

**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
“Проектування та адміністрування БД і СД”**

**на тему:
«Інформаційна система для аудиторської
компанії»**

спеціальність: 051 «Економіка»

спеціалізація: «Інформаційні технології в бізнесі»

освітній ступінь: бакалавр

Науковий керівник:

к.ф.-м.н., доц. Депутат Б.Я.

(прізвище, ім'я, по-батькові)

_____ (підпис)

Виконавець:

Шевчук Ю.І.

(прізвище, ім'я, по-батькові)

УФЕ-31с група
_____ (підпис)

“ ____ ” _____ 2020 р.

“ ____ ” _____ 2020 р.

Загальна кількість балів _____
(підписи, ППІ 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	5
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.....	8
2.1 Базы даних в проектуванні та реалізації інформаційних систем	8
2.2 Опис моделі даних ІС для аудиторської компанії	10
2.2.1. Перелік таблиць бази даних	13
2.2.2. Перелік полів таблиць бази даних.....	13
2.3 Нормалізація реляційних відношень.....	14
2.4 Визначення типів даних	17
2.5 Обмеження цілісності даних.....	19
2.6 Запити до таблиць бази даних	19
РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ	22
3.1. Інструменти для розробки веб-сайту	22
3.2 Структура веб-сайту	23
3.3 Макети сторінок веб-сайту.....	24
3.3 Програмування серверної частини.....	26
3.4 Програмування клієнтської частини	27
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	30
ДОДАТКИ.....	32

ВСТУП

У сучасних реаліях спостерігається активний вплив ІТ-технологій у всіх сферах життя. Інформаційні системи (ІС) широко застосовуються в сфері послуг, щоб автоматизувати обмін інформацією між клієнтом та постачальником відповідних послуг, а також для швидкого обміну та передачі інформації. Використання інформаційної технології є джерелом підвищення ефективності роботи фірми та її управління.

При розробці сучасної інформаційної системи питання проектування зберігання і обробки даних стає найважливішим і, з огляду на те, що автоматизуються бізнес-процеси, база даних розглядається її невід'ємною частиною. Такий розгляд бази даних ґрунтується на необхідності передачі інформації в бізнес-процесах і роботи з документами.

Зважаючи на вище викладане, обрана тема курсової роботи є актуальною, оскільки інформаційна система допомагає забезпечити оптимізацію роботи підприємства.

Метою курсової роботи є дослідження теоретичних основ ІС та реалізація WEB-орієнтованої системи замовлення послуг аудиторської компанії з використанням баз даних.

Відповідно до мети було визначено такі завдання:

- проаналізувати предметну область та здійснити проектування БД аудиторської компанії на основі системи управління базами даних MYSQL;
- забезпечити редагування даних таблиць БД та взаємозв'язків між різними таблицями;
- розробити інтерфейс web-сторінки і організувати взаємодію її з БД на основі PHP.
- визначити шляхи подальшого вдосконалення розробленої ІС, навести рекомендації стосовно її експлуатації;

Об'єктом дослідження курсової роботи є особливості розробки інформаційної системи для аудиторської компанії з використання системи управління базами даних MySQL.

Предметом дослідження є теоритичні основни та практичне застосування технологій MySQL, PHP, Java Script, HTML.

При написанні курсової роботи використовувались загальнонаукові методи дослідження такі як: аналіз, індукція, конкретизація, систематизація і узагальнення, порівняльного аналізу, графічний метод.

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

Структурно робота складається з трьох розділів. У першому розділі описано бізнес-ідею та здійснено аналіз вимог. Другий розділ містить опис побудови навчальної бази даних. У третьому розділі продемонстровано підходи та реалізацію WEB-орієнтованої системи замовлення послуг аудиторської компанії. Загальний обсяг роботи становить – 40 сторінок.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

Інформаційна система для аудиторської компанії допомагає оптимізувати процес замовлення послуг. Вона оперує інформацією про послуги, які надає дана компанія.

Інформація про користувача зберігається у вигляді емейла та імені. ІС дозволить проводити замовлення без участі другої сторони, що зменшить затрати людських ресурсів. Основним завдання ІС є формування замовлення та передача його до бази даних та сторінки адміністратора.

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

- забезпечити реєстрацію користувача та адміністратора;
- забезпечити вхід в систему та перегляд послуг користувачем;
- забезпечити підтвердження замовлення користувачем;
- створити можливість адміністратору перегляд замовлень.

При здійсненні замовлення може виникнути потреба коректування, тобто варто забезпечити редагування вмісту замовлення. При виникненні запитань користувач матиме можливість скористатись контактною формою для швидкого реагування. Також для зручності додано підсумкову суму замовлення.

1.2 Розробка моделі варіантів використання веб-сайту

Головним користувачем системи є клієнт, кінцева мета якого є здійснити замовлення. Клієнт передає свої дані: емейл та ім'я в систему, натомість з бази даних він отримує список послуг та формує власне замовлення. Вигляд діаграми варіантів використання клієнтом продукту продемонстровано на рисунку 1.1.

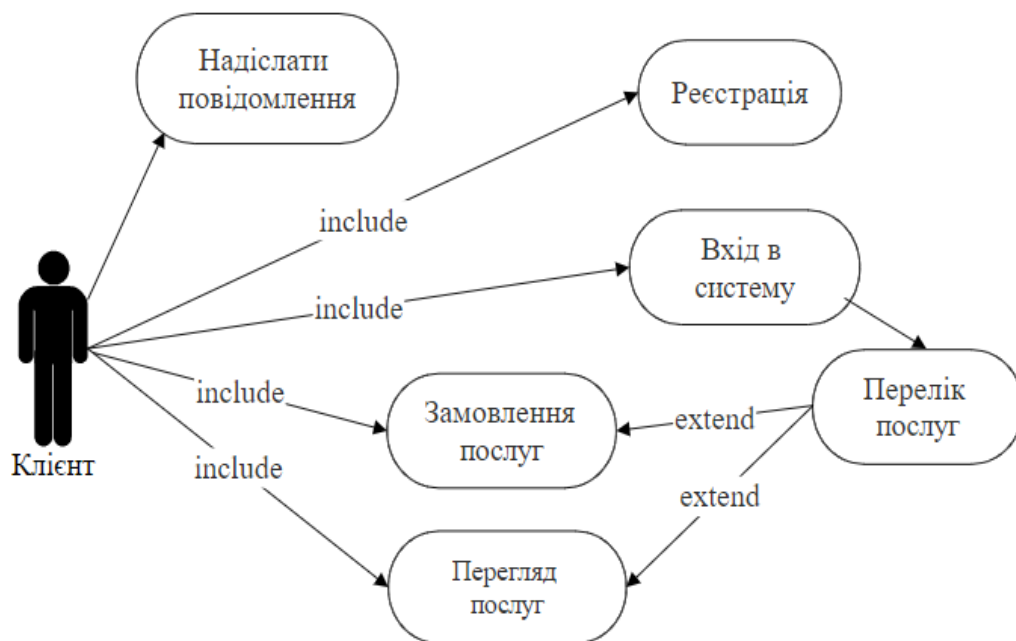


Рис. 1.1 – Діаграма варіантів використання для користувача

Другим користувачем системи є адміністратор, який переглядає замовлення клієнта. Вигляд діаграми варіантів використання адміністратором продукту продемонстровано на рисунку 1.2.

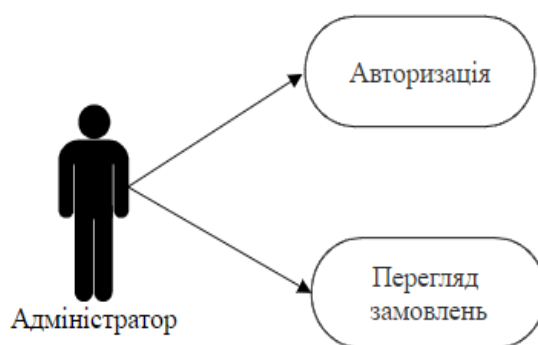


Рис. 1.2 – Діаграма варіантів використання для адміністратора

Отже, метою інформаційної системи для аудиторської компанії є формування замовлення зовнішнім користувачем, тобто клієнтом та перегляд замовлення внутрішнім користувачем – адміністратором.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Базы даних в проектуванні та реалізації інформаційних систем

Реляційна база даних являє собою тип бази даних, яка зберігає і надає доступ до точок даних, які пов'язані один з одним. Реляційні бази даних базуються на реляційній моделі, інтуїтивно зрозумілому, прямому способі подання даних у таблицях. Основними поняттями реляційних баз даних є: тип даних, домен, атрибут, кортеж, первинний ключ і відношення.

У реляційній базі даних кожен рядок таблиці - це запис із унікальним ідентифікатором, який називається ключем. Стовпці таблиці містять атрибути даних, і кожен запис зазвичай має значення для кожного атрибута, що полегшує встановлення зв'язків між точками даних.

Атрибут – це поименований стовпець відношення. Відношення зазвичай має вигляд двовимірної таблиці, у якій рядки відповідають окремим записам, а стовпці – атрибутам [1].

Атрибут може бути багатьох типів, які є визначені в моделі бази даних ER:

1. Простий атрибут: атрибути зі значеннями, які є атомарними і не можуть бути розділені в подальшому. Наприклад, вік студента .
2. Складений атрибут: складається з декількох простих атрибутів. Наприклад, адреса студента може містити номер будинку, назву вулиці, пінкод тощо.
3. Похідний атрибут: це атрибут, який отриманий за допомогою інших атрибутів. Наприклад, середній вік учнів у класі.
4. Однозначний атрибут: як впливає з назви, вони мають єдине значення.
5. Багатозначний атрибут: може мати кілька значень.

Поняття тип даних в реляційній моделі даних повністю відповідають поняттю типу даних в мовах програмування. Реляційні БД підтримують наступні типи даних:

- числові;
- символічні;
- великі двійкові об'єкти (малюнки та медіа-файли);
- бітові рядки;
- спеціалізовані числові дані (такі як «гроші»);
- спеціальні «темпоральні дані» (дата, час та часовий інтервал).

Ключ відносини - це підсхема вихідної схеми відносини, що складається з одного або декількох атрибутів, для яких декларується умова унікальності значень в кортежі відносин. При визначенні схеми відносини можуть бути задані декілька ключів.

Ключі в базах даних можуть бути представлені наступними видами:

- суперключ;
- потенційний ключ;
- первинний ключ;
- зовнішній ключ;
- сурогатний ключ.

Ключ відносини може бути простим або складеним. Простий ключ - це ключ, що складається з одного атрибута. Складним є ключ, що складається з двох і більше атрибутів [2].

Суперключ - це атрибут або безліч атрибутів, який єдиним чином ідентифікує кортеж даного відношення. Він може включати додаткові атрибути.

Потенційний ключ - це підмножина атрибутів відносини, що задовольняє вимогам унікальності і ненадлишкових. Він має такі властивості: 1) унікальність: в таблиці немає двох різних рядків з однаковими значеннями в потенційному ключі; 2) ненадмірність: не можна прибрати один з стовпців з ключа, так, щоб він

не втратив унікальності; 3) у відношенні може бути більше одного потенційного ключа.

Первинний ключ (primary key, PK) - це один з потенційних ключів відношення, обраний в якості основного ключа. Атрибути первинного ключа не можуть набувати значень Null.

Зовнішній ключ (foreign key, FK) - це ключ, визначений в базовому відношенні, який при цьому посилається на первинний того ж самого або іншого базового відношення.

Сурогатний ключ - це службовий атрибут, доданий до вже наявних інформаційних атрибутів відносини. Призначенням сурогатного ключа є служити первинним ключем відношення. Значення цього атрибута генерується штучно [2].

2.2 Опис моделі даних ІС для аудиторської компанії

Етап концептуального проектування полягає в описі та синтезі інформаційних вимог користувачів у початковий проект БД. Вихідними даними можуть бути сукупність документів користувача при класичному підході або алгоритми додатків (алгоритми бізнесу) при сучасному підході. Результатом цього етапу є високорівневе подання, а саме у вигляді системи таблиць БД інформаційних вимог користувачів на основі різних підходів.

Спочатку вибирається модель БД. Потім створюється структура БД, яка заповнюється даними за допомогою систем меню, екранних форм або в режимі перегляду таблиць БД. Тут же забезпечується захист і цілісність (у тому числі посилальна) даних за допомогою СУБД або шляхом побудови тригерів[3].

Виконати концептуального проектування бази даних передбачає визначення логіки функціонування сайту. Функціонал web-сайту можна забезпечити через такі процеси:

- реєстрація та авторизація користувача сайту;
- безпосереднє здійснення замовлення послуг.

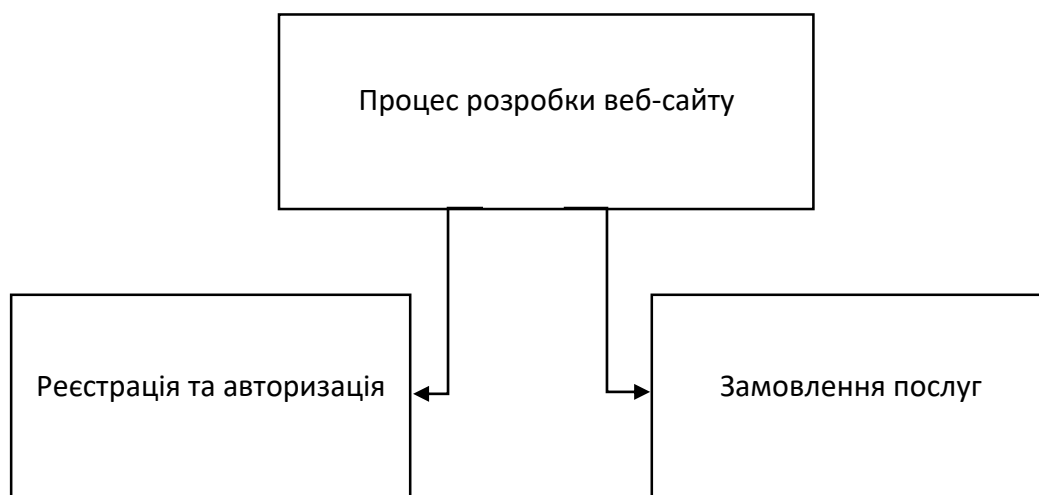


Рис. 2.1 – Діаграма функцій процесу роботи веб-сайту

Процес замовлення послуг пов'язаний із реєстрацією та авторизацією користувача, де безпосередньо представлена можливість переглянути послуги, які надаються компанією та здійснити їх замовлення. Такий процес був розроблений з метою швидкого збору інформації про клієнта, який здійснює замовлення, а також про послуги, які він обрав. Це допомагає зменшити витрати часу користувача на заповнення даних та цілісної передачі даних про замовлення.

Для здійснення реєстрації користувач повинен вказати ім'я, електронну пошту та пароль.

Таблиця 2.1

Процес реєстрації

Назва характеристики	Значення характеристики
Назва процесу	Реєстрація користувача
Основні учасники	Користувач
Вхідна подія	Запит на реєстрацію
Вихідна подія	Повідомлення про успішну реєстрацію
Клієнт процесу	Користувач

Далі розглянемо процес авторизації на основі вищезазначеного процесу реєстрації.

Таблиця 2.2

Процес авторизації користувача

Назва характеристики	Значення характеристики
Назва процесу	Вхід користувача
Основні учасники	Користувач
Вхідна подія	Запит на вхід
Вихідна подія	Перехід на користувацьку сторінку
Клієнт процесу	Користувач

Наступним процесом є замовлення. Цей процес є найважливішим, оскільки сам сайт створений для забезпечення цього процесу.

Таблиця 2.3

Процес замовлення

Назва характеристики	Значення характеристики
Назва процесу	Замовлення
Основні учасники	Користувач
Вхідна подія	Запит на замовлення
Вихідна подія	Повідомлення про успішне замовлення
Клієнт процесу	Користувач

Завершальним етапом проектування є створення концептуальної моделі бази даних, яка зображена на рисунку 2.2.

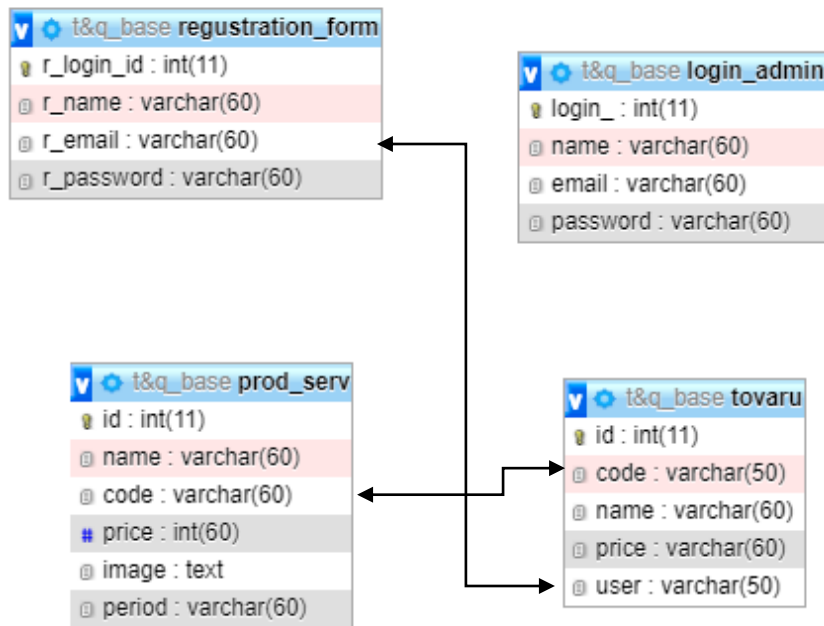


Рис. 2.2 – Концептуальна модель

2.2.1. Перелік таблиць бази даних

База даних 't&q_base' містить такі таблиці :

- 1) registration_form – таблиця, у якій міститься інформація про користувача, яка передається при реєстрації та використовується при процесі авторизації в систему;
- 2) prod_serv – таблиця, у якій знаходиться доступні послуги для користувача;
- 3) tovaru – таблиця, у яку вносяться дані про замовлення та користувача сайту.
- 4) login_admin – таблиця, у якій знаходяться дані про адміністратора, що необхідні при авторизації.

2.2.2. Перелік полів таблиць бази даних

Поля, які ідентифікують властивості таблиці «registration_form»:

1. r_login_id - ідентифікаційний код користувача;

2. r_name – ім'я користувача;
3. r_email – емейл користувача;
4. r_password – пароль користувача.

Поля, які ідентифікують властивості таблиці «prod_serv»:

1. id – індифікаційний код послуги;
2. name – назва послуги;
3. code – код послуги;
4. price – ціна послуги;
5. image – візуальна передача назви послуги;
6. period – період виконання послуги (вказується в тижнях).

Поля, які ідентифікують властивості таблиці «tovaru»:

1. id - індифікаційний код замовлення;
2. code – код замовлення;
3. name – назва замовлення;
4. price – ціна замовлення;
5. user – емейл користувача;

Поля, які ідентифікують властивості таблиці «login_admin»:

1. login_ - ідентифікаційний код адміністратора;
2. name – ім'я адміністратора ;
3. email – емейл адміністратора;
4. password – пароль адміністратора.

2.3 Нормалізація реляційних відношень

Нормалізація - це розділення таблиці на дві чи більше частин, які характеризуються кращими властивостями при доповненні, зміні і вилученні даних. Кінцева мета нормалізації полягає в отримання такого проекту бази даних, у якому кожний факт з'являється лише в однім місці, тобто надлишковість

інформації є виключеною. Це робиться з метою виключення можливої суперечливості збережених даних.

Перша нормальна форма (1НФ) - це звичайне відношення, на перетині рядків і стовпів якого розташоване атомарне значення відповідного атрибута, що має унікальне значення ключового атрибута, який не допускає дублювання кортежів. Таке відношення автоматично вже знаходиться в 1НФ.

Відношення знаходиться в другій нормальній формі (2НФ) тоді, коли воно знаходиться в 1НФ і немає неключових атрибутів, залежних від частини складного ключа. Неключовим атрибутом називається атрибут, який не входить до складу жодного потенційного ключа. Варто зазначити, що якщо потенційний ключ відношення є простим, то відношення автоматично перебуває під 2НФ.

Відношення знаходиться в третій нормальній формі (3НФ) тоді, коли відношення знаходиться в 2НФ і всі неключові атрибути є взаємно незалежні. Атрибути є взаємно незалежними тоді, коли жоден з них не є функціонально залежним від іншого[4].

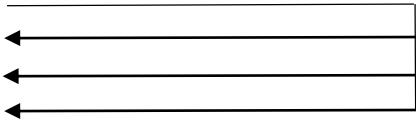
Поняття функціональної залежності є базовим, так як на його основі формуються визначення всіх інших видів залежностей. Функціональна залежність - це зв'язок між атрибутами.

У відношенні R атрибут Y функціонально залежить від атрибута X (X і Y можуть бути складовими) у тому випадку, якщо кожному значенню X відповідає в точності одне значення Y.

Функціональні залежності між атрибутами в таблиці «registration_form», котра містить усю інформацію про користувача зображено у Таблиці 2.4. У ній зображено, що ключовим полем є поле ідентифікаційний код користувача, тобто поле «r_login_id», із значенням «Primary». Поля «r_name», «r_email», «r_password», залежні від поля «r_login_id».

Таблиця 2.4


Функціональні залежності між атрибутами сутності «registration_form»

Найменування атрибутів	Функціональні залежності
r_login_id r_name r_email r_password	

Далі розглянемо функціональні залежності між атрибутами в таблиці «prod_serv», у якій міститься інформація про послуги. Графічно вона представлена в Таблиці 2.5. Поля «name», «code», «price», «image», «period», залежні від поля «id».

Таблиця 2.5

Функціональні залежності між атрибутами сутності «prod_serv»

Найменування атрибутів	Функціональні залежності
id name code price image period	

Функціональна залежність між атрибутами в таблиці «tovaru», у якій міститься інформація про замовлення, графічно відображається в Таблиці 2.6. У ній зображується, що ключовим полем є ідентифікаційний код замовлення, тобто поле «id», із значенням «Primary». Усі інші поля підпорядковуються полю «id».

Таблиця 2.6

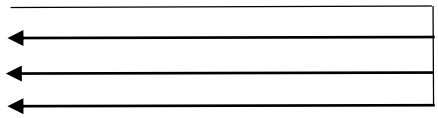
Функціональні залежності між атрибутами сутності «tovaru»

Найменування атрибутів	Функціональні залежності
id code name price user	

Функціональні залежності між атрибутами в таблиці «login_admin», котра містить усю інформацію про адміністратора зображено у Таблиці 2.7.

Таблиця 2.7

Функціональні залежності між атрибутами сутності «login_admin»

Найменування атрибутів	Функціональні залежності
r_login	
name	
email	
password	

2.4 Визначення типів даних

Структура MySQL в своїй структурі містить такі елементи:

- база даних;
- таблиці;
- записи таблиць.

Першим етапом є створення бази даних в кореневому каталозі phpMyAdmin з назвою “t&q_base”. Наступний етап полягає у наповненні бази даними таблицями: registration_form, prod_serv, tovaru, login_admin, які використовуються для функціонування сайту. Завершальним етапом є наповнення таблиць даними. Вміст таблиць продемонстрований в Таблицях 2.8-2.11.

Таблиця 2.8

Склад таблиці «registration_form»

№п/п	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	r_login_id	int	11	Первинний ключ
2	r_name	varchar	60	Обов’язкове
3	r_email	varchar	60	Обов’язкове
4	r_password	varchar	60	Обов’язкове

Таблиця 2.9*Склад таблиці «prod_serv»*

№п/п	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	id	int	11	Первинний ключ
2	name	varchar	60	Обов'язкове
3	code	varchar	60	Обов'язкове
4	price	int	60	Обов'язкове
5	image	text	-	Обов'язкове
6	period	varchar	60	Обов'язкове

Таблиця 2.10*Склад таблиці «tovaru»*

№п/п	Найменування атрибутів	Тип полів	Тип полів	Додаткові обмеження цілісності
1	id	int	11	Первинний ключ
2	code	varchar	50	Обов'язкове
3	name	varchar	60	Обов'язкове
4	price	varchar	60	Обов'язкове
5	user	varchar	50	Обов'язкове

Таблиця 2.11*Склад таблиці «login_admin»*

№п/п	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	login_	int	11	Первинний ключ
2	name	varchar	60	Обов'язкове
3	email	varchar	60	Обов'язкове
4	password	varchar	60	Обов'язкове

2.5 Обмеження цілісності даних

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.

У навчальній базі “t&q_base” застосовано наступні обмеження щодо первинних ключів:

- not null;
- primary key;
- auto_increment;

Not null – обмеження, яке застосовується до всіх полів таблиць бази даних.

2.6 Запити до таблиць бази даних

Навчальний сайт реалізує такі запити до бази даних:

1. Запити типу «SELECT»:

- SELECT * FROM prod_serv WHERE code="" . \$_GET["code"] . "" – даний запит виконує виведення всіх послуг;
- SELECT * FROM prod_serv ORDER BY id ASC – запит виконує виведення усіх послуг, впорядкованих по полю “id”;
- SELECT r_email FROM regustration_form WHERE r_email =""\$_REQUEST['rEmail']."" – виконується виведення емейла користувача для перевірки унікальності значення;
- SELECT r_email, r_password FROM regustration_form WHERE r_email = "".\$rEmail."" AND r_password = "".\$rPassword."" limit 1 – запит виконує виведення

емейла та пароля користувача, щоб перевірити чи даний користувач зареєстрований.

- "SELECT email, password FROM login_admin WHERE email = ".\$aEmail." AND password = ".\$aPassword." limit 1" – запит перевіряє правильність емейла та пароля адміністратора.
- SELECT * FROM tovaru – даний запит виводить з таблиці “tovaru” всі замовлення, які були зроблені користувачами.

Приклад використання запиту SELECT наведено в лістингу 2.1.

Лістинг 2.1

Програмний код для демонстрації використання запиту SELECT

```
if(!empty($_POST["quantity"])) {
$productByCode = $db_handle->runQuery("SELECT * FROM prod_serv WHERE code='".$_
$_GET["code"] . "'");
$itemArray=array($productByCode[0]["code"]=>array('name'=>$productByCode[0]["name"],
'code'=>$productByCode[0]["code"], 'period'=>$productByCode[0]["period"],
'quantity'=>$_POST["quantity"],'user'=>$_SESSION['rEmail']
,'price'=>$productByCode[0]["price"],'image'=>$productByCode[0]["image"]));
```

2. Запит типу «INSERT»:

- INSERT INTO tovaru (code, name, price, user) VALUES ('\$code', '\$name', '\$price', '\$user') – виконується запис даних замовлення з кошика в таблицю;
- INSERT INTO registration_form(r_name, r_email, r_password) VALUES('\$rName','\$rEmail','\$rPassword') – запит виконує запис даних користувача при реєстрації.

Приклад використання запиту INSERT наведено в лістингу 2.2.

Лістинг 2.2

Програмний код для демонстрації використання запиту SELECT

```
if(isset($_REQUEST['submit']))
{
$user= $_POST['user'];
$name= $_POST['name'];
```

Продовження лістингу 2.2

```
$price= $_POST['price'];  
$code= $_POST['code'];  
$sql = "INSERT INTO tovaru ( code, name, price, user) VALUES ( '$code', '$name', '$price', '$user' )";  
if($conn->query($sql)==TRUE){  
}else{ echo 'Order was made successfully';  
}  
}  
else{  
echo 'query doesnt execute';  
}
```

РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ

3.1. Інструменти для розробки веб-сайту

Для розробки сайту використано наступні технології:

- MySQL - вільна реляційна система керування базами даних (СКБД), яка працює як сервер для забезпечення багатокористувацького доступу до великої кількості баз даних.
- PHP (англ. PHP: Hypertext Preprocessor - «PHP: препроцесор гіпертексту») - скриптова мова програмування, застосовується для розробки додатків і сайтів. Дана мова підтримується сервером ХАМРР, на якому буде розгорнуто web-орієнтована програма.
- HTML (Hypertext Markup Language) — це мова опису структури сторінок документів, яка дозволяє звичайний текст формувати в абзаци, заголовки, списки та інші структури, створювати посилання на інші сторінки. Це текстова мова, в якій інструкції з форматування, що називаються тегами, вбудовані в розділи документа, які містять конкретну інформацію. Теги повідомляють браузерам, як формувати і представляти інформацію на екрані.
- JavaScript – це мова програмування, що дозволяє зробити Web -сторінку інтерактивною, тобто такою що реагує на дії користувача.

Отже, для реалізації даного проекту здійснена за допомогою скрипкової мови програмування PHP та СУБД MySQL.

Для адміністрування СКБД MySQL буде використано phpMyAdmin – веб-застосунок з відкритим кодом, написаний на мові PHP, що представляє собою веб-інтерфейс. PhpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL та переглядати вміст таблиць і баз даних. Система користується великою популярністю у веб-

розробників, оскільки дозволяє керувати СКБД MySQL без безпосереднього вводу SQL команд, надаючи зручний інтерфейс.

Розгортання програмного продукту буде відбуватися за допомогою локального сервера XAMP, який являє собою найпопулярнішим середовищем розробки PHP. XAMPP повністю безкоштовний і простий в установці дистрибутив Apache, що містить MariaDB, PHP і Perl. XAMPP створений з відкритим вихідним кодом, що дозволяє йому бути простим в установці та у використанні.

3.2 Структура веб-сайту

Дана програма реалізована у вигляді web-сайту, який працює через будь-який доступний браузер. Доступ до сайту відбувається через адресу: <http://localhost/index.html>.

Готовий програмний продукт має такі сторінки:

- Home – дозволяє повернутись на головну сторінку сайту з будь-якої іншої сторінки. Містить кредо компанії, та можливість перейти на вкладку About us за допомогою кнопки “Explore”.
- About us – містить інформацію про компанію: історії, пріоритети та команду.
- Contact – розміщено форму зворотнього зв'язку.
- Sign up – дозволяє користувачу зареєструватися на сайті для подальшої роботи з ним.
- Log in – забезпечує вхід зареєстрованим користувачам у їхній профіль.
- Profail – сторінка, яка містить головний функціонал програми, забезпечує перегляд послуг та їх покупку.
- Log out – за допомогою цієї сторінки користувач завершує роботу з покупкою та переходить на сторінку початкової сторінку Home.
- Admin – дозволяє адміністратору переглядати замовлення користувачів.

3.3 Макети сторінок веб-сайту

Сторінки веб-сайту мають спільне меню, яке розміщене в верху сторінки, основними елемента якого є логотип, кредо, та безпосередньо перелік сторінок.

Вигляд меню показано на рисунку 3.1.

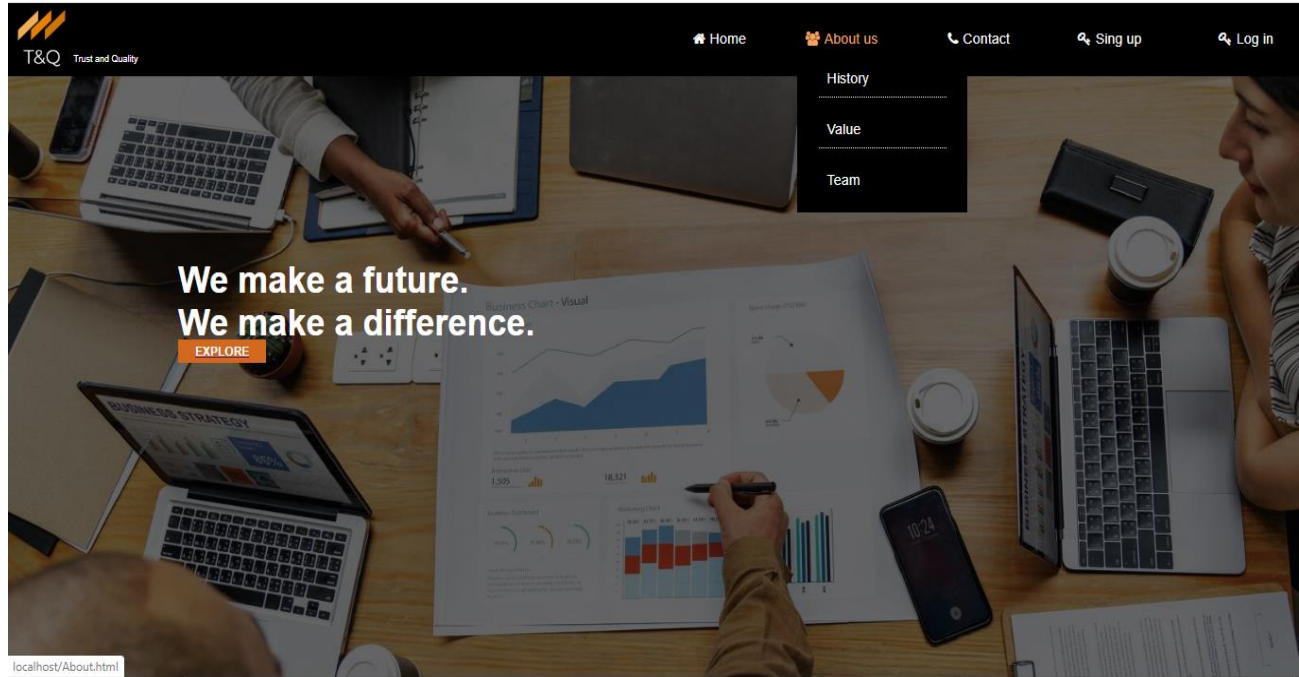


Рисунок 3.1 Вигляд меню

Сторінка About us, як показано на Рисунок 3.2 складається з 3 блоків, перехід між якими можна здійснювати за допомогою вибору підменю або прокручуванням мишки.

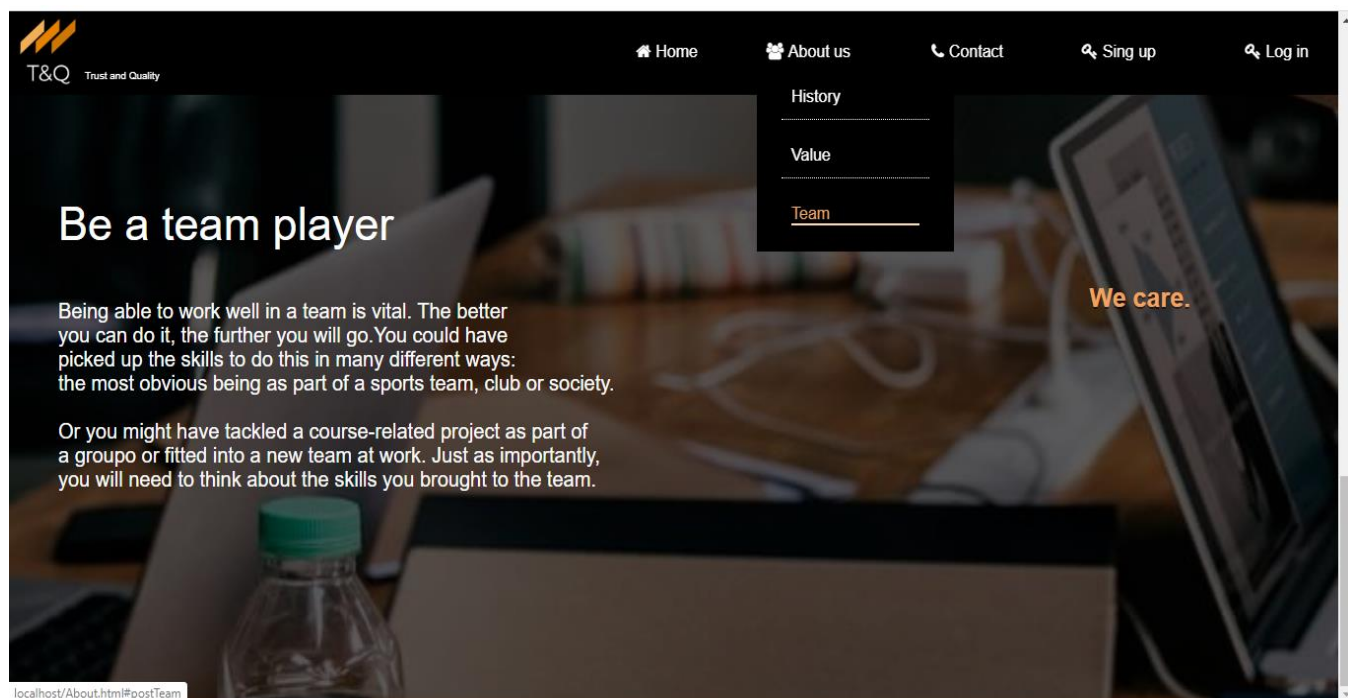


Рисунок 3.2 Вигляд сторінки “About us”

Сторінка Contact містить форму зворотнього зв'язку для відправлення листа, яка розташована по середині сторінки. Сторінки Sign up та Log in створені теж за даним макетом.

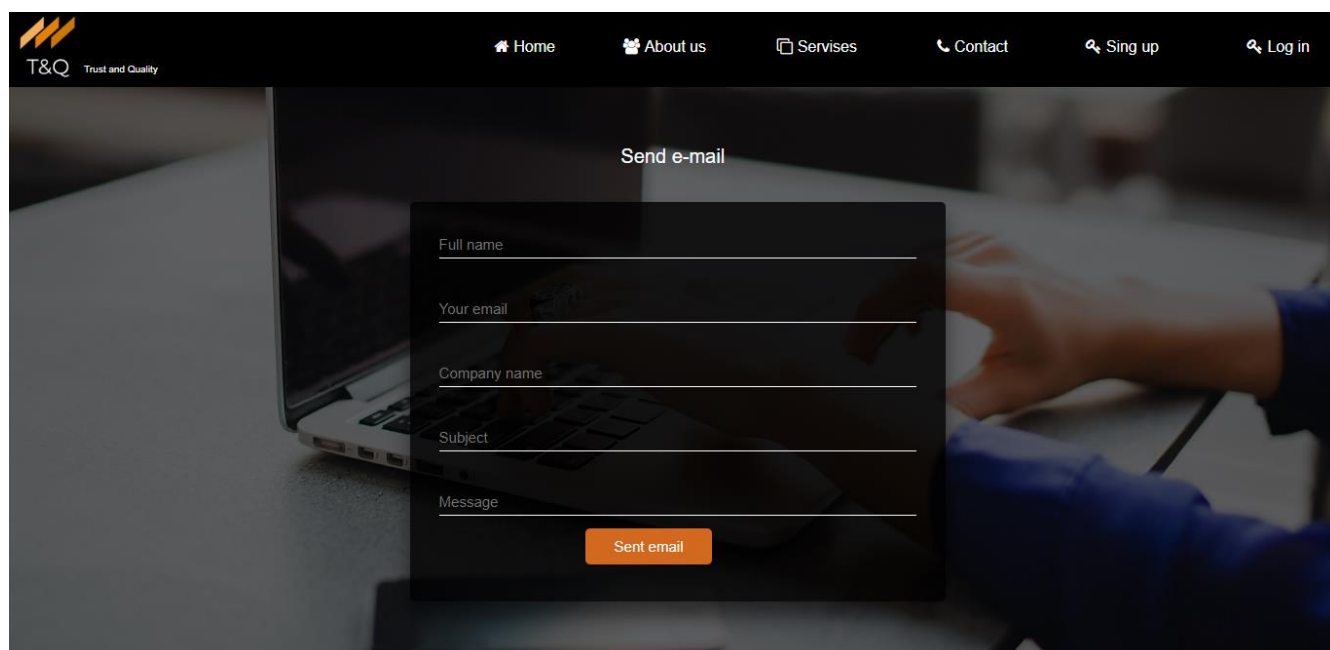


Рисунок 3.3 Вигляд сторінки “Contact”

Сторінку Profail умовно можна поділити на 2 частини: 1) відображення вмісту кошика; 2) перелік доступних послуг для замовлення.

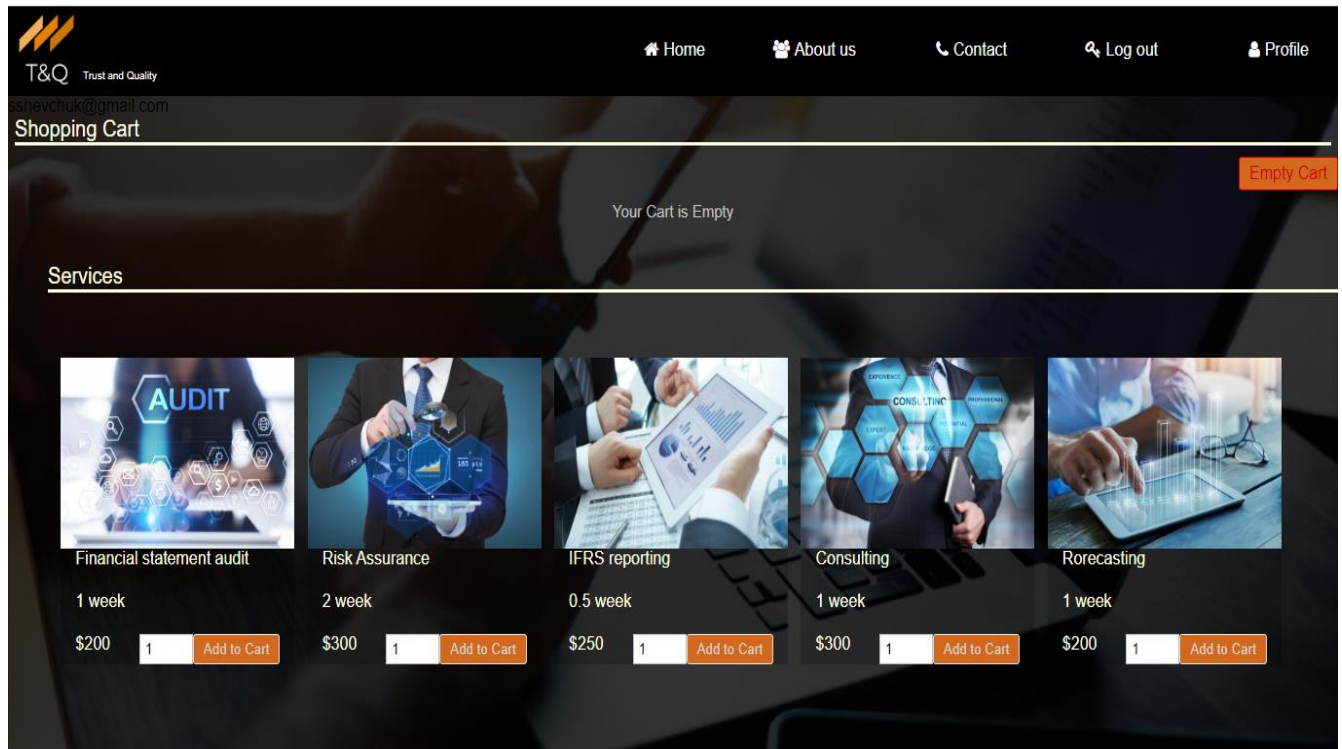


Рисунок 3.4 Вигляд сторінки "Profile"

3.3 Програмування серверної частини

Для програмування серверної частини використано мову програмування PHP.

PHP - скриптова мова програмування, застосовується для розробки додатків і сайтів. В даний час він є одним з лідерів серед мов програмування, які застосовуються для написання динамічних веб-сторінок.

За допомогою серверні скриптів організовано такі операції для функціонування сайту:

1. Зберігання даних користувача: ім'я, емейл, пароль;
2. Підсумування ціни та одиниць замовлення;
3. Редагування вмісту кошика, тобто самого замовлення.

Приклад використання PHP для додання товару наведено в лістингу 3.1:

Лістинг 3.1

Програмний код для демонстрації використання PHP

```
<form method="post" action="UserProfail.php"> <?php
foreach ($_SESSION["cart_item"] as $item){
$item_price = $item["quantity"]*$item["price"];?>
<tr>
<td style="color: #FFFFFFE0;"><?php echo $item["name"]; ?>
<input type="hidden" name="name" value="<?php echo $item['name'];?>"></td>
<td style="color: #FFFFFFE0;"><?php echo $_SESSION['rEmail']; ?>
<input type="hidden" name="user" value="<?php echo $item['user'];?>"></td>
<td style="color: #FFFFFFE0;"><?php echo $item["code"]; ?><input type="hidden" name="code"
value="<?php echo $item['code'];?>"></td>
<td style="color: #FFFFFFE0;"><?php echo $item["period"]; ?></td>
<td style="text-align:right;color: #FFFFFFE0;"><?php echo $item["quantity"]; ?></td>
<td style="text-align:right;color: #FFFFFFE0;"><?php echo "$ ".$item["price"]; ?>
<input type="hidden" name="price" value="<?php echo $item['price'];?>"></td>
<td style="text-align:right;color: #FFFFFFE0;"><?php echo "$ ". number_format($item_price,2);
?></td>
<td style="text-align:center;color: #FFFFFFE0;"><a
href="UserProfail.php?action=remove&code=<?php echo $item["code"]; ?>"
class="btnRemoveAction"></a></td>
</tr>
<?php
$total_quantity += $item["quantity"];
$total_price += ($item["price"]*$item["quantity"]);?>
```

3.4 Програмування клієнтської частини

Для розробки клієнтської частини сайту (frontend), використано стандарти розмітки сторінок HTML, стилі CSS та JavaScript.

Вигляд сторінок подано у додатках. Опис наведено нижче:

1. Web-сторінка «<http://localhost/index.html>» є головною сторінкою, де користувач може за допомогою меню перейти на будь-яку іншу вкладку (Додаток А).

2. Web-сторінка «<http://localhost/About.html>» дозволяє користувачу ознайомитись з історією, цінностями та командою компанії (Додаток Б).
3. Web-сторінка «<http://localhost/contact.php>» пропонує заповнити контактну форму, якщо в користувача виникли запитання (Додаток В).
4. Web-сторінка «<http://localhost/signup.php>» дозволяє користувачу зареєструватися на сайті (Додаток Г).
5. Web-сторінка «<http://localhost/login.php>» дозволяє користувачу здійснити авторизацію (Додаток Д).
6. Web-сторінка «<http://localhost/UserProfile.php>» є сторінка користувача, де він може переглянути послуги та здійснити замовлення (Додаток Е).
7. Web-сторінка «<http://localhost/admin.php>» знаходиться в адміністраторській частині сайту, дозволяє адміністратору здійснити авторизацію (Додаток Є).
8. Web-сторінка «http://localhost/admin_profile.php» знаходиться в адміністраторській частині сайту, за допомогою цієї сторінки адміністратор здійснює перегляд замовлення клієнтами (Додаток Ж).

ВИСНОВКИ

У результаті опрацювання Розділу 1 було розроблено бізнес-ідею сайту, а саме: надання можливості замовлення послуг аудиторської компанії користувачем; та поставленні основні технічні завдання для її досягнення.

При розробці моделей варіантів використання веб-сайту було визначено два користувача сайту:

- 1) Зовнішній – користувач, метою якого є здійснити замовлення.
- 2) Внутрішній – адміністратор, який має можливість переглядати замовлення клієнта.

У Розділі 2 описано навчальну базу даних, її таблиці та записи. Визначено типи даних, функціональні залежності та нормалізаційні відношення. А також розроблено типи запитів та безпосередньо самі запити для реалізації технічних завдань, які представлені у Розділі 1.

В ході виконання практичного завдання було створено новий програмний продукт – інформаційна система для аудиторської компанії. На початковому етапі був здійснений аналіз предметної області та визначено основні інструменти для втілення бізнес-ідеї. Далі створено web-сторінки, використовуючи мови програмування PHP та Java-script. Для реалізації бази даних використано систему управління базами даних MySQL. Здійснено опис серверної та клієнтської частин сайту.

Слід зазначити, що web-сайт завжди потребує допрацювання та оновлення.

В подальшому планується:

- додати сторінкам динамічності;
- надання адміністратору можливості керувати замовленнями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

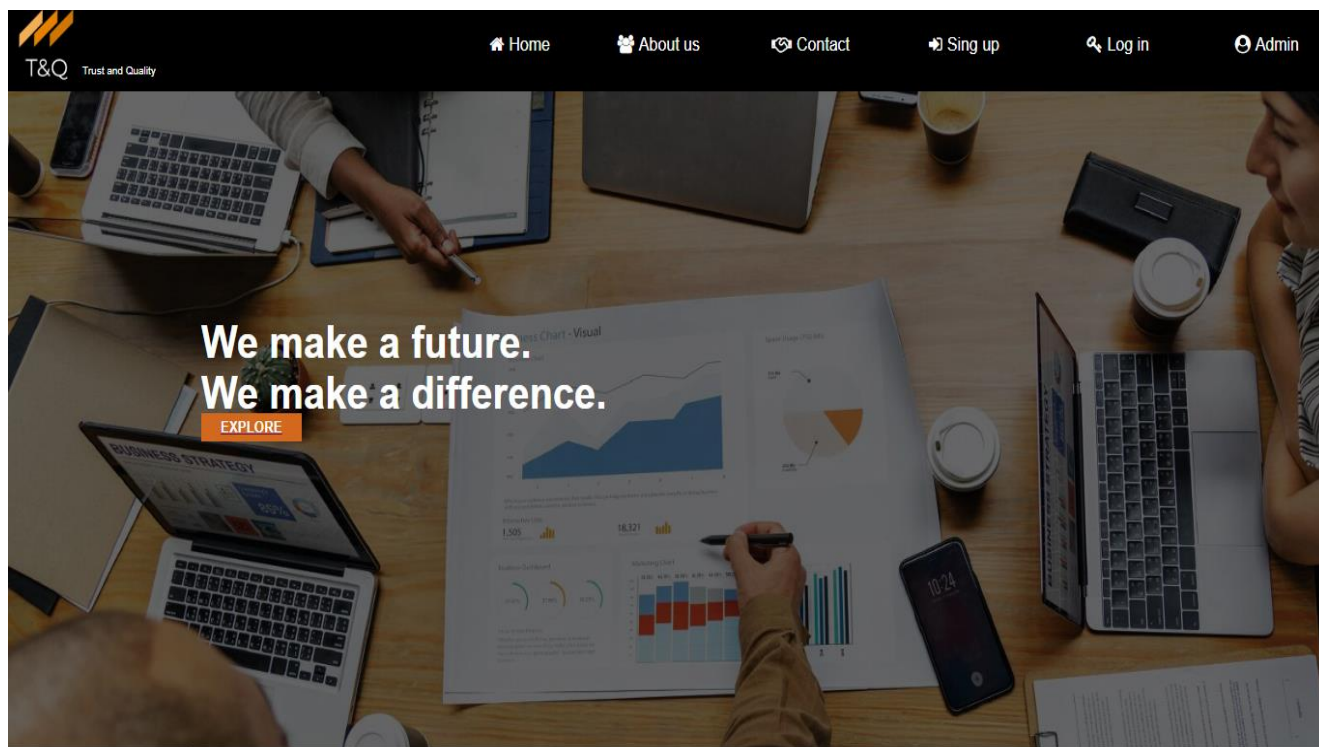
1. Проектування БД [Електронний ресурс]. – Режим доступу: https://rdb.dp.ua/uk/chapter_02
2. Реляційна модель бази даних. Історія розвитку реляційної моделі. Структура реляційних даних. Відношення в базі та їх властивості. Типи даних. [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5454386/page:7/>
3. Етапи проектування баз даних [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5194057/page:2/>
4. . Проблеми проектування бд. Метод нормальних форм [Електронний ресурс]. – Режим доступу: http://dn.khnu.km.ua/dn/k_default.aspx?M=k0537&T=04&lng=1&st=0
5. Клименко А. Профессии в сфере использования информационных технологий // Менеджер по персоналу. – 2016. – № 11. – С. 46 -56
6. Ковалева Т. Ю. Статистическое исследование динамики и структурных изменений на рынке труда [Электронный ресурс] / Т. Ю. Ковалева // Проблемы современной экономики. – 2011. – № 2 (38). – Режим доступу: <http://www.m-economy.ru/art.php?nArtId=3577>.
7. Коннолли Т.: Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Коннолли Т., Бегг К. – [3-е издание.: Пер. с англ.] – М.: Издательский дом «Вильямс», 2003. – 1440 с.
8. 4. Питер Лабберс, Брайан Олберс, Френк Салим: HTML5 для профессионалов. Потужні інструменти для розробки сучасних веб-додатків – СПб.: Издательский дом «Питер», 2011. – 272 с.
9. Пасічник В. В. Організація баз даних та знань / В.В. Пасічник, В.А. Резніченко. – К.: Видавнича група ВНУ, 2006. – 384 с.

- 10.Романюк О.Н. Організація баз даних та знань / О.Н. Романюк, Т.О. Савчук. – Вінниця: ВДТУ, 2001.
- 11.Ситник Н.В. Проектування баз та сховищ даних . – К.: КНЕУ, 2004.
- 12.Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – СПб.: Питер, 2014. – 304 с.
- 13.Node.js v4.5.0 Documentation [Електронний ресурс]. – Режим доступу: <https://nodejs.org/dist/latest-v4.x/docs/api/>
- 14.Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. — Х.: ХНАМГ, 2010. — 222 с.
- 15.Методи та засоби мультимедійних інформаційних систем: навч. посіб. / Т. М. Басюк, П. І. Жежнич; Нац. ун-т «Львів. політехніка». — Львів: Вид-во Львів. політехніки, 2015. — 426 с. — Бібліогр.: с. 413—416.
- 16.Нанда А. и др. Oracle PL/SQL для администраторов баз данных; Символ, 2008. – 496 с.
17. Наумов А. Н., Вендров А. М. і ін., "Системи керування базами даних і знань", М.:Фінанси і статистика, 1991 р. 28. Неня А. В. Організація баз даних та знань: конспект лекцій для студентів заочної форми навчання. – Суми:Вид-во СумДУ, 2010. – 109 с.
18. Шнайдер Р. Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных, – М.: Лори, 2010. – 361 с. 37. Яргер Р.Дж., Риз Дж., Кинг Т. MySQL и mSQL: Базы данных для небольших предприятий и Интернета; СПб: СимволПлюс, 2013. – 560 с.
19. PHP [Електронний ресурс]. - <https://en.wikipedia.org/wiki/PHP>.
20. Нормалізація баз даних [Електронний ресурс]. - <https://uk.wikipedia.org/wiki>.
21. Реляційна база даних [Електронний ресурс]. - <https://uk.wikipedia.org/wiki>.

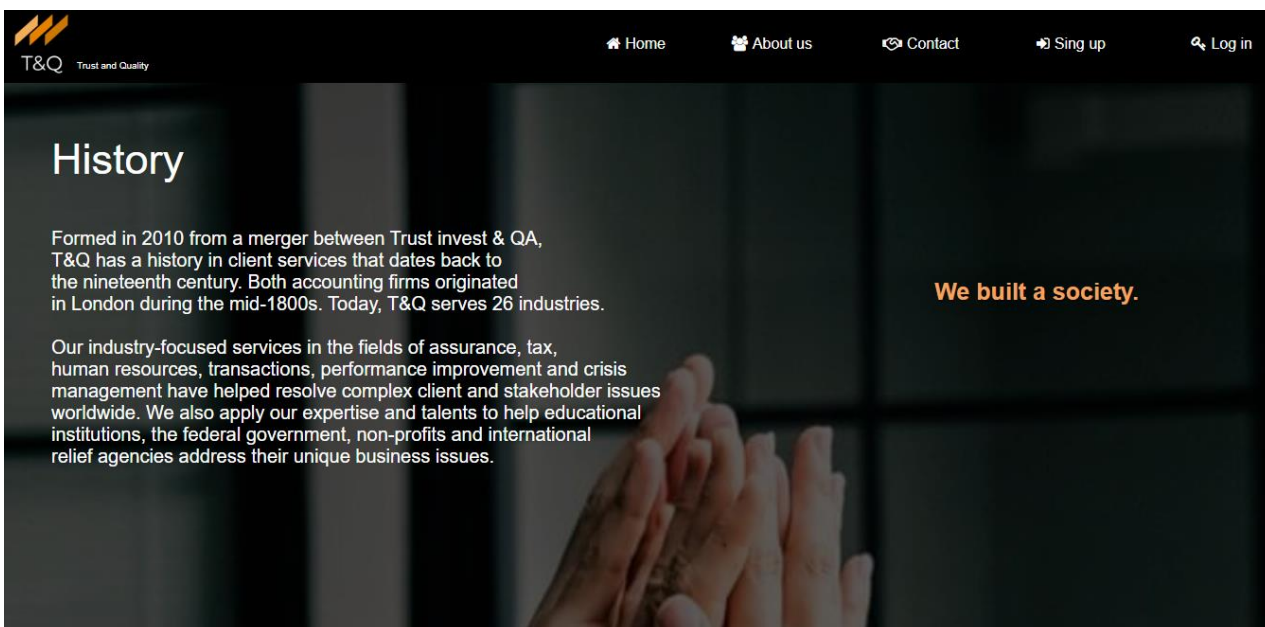
ДОДАТКИ

Додаток А

Стартова сторінка



Сторінка з інформацією про компанію



T&Q Trust and Quality

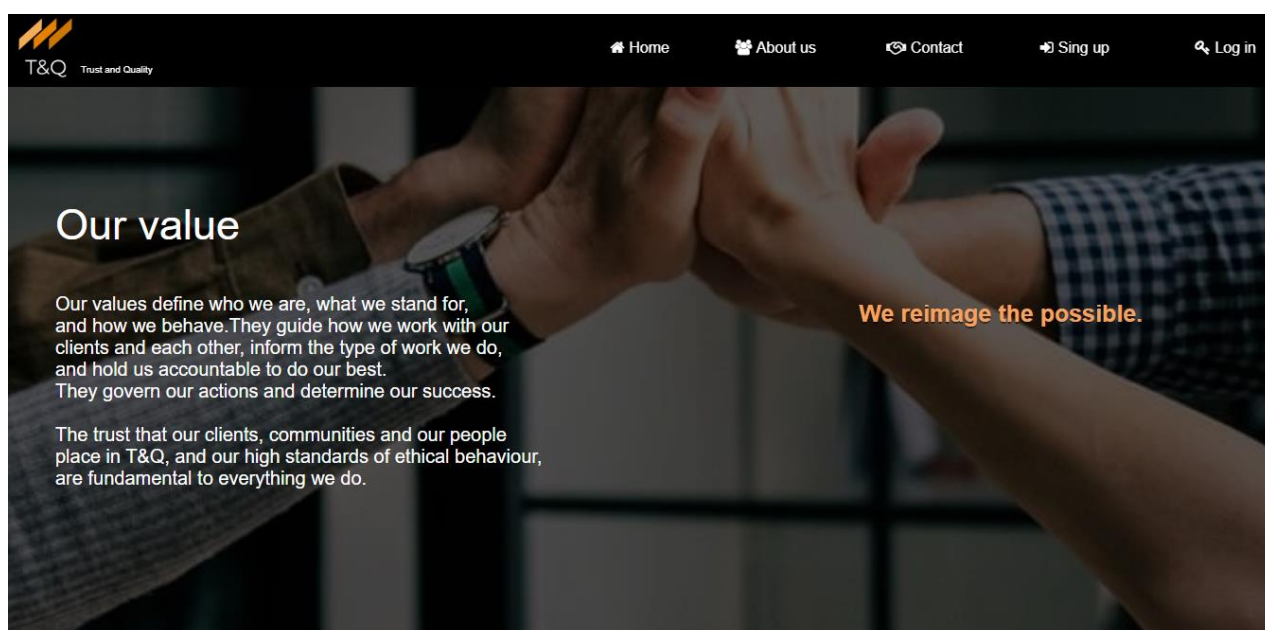
Home About us Contact Sing up Log in

History

Formed in 2010 from a merger between Trust invest & QA, T&Q has a history in client services that dates back to the nineteenth century. Both accounting firms originated in London during the mid-1800s. Today, T&Q serves 26 industries.

Our industry-focused services in the fields of assurance, tax, human resources, transactions, performance improvement and crisis management have helped resolve complex client and stakeholder issues worldwide. We also apply our expertise and talents to help educational institutions, the federal government, non-profits and international relief agencies address their unique business issues.

We built a society.



T&Q Trust and Quality

Home About us Contact Sing up Log in

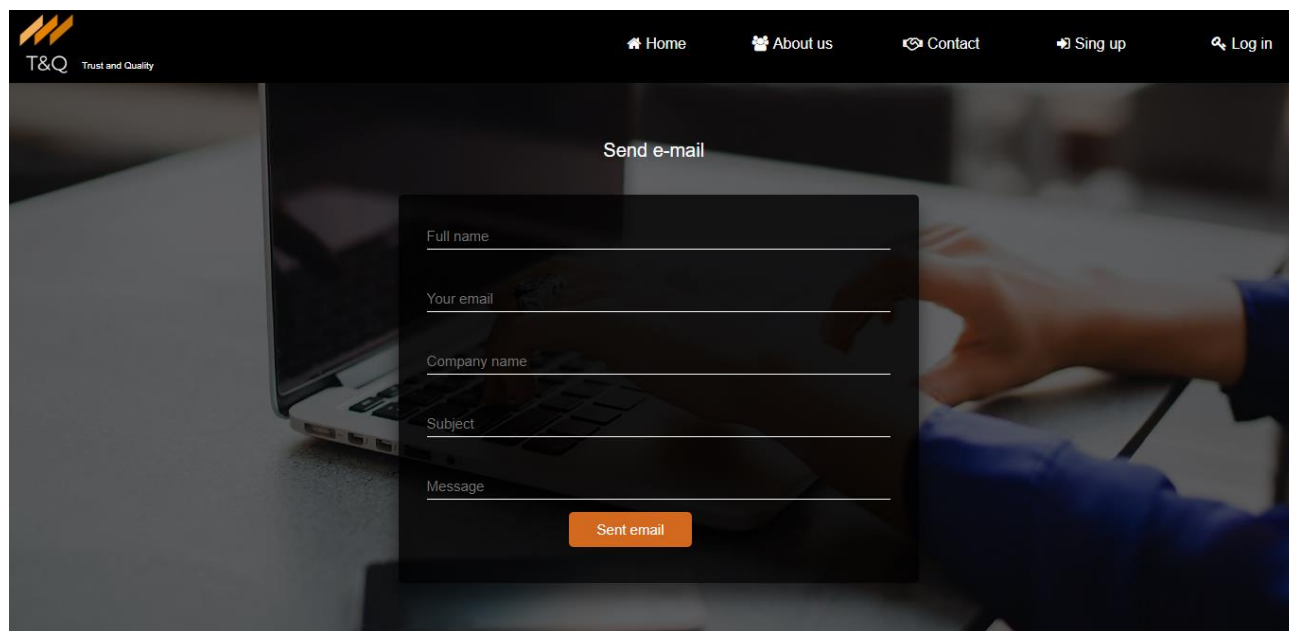
Our value

Our values define who we are, what we stand for, and how we behave. They guide how we work with our clients and each other, inform the type of work we do, and hold us accountable to do our best. They govern our actions and determine our success.

The trust that our clients, communities and our people place in T&Q, and our high standards of ethical behaviour, are fundamental to everything we do.

We reimage the possible.

Контактна форма



T&Q Trust and Quality

Home About us Contact Sing up Log in

Send e-mail

Full name

Your email

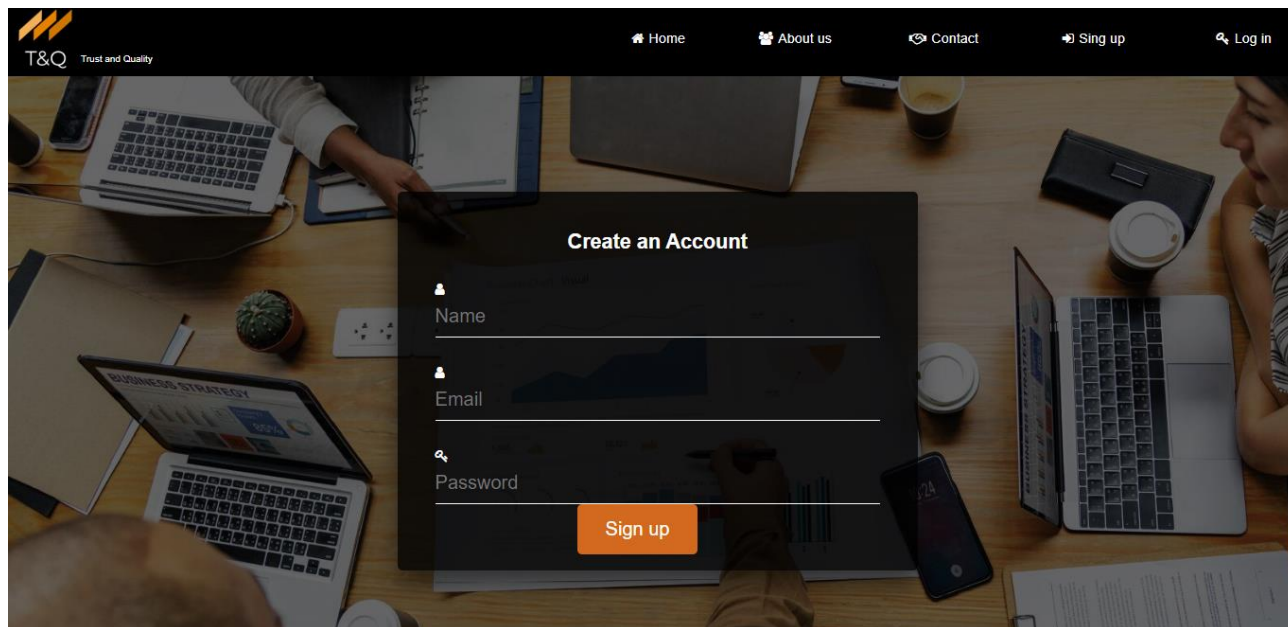
Company name

Subject

Message

Sent email

Створення акаунту користувача



The image shows a web application interface for creating a user account. The background is a blurred photograph of a modern office desk with two laptops, a smartphone, a coffee cup, and people working. Overlaid on this is a dark, semi-transparent modal box titled "Create an Account". Inside the modal, there are three input fields, each preceded by a small icon: a person icon for "Name", an envelope icon for "Email", and a key icon for "Password". Below these fields is an orange button labeled "Sign up". At the top left of the page, there is a logo with three orange diagonal lines above the text "T&Q Trust and Quality". At the top right, there is a navigation bar with links: "Home", "About us", "Contact", "Sing up", and "Log in".

T&Q Trust and Quality

Home About us Contact Sing up Log in

Create an Account

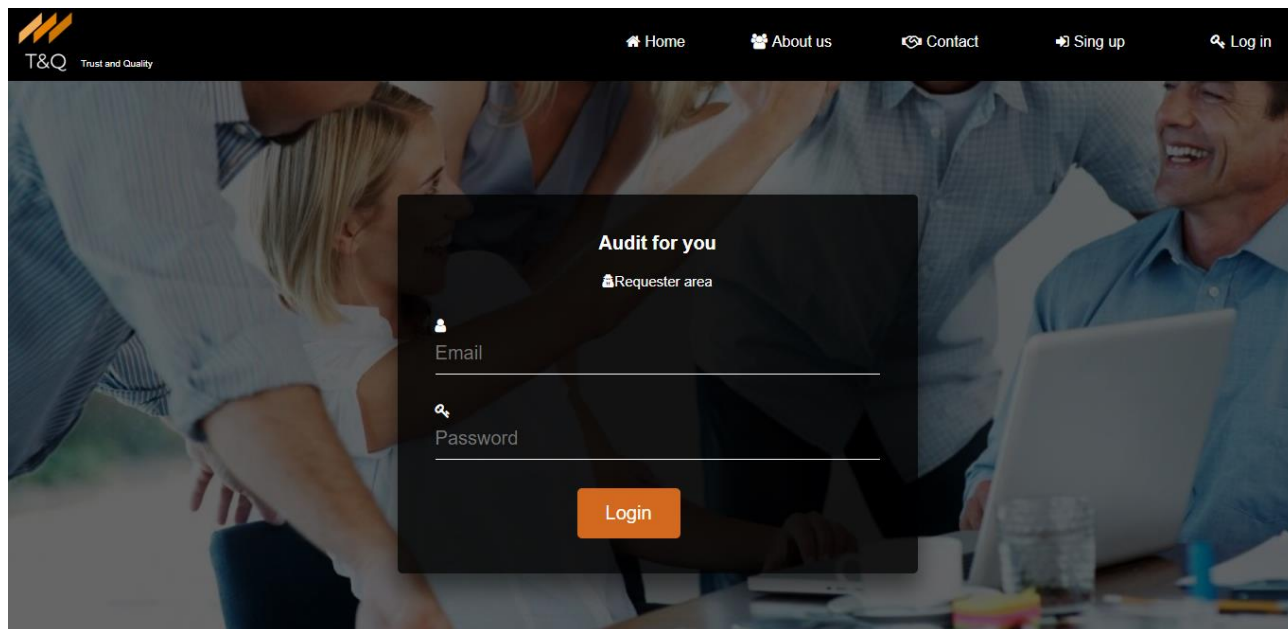
Name

Email

Password

Sign up

Авторизація користувача



The screenshot shows a web application interface for 'Audit for you'. The background is a blurred image of a group of people in a meeting. A dark overlay contains the login form. At the top left is the 'T&Q Trust and Quality' logo. The top navigation bar includes links for Home, About us, Contact, Sing up, and Log in. The login form is titled 'Audit for you' and includes a 'Requester area' label. It features input fields for 'Email' and 'Password', each with an icon (person and key respectively). An orange 'Login' button is positioned below the password field.

T&Q Trust and Quality

Home About us Contact Sing up Log in

Audit for you


Requester area

Email

Password

Login

Сторінка профілю






**T&Q** Trust and Quality
sshevhchuk@gmail.com
Shopping Cart

HomeAbout usContactLog outProfile

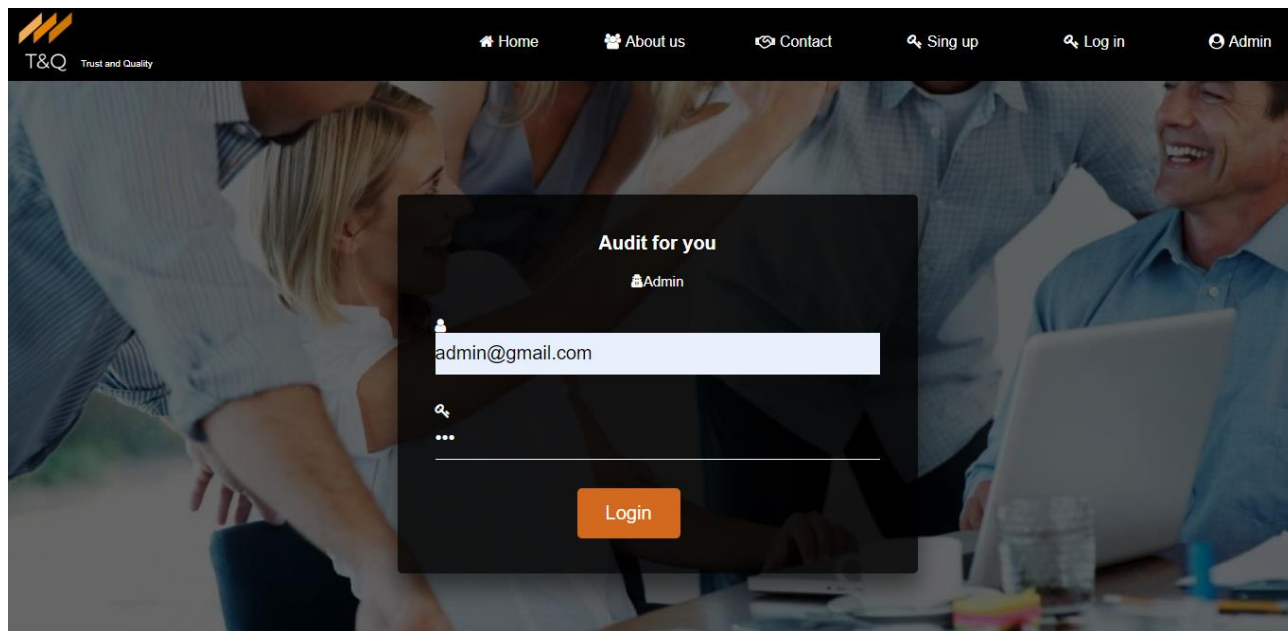
Empty Cart

Your Cart is Empty

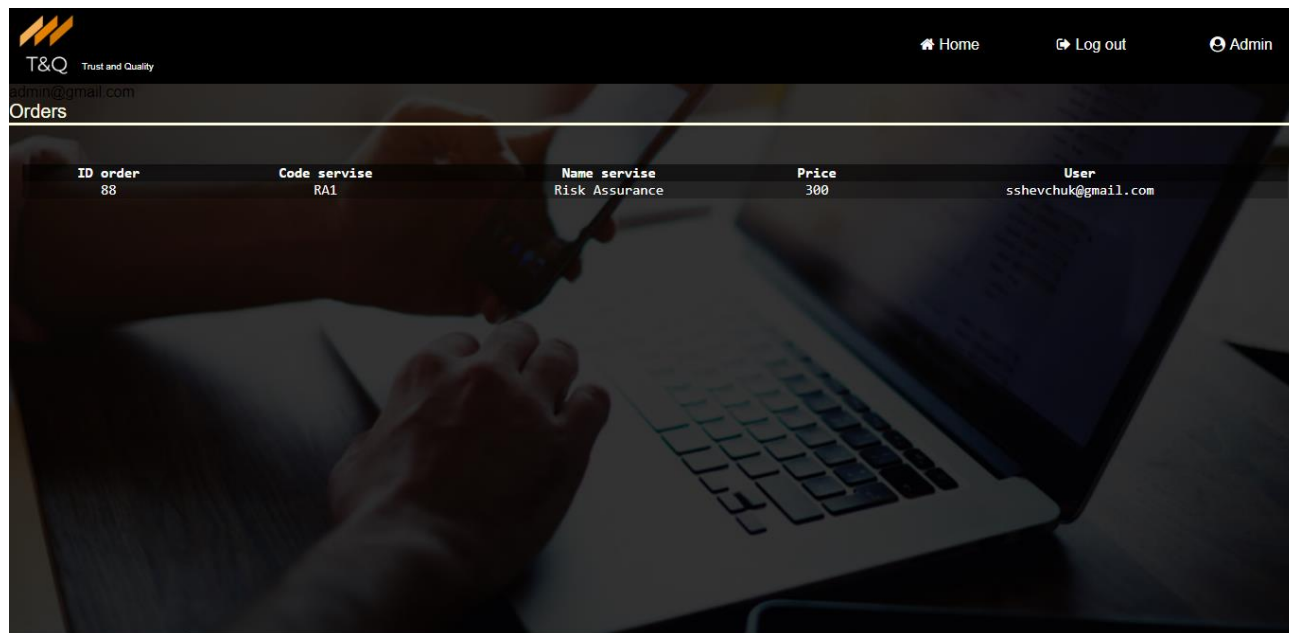
Services

				
Financial statement audit	Risk Assurance	IFRS reporting	Consulting	Rorecasting
1 week	2 week	0.5 week	1 week	1 week
\$2001Add to Cart	\$3001Add to Cart	\$2501Add to Cart	\$3001Add to Cart	\$2001Add to Cart

Авторизація адміністратора



Сторінка адміністратора



Використання Java Script для динаміки сторінки “About us”

```
$(function() {  
  $(".sub-menu-1 a").on('click', function(){  
    $("html, body").animate({  
      scrollTop: $($.attr(this, 'href')).offset().scrollTop  
    }, 500);  
  });  
});  
$(window).on("scroll", function(){  
  if($(window).scrollTop()){  
    $('menu-bar').addClass('black');  
  }  
  else{  
    $('menu-bar').removeClass('black');  
  }  
});  
$document
```