

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЛЬВІВСЬКИЙ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА ФАКУЛЬТЕТ  
УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

Кафедра цифрової економіки та бізнес-аналітики  
(назва кафедри)

КУРСОВА РОБОТА

з «Проектування та адміністрування БД і СД»  
(назва дисципліни/дисциплін)

на тему:

«Інформаційна система для фотосалону»

спеціальність: Економіка  
(код та найменування спеціальності)

освітній ступінь: бакалавр  
(бакалавр/магістр)

Науковий керівник:

К. ф.-м.н., доц. Депутат Б. Я.

(науковий ступінь, посада, прізвище, ініціали)

\_\_\_\_\_ “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.  
(підпис)

Виконавець:

студент(ка) групи УФЕ-31с

Стець С. І.

(прізвище, ініціали)

\_\_\_\_\_ “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.  
(підпис)

Загальна кількість балів \_\_\_\_\_

(підпис, ПП членів комісії)

ЛЬВІВ 2020

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ.....	5
1.1 Постановка завдання.....	5
1.2 Розробка моделі варіантів використання веб-сайту.....	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.....	9
2.1 Опис моделі даних.....	9
2.2 Теорія нормалізації реляційної моделі даних.....	9
2.3 Визначення типів даних.....	14
2.4 Створення бази даних.....	14
2.5 Даталогічне проектування.....	16
2.6 Склад таблиць бази даних.....	17
2.7 Запити до таблиць бази даних.....	18
РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ.....	19
3.1 Інструменти для розробки веб сайту.....	19
3.2 Структура веб сайту.....	20
3.3 Програмування серверної частини.....	23
3.4 Програмування клієнтської частини.....	24
3.5 Розміщення веб-сайту на сервері.....	26
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
ДОДАТКИ.....	31

## ВСТУП

На сьогоднішній день сучасні організації, що надають будь-які послуги, вже неможливо уявити без застосування інформаційних технологій. Адже, автоматизація процесу управління і контролю стала закономірним етапом розвитку таких структур.

Основи сучасної інформаційної технології складають бази даних (БД) і системи управління базами даних (СКБД), роль яких як єдиного засобу зберігання, обробки і доступу до великих обсягів інформації постійно зростає. При цьому істотним є постійне підвищення обсягів інформації, яка зберігається в БД, що тягне за собою вимогу збільшення продуктивності таких систем.

За допомогою програмного продукту, як у працівників організації, так і у клієнтів з'явиться можливість оперативно і точно переглянути всі необхідні їм дані про реалізацію фотозйомок, тим самим уникнути можливих помилок і скоротити час, який було б витрачено при ручному виконанні даних операцій.

База даних повинна бути зручною у використанні та задовольняти усім стандартам розробки баз даних. Однією з найважливіших функцій бази даних є швидкість обробки запитів від користувача та швидке видання інформації для нього.

Тема даної курсової роботи є дуже актуальною, так як інформаційні системи охоплюють більшість аспектів роботи фотосалонів. База даних має зручний інтерфейс з користувачем. Завдяки цьому користувач може зручно виконувати пошук інформації яка його зацікавить.

Метою курсової роботи є розробка бази даних, яка забезпечує роботу з інформацією, що стосується структури фотосалону.

Об'єктом проведеного дослідження даної курсової роботи є практичне використання наступних засобів розробки: PHP, MySQL, JavaScript та css.

Предметом дослідження є проектування БД та створення ІС автоматизації обліку клієнтів фотосалону.

Завданнями курсової роботи є систематизація, закріплення і розширення теоретичних та практичних знань зі створення баз даних та розробки сайту для фотосалону.

Для досягнення поставлених цілей необхідно вирішити наступні завдання:

1. розглянути предметну область;
2. реалізувати проектування предметної області;
3. створити програмний продукт для роботи в сфері фотосалонів.

Створення сайту для фотосалону має практичне значення в діяльності організацій, що займаються фотографіями. При створенні бази даних дуже важливою частиною та етапом розробки було видання відповідей на запит для користувача. Тому в цій базі користувач має змогу отримати чіткі та якщо це необхідно відсортовані дані.

Даний програмний продукт покликаний спростити роботу, завдяки чому мінімізується час виконання запису на зйомку.

## РОЗДІЛ 1. АНАЛІЗ ВИМОГ

### 1.1 Постановка завдання

Завдання курсової роботи полягає у розробці інформаційної системи фотосалону, яка буде доступною усім клієнтам. Отже, для виконання поставленого завдання потрібно створити веб-сайт, який буде містити у собі базу даних, що стосується діяльності фотосалону.

База даних (БД) — це впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації (т.зв. записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Кожен користувач матиме можливість зареєструватися на сайті, переглянути пропоновані послуги, резервування та зв'язку з адміністратором через повідомлення, яке він може вписати у спеціальній формі, яка розміщена на сайті, а адміністратор отримає його у себе на пошті.

Існує інформація про клієнтів фотосалону. Перед нами стоїть завдання стосовно реалізації:

- а) Введення, редагування та збереження інформації;
- в) Запити.

У даній курсовій роботі було створено запити саме для спрощення роботи фотосалону, які міститимуть у собі основну інформацію, що стосується діяльності фотосалону та клієнтів, які користуються його послугами.

Завдання запитів полягає у швидкому пошуку інформації про потрібного клієнта та виду його замовлення.

Створена нами інформаційна система повинна бути доступною та легкому у користуванні, щоб структура веб-сайту була відразу зрозумілою та містила у собі найпотрібніші функції.

Серед них потрібно реалізувати такі форми, як:

- Форма для реєстрації клієнта;
- Форма для входу на веб-сайт;
- Форма для введення повідомлення для зв'язку клієнтів з адміністратором.

Отже, основне завдання для сайту – це відкритий доступ для будь-якого користувача, який хоче переглянути інформацію про фотосалон та фотогалерею чи скористатися його послугами.

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

При створенні веб-сайту важливо добре обдумати його зручність у використанні та вільний доступ для кожного клієнта. Дана програма працює через будь-який браузер. Щоб потрапити на сайт, потрібно перейти за адресою:

<https://kurs222.000webhostapp.com/index.php>

Можливі такі варіанти використання веб сайту:

- адміністратор (авторизація та перегляд замовлень);
- користувач (для перегляду та замовлення послуг).

Розглянемо детальніше подані варіанти. Отже, на сайті є форми для авторизації адміністратора та користувача. Адміністратор може змінювати інформацію, яку подано на сайті та переглядати замовлення від користувачів.

Користувач після реєстрації та входу в систему матиме можливість надсилання повідомлення адміністратору, перегляду послуг і вартість на них, після чого може здійснити замовлення.

Вигляд діаграми варіантів використання для користувачів продукту (клієнт, адміністратор) покажемо на рисунках:

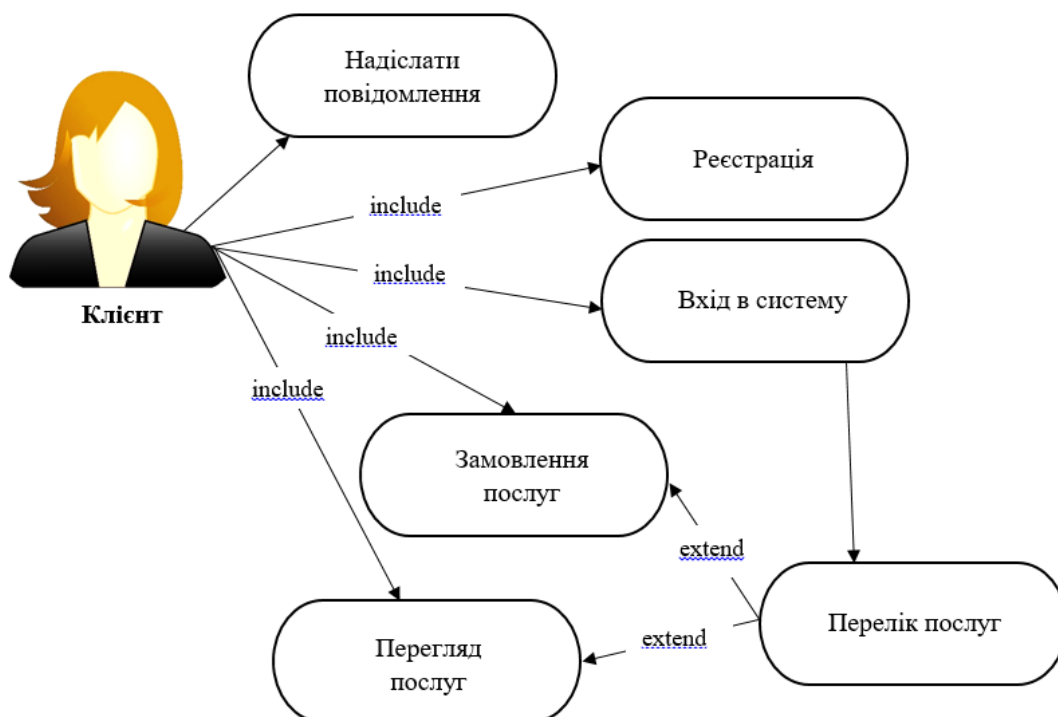


Рис. 1.1 – Діаграма варіантів використання для користувача



Рис. 1.2 – Діаграма використання для адміністратора

Основними сторінками клієнтської частини сайту є:

- Album – дозволяє повернутись на головну сторінку сайту з будь-якої іншої сторінки. Містить форму для резервування (Online Reservation),

фотогалерею, форму для написання повідомлення, а також контактну інформацію;

- Login – дозволяє заходити на сайт від імені зареєстрованого користувача;
- Register – безпосередня реєстрація клієнтів.

Адміністраторська частина дозволяє проводити операції стосовно зміни інформації в базі даних, безпосередньо на веб-сайті.

Дана частина містить такі пункти меню:

- зміни послуг та цін на них;
- зміни замовлення;
- перегляд повідомлення;
- зміни фотографій у портфоліо.

Вигляд сторінок знаходиться у додатках.

Наведемо опис до них:

1. Веб-сторінка клієнтської частини сайту:

<https://kurs222.000webhostapp.com/index.php>

Являє собою головну сторінку, де користувач може ознайомитись із інформацією про фотосалон та переглянути фотографії.

2. Веб-сторінка для реєстрації користувачів:

<https://kurs222.000webhostapp.com/register.php>

Потребує введення імені, пошти та вигаданого паролю.

3. Веб-сторінка для входу на сайт:

<https://kurs222.000webhostapp.com/login.php>

Для входу на сайт потрібно ввести дані пошти та паролю, який був вигаданий користувачем при безпосередній реєстрації.



## **РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ**

### **2.1 Опис моделі даних**

При написанні бази даних було використано реляційну модель даних. Дана модель є простою та доступною для використання, адже, єдиною використованою інформаційною конструкцією є таблиця.

Також перевагами реляційної моделі є повна незалежність даних та суворі правила проектування, які базуються на математичному апараті.

Метою створення реляційної моделі є:

1. Забезпечення максимально можливого ступеня незалежності прикладних програм від даних.
2. Створення міцного фундаменту для розв'язання семантичних питань, а також проблем несуперечності й надмірності даних.
3. Розширення мов управління даними за рахунок включення операцій над множинами.

Побудова реляційної моделі даних ґрунтується на тому розумінні, що будь-який набір даних може бути представлений у вигляді відношення, який оформлюється, але формі таблиці, де дані представляються атрибутами і значеннями на перетині відповідного атрибута із записом (кортежем).

### **2.2 Теорія нормалізації реляційної моделі даних**

З одного боку, процес проектування структури БД є процесом творчим, неоднозначним, з іншого боку, вузлові його моменти можуть бути формалізовані.

У процесі розробки логічна модель даних постійно тестується й перевіряється на відповідність вимогам користувачів. Коректність логічної моделі даних забезпечується процедурою нормалізації.

Процедура нормалізації БД полягає в усуненні надмірності даних та виявленні функціональних залежностей. Усування надмірності даних гарантує компактність наборів даних за рахунок уникання їх зайвого дублювання та унеможливорює виникнення аномалій вставки, вилучення й оновлення кортежів після їхньої фізичної реалізації.

Функціональна залежність – це залежність, що пов'язує атрибути в одному відношенні з єдиним значенням в іншому відношенні. Функціональну залежність для відношень  $A$  і  $B$  прийнято позначати як  $A \rightarrow B$ .

Це поняття підводить “на один крок” концепції об'єднання відношень зв'язками типу один до одного (1:1) або один до багатьох (1:M). Правила нормалізації або правила Кодда, як їх тепер прийнято називати, дуже прості й нечисленні, але досить суворі.

У випадку застосування до відношень з даними кожне правило описує наступний рівень відповідності вимогам теорії реляційних БД і різні ступені нормалізації. Існує п'ять різних рівнів нормалізації: перша нормальна форма (1НФ), 2НФ, 3НФ, нормальна форма Бойса-Кодда (БКНФ), 4НФ, 5НФ. Але дотепер жодна з реляційних СУБД не надає підтримки для всіх п'яти нормальних форм. Це відбувається через жорсткі вимоги до продуктивності.

Суть справи полягає в тому, що в повністю нормалізованій БД для виконання запиту Вам буде потрібно з'єднати настільки багато таблиць, що продуктивність такої системи не зможе задовольнити при виконанні запитів користувачів. Тому на практиці використовують лише перші три – 1НФ, 2НФ, 3НФ.

Щоб виконати нормалізацію, структура бази даних послідовно приводиться до різних форм. Взагалі кажучи, кожна наступна форма стосується і попередньої. Наприклад, щоб схема бази даних відповідала другій нормальній формі, вона повинна також відповідати і першій нормальній формі. На кожній стадії додається все більше правил, яким повинна відповідати схема.

Перша нормальна форма. Відношення представлене в першій нормальній формі (1НФ) тоді й тільки тоді, коли всі його атрибути містять тільки неподільні

(атомарні) значення й у ньому відсутні групи атрибутів з однаковими за змістом значеннями, які повторюються у межах одного кортежу. Неподільність значення атрибута говорить про те, що його не можна розділити на більш дрібні частини.

Наприклад, якщо у полі "Прізвище Ім'я По батькові" міститься прізвище, ім'я та по батькові читача, вимога неподільності не дотримується. Тоді необхідно виділити в окремі атрибути ім'я та по батькові. У результаті вийде три атрибути відношення «Читачі»: «Прізвище», «Ім'я» і « По батькові».

Друга нормальна форма. Відношення представлене у другій нормальній формі (2НФ) тоді й тільки тоді, коли воно є в першій нормальній формі, і кожний неключовий атрибут повністю визначається первинним ключем, тобто щоб первинний ключ однозначно визначав кортеж і не був надлишковим (збігався із суперключем). Ті атрибути, які залежать тільки від частини суперключа, повинні бути виділені до окремих таблиць.

Третя нормальна форма. У загальному випадку 1НФ і 2НФ розглядаються як проміжні етапи в процесі нормалізації БД. Більша частина СУБД орієнтована на досягнення наступного ступеня нормалізації – третьою нормальною формою (3НФ). Це пов'язано з тим, що представлення відношень у 3НФ цілком достатньо майже для всіх практичних задач.

При розробці винятково великих систем на надшвидкодіючих комп'ютерах, коли необхідно забезпечити максимальне скорочення обсягів даних, бажане провести подальшу нормалізацію відношень. Відношення представлене у 3НФ тоді й тільки тоді, коли воно є у 2НФ й у ньому немає транзитивних залежностей між неключовими атрибутами, тобто значення будь-якого атрибута відношення, що не входить до первинного ключа, не залежить від значення іншого атрибута, що не входить до первинного ключа.

Це визначення – усього лише оригінальний спосіб виразити необхідність представлення системи пов'язаних відношень у такому виді, щоб значення атрибутів кожного відношення безпосередньо визначалися або суперключем, або потенційним ключем цього відношення. Відношення X представлено в

нормальній формі, якщо в кожній нетривіальній функціональній залежності  $B \rightarrow A$   $B$  є суперключем.

Четверта та п'ята нормальні форми. Вони призначені для усунення ще двох аномалій: багатозначна залежність та об'єднуюча залежність. У відношенні  $X$  існує багатозначна залежність  $A \rightarrow B$ , якщо в ньому можна виявити ситуації, де пара кортежів містить значення  $A$ , які дублюється, і одночасно існують інші пари кортежів, що отримані шляхом перестановки значень  $B$ , присутніх у першій парі.

Насамперед, для існування багатозначної залежності потрібне існування пар кортежів.  $A$  й  $B$  можуть бути як окремими атрибутами, так і об'єднанням деякого набору атрибутів. 4НФ усуває нетривіальну багатозначну залежність у відношенні за допомогою створення менших відношень. Процес нормалізації полягає в створенні як можна більшого числа усе більш дрібних відношень з метою скорочення надмірності даних.

Відношення  $X$  представлено в 4НФ тоді й тільки тоді, коли воно презентовано в БКНФ і для будь-якої багатозначної залежності  $A \rightarrow B$  щодо цього можна сказати, що ця залежність або є тривіальною, або  $A$  є суперключем таблиці  $X$ . П'ята нормальна форма досягається в тому випадку, коли відношення не може далі розбиватися на більш дрібні відношення за допомогою операції проектування. Під операцією проектування розуміється декомпозиція, що не викликає втрати даних, і при якій відношення розбивається на частини таким чином, щоб для його одержання залишалася можливість об'єднання отриманих менших відношень.

Отже, нормалізація відношень БД покликана усунути з них надлишкову інформацію. Як видно з наведених вище прикладів, нормалізовані відношення БД містять тільки один елемент надлишкових даних – це атрибути зв'язку, які присутні одночасно в батьківському та дочірньому відношеннях. Оскільки надлишкові дані у відношеннях не зберігаються, заощаджується місце на носіях інформації.

Однак у нормалізованій БД є й недоліки, насамперед практичного характеру.

Чим ширше число суб'єктів (сутностей), що охоплюються предметною областю, тим з більшого числа відношень буде полягати нормалізована БД.

БД у складі великих систем, які керують життєдіяльністю організацій і підприємств, можуть містити сотні зв'язаних між собою відношень.

Оскільки поріг людського сприйняття не дозволяє одночасно охоплювати велику кількість об'єктів з обліком їх взаємозв'язків, можна затверджувати, що зі збільшенням числа нормалізованих відношень зменшується цілісне сприйняття БД як системи взаємозалежних даних.

Іншим недоліком нормалізованої БД є необхідність зчитування з відношень зв'язаних даних при виконанні складних запитів, які надають інформацію про взаємодію сутностей технологічного процесу між собою. При великих обсягах це приводить до збільшення часу доступу до даних.

Третя нормальна форма та нормальна форма Бойса-Кодда є теоретичними конструкціями, у той час як більшість розроблювачів БД працюють у реальному світі. Тому доречно зробити кілька зауважень про недоліки, що властиві відношенням, які представлені у ЗНФ.

Існують варіанти, коли має сенс розділити відношення на декілька дрібних, якщо частина представлених у ньому даних непостійна й часто оновлюється (оперативна інформація), а інші дані пасивні та змінюються в рідких випадках (довідкова інформація). Також є сенс об'єднати відношення при необхідності забезпечити високу швидкість реакції на запит. Необхідно також відзначити, що зв'язок між відношеннями доцільно здійснювати по атрибутах, повністю звільнених від семантичної залежності, яка породжується особливостями реалізації реальних процесів, що віддзеркалює БД.

Для цього використовують спеціально виділені інкрементні атрибути однозначної ідентифікації кортежів усередині відношень. Такий прийом дозволяє звільнитися від необхідності перевизначення зв'язків між відношеннями при зміні в реальному житті правил обліку різних суб'єктів діяльності організації. Наведені вище міркування не слід сприймати як заклик зовсім не нормалізувати дані.

Вони лише покликані показати, що при роботі з даними великого обсягу доводиться шукати компроміс між вимогами нормалізації (тобто "логічності" даних і економії місця на носіях інформації) та необхідністю поліпшення швидкодії. При цьому треба звертати увагу на вимоги користувачів, щоб уникати зайвої деталізації суб'єктів реальних процесів, які відбуваються на підприємстві.

### **2.3 Визначення типів даних**

Залежно від того, яка інформація буде розміщена в кожному полі таблиці, полю приписується відповідний їй тип даних.

MySQL має три основні типи даних: текстові, числові, та часові. В даній курсовій роботі використовувалися такі типи даних:

- серед текстових VARCHAR(size) - містить рядок змінної довжини. Найбільша довжина задається в дужках. Може зберігати до 255 символів;
- серед числових типів даних використовується:  
INT – діапазон -2 147 483 648 до 2 147 483 647  
Максимальне число цифр задається в дужках;
- серед типів даних та часу: 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.

### **2.4 Створення бази даних**

Створення бази даних здійснено на основі вільної системи керування реляційними базами даних - MySQL.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам.

MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних.

Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Для некомерційного використання MySQL є безкоштовним. Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Для того, щоб база даних могла функціонувати у неї потрібно ввести набір даних що відповідає предметній області.

Для наповнення бази даних створимо таблиці із даними про резервування, самого користувача.

```
--  
-- Структура таблиці `ticket`  
--  
  
CREATE TABLE `ticket` (  
  `id` bigint(20) UNSIGNED NOT NULL,  
  `name` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,  
  `email` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Рис. 2.1 – Структура таблиці «ticket»

```
--
-- Структура таблиці `users`
--

CREATE TABLE `users` (
  `id` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `email_verified_at` timestamp NULL DEFAULT NULL,
  `password` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `remember_token` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Рис. 2.2 – Структура таблиці «users»

```
--
-- Структура таблиці `table_content`
--

CREATE TABLE `table_content` (
  `id` bigint(20) UNSIGNED NOT NULL,
  `title` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `decs` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `photourl` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Рис. 2.3 – Структура таблиці «table\_content»

## 2.5 Даталогічне проектування бази даних

Даталогічний аналіз і проектування – це остаточне проектування реляційної схеми даних з врахуванням середовища обраної СУБД.

Проектування виконується на основі інфологічної схеми і полягає у виконанні таких пунктів:

- виділення таблиць (R-відношень);
- визначення фізичних форматів атрибутів на основі типів СУБД;
- визначення складу індексованих полів;
- визначення зв'язків навігації і логічної цілісності; - проектування представлень;



- проведення нормалізації реляційних схем (усунення надлишковості, багатозначності). Результат даталогічного проектування – реляційна схема бази даних.

## 2.6 Склад таблиць бази даних

Структура MySQL трирівнева: бази даних – таблиці – записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями `frm`, `MYD`, `MYI`.

Ім'я бази даних MySQL унікальна в межах системи, а таблиці - в межах бази даних, поля - в межах таблиці. База даних з точки зору MySQL - це звичайний каталог, що містить файли певного формату - таблиці.

Таблиці складаються із записів, а записи, у свою чергу, складаються з полів. Поле має два атрибути - ім'я і тип даних.

Основні типи даних, які використовуються в базі даних MySQL:

- `VARCHAR` – може зберігати не більше 255 символів. На відміну від `CHAR`, для зберігання значення даного типу виділяється необхідна кількість пам'яті;

- `TEXT` – може зберігати не більше 65 535 символів;

- `CHAR` – використовується для зберігання рядків. Параметр в дужках дозволяє фіксувати довжину збереженої рядка;

- `INT` – діапазон -2 147 483 648 до 2 147 483 647;

- `DATE` – дата в форматі `YYYY-MM-DD`;

- `TIMESTAMP` – зберігаються як кількість секунд з початку епохи Unix ('1970-01-01 00:00:00' UTC). Формат: `YYYY-MM-DD HH:MM:SS`.

Наша інформаційна система запрограмована на виведення у таблицю усіх користувачів, які в подальшому зареєструються, після чого до цієї таблиці можна буде застосовувати запити.

## 2.7 Запити до таблиць бази даних

Створення бази даних для фотосалону значно спрощує роботу для працівників, адже, база даних ціниться не лише зберіганням важливої інформації про діяльність фотосалону, а й можливістю швидкого виведення потрібних даних за допомогою поставлених запитів.

Під час виконання курсової роботи в основному було використано такі запити, як:

- запит на отримання фотографії з портфоліо фотосалону, який саме розміщено на сайті.

```
// Запит на отримання фотографії портфоліо
$qry = "SELECT * FROM kurs.table_content";
```

Рис. 2.4 – Запит отримання фотографії портфоліо

- запит на пошук користувача за іменем, який дозволить швидко знайти потрібну інформацію про клієнта та його замовлення:

```
//запит на пошук користувача з іменем '%name%'
$qry = "SELECT * FROM `kurs`.`users` WHERE `email` LIKE '{${email}}' AND `password` LIKE '{${pass}}'";
```

Рис. 2.5 – Запит пошуку користувача за іменем

- запит на додавання бронювання

```
//додавання тикета на бронювання
$qry = "INSERT INTO `ticket` (`name`, `email`, `created_at`, `updated_at`) VALUES ('{${name}}', '{${email}}', NULL, NULL);";
```

Рис. 2.6 - Запит на додавання бронювання

- запит на додавання фотографії до портфоліо

```
//додавання фотографії до портфоліо
$qry = "INSERT INTO `table_content` (`title`, `decs`, `photourl`) VALUES ('{${title}}', '{${decs}}', '{${photourl}}');";
```

Рис. 2.7 - Запит на створення користувача

Запити до бази даних значно спрощують роботу працівникам фотосалону, адже при потребі швидкого пошуку конкретного клієнта фотосалону, можна скористуватися саме пошуком через запити у базі даних усіх клієнтів, а не вручну здійснювати пошук. Використання бази даних є важливою частиною для ведення бізнесу та роботи в цілому.

## РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ

### 3.1 Інструменти для розробки веб сайту

Практичним завданням курсової роботи є розробка веб-сайту для фотосалону. Метою роботи є створення програмного продукту, який дозволить клієнтам здійснення резервацію, переглядати інформацію про послуги фотосалону, що зменшить затрати часу, як і клієнтам, так полегшить роботу персоналу.

У ході створення web-системи використовувались такі технології:

- MySQL — це система керування та роботою з базами даних, яка характеризується високою швидкістю, надійністю та простотою у використанні. MySQL надає великий набір функціональних можливостей, які мають безпечне середовище для зберігання, редагування та отримання даних;

- PHP — це мова програмування, яка дозволяє генерувати HTML-сторінки на стороні сервера;

- JavaScript — це динамічна, об'єктно-орієнтована мова програмування, яка надає можливість коду виконуватись на стороні клієнта, взаємодіяти із користувачем, обмінюватись даними із сервером, змінювати зовнішній вигляд web-сторінки;

- CSS — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript;

- HTML (англ. HyperText Markup Language – мова розмітки гіпертекстових документів) — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

Отже, для реалізації веб-сайту було обрано скрипкову мову програмування PHP та СУБД MySQL. MySQL складається з двох частин:

- серверної;
- клієнтської.

Сервер MySQL постійно працює на комп'ютері. Клієнтська програма, якою в даному випадку є PHP, посилає серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережеских засобів), сервер їх обробляє і запам'ятовує результат.

Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

Для роботи з MySQL буду використовувати PHP-функції, що призначені для роботи з СУБД. Також для обробки даних будуть використовуватись такі можливості PHP, як створення змінних, класів, використання логічних операцій порівняння даних тощо.

Для розділення інформації, яка міститься на сайті та програми, яка її видає, використовуються бази даних MySQL. Вся інформація сайту буде зберігатися в базі даних, а скрипт PHP буде здійснювати вибірку необхідної інформації для конкретної веб-сторінки.

### **3.2 Структура веб сайту**

Структура відповідає за два важливих складових успішності сайту. Від її правильності, логічної побудови залежить зручність користувача.

Якщо структура розроблена неправильно, навігація незручна для пошуку споживачем необхідної категорії, то він надовго не затримається на сайті, закривши вкладку з ним в браузері.

Структура веб-сайту складається із:

- головної сторінки;
- меню сайту;

- гіперпосилання на інші сторінки або сайти.

Структура нашого веб-сайту має ієрархічний вид. Подібне структурування добре уявляти у вигляді дерева, де наочно демонструється ступінь вкладеності сторінок, визначених у каталоги високого і низького рівня.

Ця модель є найпопулярнішою при веб-проектуванні. Адже, вона є зручною у використанні.

Покажемо ієрархічний вид сайту за допомогою графіка:

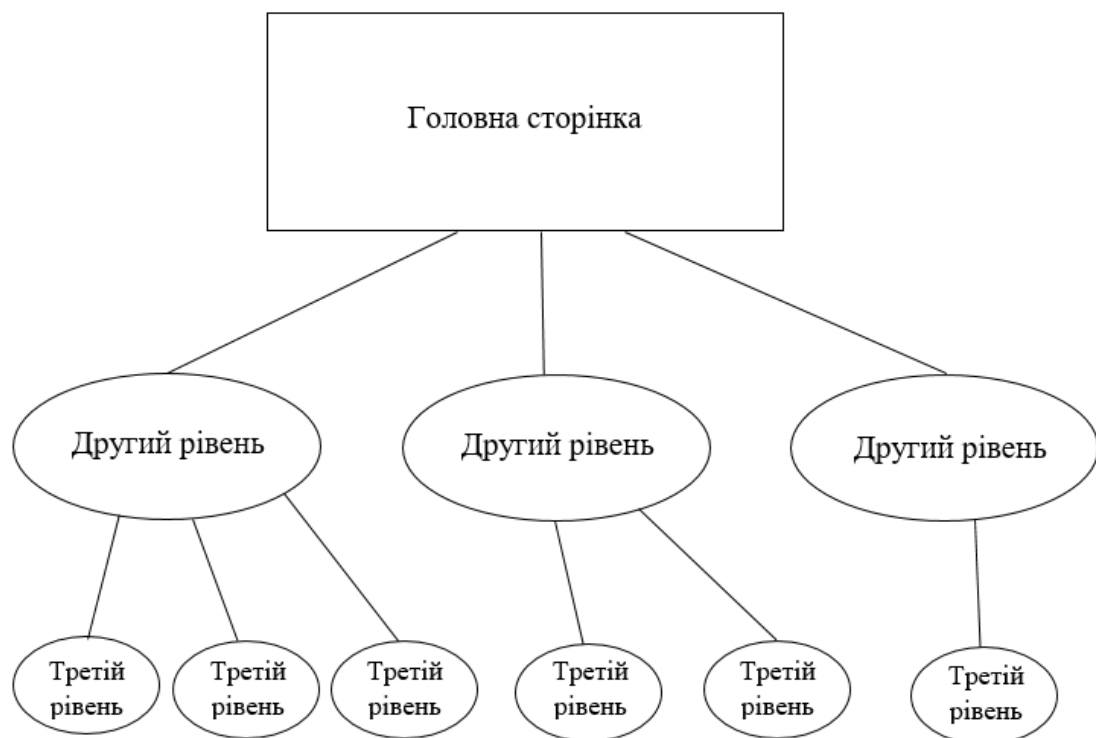


Рис. 3.1 – Графік ієрархічного виду сайту

Головна сторінка сайту містить у собі портфоліо фотографій, поля для резервування та поля для надсилання повідомлення адміністратору сайту.

Меню головної сторінки має такі пункти, як:

- альбом;
- реєстрація;
- вхід на сайт (введення особистих даних).

Реєстрація на сайті потребує введення власного імені, пошти та придуманого паролю для подальшого входу на сайт.

Для входу на сайт потрібно вводити дані про пошту та пароль, який зазначався при реєстрації на сайті.

Отже, при написанні даної курсової роботи використовувалась стандартна структура сайту, яка містить у собі найбільш затребувані функції щодо використання сайту для клієнтів фотосалону.

Таблицями бази даних є:

- а) users – таблиця, у якій міститься інформація про користувачів;
- б) tickets – таблиця, у якій міститься інформацію резервування;
- в) table\_content – таблиця, у якій знаходяться дані про наповнення сайту.

Усі сторінки сайту пов'язані між собою та взаємодіють одна з одною. Відобразимо логічну побудову усіх сторінок веб-сайту за допомогою діаграми сутностей та зв'язків:

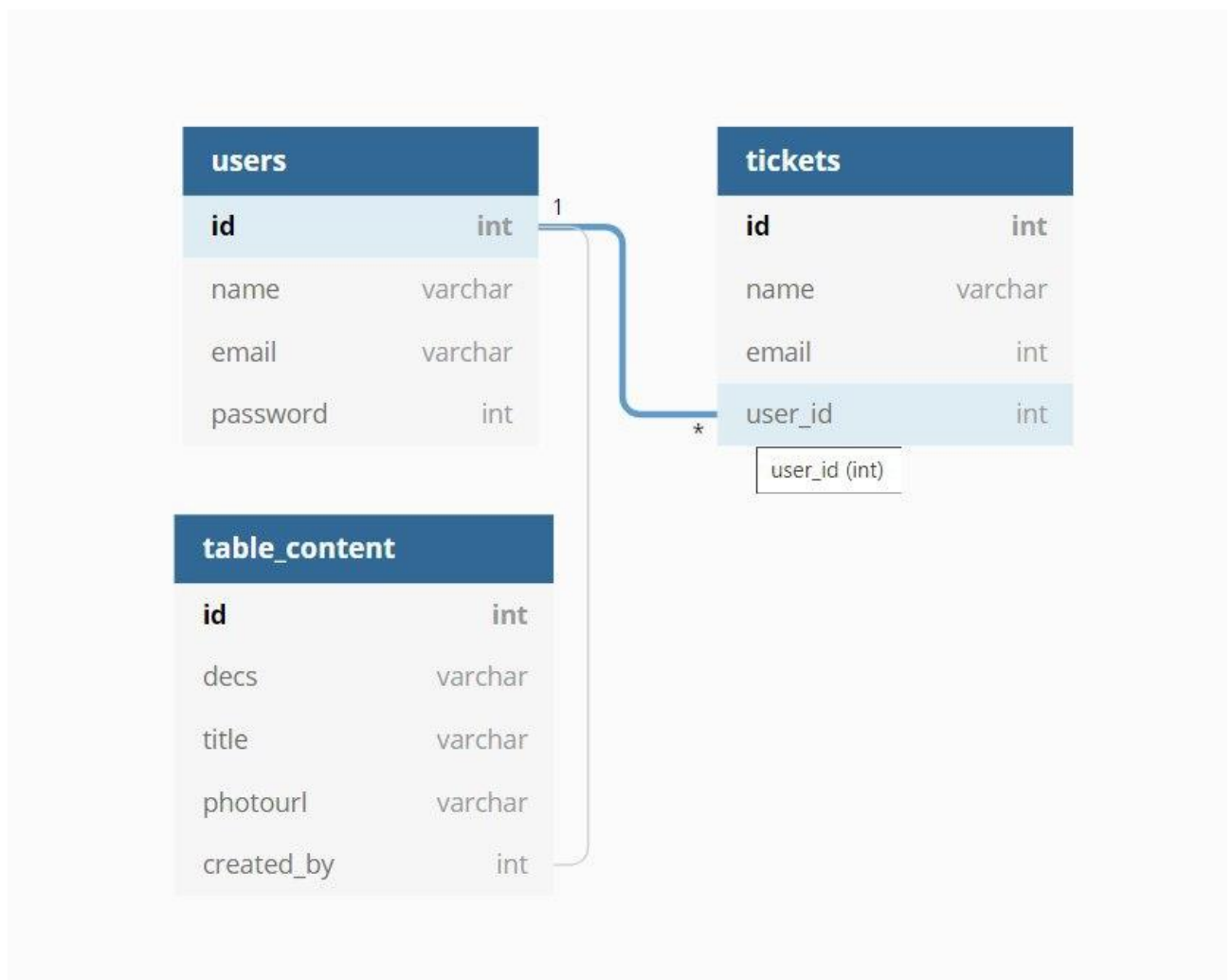


Рис. 3.2 - Діаграма сутностей та зв'язків

Основна веб-сторінка містить посилання на інші документи веб-сайту, а документи містять посилання, відповідно, на основну веб-сторінку. Це найпростіший і найпоширеніший спосіб організації веб-сайту.

Така структура сайту є дуже зручною, усі сторінки фотосалону послідовно пов'язані між собою.

### **3.3 Програмування серверної частини (Backend)**

Веб-розробку умовно можна розділити на дві частини — фронтенд (англ. front-end) і бекенд (англ. back-end).

Програмування серверної частини здійснюється на основі. Для розробки сайту ми використовували IDE Php Storm, Php7.3 і xDebug.

PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA. PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript.

Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни. Вся функціональність WebStorm включена в PhpStorm.

PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

Xdebug – розширення можливості зневадження і профілювання.

Зневаджувальна інформація, яку надає Xdebug, включає трасування стеку і функцій в повідомленнях про помилки з:

- повним відображенням параметрів для визначених користувачем функцій
- ім'я функції, назва файлу і рядка в ньому
- підтримка функцій-членів
- розподілення пам'яті
- захист від нескінченної рекурсії

Xdebug також надає:

- інформацію профілювання для PHP скриптів
- аналіз коду
- можливість зневаджувати ваші скрипти інтерактивно з фронт-ендом зневаджувача.

### **3.4 Програмування клієнтської частини (Frontend)**

До фронтенду (клієнтська частина) нашого сайту фотосалону відноситься HTML-верстка зі CSS-стилями і JavaScript.



Програмування клієнтської частини здійснюється на основі Css та Js. Для клієнтської частини використовувались бібліотеки Bootstrap для Css, і jQuery для js.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів.

Загальноприйнятим є положення, що клієнти та сервери — це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми — і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

При написанні клієнтської частини ми використовували CSS.

CSS - це мова стилів, що визначає відображення HTML-документів. Наприклад, у проектуванні нашої інформаційної системи, CSS працює з шрифтами, кольором, полями, рядками, висотою, шириною, фоновими зображеннями, позиціонуванням елементів.

CSS надає можливість створювати правила, які легко змінювати, редагувати и застосовувати до усіх визначених нами елементів. Кожне правило, складається з двох частин. У лівій частині міститься селектор (selector), а у правій блок визначення (declaration block). Блок визначення складається з набору властивостей (property) та їх значень (value).

JavaScript – це мова програмування, що дозволяє зробити Web-сторінку інтерактивною, тобто такою що реагує на дії користувача.

Послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний Web -браузер. Іншими словами, код програми вбудовується в HTML - документ і виконується на боці клієнта. Для виконання програми не потрібно навіть перезавантажувати Web-сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі значення і, якщо значення не відповідають очікуваним, заборонити відправлення даних.

### **3.5 Розміщення веб-сайту на сервері**

Для того, щоб сайт був доступний із мережі Інтернет необхідно розмістити його на хостингу. Для цього було проведено реєстрацію на безкоштовному хостингу <https://ru.000webhost.com/> та розміщено на ньому всі файли курсової роботи.

Даний хостинг є одним з найбільших хостинг-провайдерів, який можна безкоштовно використовувати. Довіра клієнтів обумовлена перевіреною часом, технічною підтримкою, надійністю і розвитком власних програмних продуктів.

Безкоштовний хостинг передбачає надання хостинг-провайдером безкоштовного дискового простору для розміщення сайту в Інтернеті. Безкоштовний хостинг, зазвичай, існує за рахунок реклами, що розміщується на сторінках сайтів. Ця реклама може бути у вигляді банерів, текстових посилань, рекламних фреймів, спливаючих вікон, хоча існують безкоштовні хостинги, які не розміщують на сайтах жодної реклами.

Працювати із цим хостингом досить комфортно і вигідно. Різноманіття способів оплати або навіть безкоштовне використання, миттєва реєстрація і функціональність нашої панелі управління зробить роботу максимально швидкою.

Завдяки співпраці зі світовими лідерами високих технологій, такими як: Intel, Supermicro, Cisco, Juniper, WD, Seagate пропонується розміщення сайтів тільки на найпотужнішому, надійному і швидкому обладнанні.

Надійність і безпека роботи наших серверів забезпечується:

- Багаторівневої перевіркою серверів;
- Регулярними профілактичними роботами;
- Системою резервного копіювання;
- Резервними каналами інтернету;
- Надійним ЦОД.

Основні недоліки безкоштовного хостингу:

- Невеликий об'єм, що надається для сайту.
- Низька надійність і стабільність серверного майданчика.
- Повільне завантаження сайтів.
- Присутність реклами.
- Часто відсутня підтримка PHP, баз даних та інших даних, що необхідні для повноцінного функціонування сайту.
- Відсутність гарантій якісного та постійного надання послуг.
- Хостинг є досить привабливим для малобюджетних, любительських чи тимчасових сайтів. Великим попитом користується серед юних розробників-початківців чи щойно створених спільнот.

У нашому випадку недоліком обраного безкоштовного хостингу є обмеження роботи з файлами. Тому деякі функції на сайті обмежені через дану особливість роботи на цьому хостингу.

## ВИСНОВКИ

Отже, база даних (БД) - це впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації (т.зв. записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин : збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Відповідно до поставленої мети в ході виконання роботи були отримані наступні результати:

- виконано дослідження предметної області;
- виконано проектування бази даних.

Існує інформація про клієнтів фотосалону. Було виконано завдання стосовно реалізації:

- а) Введення, редагування та збереження інформації;
- в) Запити.

У даній курсовій роботі було використано запити на створення користувача і таблиць, які міститимуть у собі основну інформацію, що стосується діяльності фотосалону та клієнтів, які користуються його послугами.

Наш фотосалон знаходиться за зазначеною адресою, де й саме займаються прийомом замовлень. Приміщення обладнане інструментами для проявлення плівок і друку фотографіями. У замовленні на друк вказується кількість фотографій з кожного кадру, загальна кількість, формат, тип паперу та терміновість виконання замовлення. При замовленні великої кількості фотографій надаються знижки.

Клієнтів можна розділити на професіоналів і любителів. Професіоналам, які часто приносять замовлення у фотосалоні, можуть бути запропоновані персональні знижки.

Відомості про виконані замовлення збираються і обробляються за допомогою бази даних, і вже на основі цієї інформації складають загальне замовлення на поставку витратних матеріалів (фотопапір, фотоплівка, хімічні реактиви), і обладнання. У фотосалону може бути декілька постачальників, які спеціалізуються на різноманітним поставках.

Система розроблена з метою економії часу при виконанні різних операцій з інформацією, наприклад, введення, переміщення, зміна даних, складання і роздруківка звітів.

Практичним завданням курсової роботи є розробка веб-сайту для фотосалону. Метою роботи є створення програмного продукту, який дозволить клієнтам здійснення перегляду та замовлення послуг фотосалону.

Таким чином, можна зробити висновок, що всі завдання курсового проектування вирішені і його мета досягнута.

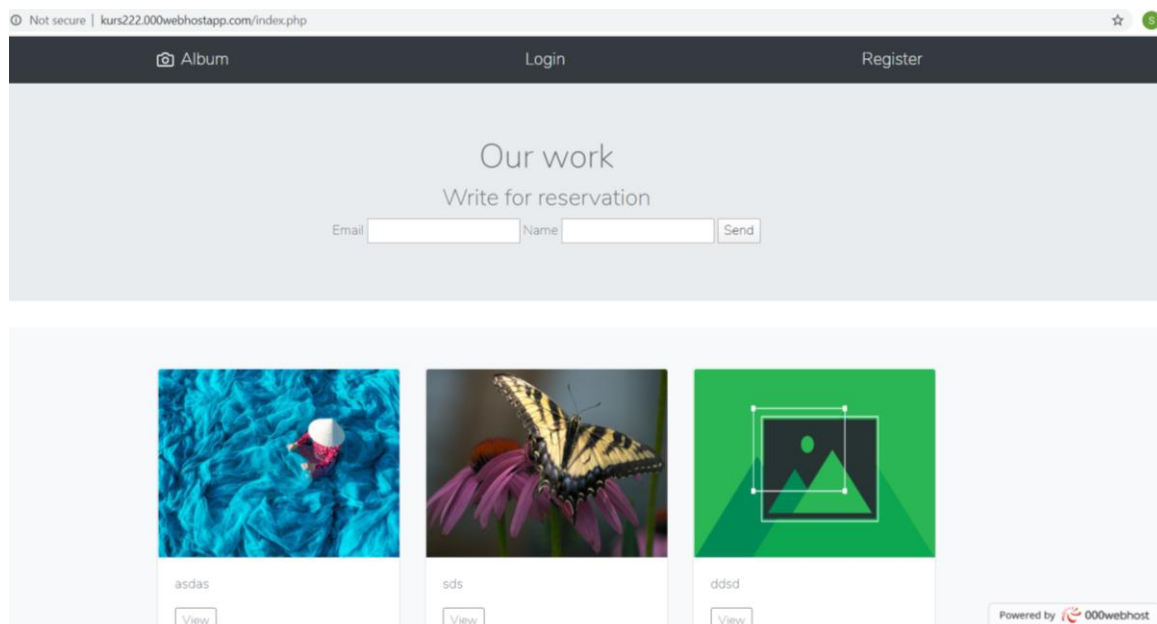
## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пасічник В. В. Організація баз даних та знань / В. В. Пасічник, В. А. Резніченко. – К.: Видавнича група BHV, 2006. – 384 с.
2. Документация по MySQL [Електронний ресурс]. – Режим доступу: <http://www.mysql.ru/docs/>
3. Гоше Х.Д. HTML5. Для профессионалов / Х.Д. Гоше. – [2-е изд.]. – СПб.: Питер, 2015. – 560 с.
4. Роббинс Д.Н. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Д.Н. Роббинс. – [4-е изд.]. – М.: Эксмо, 2014. – 516 с.
5. Флэнаган Д. JavaScript. Подробное руководство / Д. Флэнаган. – [6-е изд.]. – СПб.: Символ-Плюс, 2012. – 1080 с.
6. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – СПб.: Питер, 2014. – 304 с.
7. WebReference.ru [Електронний ресурс]. – Режим доступу: <https://webref.ru/>
8. Современный учебник JavaScript [Електронний ресурс]. – Режим доступу: <http://learn.javascript.ru/>
9. Mozilla Developer Network [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/ru/>
10. Node.js v4.5.0 Documentation [Електронний ресурс]. – Режим доступу: <https://nodejs.org/dist/latest-v4.x/docs/api/>
11. Express [Електронний ресурс]. – Режим доступу: <http://expressjs.com/ru/>


## ДОДАТКИ

### Додаток А

#### Головна сторінка веб-сайту для користувачів




## Головна сторінка веб-сайту для адміністратора

 Album [Logout](#)

## Our work


Add new photo to Portfolio

Title  Description  Photo url




asdas

[Back to top](#)



sdasd



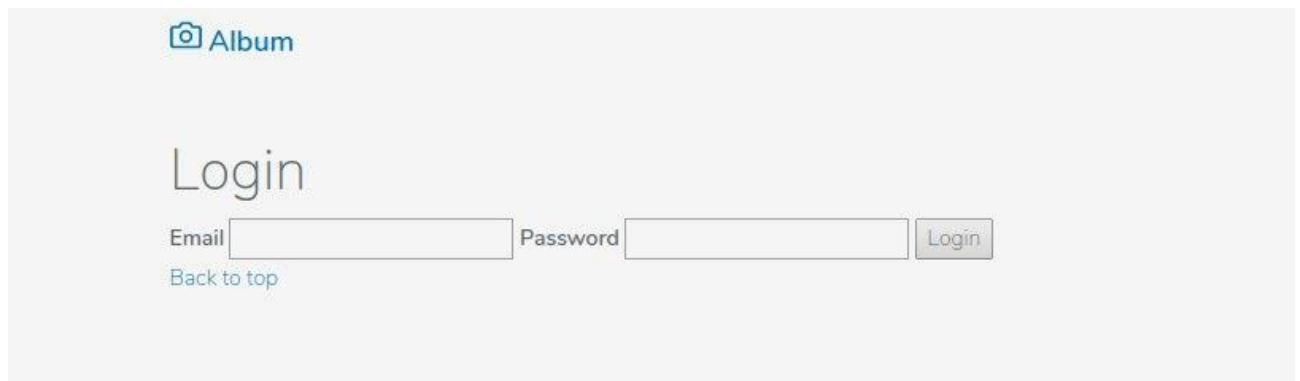
sds

Album example is © Bootstrap, but please download and customize it for yourself!

New to Bootstrap? Visit the [homepage](#) or read our [getting started guide](#).



## Сторінка для входу на веб-сайт



The image shows a login page for a website called "Album". At the top left, there is a camera icon followed by the word "Album". Below this, the word "Login" is displayed in a large, dark font. Underneath "Login", there are two input fields: one labeled "Email" and another labeled "Password". To the right of the "Password" field is a button labeled "Login". Below the "Email" field, there is a link that says "Back to top".

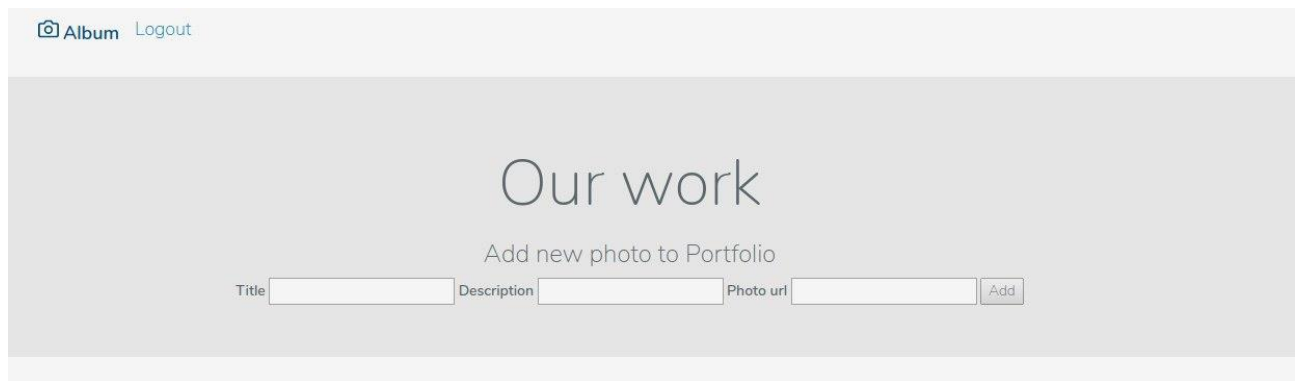
Album

# Login

Email  Password

[Back to top](#)

## Форма для додавання нової фотографії в портфоліо



The screenshot shows a web interface for adding a new photo to a portfolio. At the top left, there is a logo with a camera icon and the text "Album", followed by a "Logout" link. The main heading "Our work" is centered in a large, light gray font. Below it, the text "Add new photo to Portfolio" is centered in a smaller, light gray font. At the bottom, there is a form with three input fields: "Title", "Description", and "Photo url". Each field is preceded by its label. To the right of the "Photo url" field is a small "Add" button.

Album Logout

# Our work

Add new photo to Portfolio

Title  Description  Photo url

## Форма реєстрації нового користувача



The image shows a registration form for a new user. At the top left, there is a camera icon and the word "Album". Below this, the word "Registration" is displayed in a large, light gray font. Underneath, there are three input fields labeled "Name", "Email", and "Password". To the right of the "Password" field is a "Login" button. Below the "Name" field, there is a link that says "Back to top".

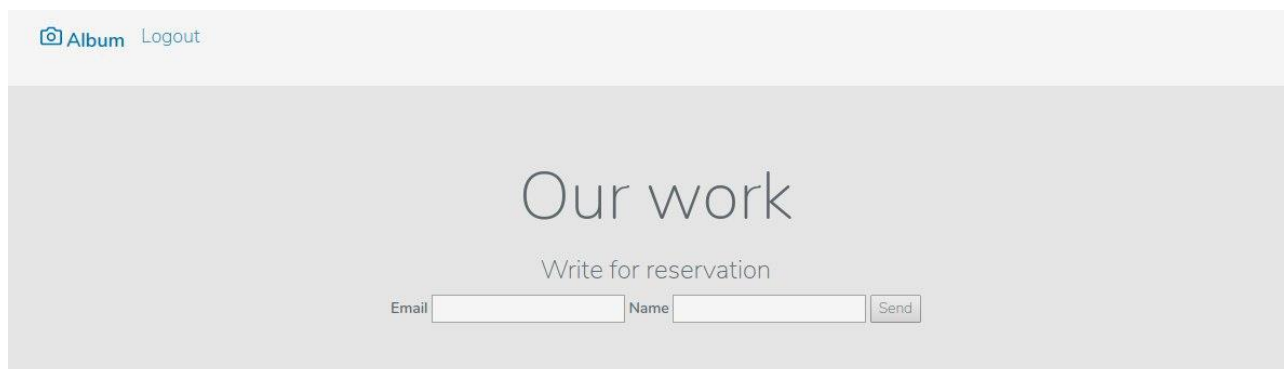
Album

## Registration

Name  Email  Password

[Back to top](#)

## Форма для зворотнього зв'язку (резервування)



The screenshot shows a web interface with a light gray background. At the top left, there is a logo with a camera icon and the text 'Album', followed by a 'Logout' link. In the center, the text 'Our work' is displayed in a large, dark font. Below it, the text 'Write for reservation' is shown in a smaller font. At the bottom, there are two input fields: the first is labeled 'Email' and the second is labeled 'Name'. To the right of the 'Name' field is a 'Send' button.


Album Logout

# Our work

Write for reservation


Email  Name

## Ціни на послуги фотосалону




asdas

[View](#)



sdsad

[View](#)



sds

[View](#)

### Pricing

Cool

\$0 / h

10 photos  
Email support  
Help center access

[Sign up for free](#)

Elite

\$15 / h

30 photos  
Priority email support  
Help center access

[Get started](#)

Luxury

\$29 / h

30 photos  
Phone and email support  
Help center access

[Contact us](#)

## Форма для надсилання повідомлення

Contact us

Name

Your E-mail

Your message

Please enter your message here....

## Про нас

### ЧОМУ МИ?

Завдяки новітній професійній фототехніці у нас ви зможете замовити будь-які види фотопослуг, отримавши кваліфіковану консультацію і результат, зробити термінове фото на документи, що отримаєте вже за 5-10 хвилин. Ми приймаємо як друк цифрових фотографій, так і фотографій з фотоплівок, оцифровуємо відео з записом на DVD-диск, роздруковуємо тексти, робимо ксерокопії.

## Контактна інформація

### Контакти

Адреса: 01001, Україна, м. Львів, вул. Личаківська, 25 А

E-mail: [sophiast1506@gmail.com](mailto:sophiast1506@gmail.com)

[sophiast1506@gmail.com](mailto:sophiast1506@gmail.com)

Телефон: (044)279-47-96