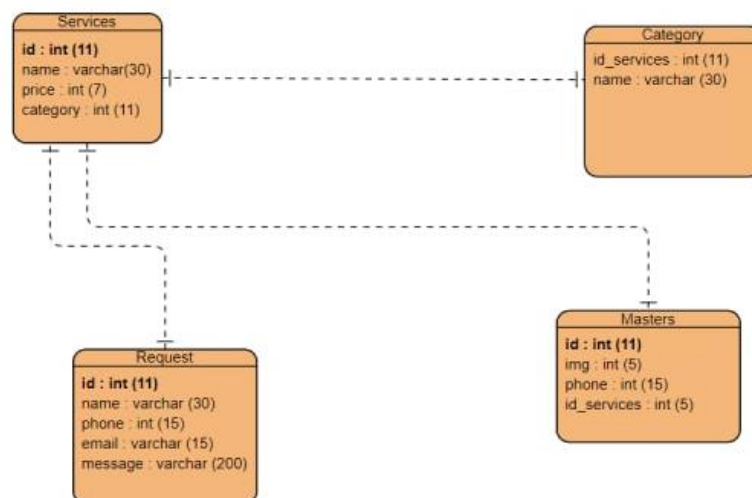


Рис. 2.1. Діаграма сутностей і зв'язків

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

*В таблиці Services, є такі поля:*

- id
- name
- price
- category

*В таблиці Masters, є такі поля:*

- id
- img
- phone
- id\_services

*В таблиці Request, є такі поля:*

- id
- name
- number
- email
- message

*В таблиці Category, є такі поля:*

- id\_services
- name

Існують такі види моделей даних:

– **Ієрархічна структура (модель)** будується у вигляді ієрархічної деревоподібної структури, у якій для кожного головного об'єкта існує кілька підлеглих, а для кожного підлеглого об'єкта може бути тільки один головний. На найвищому рівні ієрархії перебуває кореневий об'єкт. Прикладом ієрархічної структури даних може бути організація каталогів на диску, різного роду класифікації, структура державної влади тощо. Концептуальна схема ієрархічної моделі являє собою сукупність типів записів, пов'язаних типами зв'язків в одним чи кількома деревами. Усі типи зв'язків цієї моделі належать до виду «один до декількох» і зображуються у вигляді стрілок. Таким чином, взаємозв'язки між об'єктами нагадують

взаємозв'язки в генеалогічному дереві, за єдиним винятком: для кожного породженого (підлеглого) типу об'єкта може бути тільки один вхідний (головний) тип об'єкта. Тобто ієрархічна модель даних допускає тільки два типи зв'язків між об'єктами: «один до одного» і «один до декількох». Ієрархічні бази даних є навігаційними, тобто доступ можливий тільки за допомогою заздалегідь визначених зв'язків.

При моделюванні подій, як правило, необхідні зв'язки типу «багато до декількох». Як одне з можливих рішень зняття цього обмеження можна запропонувати дублювання об'єктів. Однак дублювання об'єктів створює можливості неузгодженості даних. Перевага ієрархічної бази даних полягає в тому, що її навігаційна природа забезпечує швидкий доступ при проходженні вздовж заздалегідь визначених зв'язків. Однак негнучкість моделі даних і, зокрема, неможливість наявності в об'єкта декількох батьків, а також відсутність прямого доступу до даних роблять її непридатною в умовах частого виконання запитів, не запланованих заздалегідь. Ще одним недоліком ієрархічної моделі даних є те, що інформаційний пошук з нижніх рівнів ієрархії не можна спрямувати по вище розміщених вузлах.

– **Мережева модель.** У мережній моделі один і той же об'єкт може одночасно виступати як у ролі головного, так і підлеглого елемента. Це означає, що кожний об'єкт може брати участь у довільній кількості зв'язків. Зв'язок у цьому випадку може встановлюватися явно, коли значення деяких полів є посилання на дані, що містяться в іншому файлі. Подібно до ієрархічної, мережну модель також можна подати у вигляді орієнтованого графа. Але в цьому випадку граф може містити цикли, тобто вершина може мати кілька батьківських вершин. Така структура набагато гнучкіша і виразніша від попередньої і придатна для моделювання ширшого класу завдань.

Ієрархічні і мережні бази даних часто називають базами даних з навігацією. Ця назва відбиває технологію доступу до даних, використовувану при написанні програм обробки мовою маніпулювання

даними. При цьому доступ до даних по шляхах, не передбачених при створенні бази даних, може потребувати нерозумно тривалого часу.

– **Реляційна модель.**

У реляційній моделі дані й взаємозв'язки між ними подаються за допомогою прямокутних таблиць. Рядки в реляційній базі даних називають записами, а стовпці — полями. Модель реляційної бази даних була вперше розроблена доктором Е. Ф. Коддом на початку 70-х років ХХ ст. як більш зручний засіб збереження, вибірки й маніпулювання даними, ніж ієрархічні й мережні бази даних. Модель двовимірної таблиці дозволяє звертатися до даних як по рядках, так і по стовпцях, що є значною перевагою. 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.

## **2.2. Нормалізація відношень**

Нормалізація відношень - покроковий процес розділення (декомпозиції) початкових відношень БД на простіші. Кроки цього процесу переводять схему відношення БД в послідовні нормальні форми. Кожна наступна форма володіє кращими властивостями ніж попередня. Кожній нормальній формі відповідає 12 певний набір обмежень.

При переведенні структури відношення у форми вищого порядку досягають видалення з таблиць надмірної описової інформації. Процес нормалізації заснований на понятті функціональної залежності атрибутів.

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



Виділяються шість нормальних форм:

*Перша нормальна форма.* Відношення відповідає 1NF тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні (неподільні) значення атрибутів і не містяться групи, що повторюються.

*Друга нормальна форма.* Відношення знаходиться в 2NF, якщо виконуються обмеження 1NF і кожен описовий атрибут функціонально повно залежить від первинного ключа (у тому числі і складеного).

*Третя нормальна форма.* Відношення знаходиться у 3NF, якщо виконуються обмеження 2NF і всі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут не транзитивно залежить від ключа.

*Четверта нормальна форма (4НФ, 4NF)* вимагає, аби в схемі баз даних не було нетривіальних багатозначних залежностей множин атрибутів від чого, окрім надмножини ключа-кандидата. Вважається, що таблиця знаходиться у 4НФ тоді, і тільки тоді, коли вона знаходиться в НФБК, та багатозначні залежності є функціональними залежностями. Четверта нормальна форма усуває небажані структури даних — багатозначні залежності.

*П'ята нормальна форма (5НФ, 5NF, PJ/NF)* вимагає, аби не було не тривіальних залежностей об'єднання, котрі б не витікали із обмежень ключів. Вважається, що таблиця в п'ятій нормальній формі, тоді, і тільки тоді, коли вона знаходиться в 4НФ, та кожна залежність об'єднання зумовлена її ключами-кандидатами.

*Шоста нормальна форма.* Таблиця знаходиться у 6nf, якщо вона знаходиться у 5nf та задовольняє вимозі відсутності нетривіальних залежностей. Зазвичай 6nf ототожнюють з dknf.

Цілі нормалізації наступні: Виключити дублювання інформації в таблицях. Забезпечити можливість змін у структурі таблиць. Зменшити вплив структурних змін бази даних на роботу додатків, які забезпечують користувачам доступ до даних.

### **Основні правила нормалізації:**

- В кожній таблиці повинен бути первинний ключ;
- Зміна значень в полях не повинна впливати на інформацію в інших полях;
- В кожній базі даних не повинно бути повторюваних полів;
- Кожному значенню первинного ключа повинна відповідати інформація про тип суті або про об'єкт таблиці.

### **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, FKвиділені курсивом, а PK— жирним шрифтом.

Які є види ключів і для чого вони потрібні?

Ключ — це стовпець (може бути декілька стовпців), що додається до таблиці і дозволяє встановити зв'язок із записами в іншій таблиці.

Існують ключі двох типів: первинні і вторинні (зовнішні).

Первинний ключ — це одне або кілька полів (стовпців), комбінація значень яких однозначно визначає кожний запис у таблиці. Первинний ключ не допускає значень Null і завжди повинен мати унікальний індекс. Первинний ключ використовується для зв'язування таблиці з зовнішніми ключами в інших таблицях.

Зовнішній (вторинний) ключ — це одне або кілька полів (стовпців) у таблиці, що містять посилання на поле або поля первинного ключа в іншій таблиці. Зовнішній ключ визначає спосіб об'єднання таблиць.

З двох логічно пов'язаних таблиць одну називають таблицею первинного ключа або головною таблицею, а іншу таблицею вторинного (зовнішнього) ключа або підпорядкованою таблицею. СУБД дозволяють зіставити споріднені записи з обох таблиць і спільно вивести їх у формі, звіті або запиті.

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 «Services»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
name	VARCHAR	30	-
price	INT	7	-
category	INT	11	-

Таблиця 2.3.2

**Тип даних реляційного відношення «Masters»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
img	INT	5	-
phone	INT	15	-
id_services	INT	5	-

Таблиця 2.3.3

**Тип даних реляційного відношення «Request»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id	INT	11	Первинний ключ
name	VARCHAR	30	-
phone	INT	15	-
email	VARCHAR	15	-
messange	VARCHAR	200	-

Таблиця 2.3.4

**Тип даних реляційного відношення «Category»**

Атрибут	Тип даних	Розмір	Додаткові обмеження цілісності
id_services	INT	11	Первинний ключ
name	VARCHAR	30	-

Які взагалі бувають типи даних в SQL? MySQL має три основні типи даних: текстові, числові, та часові.

Таблиця 2.3.5

## Числові типи даних в SQL

Тип	Опис
TINYINT(size)	Цілий від -128 до 127 . Від 0 до 255 UNSIGNED Максимальне число цифр задається в дужках.
SMALLINT(size)	Від -32768 до 32767. Від 0 до 65535 UNSIGNED Максимальне число цифр задається в дужках.
MEDIUMINT(size)	Від -8388608 до 8388607. Від 0 до 16777215 UNSIGNED Максимальне число цифр задається в дужках.
INT(size)	Від -2147483648 до 2147483647. Від 0 до 4294967295 UNSIGNED Максимальне число цифр задається в дужках.
BIGINT(size)	Від -9223372036854775808 до 9223372036854775807. Від 0 до 18446744073709551615 UNSIGNED Максимальне число цифр задається в дужках.
FLOAT(size,d)	Число з плаваючою крапкою. Максимальне число цифр задається в параметрі size. Максимальне число цифр після десяткової крапки задається в параметрі d.
DOUBLE(size,d)	Точніше число з плаваючою крапкою. Максимальне число цифр задається в параметрі size. Максимальне число цифр після десяткової крапки задається в параметрі d.
DECIMAL(size,d)	DOUBLE, що зберігається як рядок з фіксованою крапкою.. Максимальне число цифр задається в параметрі size. Максимальне число цифр після десяткової крапки задається в параметрі d.

Таблиця 2.3.6

## Типи дати і часу в SQL

Тип	Опис
DATE()	Дата. Формат: YYYY-MM-DD Зауваження: Підтримується діапазон від '1000-01-01' до '9999-12-31'
DATETIME()	Формат: YYYY-MM-DD HH:MM:SS Зауваження: Підтримується діапазон від '1000-01-01 00:00:00' до '9999-12-31 23:59:59'
TIMESTAMP()	Значення TIMESTAMP зберігаються як кількість секунд з початку епохи Unix ('1970-01-01 00:00:00' UTC). Формат: YYYY-MM-DD HH:MM:SS Зауваження: Підтримується діапазон від '1970-01-01 00:00:01' UTC до '2038-01-09 03:14:07' UTC
TIME()	Час. Формат: HH:MM:SS Зауваження: Підтримується діапазон від '-838:59:59' до '838:59:59'
YEAR()	Рік в двоцифровому, або чотирицифровому форматі. Зауваження: Значення, що дозволені в чотирицифровому форматі: від 1901 до 2155. Значення дозволені в двоцифровому форматі: від 70 до 69, що відповідає 1970 та 2069.

Таблиця 2.3.7

**Текстові типи даних в SQL**

Тип	Опис
CHAR(size)	Містить рядок фіксованої довжини (може містити букви, цифри, та інші символи). Фіксована довжина задається в дужках. Може зберігати до 255 символів.
VARCHAR(size)	Містить рядок змінної довжини. Найбільша довжина задається в дужках. Може зберігати до 255 символів. Примітка: Якщо ви покладете туди значення більше за 255, тип буде перетворений на TEXT.
TINYTEXT	Рядок з найбільшою довжиною 255 символів
TEXT	Зберігає рядок з найдовшою довжиною 65,535 символів
BLOB	Великий двійковий об'єкт (Binary Large Object). Зберігає до 65,535 байт даних
MEDIUMTEXT	Зберігає рядок з максимальною довжиною в 16,777,215 символів.
MEDIUMBLOB	Великий двійковий об'єкт. 16 Мегабайт даних
LONGTEXT	Рядок з найбільшою довжиною в 4,294,967,295 символів.
LOBLOB	Великий двійковий об'єкт. 4 Гігабайти даних
ENUM(x,y,z,i т.д.)	Дозволяє ввести список можливих значень. Можна перелічити до 65535 різних значень типу. Якщо значення що вставляють в поле не буде належати списку, вставиться порожнє значення. Зауваження: Значення будуть відсортовані в тому порядку в якому ви їх запишете. Можливі значення вводяться в такому форматі: ENUM('X','Y','Z')
SET	Подібно до ENUM окрім того, що SET може містити до 64 значень списку, і не може зберігати більше одного вибору.

**2.4. Обмеження цілісності даних**

Терміном цілісність даних позначають достовірність і точність інформації, що зберігається в базі. Цілісність досягається забезпеченням відповідності даних певним додатковим обмеженням, крім тих, які накладаються схемою бази на структуру 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.

Цілісність даних - важлива властивість SQL. При правильному використанні воно забезпечує коректність і валідність даних, що зберігаються в будь-який момент часу. Також, з їх допомогою можна виявляти помилки в додатках, які важко знайти іншими способами. Цілісність даних підтримується за допомогою обмежень.

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

PRIMARY KEY - набір полів (1 або більше), значення яких утворюють унікальну комбінацію і використовуються для однозначної ідентифікації запису в таблиці. Для таблиці може бути створене тільки одне таке обмеження. Дане обмеження використовується для забезпечення цілісності сутності, яка описана таблицею.

CHECK використовується для обмеження безлічі значень, які можуть бути поміщені в даний стовпець. Це обмеження використовується для забезпечення цілісності предметної області, яку описують таблиці в базі.

Обмеження UNIQUE забезпечує відсутність дублікатів в стовпці або наборі стовпців. Різниця між PRIMARY KEY і UNIQUE описана в primary і unique ключі.

Обмеження FOREIGN KEY захищає від дій, які можуть порушити зв'язки між таблицями. FOREIGN KEY в одній таблиці вказує на PRIMARY KEY в іншій. Тому це обмеження націлене на те, щоб не було записів FOREIGN KEY, яким не відповідають записи PRIMARY KEY. Таким чином, FOREIGN KEY підтримує кількість посилань цілісність даних.

## 2.5. Реалізація SQL-скрипту

SQL скрипт об'єкта бази даних - це SQL інструкція, за допомогою якої створюється цей об'єкт, збережена в текстовому файлі.

Іншими словами, це простий SQL запит, зазвичай збережений у текстовому файлі з розширенням .sql. У цьому SQL запиті містяться всі

необхідні інструкції створення об'єкта (або об'єктів), включаючи інструкції наповнення його даними.

Такий SQL скрипт можна відкрити будь-яким текстовим редактором, скопіювати текст SQL запиту і виконати, наприклад, в середовищі SQL Server Management Studio, таким чином, створивши об'єкт бази даних, що не розробляючи відповідні SQL інструкції самостійно.

Так як SQL скрипт - це звичайний текстовий файл, його можна створити вручну, наприклад, коли Ви розробляєте об'єкт БД, Ви просто зберігаєте SQL інструкцію в файл, і додаєте в нього в міру необхідності інші SQL інструкції.

Приклад створення таблиці:

```
CREATE TABLE `clients` (
  'id_client' int(11) NOT NULL,
  'name' varchar(255) NOT NULL,
  'phone' varchar(255) NOT NULL,
  'email' varchar (255) NOT NULL,

) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_FORMAT=DYN
AMIC;

-- Дані таблиці 'clients'
```

```
INSERT INTO 'clients' ('id','name','phone','email') VALLUES
(1,'Ірина',09645446,'ira@gmail.com');
(2,'Світлана',096963246,'svitl@gmail.com');
(3,'Вероніка',099353246,'veronik@gmail.com');
(4,'Катерина',0796853246,'katya@gmail.com');
(5,'Влада',095363246,'vlada@gmail.com');
(6,'Адріана',099353246,'adriana@gmail.com');
(7,'Анастасія',098623246,'anastas@gmail.com');
(8,'Богдана',098853246,'bohd@gmail.com');
(9,'Тетяна',096363246,'tetyana@gmail.com');
```

```
(10,'Софія',099753246,'sofia@gmail.com');
(11,'Марія',092053246,'maria@gmail.com');
(12,'Ольга',080853246,'olia@gmail.com');
(13,'Надія',096853246,'nadia@gmail.com');
(14,'Олена',099653246,'olena@gmail.com');
```

--Структура таблиці 'product'

```
CREATE TABLE 'product' (
    'id_product' int (10) UNSIGNED NOT NULL,
    'name' varchar (255) DEFAULT NULL,
    'price' int (2000) NOT NULL,
    'master' varchar (255) DEFAULT NULL
)ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_FORMAT=DYNAMIC;
```

-- Дані таблиці 'product'

```
INSERT INTO 'product' (`id_product`,`name`,`price`,`master`) VALUES
(1, 'Манікюр', 300, 'Юлія'),
(2, 'Педикюр', 350, 'Наталія'),
(3, 'Макіяж', 400, 'Ірина'),
(4, 'Зачіска', 300, 'Світлана'),
(5, 'Ламінування вій', 200, 'Анастасія'),
(6, 'Брови', 180, 'Катерина'),
(7, 'Масаж', 500, 'Борис'),
(8, 'Покраска волосся', 680, 'Адріана'),
(9, 'Косметологічні процедури', 800, 'Христина'),
(10, 'Чистка обличчя', 700, 'Віолетта'),
(11, 'Кератин для волосся', 900, 'Анна'),
(12, 'Розслабляючий масаж', 650, 'Інна');
```



## РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ

### 3.1 Структура веб-сайту

Веб-сторінка — це документ, визначений унікальною URL-адресою на веб-сайті. Вона є одним із основних об'єктів веб-сайту. Веб-сторінки будують ієрархію веб-сайту за допомогою батьківських та дочірніх зв'язків з іншими веб-сторінками.

#### *Завдання й цілі сучасного веб-сайту*

Веб-сайт виконує такі основні завдання:

- реклама продукції, послуг, ідей. Правильно зроблений веб-сайт із легкістю приведе клієнта до висновку про необхідність покупки товару, або послуг, або ідей, що пропагуються на ньому;
- продаж товарів, послуг, інформації, ідей. У сучасної людини немає багато часу для ходіння по магазинах. Тому можливість замовлення товарів і послуг, не відходячи від комп'ютера, значно розширює можливості і клієнта, і продавця;
- безкоштовне надання інформації або послуг. Насправді надання інформації або послуг — це засіб залучення відвідувачів до даного ресурсу для здобуття, наприклад, статистичної інформації або ж для показу реклами, якщо це рекламний майданчик;
- підтримка клієнтів.

Базові елементи сторінки веб-сайту

#### **Тема веб-сторінки**

Як правило велика смуга зверху з великим заголовком і / або логотипом. Саме тут зазвичай розміщується базова інформація про веб-сайті.

#### **Панель навігації сайту**

Посилання на розділи сайту: зазвичай представлені кнопками меню, посиланнями або вкладками. Як і заголовок, цей контент зазвичай залишається неізмінним при переході з однієї веб-сторінки на іншу - неузгоджена навігація

на вашому веб-сайті просто призведе до розгубленості і розчарування користувачів. Багато веб-дизайнери вважають панель навігації частиною заголовка, а не окремим компонентом, але це не є обов'язковою вимогою; насправді деякі також стверджують, що реалізація цих двох складових окремо краще для доступності, так як програми читання з екрану краще читають дві функції, якщо вони розділені.

### **Основний зміст веб-сторінки**

Велика область в центрі, яка містить більшу частину унікального контенту даної веб-сторінки, наприклад, відео, яке ви хочете подивитися, чи розповідь, який ви читаєте, або карту, яку ви хочете переглянути, заголовки новин і т. д. Це та частина сайту, яка безумовно буде змінюватися від сторінки до сторінки!

### **Бічна панель сторінки**

Тут розміщується периферійна інформація, посилання, цитати, реклама і т.д. Зазвичай це контекстно до того, що міститься в основному контенті (наприклад, на сторінці новинний статті, бічна панель може містити біографію автора або посилання на пов'язані статті), але є також випадки, коли можна знайти повторювані елементи, такі як вторинна навігаційна система.

### **Нижній колонтитул**

Смуга в нижній частині сторінки, яка часто містить дрібний шрифт, повідомлення про авторські права та контактну інформацію. Це місце для розміщення загальної інформації, але як правило ця інформація не є критичною по відношенню до самого сайту. Нижній колонтитул також іноді використовується в цілях SEO, надаючи посилання для швидкого доступу до популярного контенту.

Вище описано зовнішня структура сайту, ще є внутрішня частина. Внутрішня структура сайту, як правило, представлена картою сайту – діаграмою, що візуально відображає ієрархію сторінок сайту, схему зв'язків і переходів між ними.

В якості середовища для написання коду було обрано програмний продукт VisualStudioCode — безкоштовний редактор з відкритим програмним кодом, використовується для редагування коду JavaScript, HTML, CSS, PHP.

Програмна реалізація веб-сервісу почалася з верстки шаблону сайту.

Верстка виконувалася мовою розмітки гіпертексту HTML. Для надання розмітці стилів використовувалися каскадні таблиці стилів CSS. Дана програма реалізована у вигляді web-сайту, який працює через будь-який доступний браузер.

Для адміністрування системи управління базами даних MySQL використовується веб-застосунок з графічним веб-інтерфейсом, розроблений на мові програмування PHP – phpMyAdmin.

*Приклад підключення до бази даних:*

```
<?php

$server = '127.0.0.1';
$login = 'root';
$pass = 'root';
$name_db = 'mybeauty_salon';

$link = mysqli_connect($server, $login, $pass, $name_db);

if ($link = FALSE)
{
    echo "З'єднання не вдалось";
}

?>
```

### 3.2 Макет сторінок веб сайту

Макет сайту - щось середнє між блокової схемою і прототипом. Це оптимальний варіант для клієнта, якому не доведеться переплачувати за інтерактивні ефекти прототипу. З іншого боку, макет не виглядає занадто простим як блокова схема. З його допомогою верстальщику простіше зрозуміти, яким саме має бути сайт. Залежно від домовленості між клієнтом і дизайнером, макет виглядає як блок із зазначенням розмірів або як майже готовий дизайн.

*Переваги використання макета:*

1. Визначення візуальних недоліків на перших стадіях роботи над проектом.
2. Можливість наочно показати клієнту дизайнерські ідеї.
3. Спрощення роботи програміста.
4. Розуміння того, як буде виглядати готовий сайт.
5. Зведення до мінімуму непорозумінь між дизайнером і замовником.

*Вимоги до готового макету:*

1. Симетричне відображення візуальних елементів.
2. Вказівка параметрів макету.
3. Створення елементів в різних варіантах відображення.
4. Використання сітки для вирівнювання блоків та інших елементів.
5. Окремий файл для повторів.
6. Використання стандартних шрифтів.
7. Використання окремих шарів для всіх елементів.

Сторінки веб-сайту мають меню, яке розміщене праворуч і вверху сторінки, основними елемента якого є логотип та перелік сторінок ( Рис 3.2.1).

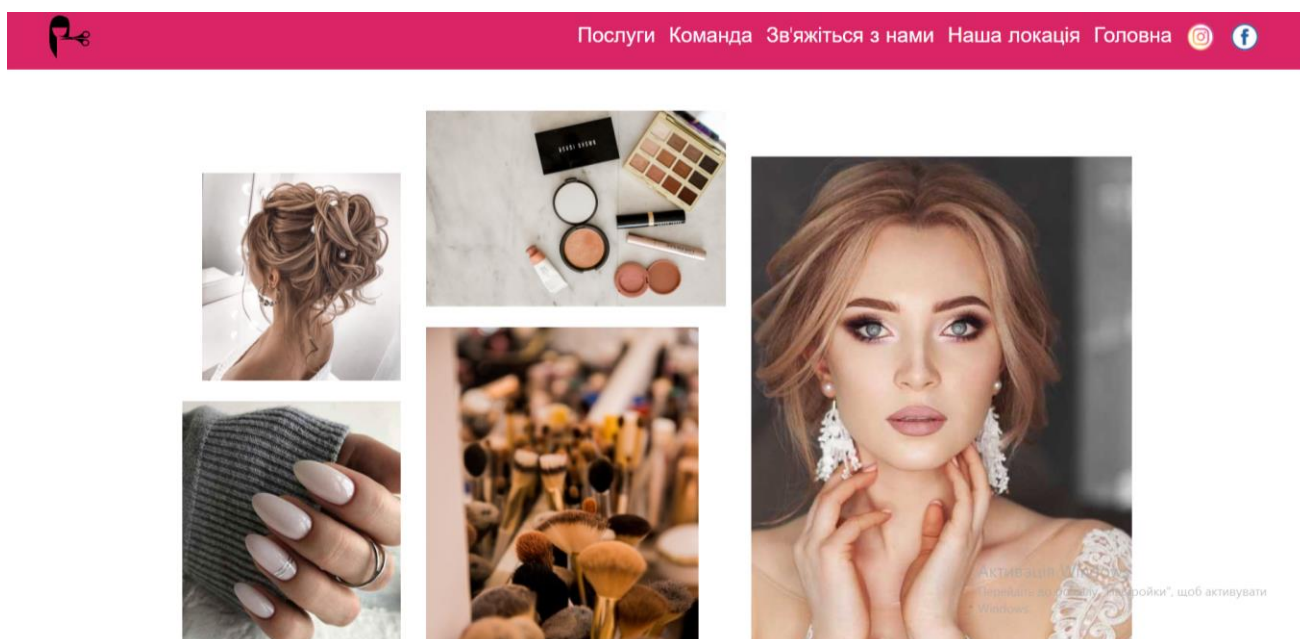


Рис. 3.2.1 Вигляд меню

На вкладці Наші послуги(Рис.3.2.2), ми можемо побачити весь асортимент послуг, який пропонує салон краси «Beauty».

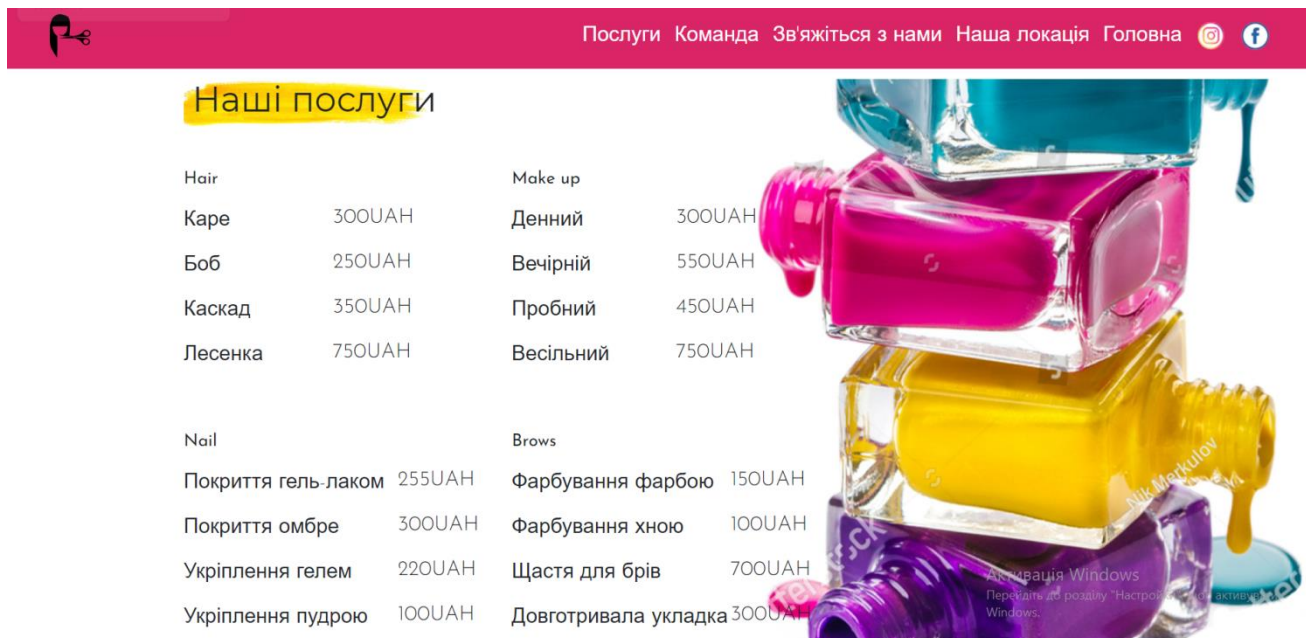


Рис. 3.2.2 Вигляд сторінки на вкладці Наші послуги

Сторінка Наша локація, яку показано на Рис. 3.4 складається з блоку контактної інформації, час та графік роботи салону.

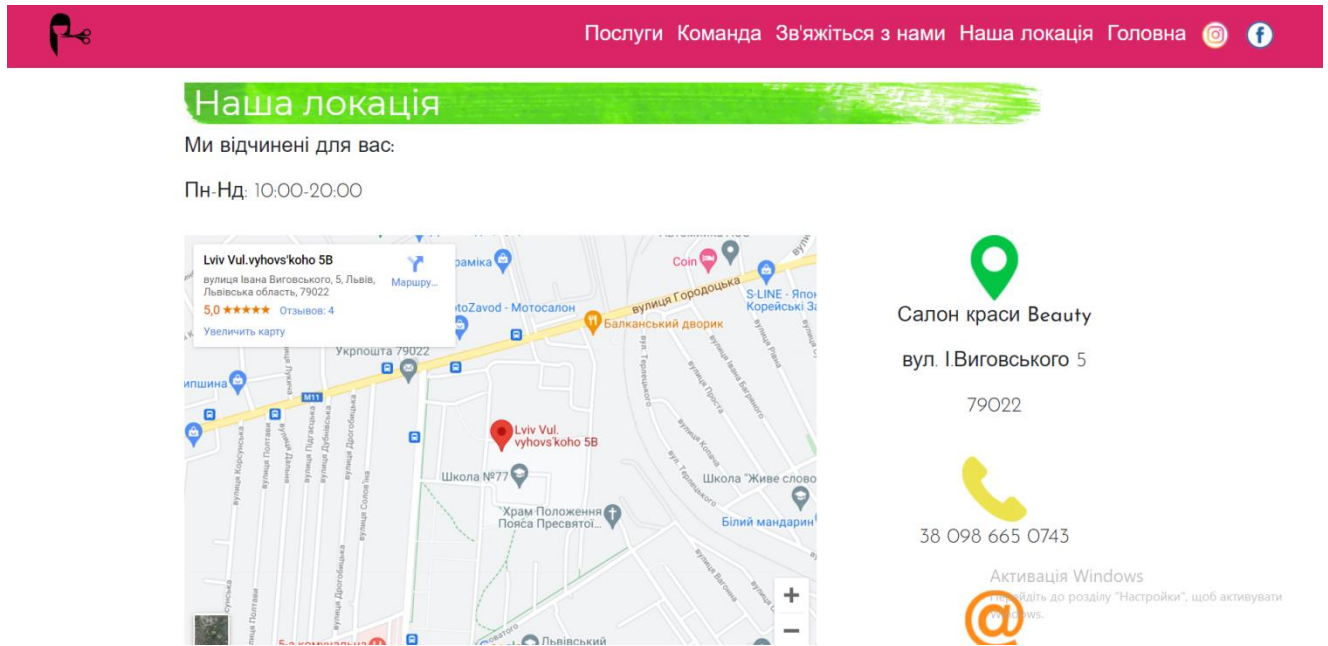


Рис. 3.2.3 Вигляд вкладки Наша локація

### 3.3. Програмування серверної частини

Процес створення сайту розділений на дві частини, клієнтську і серверну (Frontend та Backend). Клієнтська частина це Frontend розробка, до неї можна віднести CSS верстку, створення макета і шаблонів для сайту, а також призначеного для користувача інтерфейсу і спеціальних скриптів, що відповідають за візуалізацію і анімацію. До серверної частини (Backend розробка) відноситься створення ядра сайту, розробка платформи, основного функціоналу та адміністративної частини.

Наприклад, уявіть собі салон краси. Декорації, меблі, а також музика є клієнтською частиною (Frontend), а серверна частина (Backend) включає в себе майстрів, перукарів і менеджерів, які забезпечують безперебійну роботу салону.

Front-end - це та частина програмного забезпечення з якою безпосередньо контактує користувач.

Для прикладу у веб-розробці в якості Front-end виступають:

- HTML-верстка;
- Стили CSS;

- JavaScript.

Власне ці технології створюють картинку для користувача і з ними він контактує безпосередньо виконуючи ті чи інші дії на сайті.

PHP- скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. Для програмування серверної частини веб-сайту і було використано PHP.

За допомогою скриптів PHP організовано такі операції для функціонування сайту:

1. Зберігання даних користувача, а саме: email, name, phone.
2. Графік роботи салону краси.
3. Googleкарта, де ми знаходимося.

### 3.4. Програмування клієнтської частини

В даній роботі написано багато коду, який відповідає за оформлення сторінок та стилізацію з метою покращення вигляду Front-end частини. Завдання клієнтської частини (програми-клієнта) полягає у взаємодії з користувачем, передачі запиту користувача серверу, отримання результату запиту від серверної частини (програми-сервера) і представлення його взручному для користувача вигляді.

Для прикладу розглянемо використання мови HTML з метою створення розділу Наша локація.

```
<section class="container my-md-5 my-3">

<div class="row">
<div class="col-md my-3">
<div class="bg-green">
<h3 style="color:#fff">Наша локація</h3>
</div>

<p><strong>Ми відчинені для вас:</strong></p>
```

```

<p> Пн-Нд: 10:00-20:00</p>
</div>
</div>
<div class="row">
<div class="col-md-8">
<div id="map-container-google-1" class="z-depth-1-half map-container mb-4"
style="height: 500px">
<iframe

src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2573.70236054
9504!2d23.968032915524386!3d49.82925907939438!2m3!1f0!2f0!3f0!3m2!1i1024!
2i768!4f13.1!3m3!1m2!1s0x473ae738b04fd43d%3A0x30ec96bc683b322e!2sLviv%
20Vul.vyhovs%CA%B9koho%205B!5e0!3m2!1sru!2sua!4v1621542884221!5m2!1s
ru!2sua" width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"></iframe> "
frameborder="0" style="border:0;" allowfullscreen=""></iframe>
</div>
</div>

```

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

Для розміщення сайту салону краси використано локальне середовище Open Server.

Open Server Panel - це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників.

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і



тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

Хоча спочатку програмні продукти, що входять до складу комплексу, не розроблялись спеціально для роботи один з одним, така зв'язка стала дуже популярною серед користувачів Windows, в першу чергу через те, що вони отримували безкоштовний комплекс програм з надійністю на рівні Linux серверів.

Зручність і простота управління безумовно не залишать вас байдужими, за час свого існування Open Server зарекомендував себе як першокласний і надійний інструмент необхідний кожному веб-майстру.

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.

## ВИСНОВОК

На сьогоднішній день майже кожна людина використовує веб-сайти чи веб-додатки для своїх потреб і кількість таких людей зростає щороку. Цей прогрес відбувся доволі швидко, так, якщо проаналізувати розвиток інтернету за останніх 10-15 років то можна помітити, як різко розвивались веб технології та скільки різних можливостей добавилось з плином часу. Тому на перший погляд мережа Internet подібна до звичайних глобальних мереж. Але насправді вона має ряд абсолютно специфічних особливостей і унікальну структуру.

Програмування клієнської частини, як і серверної відіграє важливу роль. Оскільки клієнту необхідно забезпечити комфортне зовнішнєсприйняття інформації, тобто сайт повинен не тільки швидко „думати” і мати багато функціоналу, але й повинен мати хороший,привабливий дизайн. Оскільки саме дизайн сайту, це перше що бачить користувач, коли заходитьна той чи інший сайт.

Проаналізувавши вище зазначену інформацію, можна зробити висновки що у моїй курсовій роботі:

1. Було проаналізовано предметну область для розроблюваного продукту. Досліджено структуру та напрями діяльності об’єкту управління.Також були проаналізовані усі бізнес-процеси, які відбуваються у даному об’єкті управління.

2. Спроектовано та описано структуру БД, її таблиці та записи. Визначено типи даних та функціональні залежності. Розроблено запити та визначено атрибути полів. Для кращого розуміння архітектури продукту було розроблено діаграми, які показують взаємодію користувачів та їх функцій між собою та у системі.

3. В ході практичного завдання було розроблено веб-сайт салону краси “Beauty” використовуючи мову програмування PHP. Програмна реалізація веб-сайту почалася з верстки шаблону сайту.Верстка виконувалася мовою розмітки

гіпертексту HTML. Для надання розмітці стилів використовувалися каскадні таблиці стилів CSS.

На мій погляд, при належному просуванні сайт міг би мати успіх у певній категорії користувачів. Для цього необхідно перевести його на більш потужний веб сервер і більш містку і оперативну базу даних. Також потрібно продумати рекламні акції, спосіб оповіщення потенційних клієнтів і інші комерційні аспекти.

Надалі розроблений нами web-сайт можна буде використовувати в реальному житті. Так само можливе розширення списку послуг, що надаються, акцій, новин. Даний ресурс стабільний в роботі і не вимагає високих знань для роботи з ним.

## 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. Лаура Томсон, Люк Веллінг. - PHP & MySQL. – 300 с.
10. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования, 3-е издание = PHPObjects, PatternsandPractice, ThirdEdition. — М.: «Вильямс», 2015. — С. 560
11. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <http://learn.javascript.ru/>
12. Веб-дизайн, структура сайта [Электронный ресурс]. – Режим доступа: <https://webstudio2u.net/ua/design-web/403-structure-models.html> .
13. Bootstrap [Электронный ресурс]. — Режим доступа: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>.

14. Переваги MySQL [Електронний ресурс]. – Режим доступу:<https://studfile.net/preview/5607354/page:3/> .
15. Переваги PHP [Електронний ресурс]. – Режим доступу:<https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movy-prohramuvannia-php>.
16. Статті PHP [Електронний ресурс]. - Режим доступу:  
<https://tutorialzine.com/tag/php>
17. CSS styles [Електронний ресурс]. - Режим доступу: <http://htmlbook.ru/css>
18. Львівський IT -кластер [Електронний ресурс]. – Режим доступу:  
<http://itcluster.lviv.ua/about-us/about-cluster/>
19. Моделі даних [Електронний ресурс]. – Режим доступу:[https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%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%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).
20. 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).