

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.е.н., доц. каф. Старух А.І.
(науковий ступінь, посада, прізвище, ініціали)
_____ “___” травня 2021 р.
(підпис)

Виконавець:

студент(ка) групи УФЕ-31 с_
Артамонова О.Д.
_____ “___” травня 2021 р.
(підпис)

Загальна кількість балів _____

(підпис, ППП членів комісії)

ЗМІСТ

ЗМІСТ	2
ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ	6
1.1 Постановка завдання	6
1.2 Розробка моделі варіантів використання веб-сайту	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ	9
2.1 Опис моделі даних	9
2.2 Нормалізація відношень	12
2.3 Визначення типів даних	14
2.4 Обмеження цілісності даних	16
2.5 Реалізація SQL-скрипту	17
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ	20
3.1 Розміщення веб-сайту на локальному віртуальному середовищі або в Інтернеті.....	20
3.2 Структура та макет веб-сайту.....	21
3.3 Програмування клієнтської та серверної частин	22
ВИСНОВКИ	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	29
ДОДАТКИ	31

ВСТУП

Актуальність теми дослідження. За роки існування аудиту в Україні цей термін став елементом ринкового механізму, що працює в інтересах користувачів фінансової звітності.

Під аудиторською перевіркою багато хто в Україні розуміє перевірку компанії з погляду правильності нарахування та сплати податків. Це обумовлено тим, що основний запит до бухгалтера в нашій країні – правильно вести податкові справи. Поряд з аудитом аудиторські компанії надають і неаудиторські послуги, такі як: бухгалтерське обслуговування, податкове консультування, відновлення обліку, підготовка документації з трансфертного ціноутворення, упровадження бюджетування й управлінського обліку.

Все частіше суб'єкти господарювання будь-якої форми власності залучають аудиторські компанії для проведення перевірок і надання консультаційних послуг. Це зумовлено перш за все власним бажанням, оскільки наявність висновків незалежного аудитора свідчать про стабільність фірми, як надійного партнера, а також надають перевагу при залученні інвесторів. Часто наявність аудиторського звіту є вимогою законодавства. Отже, зовнішній аудитор у відносинах із підприємствами посідає почесне місце фінансового помічника, який сприяє правильним рішенням в управлінні підприємством. Тож попит на їх послуги росте з кожним роком все більше і більше.

Проте слід розуміти, що сфера аудиту це теж бізнес зі своїм ринковим середовищем та конкурентними умовами. Тому популярність компанії визначає її місце серед інших представників даної ніші.

У стандартних умовах комунікація між клієнтом і аудитором відбувається шляхом відвідування офісу, щоб клієнт домовився про консультацію, а це завжди займає багато часу. Тому розробка веб-сайту, який надасть можливість користувачу подати заявку на надання необхідної йому послуги в аудиторській фірмі є досить актуальним завданням. Такий програмний продукт, який буде містити в собі різноманітні функції, набагато пришвидшить та спростить процес

роботи з клієнтом.

Інтернет значно змінив просторовий і часовий масштаби ведення комерції. Він став глобальним засобом комунікації, що не має територіальних обмежень, при цьому вартість доступу до інформації не залежить від віддаленості від неї на противагу традиційним засобам, де ця залежність прямо пропорційна, вартість комунікацій, порівняно з традиційними засобами, стає мінімальною, а їх функціональність і масштабованість значно зростають.

У результаті Інтернет дозволяє перейти від масового маркетингу до маркетингу «один-одному», до спеціальних маркетингових пропозицій окремому споживачу, спеціального виробництва, індивідуальних звернень. Зниження трансформаційних витрат може досягатися за рахунок оптимального вибору структури асортименту інформаційного продукту, скорочення часу на розробку та впровадження нового інформаційного продукту, обґрунтованої політики ціноутворення, зниження числа посередників, витрат на маркетингову стратегію просування та продажу інформаційного продукту і т.д.

Мета і завдання дослідження. *Метою курсової роботи є розробка веб-сайту, на якому можна подати заявку на отримання послуг від спеціаліста в аудиторську компанію та дослідити цю інформаційну систему аудиторської компанії.*

Виходячи з цього, для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні *завдання*:

- охарактеризувати інформаційну систему для аудиторської компанії;
- проаналізувати предметну область;
- розробити архітектуру та спроектувати структуру бази даних;
- розробити інформаційну систему(веб-сайт), що дає змогу переглянути послуги аудиторської компанії та подати онлайн заявку на консультацію.

Об'єкт дослідження є особливість розробки інформаційної системи для аудиторської компанії за допомогою системи управління базами даних MySQL.

Предмет досліджень є теоретичні, методичні та практичні аспекти

розробки програмного забезпечення за допомогою різних мовою проектування.

Використане програмне забезпечення. Для досягнення результату використовувалось середовище розробки Visual Studio Code, локальний сервер — ХАМРР та мови програмування: HTML, CSS, JS і PHP.

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 – 38 сторінок.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

Інформаційно-аналітичні системи, призначені для забезпечення інформаційної підтримки, електронного обміну даними й документами, а також для надання учасникам ринку маркетингової інформації і аналітичних матеріалів для прийняття рішень на корпоративному, а також мікро- і макроекономічних рівнях.

Необхідна мінімальна інформаційна підтримка зазвичай охоплює онлайн доступ до постійно актуальної ринкової інформації на основі сучасних інформаційно-комунікаційних технологій, включаючи використання можливостей мережі Інтернет і цифрової телефонної мережі. Інформаційно-аналітичні системи забезпечують єдиний інформаційний простір для внутрішньогалузевого й міжгалузевого міжкорпоративного електронного документообігу, електронного обміну даними й створюють інтерфейс для автоматизації ділових операцій.

Сайт аудиторської компанії призначений в першу чергу для оприлюднення інформації про стан та спектр діяльності компанії. В даному ракурсі сторінка покриває потребу у нагальному інформуванні клієнтської аудиторії, оскільки вона знаходиться в публічному доступі цілодобово.

З іншої сторони функціональне призначення полягає в забезпеченні зв'язку «клієнт - компанія». Будь-який відвідувач сайту має змогу стати безпосереднім клієнтом фірми. В його розпорядженні є форма одностороннього зв'язку — запит. Заповнивши всі обов'язкові поля які вказані в заявці, дані потрапляють до керівництва, яке в найкоротші терміни відреагує на звернення.

1.2 Розробка моделі варіантів використання веб-сайту

Веб-сайт зокрема слугує точкою контакту у зв'язку «фірма — клієнт».

Наявність інформаційної системи аудиторської компанії має низку безумовних переваг:

- 1. Детальна інформація про вашу компанію.** Веб-сайт – це місце, де можна опублікувати детальну інформацію про компанію, послуги, умови замовлення та виконання послуг, контактах і реквізити.
- 2. Залучення клієнтів.** Веб-сайт допомагає залучити потенційних клієнтів і спростити початковий процес замовлення ваших послуг.
- 3. Рівень обслуговування.** Веб-сайт виконує роль платформи для збирання питань, відгуків і пропозицій, які суттєво допомагають покращити роботу компанії.
- 4. Ваш імідж.** Веб-сайт працює на ваш імідж. Наявність веб-сайту свідчить про інноваційність компанії та відповідність вимогам часу.

На даному етапі розробки система має один елемент користування. Будь який користувач мережі Інтернет може відвідати веб-сайт компанії. Він має можливість ознайомитись з інформацією про фірму, працівників та спектр послуг, що надає фірма. Також потенційний клієнт може залишити заявку на консультацію через форму зворотного зв'язку.

Для того, щоб більш точно зрозуміти як повинна працювати система, все частіше використовується опис функціональності системи через варіанти використання (Use Case). Варіанти використання це – опис послідовності дій, які може здійснювати система у відповідь на зовнішні впливи користувачів або інших програмних систем. Варіанти використання відображають функціональність системи з точки зору отримання відчутного результату для користувача, тому вони точніше дозволяють розподіляти функції за значимістю одержуваного результату.

Варіанти використання призначені в першу чергу для визначення функціональних вимог до системи і керують усім процесом розробки.

Вигляд діаграми варіантів використання для користувачів(клієнт) на рисунку 1.1.

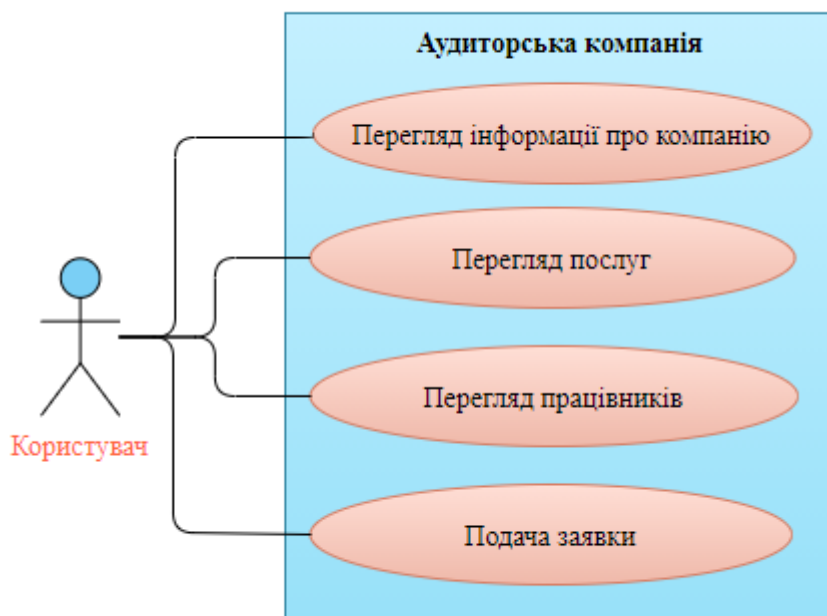


Рис 1.1 Діаграма варіантів використання для користувача

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Опис моделі даних

База даних (БД) - упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД. Система управління базами даних (СУБД) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

Централізований характер управління даними в базі даних передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі.

Головним завданням БД є гарантоване збереження значних обсягів інформації 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).

Набори принципів, що визначають організацію логічної структури зберігання даних у базі, називають моделями даних.

Модель даних – це система правил, згідно з якими створюють структури даних, здійснюють доступ до даних та змінюють їх. Розрізняють такі моделі:

- Ієрархічна 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.
- Мережна 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.

- Об'єктно-орієнтовану модель. База даних, в якій дані моделюються у вигляді об'єктів, їх атрибутів, методів і класів
- 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. 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.

Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

- кожен елемент таблиці — один елемент даних;
- всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символічний тощо);
- кожен стовпчик має унікальне ім'я;
- однакові рядки в таблиці відсутні;
- порядок проходження рядків і стовпчиків може бути довільним.

При розробці сайду аудиторської компанії було розроблено реляційну базу даних. Для візуального відображення структури створено ER – діаграму (рис.2.1)

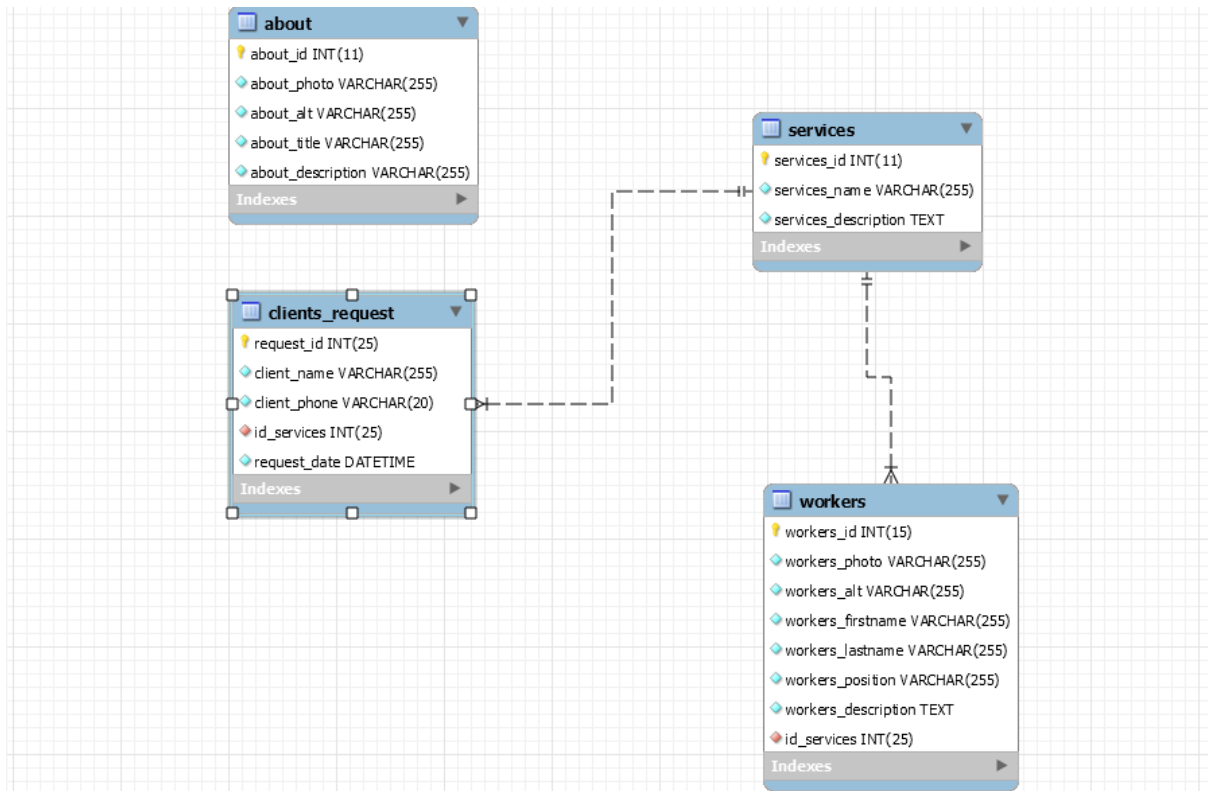


Рис 2.1. Діаграма сутностей та зв'язків

Також на рисунку(Рис.2.1) можна побачити, що дана модель складається з чотирьох таблиць: about, clients_request, services та workers.

About таблиця яка містить інформацію про компанію, полями якої є:

- a) about_id — ідентифікаційний код інформації;
- b) about_photo — назва логотипу;
- a) about_alt — альтернативний текст для логотипу;
- c) about_title — заголовок інформації;
- d) about_description — опис інформації.

Services таблиця яка містить інформацію про послуги компанії, полями якої є:

- a) services_id — ідентифікаційний код послуги;
- b) services_name — назва послуги;
- c) services_description — опис послуги.

Workers таблиця яка містить інформацію про працівників компанії, полями якої є:

- b) workers_id — ідентифікаційний код працівника;

- c) workers_photo — назво фотографії;
- d) workers_alt — альтернативний текст для фотографії;
- e) workers_firstname — ім'я працівника;
- f) workers_lastname — прізвище працівника;
- g) workers_position — посада працівника;
- h) workers_description — опис роду діяльності;
- i) id_services — ідентифікаційний код послуги.

Clients_request таблиця яка містить інформацію про заявку клієнта, полями якої є:

- a) request_id — ідентифікаційний код запиту клієнта;
- b) client_name — ім'я клієнта;
- c) client_phone — номер телефону клієнта;
- d) id_services — ідентифікаційний код послуги;
- e) request_date — дата подання запиту.

2.2 Нормалізація відношень

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

Нормальна форма — властивість відношення в реляційній моделі даних, що характеризує його з точки зору надмірності, яка потенційно може призвести до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення.

Таким чином, схема реляційної бази даних переходить у першу, другу, третю і так далі нормальні форми. Якщо відношення відповідає критеріям нормальної форми n та всіх попередніх нормальних форм, тоді вважається, що це відношення знаходиться у нормальній формі рівня n . У теорії реляційних баз даних прийнято виділяти таку послідовність нормальних форм:

- Перша нормальна форма (1НФ, 1NF) утворює ґрунт для

структурованої схеми бази даних:

- Кожна таблиця повинна мати основний ключ: мінімальний набір колонок, які ідентифікують запис.
- Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи неключові атрибути.
- Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.
- Друга нормальна форма (2НФ, 2NF) вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа:
 - Схема бази даних повинна відповідати вимогам першої нормальної форми.
 - Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.
- Третя нормальна форма (3НФ, 3NF) вимагає, аби дані в таблиці залежали винятково від основного ключа:
 - Схема бази даних повинна відповідати всім вимогам другої нормальної форми.
 - Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.
- Нормальна форма Бойса — Кодда. Відношення знаходиться в НФБК тоді і лише тоді, коли детермінант кожної функціональної залежності є потенційним ключем. Якщо це правило не виконується, то, щоб привести вказане відношення до НФБК, його слід розділити на два відношення шляхом двох операцій проєкції на кожну функціональну залежність, детермінант якої не є потенційним ключем:
 - Проєкція без атрибутів залежної частини такої функціональної залежності;

- Проекція на всі атрибути цієї функціональної залежності.
- Четверта нормальна форма (4НФ, 4NF) потребує, аби в схемі баз даних не було нетривіальних багатозначних залежностей множин атрибутів від будь чого, окрім надмножини ключа-кандидата. Вважається, що таблиця знаходиться у 4НФ тоді і лише 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 нетривіальних залежностей об'єднання, котрі б не витікали із обмежень ключів. Вважається, що таблиця в п'ятій нормальній формі тоді і лише 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.

2.3 Визначення типів даних

Структура MySQL трирівнева: бази даних – таблиці – записи. Бази даних і таблиці. 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 використовувались:

- 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;

- INT – діапазон від -2 147 483 648 до 2 147 483 647;
- DATETIME — Формат: YYYY-MM-DD HH:MM:SS. Підтримується діапазон від '1000-01-01 00:00:00' до '9999-12-31 23:59:59'

У кореневому каталозі phpMyAdmin створимо нову базу даних «audit». Вона містить вище описані таблиці з наступними типами даних полів.

Таблиця 2.1

Склад таблиці «about»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	about_id	INT	11
2	about_photo	VARCHAR	255
3	about_alt	VARCHAR	255
4	about_title	VARCHAR	255
5	about_description	VARCHAR	255

Таблиця 2.2

Склад таблиці «services»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	services_id	INT	11
2	services_name	VARCHAR	255
3	services_description	TEXT	

Таблиця 2.3

Склад таблиці «workers»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	workers_id	INT	15
2	workers_photo	VARCHAR	255
3	workers_alt	VARCHAR	255
4	workers_firstname	VARCHAR	255
5	workers_lastname	VARCHAR	255
6	workers_position	VARCHAR	255
7	workers_description	TEXT	
8	id_services	INT	25

Таблиця 2.4

Склад таблиці «clients_request»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	request_id	INT	11

2	client_name	VARCHAR	255
3	client_phone	VARCHAR	20
4	id_services	INT	25
5	request_date	DATETIME	

2.4 Обмеження цілісності даних

Цілісності бази даних — це правила, які дають змогу уникнути введення некоректних даних у БД, а також забезпечити можливість зв'язування декількох таблиць. Ці правила можуть бути описані при створенні чи модифікації таблиці.

В sql виділяють такі типи цілісності даних:

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

PRIMARY KEY (PK) використовується для ідентифікації рядків таблиці, в нього є певні особливості:

- В одній таблиці БД може бути тільки один PK.
- Рядки, які мають це правило, не можуть мати не визначені або не задані дані.

FOREIGN KEY (FG) використовується для того, щоб зв'язати дві таблиці, в нього є такі особливості :

- Поле у головній таблиці, на яке буде адресуватися FG повинне бути оголошене PK або UNIQUE.
- Поле у головній таблиці і поле з якої відбувається адресація повинні мати один й той самий тип даних

При роздобці даної бази даних „audit” було використано: primary key, foreign key, unique.

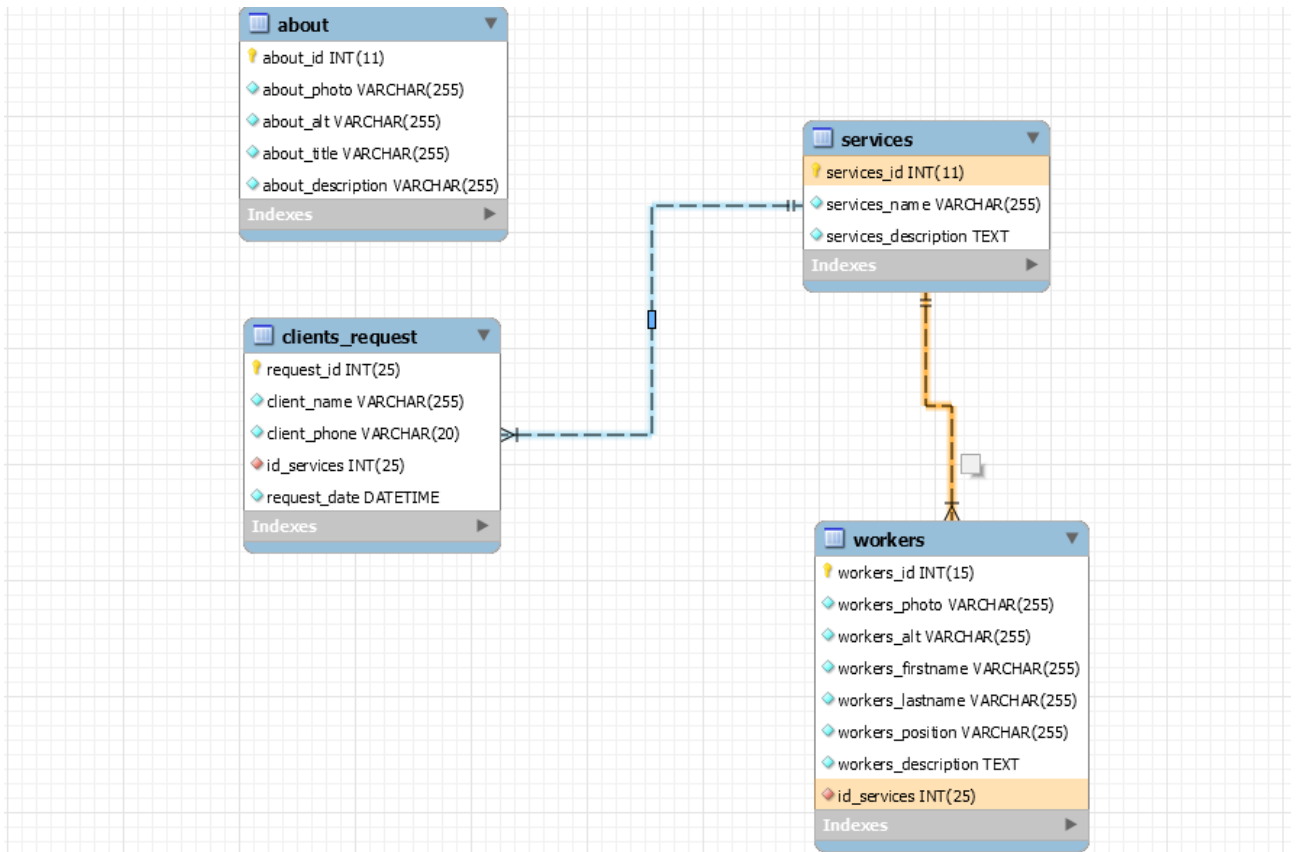


Рис 2.2. Діаграма сутностей та зв'язків, створений за допомогою
PRIMARY KEY і FOREIGN KEY

2.5 Реалізація SQL-скрипту

Веб-сайт реалізує такі запити до бази даних:

1. Запити типу «SELECT» :

- (SELECT * FROM about) – використовується для вибору всієї інформації з таблиці about;
- (SELECT * FROM services) – використовується для вибору всієї інформації з таблиці services;
- (SELECT * FROM workers) – використовується для вибору всієї інформації з таблиці workers;

Прикладом використання запиту SELECT у коді є:

```
<?php
```

```
$query = mysqli_query($sql, "SELECT * FROM services");
```

```

if ($query)
{
    while ($row = mysqli_fetch_row($query)) {
        echo "<option value='$row[0]'> $row[1]</option>";
    }
}
?>

```

2. Запити типу «INSERT» :

Заповнення таблиці `services`:

```

INSERT INTO `services` (`services_id`, `services_name`,
`services_description`) VALUES

```

(1, 'Аудиторські послуги', 'Відділ може виконати перевірку результатів фінансової і виробничої діяльності, допомогти підвищити якість зовнішньої фінансової звітності, зміцнити процедури корпоративного управління'),

(2, 'Консалтинг', 'Сформована команда експертів, які володіють практичним досвідом у наданні консультацій з реалізації стратегій, операційного управління та виробництва'),

(3, 'Послуги з оподаткування', 'Фахівці надають актуальну і точну інформацію про законодавство, що дозволяє компаніям використовувати в управлінні гнучкий і підхід і застережливий принцип'),

(4, 'Юридичні послуги', 'Команда юристів і адвокатів, що спеціалізується на питаннях національного і міжнародного законодавств'),

(5, 'Супровід угод', 'Ми надамо Вам допомогу на всіх етапах угоди - від пошуку потенційних компаній для купівлі-продажу до етапу інтеграції'),

(6, 'Форензик ', 'Група Форензик надає фінансові і економічні послуги компаніям та їх юристам, які беруть участь в судових розглядах, і в корпоративних розслідуваннях');

Прикладом використання запиту INSERT у коді є:

```

<?php
include 'config.php';

```

```

$named = isset($_REQUEST['name']) ? addslashes($_REQUEST['name']) : "";
$phoned = isset($_REQUEST['phone']) ? addslashes($_REQUEST['phone']) : "";
$idservice = isset($_REQUEST['service']) ?
addslashes($_REQUEST['service']) : "";

```

```

$msg = "INSERT INTO clients_request (client_name, client_phone,
id_services, request_date)
VALUES ('$named', '$phoned', '$idservice', NOW())";

```

```

if ($sql->query($msg) === TRUE) {
    echo "Нова заявка успішно додана";
} else {
    echo "Помилка: " . $msg . "<br>" . $sql->error;
}
?>

```

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Розміщення веб-сайту на локальному віртуальному середовищі або в Інтернеті

Для реалізації проекту було використано локальну платформу ХАМРР. ХАМРР — безкоштовна „кроссплатформена” збірка в якій міститься набір додатків для повноцінного функціонування сайту. Перевагою є відсутність потреби в налаштуванні кожного компонента окремо (Рис.3.1).

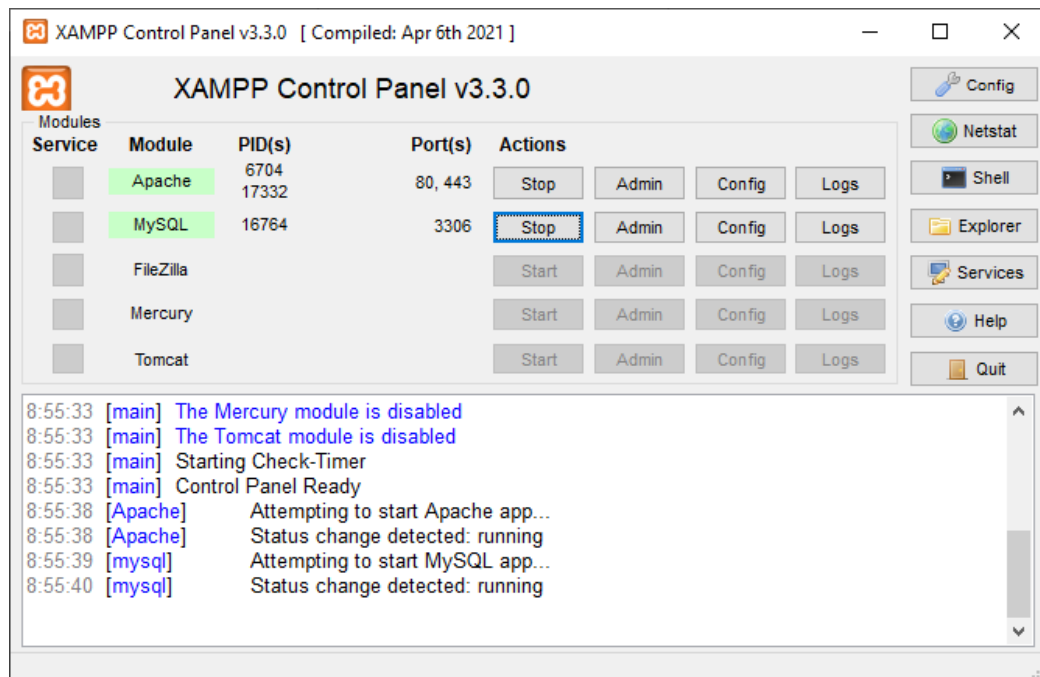


Рис 3.1 Інтерфейс Control Panel ХАМРР

Збірка дозволяє вести розробку на окремому комп'ютері без підключення до мережі. Відповідно, немає необхідності в високошвидкісному доступі в Інтернет і наявності хостингу.

Сама назва ХАМРР - аббревіатура, яка розшифровується таким чином:

- Х — позначає, що збірка буде працювати на всіх операційних системах;
- Apache — веб-сервер, що підтримує основні версії ОС, гнучко налаштовується і надійний;

- MySQL — одна з найвідоміших систем управління базами даних;
- PHP — мова програмування, що підтримується багатьма хостинг-провайдерами і є лідером серед мов, які використовуються для створення веб-сайтів;
- Perl — легка мова програмування, яка підтримує модулі сторонніх розробників.

Так, як XAMPP є легкою платформою у використанні, вона ідеально підходить для виконання завдання.

3.2 Структура та макет веб-сайту

Ключова складова веб сайту є структура. Вона впливає на його якість, зручність для користувачів і індексацію пошуковими системами. За допомогою коректно складеної структури вирішується ряд завдань.

Структура web сайту — це логічна побудова сторінок ресурсу, його архітектура, позиції елементів оформлення, зв'язок між розділами і сторінками всередині сайту. Розробка зовнішньої структури сайту передбачає логічне і звичне для користувача розташування меню, футера, обкладинки, зрозумілі форми для оформлення замовлення і входу на сайт.

Сайт аудиторської компанії за своєю структурою має вигляд лендінгу. Як правило, лендінг складається з однієї сторінки, на якій розміщена вся необхідна користувачеві інформація про товар або послугу. Основна мета такої сторінки — привертати увагу клієнта, мотивувати його зробити замовлення. В даному випадку аудиторська компанія реалізує набір послуг.

Доступ до сайту відбувається через адресу: <http://localhost/kursova/>. Потрапляючи на сторінку аудиторської компанії будь-який користувач може ознайомитись з інформацією про саму компанію, її співробітників та перелік послуг.

Структуру сайту складається з таких елементів:

- Шапка сайту. Містить логотип компанії, меню та контактний номер

(Додаток А);

- Банер. Основний вміст складається з заголовків та кнопки перенаправлення до форми заявки (Додаток Б);
- Інформаційний сегмент про компанію. Охоплює блоки інформації про саму компанію та її переваги (Додаток В);
- Секція послуг. Ознайомлює користувача з переліком послуг які надаються компанією (Додаток Д);
- Слайдер співробітників. Являє собою блок з прокруткою, який складається з ім'я та прізвища, фотографії, посади та короткого опису діяльності працівника (Додаток Є);
- Зворотній зв'язок. Блок контактів поділяється на дві частини: мапа з геолокацією компанії та формою подачі заявки (Додаток Ж);
- Футер. Дублює шапку сайту та включає посилання на соціальні мережі (Додаток З);

3.3 Програмування клієнтської та серверної частин

Для розробки проекту було використано середовище Visual Studio Code. Visual Studio Code — редактор вихідного коду, розроблений Microsoft. Позиціонується як «легкий» редактор коду для кроссплатформенної розробки веб- і хмарних додатків. Включає в себе відладчик, інструменти для роботи з Git, підсвічування синтаксису, засоби для рефакторинга та підтримує ряд мов програмування. Має широкі можливості для кастомізації та приємний інтерфейс редактора (Додаток И).

Клієнтська частина, та яку користувач бачить в першу чергу. При роботі використовуються такі мови, як:

- HTML (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.

HTML код розглянемо на прикладі блоку шапки сайту аудиторської компанії:

```
<header class="header">
  <article class="container">
    <div class="header__inner">
      <a href="index.php" class="logo">
        
      </a>
      <nav class="menu">
        <button class="menu__btn"></button>
        <ul class="menu__list">
          <li class="menu__item"><a class="menu__list-link"
href="#services">Послуги</a></li>
          <li class="menu__item"><a class="menu__list-link"
href="#about">Про компанію</a></li>
          <li class="menu__item"><a class="menu__list-link" href="#contact"
>Контакти</a></li>
        </ul>
      </nav>
      <a class="phone" href="tel:380636111985" >+38 (063) 611 19 85 </a>
    </div>
  </article>
</header>
```

- CSS/SCSS — формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. SCSS це розширення CSS, що дає можливість використовувати змінні, вкладення, міксини, імпорту і багато іншого.

CSS/SCSS код розглянемо на прикладі елемента меню сайту аудиторської

компанії:

```
.menu{
  margin-left: 70px;
  &__btn{
    display: none;
  }
  &__list{
    display: flex;
    &-link{
      text-transform: uppercase;
      padding-bottom: 5px;
      transition: all .5s;
      border-bottom: 1px solid transparent;
      color: $contrast-color;
      &:hover{
        border-bottom: 1px solid $orange-color;
      }
    }
  }
  &__item{
    margin: 0 15px;
  }
}
```

- 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

JavaScript код розглянемо на прикладі виклику функції натискання на бургер-меню сайту аудиторської компанії:

```
$(function(){
```



```

$('.menu__btn').on('click', function(){
    $('.menu__list').toggleClass('menu__list--active');
});
});

```

Для програмування серверної частини веб-сайту використано PHP. 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.

Створення зв'язку між базою даних і сайтом відбувається за допомогою файлу config.php в папці include:

```

<?php
$host = 'localhost'; // адреса сервера
$dbase = 'audit'; // ім'я бази даних
$user = 'root'; // ім'я користувача
$password = ''; // пароль

$sql = mysqli_connect($host, $user, $password, $dbase)
    or die("Помилка");
?>

```

Далі файлу index.html змінюємо розширення на .php та прописуємо наступний скрипт:

```

<?php
include($_SERVER['DOCUMENT_ROOT'].'/kursova/include/config.php');
?>

```

Таким чином створено підключення основного файлу проекту до файлу config.php. Для того щоб відкрити в браузері проект, достатньо ввести в адресному рядку `http://localhost/kursova/`.

Як було зазначено вище до сайту підключається база даних. Це багато в чому спрощує розробку, тому що вся інформація буде зберігатися в таблицях, а не в самому коді. Так швидше редагувати і додавати дані на сайт.

Наприклад, відомості про спеціалістів компанії або послуги. Якби не бази даних, довелося б зберігати всі дані в коді, а це не найкращий варіант, так як це збільшує час завантаження коду.

Для виведення даних про співробітників з бази використовується наступний код та sql запит:

```
<?php
$query = mysqli_query($sql, "SELECT * FROM workers");
if ($query)
{
    while ($row = mysqli_fetch_row($query)){
        echo "<div class='workers__item'><img class='workers__item-
icon' src='./img/$row[1]' alt='$row[2]'><p class='workers__item-title'>$row[3]
$row[4] <br/> $row[5]</p><p class='workers__item-text'>$row[6]</p></div>";
    }
    mysqli_free_result($query);
}
?>
```

Також за допомогою PHP можна доволі просто отримати інформацію, яку ввів користувач. З форми, яка також частково заповнюється даними з бази, після натискання кнопки «Залишити заявку», дані записуються в базу. Приклад коду форми:

```
<form id = "contact_form" class = "contact__form" action = " " method = "post">
    <input required class = "contact__input" type = "text" name = "name"
placeholder = "Ім'я">
    <input required class = "contact__input" type = "text" name = "phone"
placeholder = "Телефон">
    <select required class = "contact__input" name = "service">
        <option> Оберіть послугу </option>
<?php
$query = mysqli_query($sql, "SELECT * FROM services");
```

```

        if ($query)
        {
            while ($row = mysqli_fetch_row($query)) {
                echo "<option value='$row[0]'> $row[1]</option>";
            }
        }
    ?>
</select>

<button class="contact__btn aj__send" type="button" name =
"Submit" >Залишити заявку</button>

<p class="response"></p>
</form>

```

Приклад коду запису даних в базу:

```

<?php
include 'config.php';

$named = isset($_REQUEST['name']) ? addslashes($_REQUEST['name']) : "";
$phoned = isset($_REQUEST['phone']) ? addslashes($_REQUEST['phone']) :
";

$idservice = isset($_REQUEST['service']) ?
addslashes($_REQUEST['service']) : "";

$msg = "INSERT INTO clients_request (client_name, client_phone,
id_services, request_date)
VALUES ('$named', '$phoned', '$idservice', NOW())";
if ($sql->query($msg) === TRUE) {
    echo "Нова заявка успішно додана";
} else {
    echo "Помилка: " . $msg . "<br>" . $sql->error;
}
?>

```

ВИСНОВКИ

В ході виконання завдання проаналізовано всі аспекти і можливості використання HTML, CSS, JS, PHP. Для закріплення набутих знань розроблено базу даних аудиторської компанії, а також веб-сайт, який дозволяє відвідувачам дізнаватись актуальну інформацію та залишити заявку на отримання відповідних послуг.

Детальний аналіз вимог користування програмним продуктом дозволи виокремити можливі варіанти використання веб-сайту, що продемонстровано у вигляді «Use Case» діаграми.

Було спроектовано та описано структуру БД, її таблиці та записи. Визначено типи даних та функціональні залежності. Розроблено запити та визначено атрибути полів. Для кращого розуміння структури бази даних побудовано ER діаграму. Вона показує взаємодію між відношеннями і дозволяє приступити до програмування бази даних.

При розробці клієнтської та серверної частини застосовано 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 SCSS. За допомогою jQuery реалізовано адаптивний слайдер.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. W3SCHOOLS ONLINE WEB TUTORIALS [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/>
2. HTMLBOOK.RU | ДЛЯ ТЕХ, КТО ДЕЛАЕТ САЙТЫ [Електронний ресурс]. – Режим доступу: <http://htmlbook.ru/>
3. Подключение в PHP к MySQL и выполнение запросов [Електронний ресурс]. – Режим доступу: <https://metanit.com/php/tutorial/7.2.php>
4. Типы данных SQL для MySQL, SQL Server і MS Access [Електронний ресурс]. – Режим доступу: https://html5css.ru/sql/sql_datatypes.php
5. JavaScript навчальні матеріали [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
6. Зображення в HTML [Електронний ресурс]. – Режим доступу: <https://vertex-academy.com/tutorials/ru/izobrazheniya-v-html/>
7. Установка XAMPP. Покроково: скачування, установка, запуск [Електронний ресурс]. – Режим доступу: <https://ipipe.ru/info/ustanovka-xampp>
8. Бази даних та інформаційні системи — Вікі ЦДПУ [Електронний ресурс]. – Режим доступу: <https://cutt.ly/CyTXnR1>
9. БАЗЫ ДАННЫХ - УРОК 3. РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ [Електронний ресурс]. – Режим доступу: <https://site-do.ru/db/db3.php>
10. ОСНОВИ БАЗ ДАНИХ [Електронний ресурс]. – Режим доступу: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/118/118.html>
11. ЧТО ТАКОЕ СУБД - RU-CENTER [Електронний ресурс]. – Режим доступу: https://www.nic.ru/help/chto-takoe-subd_8580.html
12. РОЗДІЛ 1 . МОДЕЛЮВАННЯ РЕЛЯЦІЙНОЇ СТРУКТУРИ БД [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5391756/>
13. КОРОТКЕ ОЗНАЙОМЛЕННЯ З ІНСТРУМЕНТОМ ДЛЯ БАЗ ДАНИХ – PHPMYADMIN | SEBWEO [Електронний ресурс]. – Режим доступу:

- <https://sebweo.com/korotke-oznajomlennya-z-instrumentom-dlya-baz-danih-phpmyadmin/>
14. НОРМАЛІЗАЦІЯ ВІДНОШЕНЬ [Електронний ресурс]. – Режим доступу: https://life-prog.ru/ukr/1_331_normalizatsiya-vidnoshen.html
 15. PHP documentation [Електронний ресурс]. - Режим доступу: <http://php.net/docs.php/>
 16. Web developer information website [Електронний ресурс]. - Режим доступу: <http://www.w3schools.com/>
 17. Web developer information website [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org/wiki/PHP>
 18. Вікіпедія. [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org>
 19. Cpp-reference. [Електронний ресурс]. – Режим доступу : <http://cpp-reference.ru/patterns/behavioral-patterns/command/>
 20. Bootstrap [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>.

ДОДАТКИ

Додаток А

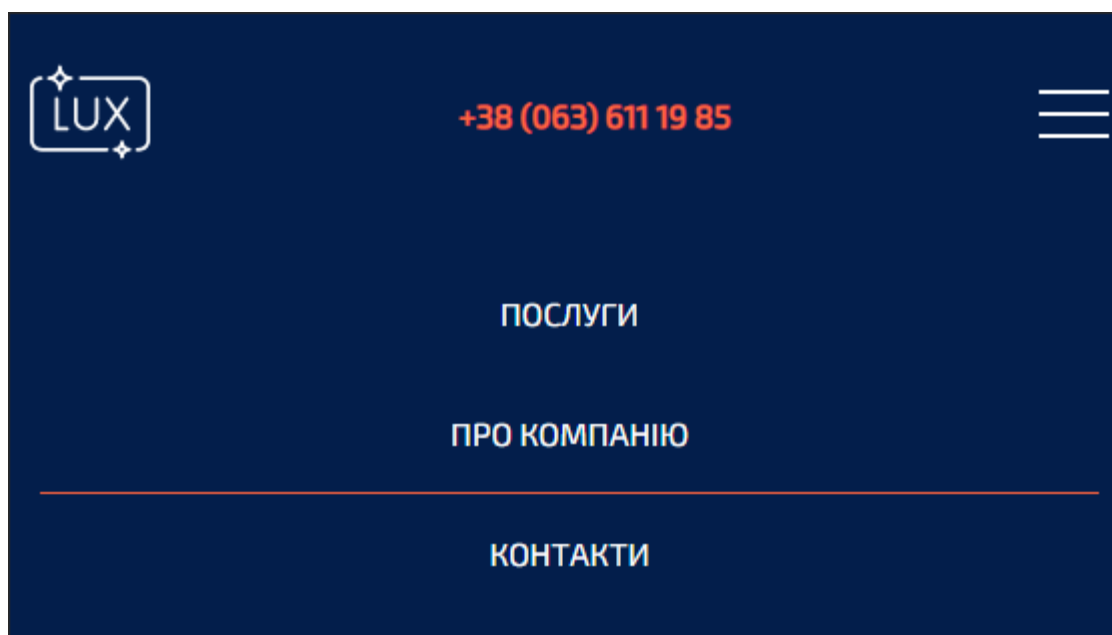


Рис 3.2 Вигляд меню сайту при адаптивному дизайні

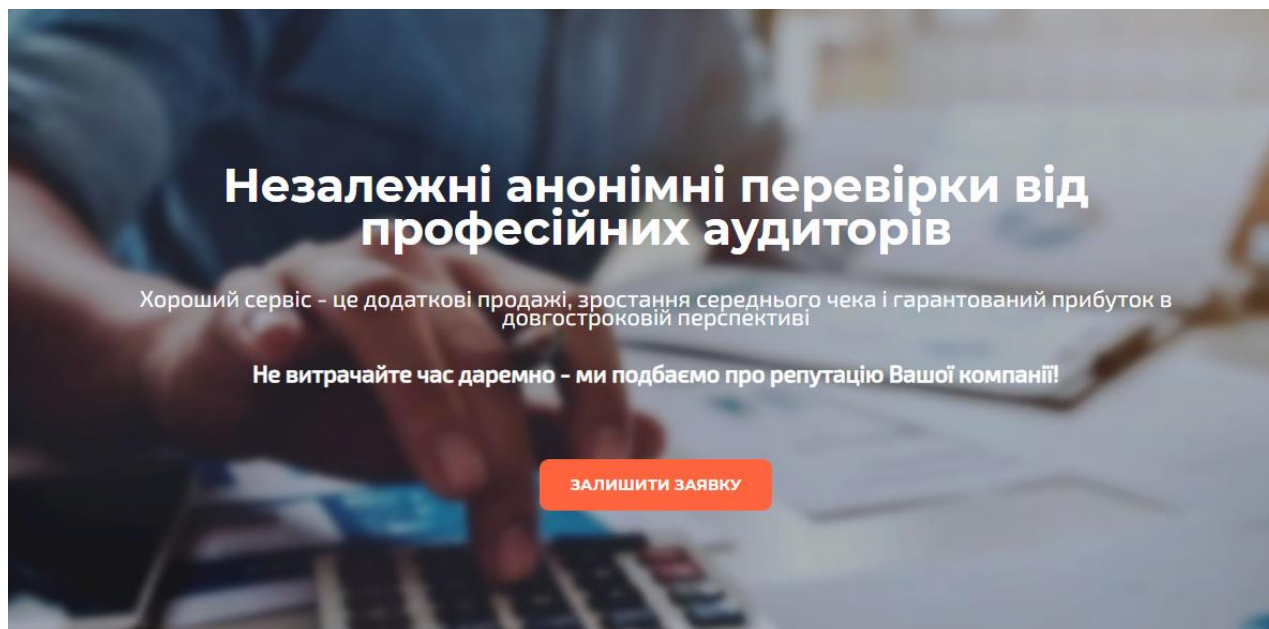


Рис 3.3 Вигляд блоку з банером



Конфіденційність інформації

Ми гарантуємо нерозголошення інформації третім особам (прописуємо в договорі)



Перевірки у всіх видах діяльності

До нас можна звернутися будь-яка компанія, яка здійснює продаж або надає послуги



Працюємо на результат

Оперативно надаємо необхідні дані, даємо докладні рекомендації, як збільшити ефективність роботи персоналу



Прозорість і деталізація вартості перевірки

Розрахунок вартості відбувається заздалегідь, ви завжди знаєте, за що платите. Детально надаємо інформацію щодо умов, методів і типів перевірки



Професіоналізм співробітників

Наша команда складається з сертифікованих фахівців, кваліфікація яких підтверджена сертифікатами ISO

Збільште ефективність роботи співробітників за допомогою перевірених даних

Об'єктивно оцінити роботу співробітників допомагає тасмний клієнт, який фіксує отримані дані, на підставі яких наша компанія формує докладний звіт, в якому відображені результати перевірки і дає докладні рекомендації, як збільшити ефективність роботи компанії.





"LUX" - ваш надійний партнер в підвищенні якості обслуговування і збільшення доходів

Ми працюємо з банками, охоронними підприємствами, ресторанами, кафе швидкого обслуговування, магазинами побутової техніки та електроніки, будівельними фірмами, фітнес-центрами, салонами краси і багатьма іншими сферами бізнесу.

Рис 3.4 Інформаційний блок компанії

Послуги

Ми надаємо широкий спектр послуг в сфері контролю персоналу, стану торгового місця, аналізу рівня обслуговування. Основні напрямки, з якими ми працюємо:

Аудиторські послуги

Відділ може виконати перевірку результатів фінансової і виробничої діяльності, допомогти підвищити якість зовнішньої фінансової звітності, зміцнити процедури корпоративного управління

Консалтинг

Сформована команда експертів, які володіють практичним досвідом у наданні консультацій з реалізації стратегій, операційного управління та виробництва

Послуги з оподаткування

Фахівці надають актуальну і точну інформацію про законодавство, що дозволяє компаніям використовувати в управлінні гнучкий і підхід і застережливий принцип

Юридичні послуги

Команда юристів і адвокатів, що спеціалізується на питаннях національного і міжнародного законодавства

Супровід угод

Ми надамо Вам допомогу на всіх етапах угоди - від пошуку потенційних компаній для купівлі-продажу до етапу інтеграції

Форензик

Група Форензик надає фінансові і економічні послуги компаніям та їх юристам, які беруть участь в судових розглядах, і в корпоративних розслідуваннях

Рис 3.5 Блок послуг, що надає компанія

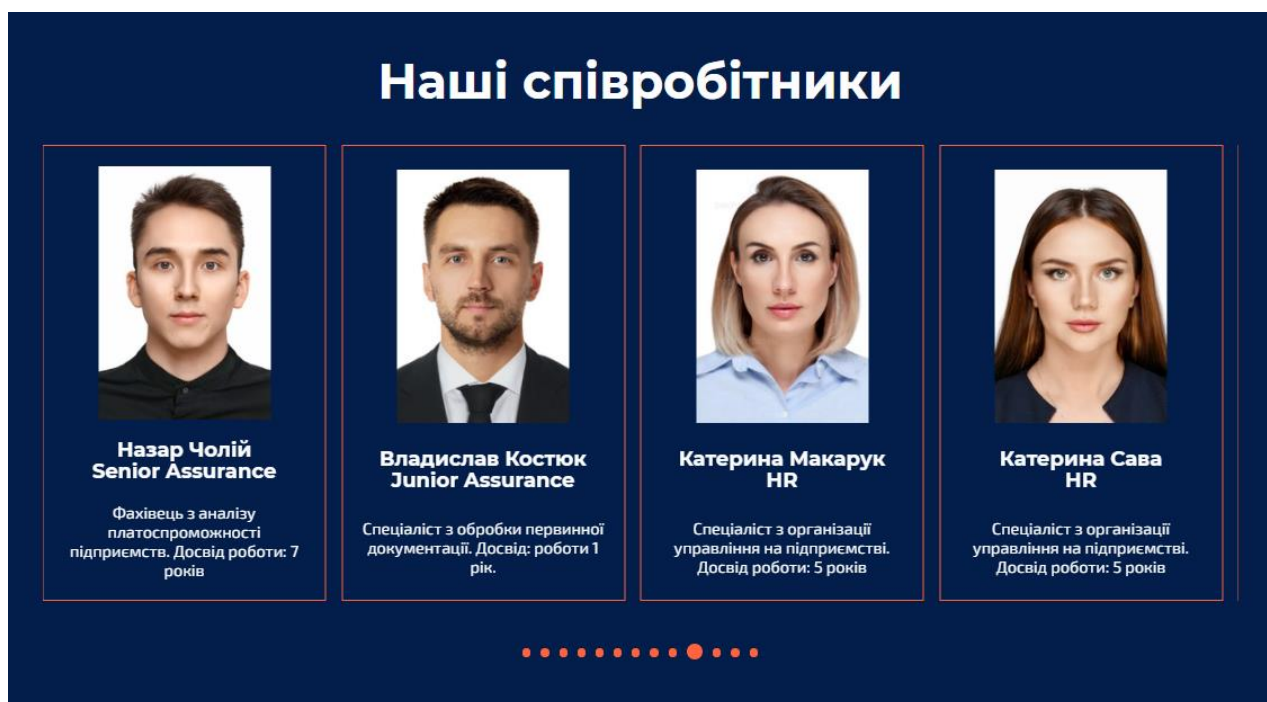
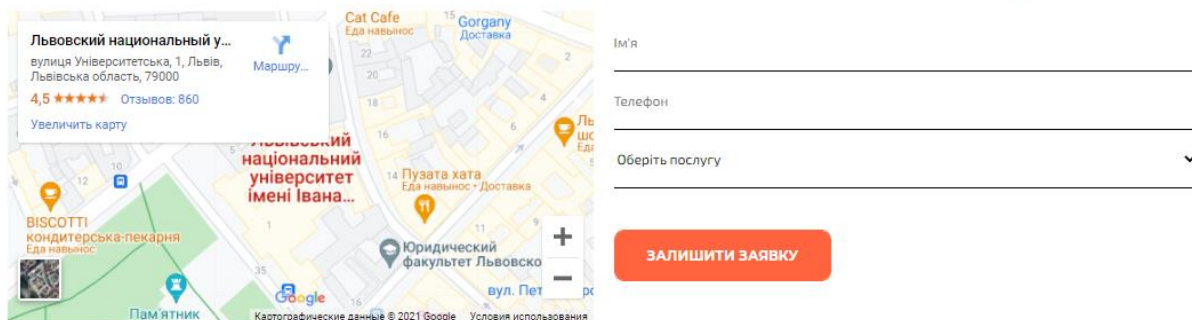


Рис 3.6 Блок співробітників компанії

Контакти

Якщо у вас виникають питання, то виможете звернутись до нас за адресою або залишити заявку на консультацію



The image shows a contact page layout. On the left is a Google Map of Lviv, Ukraine, with a search bar at the top left displaying 'Львовский национальный у...' and 'вулиця Університетська, 1, Львів, Львівська область, 79000'. The map shows various landmarks like 'Cat Cafe', 'Gorgany', 'Пузата хата', and 'Юридический факультет Львовско...'. On the right is a contact form with the following fields:

- Ім'я (Name):
- Телефон (Phone):
- Оберіть послугу (Select service):
- Залишити заявку (Submit form):

Рис 3.7 Блок контактної інформації

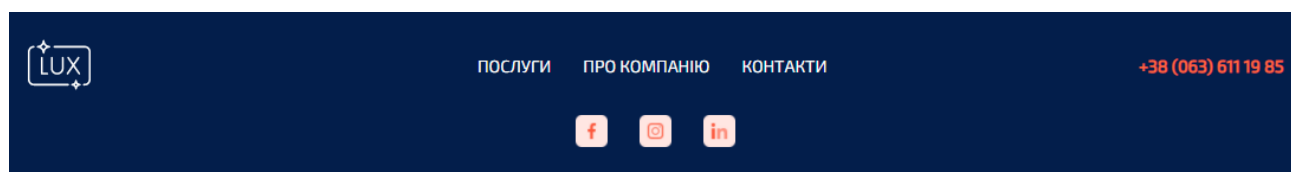


Рис 3.8 Футер сайту аудиторської компанії

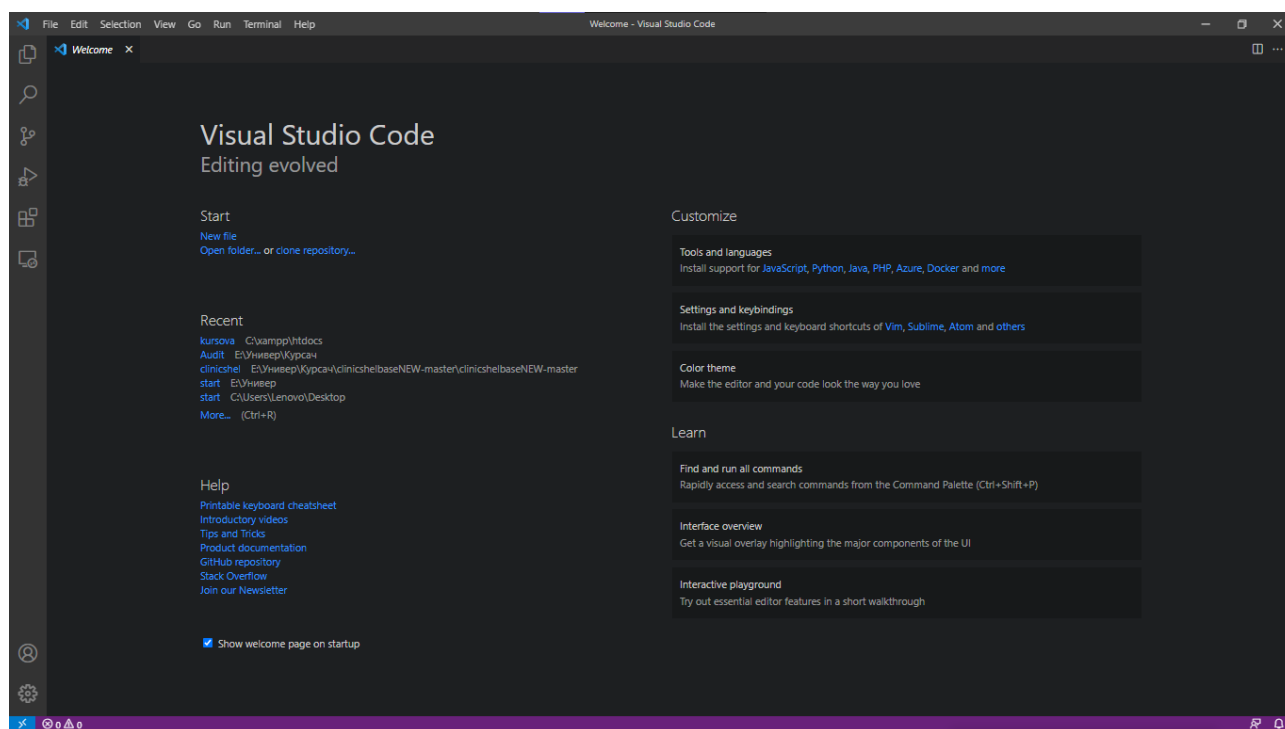


Рис 3.9 Стартовый экран Visual Studio Code