

Міністерство освіти і науки України  
Криворізький національний університет  
Факультет інформаційних технологій  
Кафедра автоматизації, комп'ютерних наук і технологій

## **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти – бакалавр  
за освітньо-професійною програмою  
«Комп'ютерні науки»  
зі спеціальності  
122 - Комп'ютерні науки

Тема роботи:

«Розробка системи управління взаємовідносинами з клієнтами для навчального  
закладу»

Виконав студент гр. \_\_\_\_\_ Назаров Н. М.

Керівник \_\_\_\_\_ Харламенко В. Ю.

Нормоконтроль \_\_\_\_\_ Маринич І. А.

Завідувач кафедри \_\_\_\_\_ Рубан С. А.

Кривий Ріг - 2024

# КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 122 - Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Зав. кафедрою: к.т.н. Рубан С.А.

« 27 » березня 2024 р.

## ЗАВДАННЯ

### на кваліфікаційну роботу бакалавра

студентові групи КН-20 Назарову Назару Максимовичу

**1. Тема кваліфікаційної роботи:** «Розробка системи управління взаємовідносинами з клієнтами для навчального закладу»

затверджено наказом по університету № 235с від 27.03.2024 р.

**2. Термін здачі кваліфікаційної роботи:** 05.06.2024 р.

**3. Склад кваліфікаційної роботи:** Пояснювальна записка обсягом 67с., додатки, презентація у Microsoft PowerPoint (15 слайдів) в електронному та друкованому вигляді

**4. Консультанти кваліфікаційної роботи:**

Розділ 1-2

ст. викладач Харламенко В. Ю.

Нормоконтроль

доц. Маринич І. А.

## 5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>23.03.2024</i>
2	<i>Розділ 1</i>	<i>30.03.24</i>
3	<i>Розділ 2</i>	<i>28.04.24</i>
4	<i>Висновки</i>	<i>11.05.24</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>15.05.24</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>20.05.24</i>
7	<i>Підготовка доповіді до захисту</i>	<i>03.06.24</i>

6. Дата видачі завдання: 29.01.2024р.

Керівник \_\_\_\_\_ /Харламенко В. Ю./

7. Запевнення: Я, Назаров Назар Максимович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент \_\_\_\_\_ / Назаров Н. М./

## АНОТАЦІЯ

Назаров Н.М. Розробка системи управління взаємовідносинами з клієнтами для навчального закладу.

Кваліфікаційна робота на здобуття ступеня вищої освіти - бакалавр, за спеціальністю 122 Комп'ютерні науки. Криворізький національний університет, Кривий Ріг, 2024.

Робота складається зі вступу, двох розділів, висновку, переліку використаної літератури з 20 позицій та 2 додатки. Загальний обсяг роботи становить 67 сторінок, з яких основний зміст роботи викладено на 51 сторінці, включає 2 таблиці і 50 рисунків.

У роботі була розглянута концепція CRM системи та їх основні функції. Був наданий опис механізмів обміну даними між компонентами системи та користувачем, а також пояснення цих самих механізмів. У рамках дослідження було проведено порівняльний аналіз сучасних інструментів та технологій CRM системи. За результатами дослідження, було вирішено використовувати ASP.NET Core та Entity Framework для розробки програми.

Система включає функціонал для управління лідами та клієнтами, що дозволяє зберігати та обробляти інформацію про потенційних і діючих студентів навчального закладу. Завдяки використанню бібліотеки Entity Framework Core реалізовано бекенд для взаємодії з базою даних проекту, що забезпечує надійне зберігання та обробку даних.

CRM, ENTITY FRAMEWORK CORE, ASP.NET CORE, УПРАВЛІННЯ КЛІЄНТАМИ, БАЗИ ДАНИХ.

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ДОСЛІДЖЕННЯ МОЖЛИВИХ ПІДХОДІВ ДО ВИРІШЕННЯ ЗАВДАННЯ РОЗРОБКИ CRM СИСТЕМИ ДЛЯ УПРАВЛІННЯ КЛІЄНТАМИ .....	7
1.1 Визначення CRM систем та їх можливості.....	7
1.2 Аналіз сучасних аналогів CRM систем .....	9
1.3 Постановка задачі дослідження .....	16
Висновки до розділу: .....	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ CRM СИСТЕМИ ДЛЯ НАВЧАЛЬНОГО ЗАКЛАДУ	20
2.1 Вибір технологій для реалізації проекту.....	20
2.2 Проектування структури бази даних .....	22
2.3 Варіанти використання системи .....	26
2.4 Класи .....	29
2.5 Інструкція користувача .....	31
2.5.1 Головна сторінка .....	33
2.5.2 Сторінка лідів .....	37
2.5.3 Сторінка клієнтів.....	38
2.5.4 Сторінка відмов.....	39
2.5.5 Сторінка управління курсами .....	40
2.5.6 Сторінка виконання плану .....	43
2.5.7 Сторінка нових користувачів.....	46
2.5.8 Сторінка дітей.....	46
2.5.9 Сторінка дорослих.....	47
Висновки до розділу: .....	48
ВИСНОВКИ .....	49
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	50
ДОДАТОК А. Лістинг коду контролера .....	52
ДОДАТОК Б. Лістинг коду сторінок .....	57

## ВСТУП

В сучасному освітньому середовищі, де створення та підтримка ефективної взаємодії між учнями, вчителями, адміністрацією та батьками стає все більш важливою задачею, виникає необхідність у використанні інноваційних інструментів. Розробка CRM-системи для навчального закладу може стати ключовим етапом у поліпшенні цієї взаємодії та оптимізації управління освітнім процесом.

Ця CRM-система може стати центральним механізмом, який об'єднує різні аспекти навчального закладу - від управління студентами та їх академічними досягненнями до взаємодії з батьками та аналізу даних для прийняття стратегічних рішень.

Створення CRM-системи для навчального закладу відкриває безліч можливостей. Вона може допомогти в оптимізації адміністративних процесів, автоматизації контролю за успішністю студентів, вдосконаленні комунікації між всіма учасниками навчального процесу, а також в управлінні фінансами та ресурсами.

Дана система може стати важливим інструментом для впровадження освітніх програм, моніторингу планування та оцінювання результатів. Вона також може сприяти розвитку індивідуального підходу до кожного студента та покращенню якості навчання та виховання.

У даному контексті, розробка CRM-системи для навчального закладу є стратегічно важливою для підвищення ефективності та якості освіти, створення сприятливої та продуктивної навчальної середовища, а також для відповіді на потреби сучасного освітнього процесу.

## РОЗДІЛ 1.

### ДОСЛІДЖЕННЯ МОЖЛИВИХ ПІДХОДІВ ДО ВИРІШЕННЯ ЗАВДАННЯ РОЗРОБКИ CRM СИСТЕМИ ДЛЯ УПРАВЛІННЯ КЛІЄНТАМИ

#### 1.1 Визначення CRM систем та їх можливості

CRM (Customer Relationship Management) - управління взаємовідносинами з клієнтами. Це підхід, стратегія та технологія, спрямовані на управління взаємовідносинами з поточними і потенційними клієнтами компанії. Основна мета CRM - підвищити рівень задоволеності клієнтів та їх лояльність, що, у свою чергу, сприяє зростанню доходів компанії.

Визначення CRM досить широке і охоплює багато, але не всі, діяльності різних компаній. Ця система обмежується діяльністю, яка відбувається в функціях, що стикаються з клієнтом, включаючи маркетинг, продажі, обслуговування клієнтів та підтримку продукту. Також можливо впровадити цю систему у компанію, яка не спрямована на клієнтів, хоча це складніше. CRM не є всеосяжним. Є багато компаній, які можуть вирішити розробити процес дослідження та розробки, спрямований на клієнта, але цей новий процес не буде частиною вашої програми CRM, тому що ця система обмежується функціями фронт-офісу. Однак немає сумніву, що новій програмі дослідження та розробки потрібна інформація про клієнтів, яка генерується функціями CRM.

Ми використовуємо це широке визначення, щоб ми могли розробити каркас, який охоплює всі функції та компоненти, необхідні для побудови вашого рішення CRM. Давайте розглянемо ключові елементи визначення:

CRM акцентується на стратегічний вплив, а не на операційний вплив. Користь, як правило, довгострокова, а не негайна (майбутній збільшений прибуток, а не негайне зменшення витрат). Це не означає, що вам не слід продавати свою програму

з деякими пропозиціями щодо зниження витрат наперед. Це означає, що вам потрібно розуміти, де дійсно відбудуться користь і встановити реалістичні очікування.

CRM - це загальна дисципліна. Щоб зрозуміти CRM, подумайте про це як про наявність тих самих компонентів, що й у будь-якого виробничого бізнесу. Воно використовує машину (технологію CRM) та потужність (ваших людей), щоб перетворити сировину (інформацію про клієнтів) на продукти (процеси та взаємодії, які зміцнюють лояльність клієнтів).

CRM включає всі функції, які безпосередньо стикаються з клієнтом протягом всього його життя з вашою компанією. Воно стикається з декількома організаціями та перетинає межі. Функції, які зазвичай включаються в зусилля з CRM, - маркетинг, продажі, обслуговування клієнтів та підтримка продукту (чи внутрішня, чи через партнера по каналу, чи в мережі).

Ми розглянули визначення CRM системи та його використання у різних сферах. Зараз ознайомимося з основними можливостями:

- Збір та аналіз даних про клієнтів:

CRM-системи забезпечують збір даних з різних джерел, таких як веб-сайти, соціальні мережі, електронна пошта та телефонні дзвінки. Це дозволяє створити єдине, цілісне уявлення про кожного клієнта.

- Управління контактами:

CRM дозволяє ефективно управляти всіма контактами компанії, що включає інформацію про клієнтів, партнерів і постачальників. Ця інформація зберігається в централізованій базі даних, що забезпечує легкий доступ до неї.

- Автоматизація маркетингу:

CRM-системи допомагають автоматизувати маркетингові кампанії, відстежувати їх ефективність та персоналізувати маркетингові повідомлення для окремих клієнтів або сегментів клієнтів.

- Управління продажами:



CRM дозволяє відстежувати весь процес продажів від першого контакту до укладення угоди. Це включає в себе управління потенційними клієнтами, відстеження етапів продажу та прогнозування доходів.

- Обслуговування клієнтів:

CRM-системи забезпечують інструменти для управління запитами клієнтів, що включає в себе відстеження проблем, забезпечення підтримки та управління зворотнім зв'язком.

- Аналіз та звітність:

CRM-системи забезпечують інструменти для аналітики та створення звітів, що дозволяє оцінювати ефективність взаємодії з клієнтами, виявляти тренди та приймати обґрунтовані рішення.

- Інтеграція з іншими системами:

CRM-системи можуть інтегруватися з іншими корпоративними системами, такими як ERP (Enterprise Resource Planning), що дозволяє об'єднати всі бізнес-процеси в єдину екосистему.

- Персоналізація взаємодії з клієнтами:

CRM-системи дозволяють персоналізувати взаємодію з кожним клієнтом, базуючись на зібраних даних про його поведінку, історію покупок та інтереси.

- Управління лояльністю клієнтів:

CRM-системи допомагають розробляти та реалізовувати програми лояльності, що сприяють утриманню клієнтів та збільшенню їхньої довічної цінності для компанії. [1-2]

## 1.2 Аналіз сучасних аналогів CRM систем

Щоб краще зрозуміти потреби користувачів та необхідний функціонал для роботи із CRM системою, я порівняв шість найпопулярніших варіантів на сьогоднішній день.

1. Salesforce - одна з найпопулярніших CRM-систем, яка пропонує широкий спектр функціональних можливостей для управління взаємодією з клієнтами, маркетингових кампаній, автоматизації продажів та підтримки клієнтів. Платформа має модульну структуру, що дозволяє компаніям вибирати тільки ті інструменти, які їм потрібні. [3]

Плюси:

- Потужні інструменти для аналітики та звітності.
- Висока гнучкість та налаштовуваність.
- Велика кількість інтеграцій з іншими сервісами.
- Підтримка мобільних додатків.

Недоліки:

- Висока вартість підписки.
- Складність налаштування для нових користувачів.
- Потребує тривалого навчання для ефективного використання.

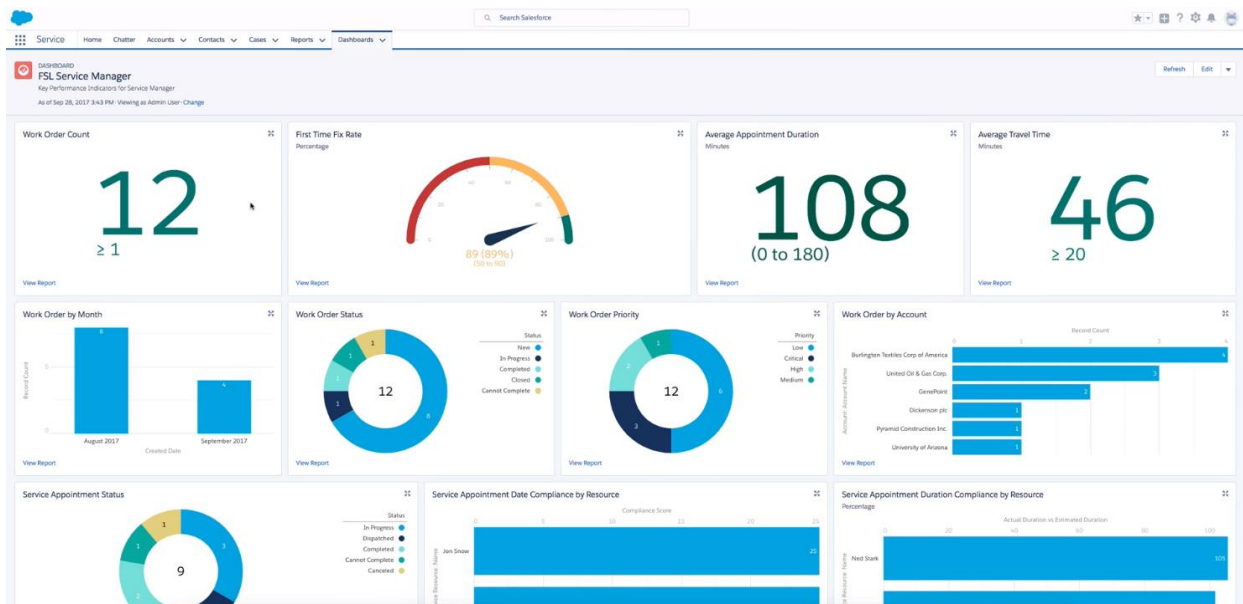


Рисунок 1.1 - Інтерфейс сторінки дашбордів системи Salesforce

2. HubSpot CRM пропонує безкоштовний базовий функціонал з можливістю розширення за рахунок платних модулів для маркетингу, продажів та обслуговування

клієнтів. Платформа відома своїм зручним інтерфейсом і широким набором інструментів для малого та середнього бізнесу. [4]

#### Плюси:

- Безкоштовна основна версія.
- Інтуїтивно зрозумілий інтерфейс.
- Інтеграція з іншими продуктами HubSpot.
- Вбудовані інструменти для автоматизації маркетингу.

#### Недоліки:

- Обмежений функціонал у безкоштовній версії.
- Вартість додаткових модулів може бути високою.
- Обмежені можливості для великих підприємств.

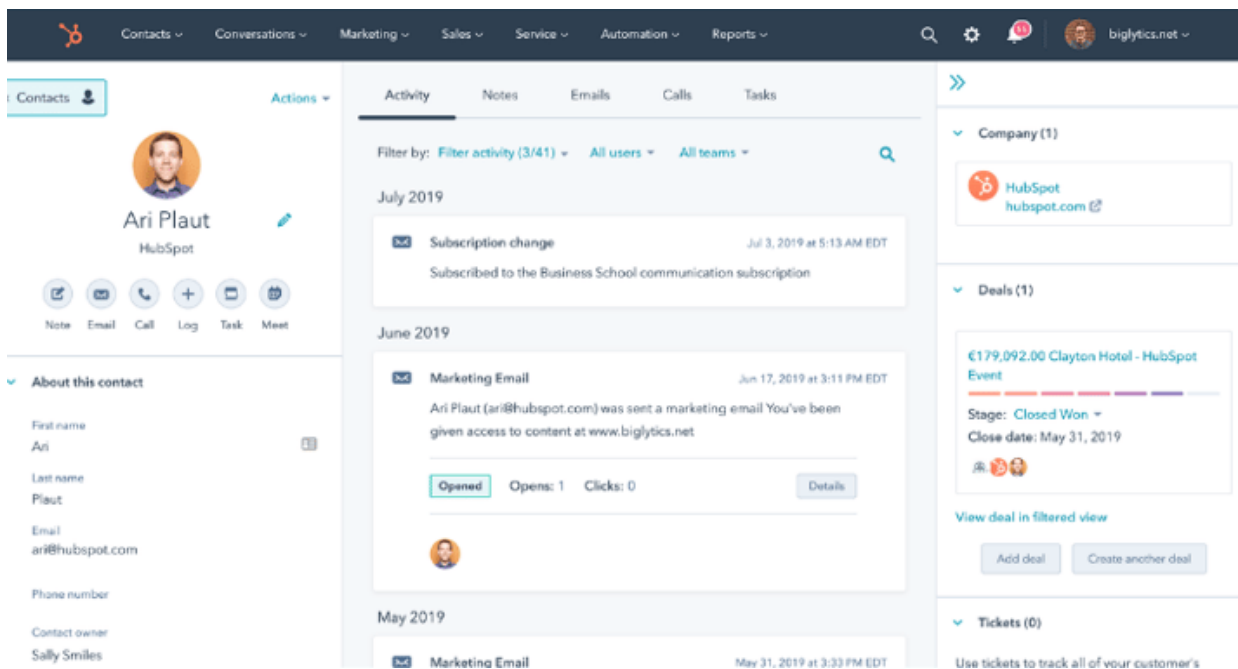


Рисунок 1.2 - Інтерфейс сторінки контактів системи HubSpot

3. Microsoft Dynamics 365 - потужна CRM та ERP-система, що об'єднує управління продажами, обслуговування клієнтів, маркетинг та фінансові операції в

одному рішенні. Вона інтегрується з іншими продуктами Microsoft, такими як Office 365 та Azure. [5]

#### Плюси:

- Широкий спектр функціональності.
- Глибока інтеграція з іншими продуктами Microsoft.
- Потужні аналітичні інструменти.
- Висока гнучкість налаштувань.

#### Недоліки:

- Висока вартість.
- Складність у налаштуванні та адмініструванні.
- Потребує значних ресурсів для впровадження.

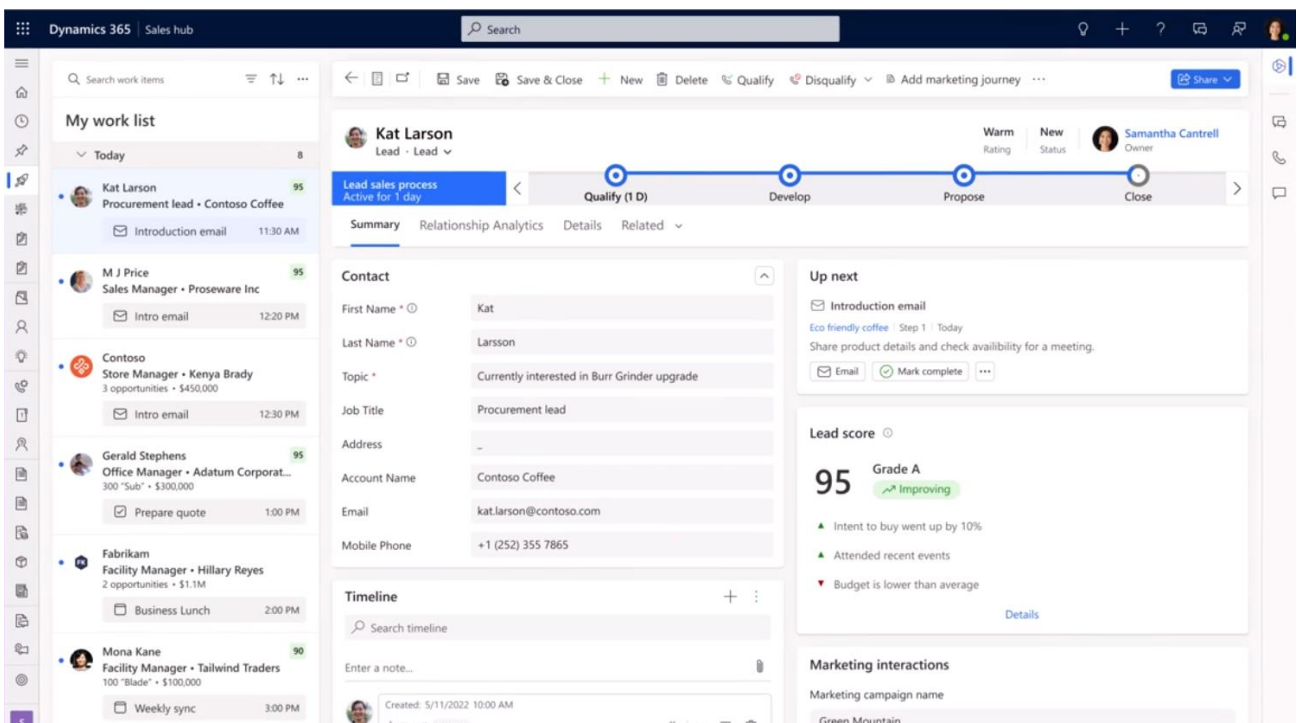


Рисунок 1.3 - Інтерфейс сторінки контактів системи Microsoft Dynamics 365

4. Monday.com - це платформа для управління проектами та командної роботи, яка також має функціонал CRM систем. Вона відома своїм візуальним та гнучким інтерфейсом, який дозволяє командам організувати роботу та співпрацювати ефективніше. [6]

#### Плюси:

- Інтуїтивно зрозумілий та візуально привабливий інтерфейс.
- Гнучкість налаштувань.
- Потужні інструменти для управління проектами.
- Інтеграція з багатьма іншими сервісами.

#### Недоліки:

- Обмежений функціонал CRM порівняно з іншими системами.
- Висока вартість для розширених функцій.
- Не підходить для великих компаній з комплексними потребами.

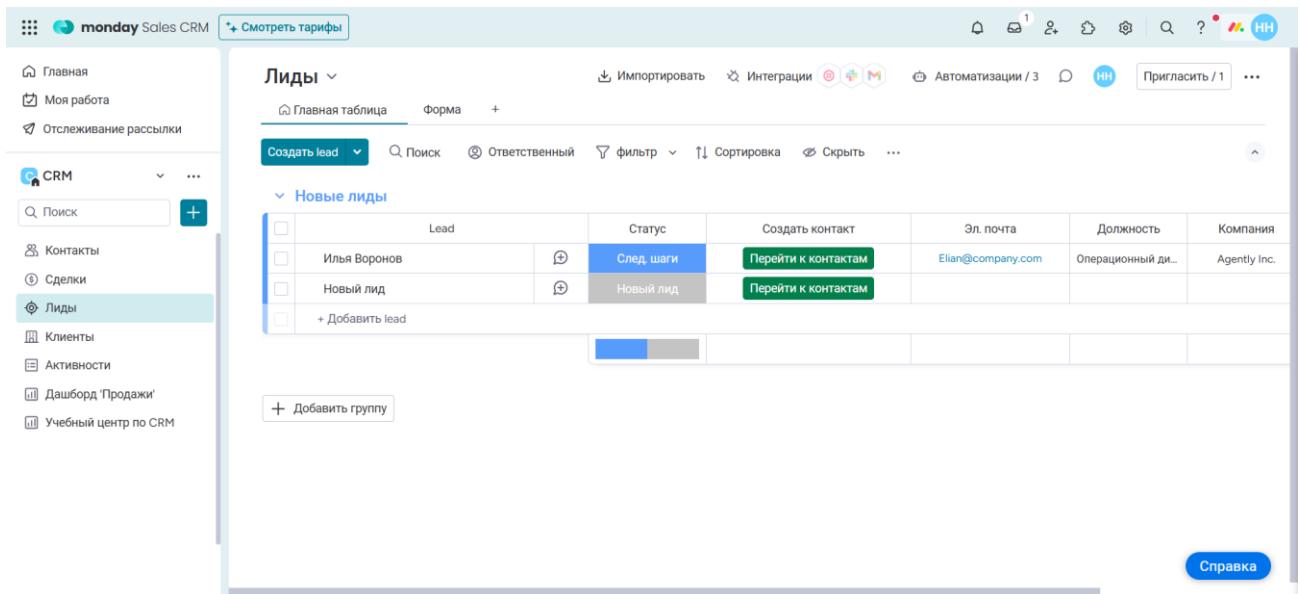


Рисунок 1.4 - Интерфейс сторінки лідів системи Monday

5. Zoho CRM - гнучка та доступна система управління взаємодією з клієнтами, яка надає широкий набір інструментів для автоматизації продажів, маркетингу та

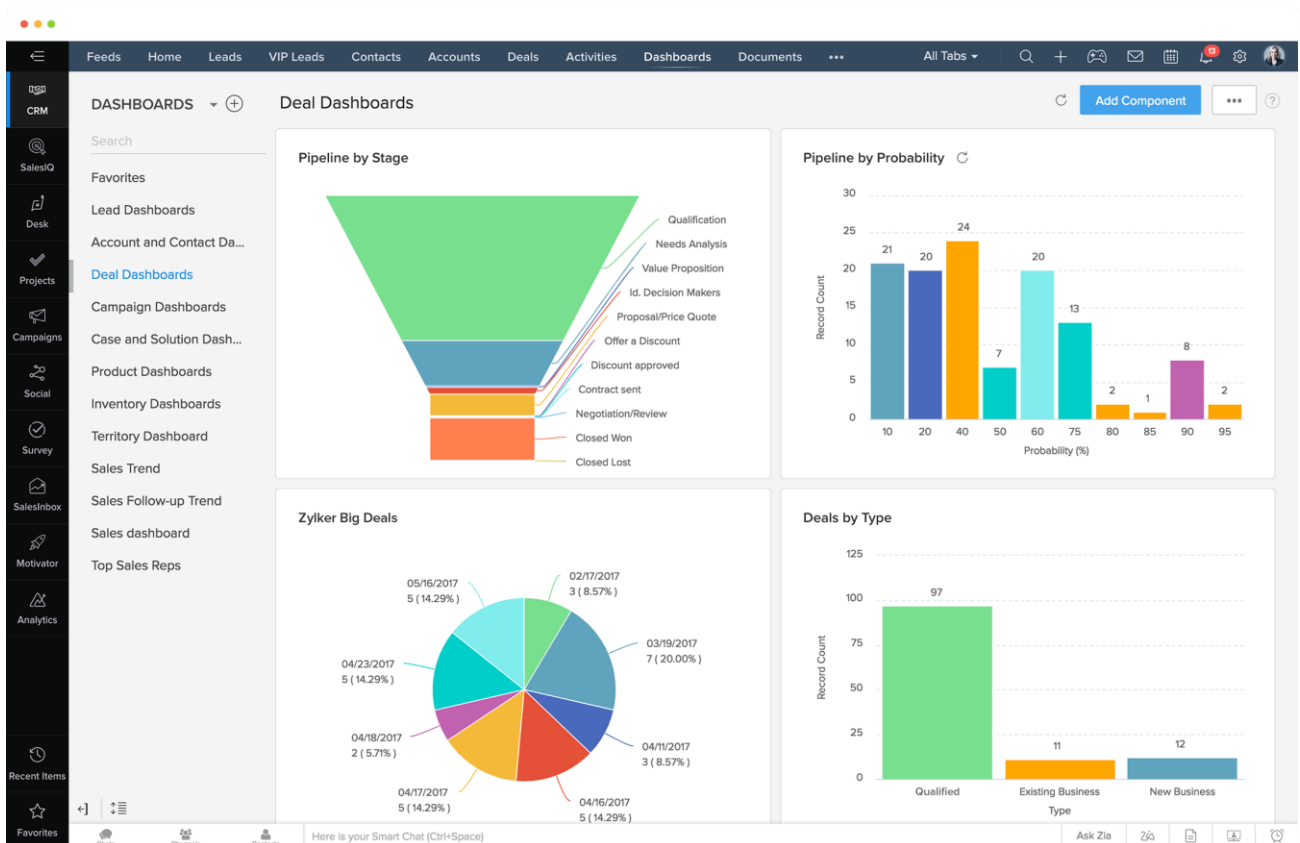
обслуговування клієнтів. Вона відома своїм співвідношенням ціни та якості. Має приємний дизайн та є однією із популярніших CRM систем. [7]

### Плюси:

- Доступна ціна.
- Широкий набір функцій для малого та середнього бізнесу.
- Інтеграція з іншими продуктами Zoho.
- Висока гнучкість налаштувань.

### Недоліки:

- Обмежена підтримка для великих підприємств.
- Інтерфейс може бути заплутаним для нових користувачів.
- Обмежені можливості аналітики порівняно з конкурентами.



## Рисунок 1.5 - Интерфейс сторінки дашбордів системи Zoho

6. Key CRM - вітчизняна розробка у сфері систем управління з клієнтами, яка є менш відомою у світі, але потужною CRM-системою, що надає базові функції управління взаємодією з клієнтами, автоматизації продажів та маркетингу. Вона орієнтована на простоту використання та базові потреби малого бізнесу. [8]

### Плюси:

- Простий та інтуїтивно зрозумілий інтерфейс.
- Доступна ціна.
- Основні функції CRM для малого бізнесу.
- Швидке налаштування та впровадження.

### Недоліки:

- Обмежений функціонал порівняно з більш відомими CRM-системами.
- Обмежена кількість інтеграцій.
- Недостатні можливості для великих підприємств.

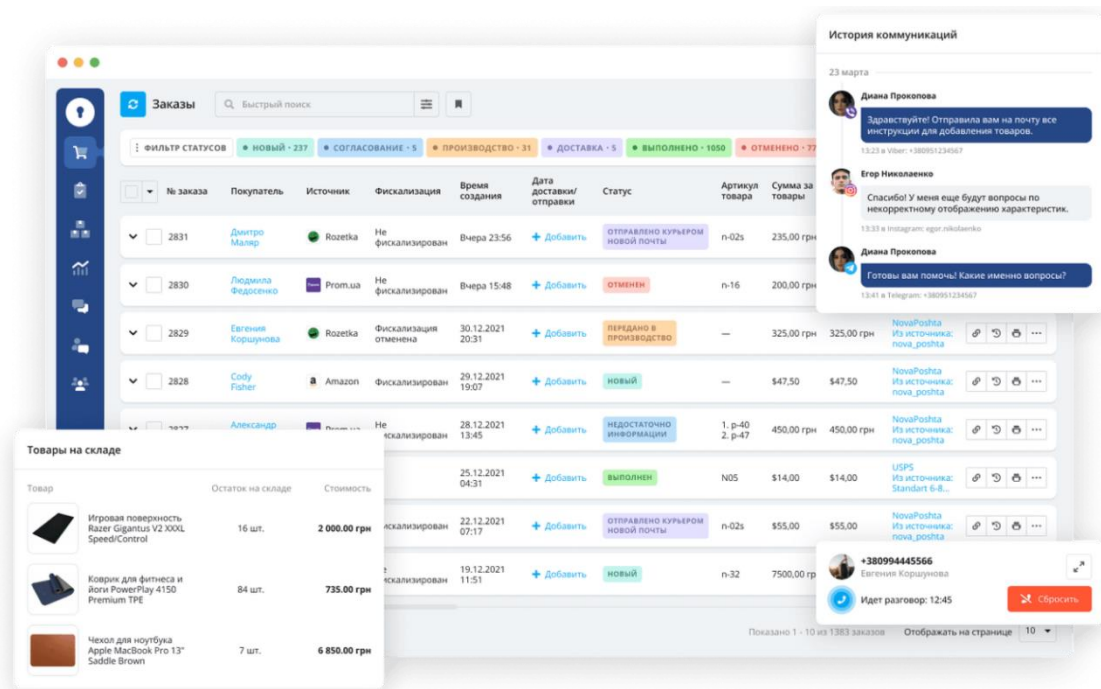


Рисунок 1.6 - Интерфейс сторінки замовлень системи Key

Таблиця 1.1 - Порівняльна таблиця можливостей CRM систем

Функції/CRM	Salesforce	HubSpot	Microsoft Dynamics 365	Monday.com	Zoho CRM	Key CRM
Збір та аналіз даних	Високий	Середній	Високий	Середній	Середній	Низький
Управління контактами	Високий	Високий	Високий	Середній	Високий	Середній
Автоматизація маркетингу	Високий	Високий	Високий	Низький	Середній	Низький
Управління продажами	Високий	Високий	Високий	Середній	Середній	Низький
Обслуговування клієнтів	Високий	Середній	Високий	Низький	Середній	Низький
Аналітика та звітність	Високий	Середній	Високий	Середній	Середній	Низький
Інтеграція з іншими системами	Високий	Високий	Високий	Середній	Середній	Низький
Персоналізація	Високий	Середній	Високий	Середній	Середній	Низький
Управління лояльністю	Високий	Середній	Високий	Низький	Середній	Низький
Мобільна підтримка	Високий	Високий	Високий	Середній	Високий	Середній
Ціна	Висока	Від низької до високої	Висока	Висока	Низька	Низька
Простота використання	Середня	Висока	Середня	Висока	Середня	Висока

### 1.3 Постановка задачі дослідження

Основними варіанти використання CRM системи навчального закладу для вивчення мов можуть бути:

Збір та обробка інформації про студентів:



- Розробка інтерфейсу для введення та збереження особистих даних студентів, таких як ім'я, контактні дані, адреса тощо.

- Перевірка та валідація введених даних для забезпечення їхньої коректності та консистентності.

- Створення інструментів для аналізу та звітності про студентів, що дозволяє легко отримувати статистичну інформацію про навчальний процес, прогрес учнів та ефективність викладачів.

Управління студентськими даними:

- Створення системи для відстеження академічної історії студентів, включаючи результати екзаменів, рейтинги, курси, які вони відвідують тощо.

- Можливість швидкого доступу до даних кожного студента для адміністраторів та викладачів.

Аналіз даних і прогнозування:

- Розробка інструментів для аналізу та візуалізації академічних даних студентів.
- Використання методів аналізу даних та машинного навчання для прогнозування можливих проблем або відставань студентів.

Комунікація зі студентами:

- Реалізація системи сповіщень про події, розсилки інформаційних повідомлень та можливості онлайн-запису на консультації чи інші заходи.

- Інтеграція з електронною поштою, системами повідомлень та соціальними мережами для ефективного спілкування.

Підтримка адміністративних процесів:

- Реалізація інструментів для ведення електронної документації, включаючи заяви на зарахування, розклади занять, звіти про успішність тощо.

- Можливість автоматичного створення та обробки різних типів документів.

Моніторинг активності студентів:

- Створення інструментів для моніторингу та аналізу активності студентів у системі, їхньої участі у викладацьких заходах, обговореннях, проектах тощо.

- Автоматичне створення звітів про активність та участь студентів.

Інтеграція з існуючими системами:

- Реалізація інтерфейсів для інтеграції з існуючими системами навчального закладу, такими як системи управління навчальними матеріалами, розкладами тощо.

- Забезпечення сумісності та обміну даними між різними системами.

Забезпечення безпеки та конфіденційності даних:

- Використання шифрування для захисту особистих даних студентів та іншої конфіденційної інформації.

- Розробка механізмів контролю доступу та автентифікації для забезпечення безпеки даних.

Підтримка мобільних пристроїв:

- Розробка мобільного додатку або адаптивного веб-інтерфейсу для забезпечення зручного доступу студентів до системи з різних пристроїв.

- Оптимізація інтерфейсу для роботи на різних типах пристроїв та екранних розмірах.

Навчання та підтримка користувачів:

- Надання навчання та документації для адміністраторів, викладачів та студентів щодо користування системою.

- Створення системи підтримки користувачів для вирішення технічних питань та надання консультацій.[9]

Висновки до розділу:

У цьому розділі був проведений аналіз того, що таке CRM система, її функціонал та можливості. Було з'ясовано, що CRM системи є потужним інструментом для оптимізації процесів взаємодії з клієнтами, включаючи збір, обробку та аналіз даних про студентів.

Також був проведений аналіз сучасних аналогів системи управління клієнтами. Після порівняння шести популярних аналогів, було виявлено їх сильні та слабкі сторони. Це дослідження показало, які функціональні можливості необхідні для ефективного управління клієнтами у контексті навчального закладу.

У розділі були визначені основні вимоги до CRM системи, окреслено цілі та завдання, які необхідно вирішити у процесі розробки. Це включає створення інтерфейсу для введення та збереження даних, механізмів фільтрації, сортування та аналізу інформації, а також інтеграцію з іншими системами для забезпечення цілісності даних та зручності користування.

## **РОЗДІЛ 2.**

### **ПРОЕКТУВАННЯ CRM СИСТЕМИ ДЛЯ НАВЧАЛЬНОГО ЗАКЛАДУ**

#### 2.1 Вибір технологій для реалізації проекту

Для створення мого проекту «CRM системи для навчального закладу», я обрав декілька технологій, які необхідні для роботи додатку.

На сьогоднішній день існує велика кількість систем управління базами даних(СУБД). Для цієї роботи я обрав Microsoft SQL Server, версія 2019 року. SQL Server - це реляційна система управління базами даних (РСУБД), розроблена компанією Microsoft. Вона призначена для зберігання та керування даними, а також для їх обробки за допомогою структурованої мови запитів (SQL). SQL Server забезпечує високий рівень продуктивності, надійності, безпеки та масштабованості для підприємств різного розміру. До недоліків цієї системи можна віднести обмежений функціонал і невеликі проблеми з надійністю.

Для роботи із СУБД необхідна мова SQL (Structured Query Language). Це стандартна мова запитів, яка використовується для управління і маніпулювання реляційними базами даних. Вона дозволяє виконувати різноманітні операції з даними, зокрема створення, читання, оновлення та видалення записів (операції CRUD). SQL використовується багатьма СУБД, такими як MySQL, PostgreSQL, SQL Server, Oracle та інші. [10]

З урахуванням вимог до функціональності CRM системи, було вирішено виконувати на веб-фреймворкі ASP.NET Core 6 на мові C#. Основними перевагами даного рішення є:

- можливості поєднання різних підходів до побудови додатку, зокрема REST Web API, архітектури Модель-Представлення-Контролер (MVC), GraphQL, gRPC;

- широкий вибір інструментів розробки веб-інтерфейсу (сторінки Razor, архітектурний шаблон MVC, Blazor та Blazor Web Assembly) та можливість створення веб-API на базі технології REST для сучасних SPA фреймворків на базі JavaScript;
- широка підтримка інструментів впровадження залежностей (Dependency Injection), зокрема впровадження в конструктор, у властивості та методи завдяки вбудованому IoC-контейнеру;
- підтримка технології вбудовування та рендерінгу коду C# на HTML-сторінках з використанням синтаксису розмітки Razor;
- зручність керування сторонніми бібліотеками з використанням пакетного менеджера Nuget;
- наявність потужної довідкової системи та ресурсів у мережі Інтернет, що постійно оновлюється та доповнюється;
- кросплатформеність як на рівні розробки додатку, так і на рівні розгортання;
- широка підтримка провайдерами послугами хостингу, у тому числі і хмарними. [11]

Також для розробки системи було вирішено використовувати наступні технології, фреймворки та бібліотеки:

- Bootstrap - бібліотека стилів, що забезпечує широкі можливості створення інтерфейсів веб-додатків у достатньо строгому, привабливому та мінімалістичному стилі з без необхідності або з мінімальним використанням власних скриптів на JavaScript; [12]
- EntityFramework Core - найпопулярніша .NET бібліотека об'єктно-реляційного відображення, що на поточний момент має підтримку великої кількості СУБД, дуже високу швидкість виконання операцій з даними та потужні можливості фільтрації, сортування та проєкції; [13]

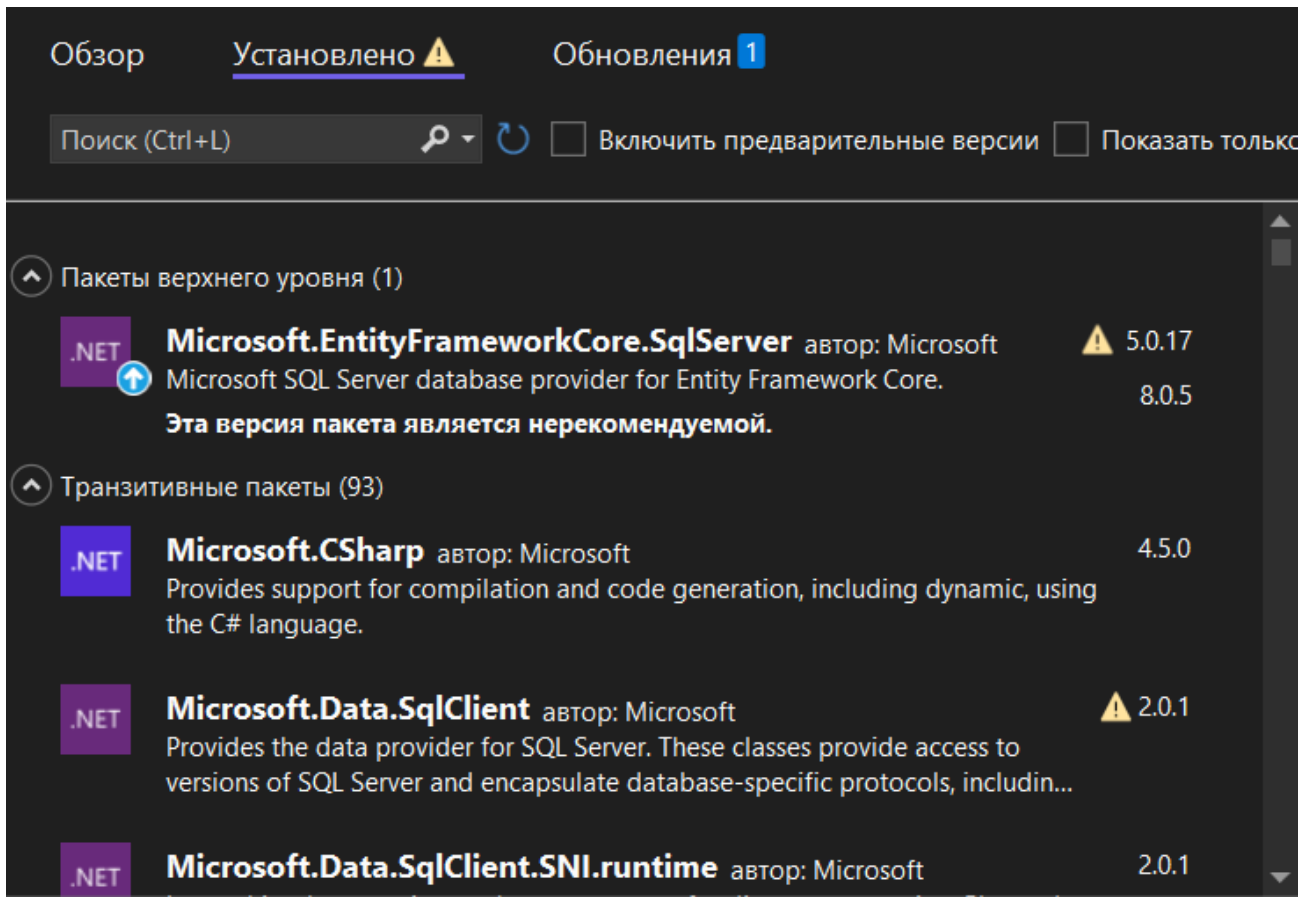


Рисунок 2.1 – Бібліотеки, що використані у проекті

## 2.2 Проектування структури бази даних

База даних, представлена на діаграмі, складається з кількох таблиць, які зв'язані між собою для зберігання і управління даними, пов'язаними з курсами, студентами, рекламними джерелами, статусами, групами, планами, мовами і типами. Це необхідно для ефективного управління даними та зручності роботи з ними. Розділення даних на різні таблиці дозволяє більш ефективно управляти ними, а також легко створювати запити для отримання необхідної інформації.

На рисунку 2.2 буде показана структура бази даних.

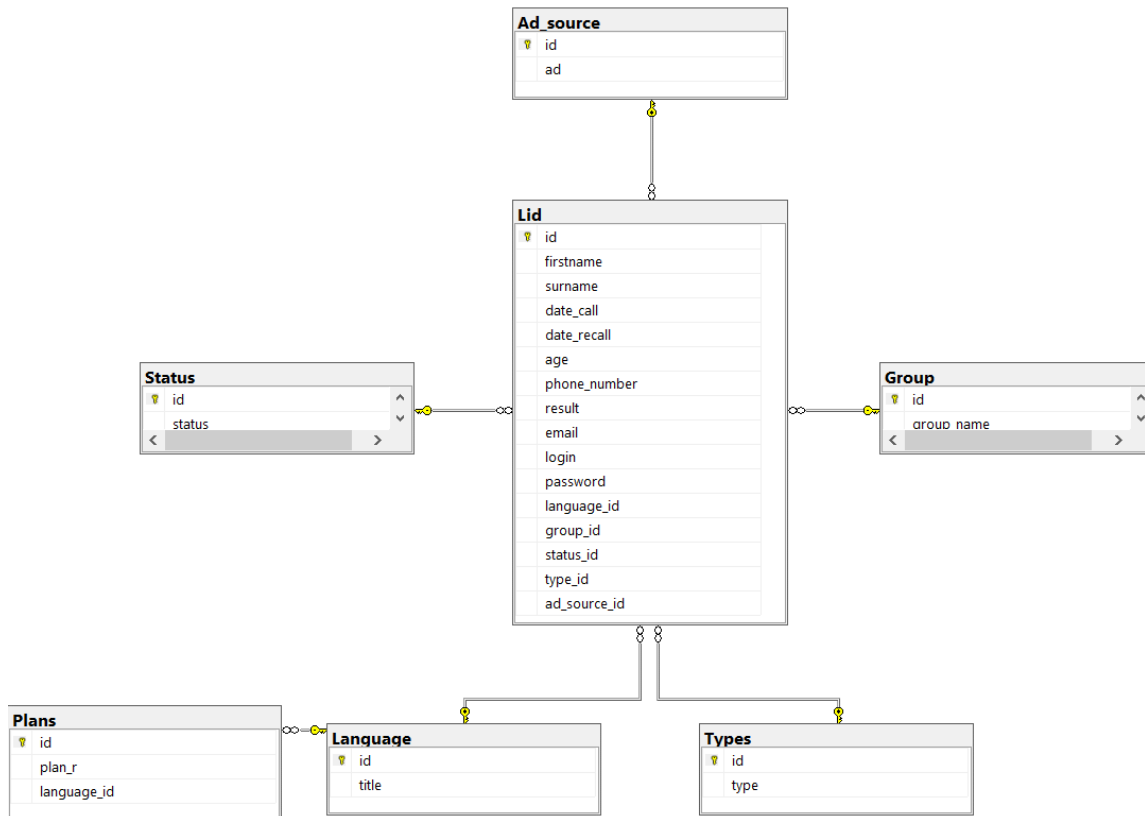


Рисунок 2.2 - Структура бази даних

Ось детальний опис кожної таблиці і зв'язків між ними:

Таблиця Lid є центральною таблицею бази даних, що містить інформацію про лідів, і використовує дані з інших таблиць для ефективної роботи. Вона має такі поля:

- id (Первинний ключ) - унікальний ідентифікатор ліда.
- firstname - ім'я ліда.
- surname - прізвище ліда.
- date\_call - дата першого дзвінка.
- date\_recall - дата повторного дзвінка.
- age - вік ліда.
- phone\_number - номер телефону ліда.
- email - електронна пошта ліда.
- result - результат комунікації з лідом.

- login - логін для доступу до системи.
- password - пароль для доступу до системи.
- language\_id - зовнішній ключ до таблиці Language, вказує на мову курсу.
- group\_id - зовнішній ключ до таблиці Group, вказує на групу, до якої належить

лід.

- status\_id - зовнішній ключ до таблиці Status, вказує на статус ліда.
- type\_id - зовнішній ключ до таблиці Types, вказує на тип навчання ліда.
- ad\_source\_id - зовнішній ключ до таблиці Ad\_source, вказує на джерело

реклами.

Таблиця Ad\_source містить інформацію про джерела реклами:

- id (Первинний ключ) - унікальний ідентифікатор джерела реклами.
- ad - назва реклами, яка привернула до себе увагу ліда (YouTube, Instagram,

Telegram, Facebook, TikTok).

Таблиця Status зберігає статуси лідів:

- id (Первинний ключ) - унікальний ідентифікатор статусу.
- status - назва статусу (клієнт, відмова, передзвонити, ігнор, провести

співбесіду).

Таблиця Group містить інформацію про групи, до яких належать ліди:

- id (Первинний ключ) - унікальний ідентифікатор групи.
- group\_name - назва групи (група, індивідуально, удвох, утрюх).

Таблиця Plans зберігає плани курсів:

- id (Первинний ключ) - унікальний ідентифікатор плану.
- plan - назва або опис плану.
- language\_id - зовнішній ключ до таблиці Language, вказує на мову, для якої

створено план.

Таблиця Language містить інформацію про мови:

- id (Первинний ключ) - унікальний ідентифікатор мови.
- title - назва мови (англійська, німецька, японська, китайська, польська)



Таблиця Types зберігає тип заняття який обирають ліди:

- id (Первинний ключ) - унікальний ідентифікатор типу.
- type - назва типу (онлайн, оффлайн).

Зв'язки між таблицями:

- Lid.language\_id - Language.id: зв'язок між таблицями Lid і Language, що дозволяє визначити мову курсу, яким цікавиться лід.

- Lid.group\_id - Group.id: зв'язок між таблицями Lid і Group, що дозволяє віднести ліда до певної групи.

- Lid.status\_id - Status.id: зв'язок між таблицями Lid і Status, що дозволяє визначити поточний статус ліда.

- Lid.type\_id - Types.id: зв'язок між таблицями Lid і Types, що дозволяє визначити тип ліда.

- Lid.ad\_source\_id - Ad\_source.id: зв'язок між таблицями Lid і Ad\_source, що дозволяє визначити джерело реклами, з якого прийшов лід.

- Plans.language\_id - Language.id: зв'язок між таблицями Plans і Language, що дозволяє визначити мову, для якої створено план курсу.

Структура цієї бази даних забезпечує ефективне управління даними лідів, необхідними для CRM-системи навчального закладу. Вона дозволяє переглядати інформацію про лідів

Структура бази даних забезпечує ефективне управління даними, необхідними для CRM-системи навчального закладу. Вона дозволяє легко продивлятися та редагувати інформацію про лідів, їх статуси, групи, плани курсів, мови навчання, типи навчання та джерела реклами. Завдяки добре продуманим зв'язкам між таблицями, система може забезпечити швидкий доступ до необхідної інформації і підтримувати високий рівень організації даних.

Таблиця 2.1 - Структура даних моделі ліда

Назва поля	Тип	Призначення
id	int	Ідентифікатор ліда
firstname	nvarchar	Зберігає ім'я
surname	nvarchar	Зберігає прізвище
date_call	datetime	Дата першого дзвінка
date_recall	datetime	Дата повторного дзвінка
age	int	Вік ліда
phone_number	int	Номер телефону
email	nvarchar	Електрона пошта
result	nvarchar	Результат дзвінка(співбесіди)
login	nvarchar	Логин адміна
password	nvarchar	Пароль адміна
language_id	int	Ідентифікатор мови ліда
group_id	int	Ідентифікатор групи ліда
status_id	int	Ідентифікатор статусу ліда
type_id	int	Ідентифікатор типу занять ліда
ad_source_id	int	Ідентифікатор реклами, яка зацікавила ліда

### 2.3 Варіанти використання системи

Діаграма використання демонструє, як різні користувачі взаємодіють з CRM системою навчального закладу.

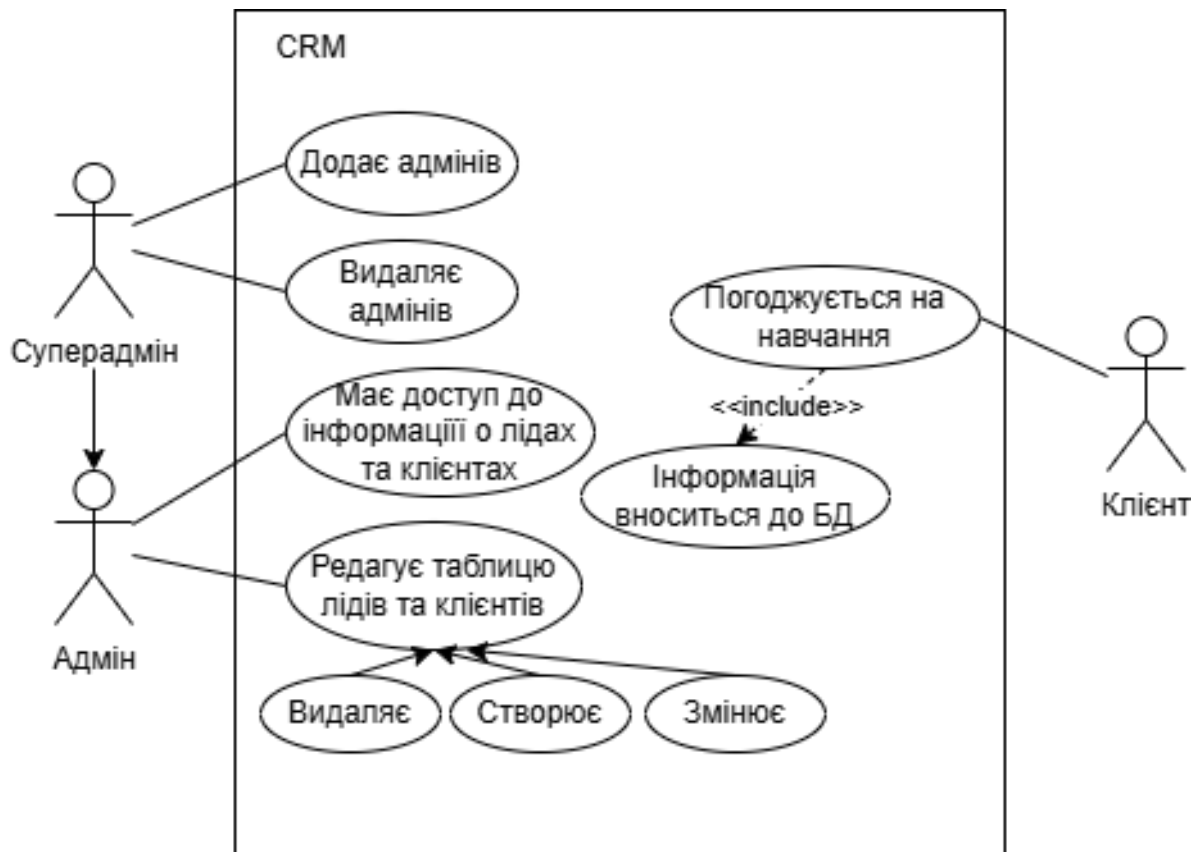


Рисунок 2.3 - Діаграма використання системи

Суперадмін та звичайний адміністратор має найбільше прав і можливостей, оскільки він може керувати всією системою, включаючи додавання, редагування та видалення записів. Клієнти мають обмежений доступ, орієнтований на редагування та перегляд інформації про свої курси. Централізоване зберігання даних у базі даних забезпечує ефективне управління інформацією та її доступність для всіх користувачів системи. Вся інформація, яка вноситься в систему, автоматично зберігається в базі даних (БД). Це забезпечує централізоване зберігання даних та доступ до них для всіх акторів.

На діаграмі зображено три актори (користувача) та їх взаємодія з різним функціоналом системи. Розглянемо кожного актора та випадки використання детально, а саме: адмін (адміністратор), суперадмін, клієнт. [14]

Випадки використання для суперадіміна:

- Додає нових адміністраторів до системи. Це забезпечує розподіл прав доступу та обов'язків між кількома адміністраторами. При наданні прав адміністраторам вони отримують ті самі можливості що і суперадміні, але без можливості додавання та видалення інших.

- Також суперадмін має можливість видаляти інших адміністраторів зі системи, відбираючи у них доступ до CRM системи.

Випадки використання для адміністратора:

- Адміністратор може переглядати інформацію про потенційних студентів (лідів) та клієнтів, що вже зареєстровані в системі. Інформація, яку він може переглядати, має такі базові пункти, як ім'я, прізвище та деякі незвичайні, наприклад: джерело реклами, результат дзвінка та інше.

- Якщо до навчального закладу звернулися нові ліди, адміністратор має можливість створювати нові записи в таблиці лідів та клієнтів додаючи будь-яку інформацію.

- Інформація з часом може змінюватися, тому адміністратор може редагувати дані лідів та клієнтів. Він може змінювати інформацію, видаляти або додавати нові записи. Актуалізувати дані можна використовуючи за основу страї, не потрібно заповнювати усі поля з нуля.

- Адміністратор може видаляти інформацію про лідів та клієнтів, якщо це необхідно.

Випадки використання для клієнта:

- Людина може проконсультуватися з представником навчального закладу, якщо вона зацікавлена у цьому, але ще не вирішила чи хоче вона навчатися, вона вважається лідом. Але якщо вона хоче навчатися та обрала свій курс, групу, формат навчання та надала інші дані, які вносяться у базу даних, вона стає клієнтом.

- Клієнт може попросити актуалізувати інформацію про себе: змінити курс, група, формат навчання та інше. Також можна відмовитися від послуг навчального

закладу, якщо він цього бажає. Але перед цим можуть попросити описати проблему, спитати що не сподобалося.

## 2.4 Класи

Діаграма класів представляє структуру і взаємозв'язки між класами в CRM-системі для навчального закладу.

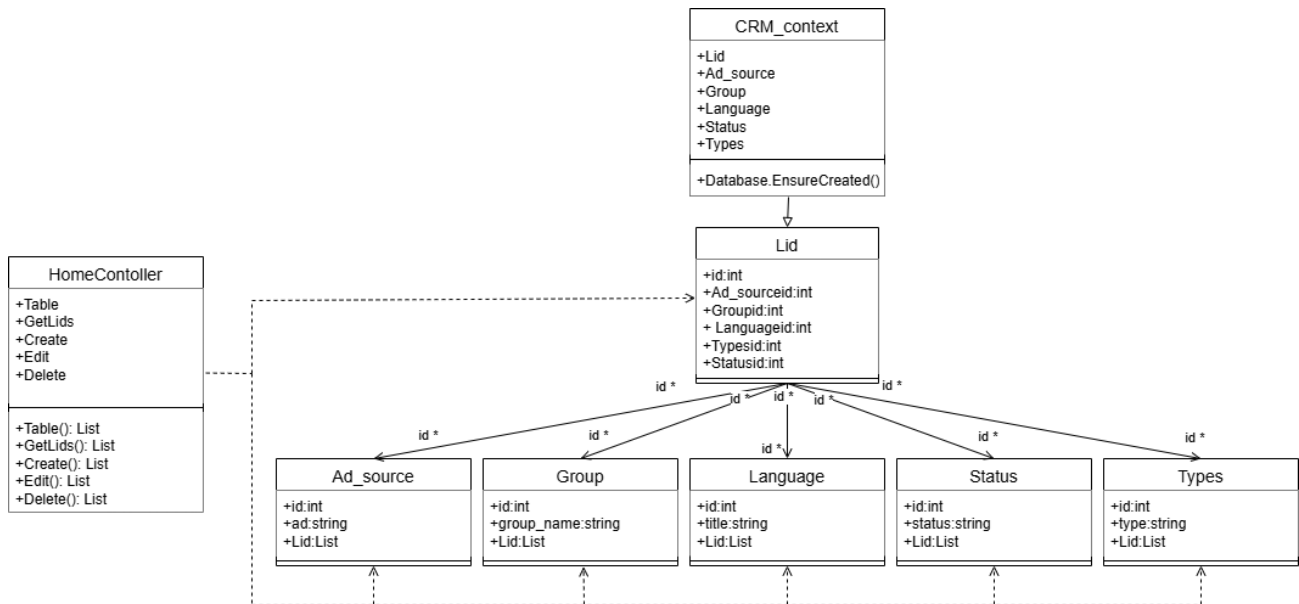


Рисунок 2.4 - Діаграма класів

Розглянемо кожен клас, його атрибути і методи детально.

Клас CRM\_context представляє контекст бази даних для Entity Framework, який відповідає за управління з'єднанням з базою даних і відстеження змін у моделях.

Атрибути:

- Lid (DbSet<Lid>) - набір даних таблиці лідів.
- Ad\_source (DbSet<Ad\_source>) - набір даних таблиці джерел реклами.
- Group (DbSet<Group>) - набір даних таблиці груп.
- Language (DbSet<Language>) - набір даних таблиці мов.
- Status (DbSet<Status>) - набір даних таблиці статусів.

- Types (DbSet<Types>) - набір даних таблиці типів.

Методи:

- Database EnsureCreated() - метод для забезпечення створення бази даних, якщо вона ще не існує.

Клас Lid представляє таблицю лідів у базі даних.

Атрибути:

- id (int) - унікальний ідентифікатор ліда.
- Ad\_sourceId (int) - зовнішній ключ до таблиці Ad\_source.
- GroupId (int) - зовнішній ключ до таблиці Group.
- LanguageId (int) - зовнішній ключ до таблиці Language.
- TypeId (int) - зовнішній ключ до таблиці Types.
- StatusId (int) - зовнішній ключ до таблиці Status.

Клас Ad\_source представляє таблицю джерел реклами.

Атрибути:

- id (int) - унікальний ідентифікатор джерела реклами.
- ad (string) - назва або опис джерела реклами.
- LidList (List<Lid>) - список лідів, пов'язаних з цим джерелом реклами.

Клас Group представляє таблицю груп.

Атрибути:

- id (int) - унікальний ідентифікатор групи.
- group\_name (string) - назва групи.
- LidList (List<Lid>) - список лідів, пов'язаних з цією групою.

Клас Language представляє таблицю мов.

Атрибути:

- id (int) - унікальний ідентифікатор мови.
- title (string) - назва мови.
- LidList (List<Lid>) - список лідів, що вибрали цю мову.

Клас Status представляє таблицю статусів.

Атрибути:

- id (int) - унікальний ідентифікатор статусу.
- status (string) - назва статусу.
- LidList (List<Lid>) - список лідів з цим статусом.

Клас Types представляє таблицю типів лідів.

Атрибути:

- id (int) - унікальний ідентифікатор типу.
- type (string) - назва типу.
- LidList (List<Lid>) - список лідів з цим типом.

Клас HomeController представляє контролер, що відповідає за взаємодію користувачів з CRM-системою через веб-інтерфейс.

Методи:

- Table() - метод для отримання списку таблиць.
- GetLids() - метод для отримання списку лідів.
- Create() - метод для створення нового ліда.
- Edit() - метод для редагування існуючого ліда.
- Delete() - метод для видалення ліда.

Ця діаграма класів показує структуру CRM-системи, де кожен клас відповідає певній таблиці бази даних або функціональності системи. Взаємозв'язки між класами відображають зовнішні ключі в таблицях бази даних, що забезпечує цілісність даних і полегшує навігацію між зв'язаними даними. [15] Клас HomeController надає методи для керування основними CRUD-операціями (створення, читання, оновлення, видалення) для лідів, що забезпечує взаємодію з користувачем через веб-інтерфейс.

## 2.5 Інструкція користувача

Цей розділ описує, як виглядають сторінки CRM системи та як ними користуватися. Цей пункт інструкції допоможе користувачам зрозуміти, як

виглядають різні сторінки системи та як ефективно ними користуватися для управління лідами, клієнтами та курсами. Буде розглянуто кожну сторінку, її інтерфейс, функціональні елементи та перелік дій та функцій, які необхідно виконати для роботи із системою.

На рис. 2.5 зображена мапа сайту. Вона надає структурований огляд основних сторінок і їх функціональних можливостей. Користувачі дізнаються, як переходити між сторінками, які дії можуть виконувати на кожній з них, та як оптимально використовувати інструменти CRM системи для роботи з нею.

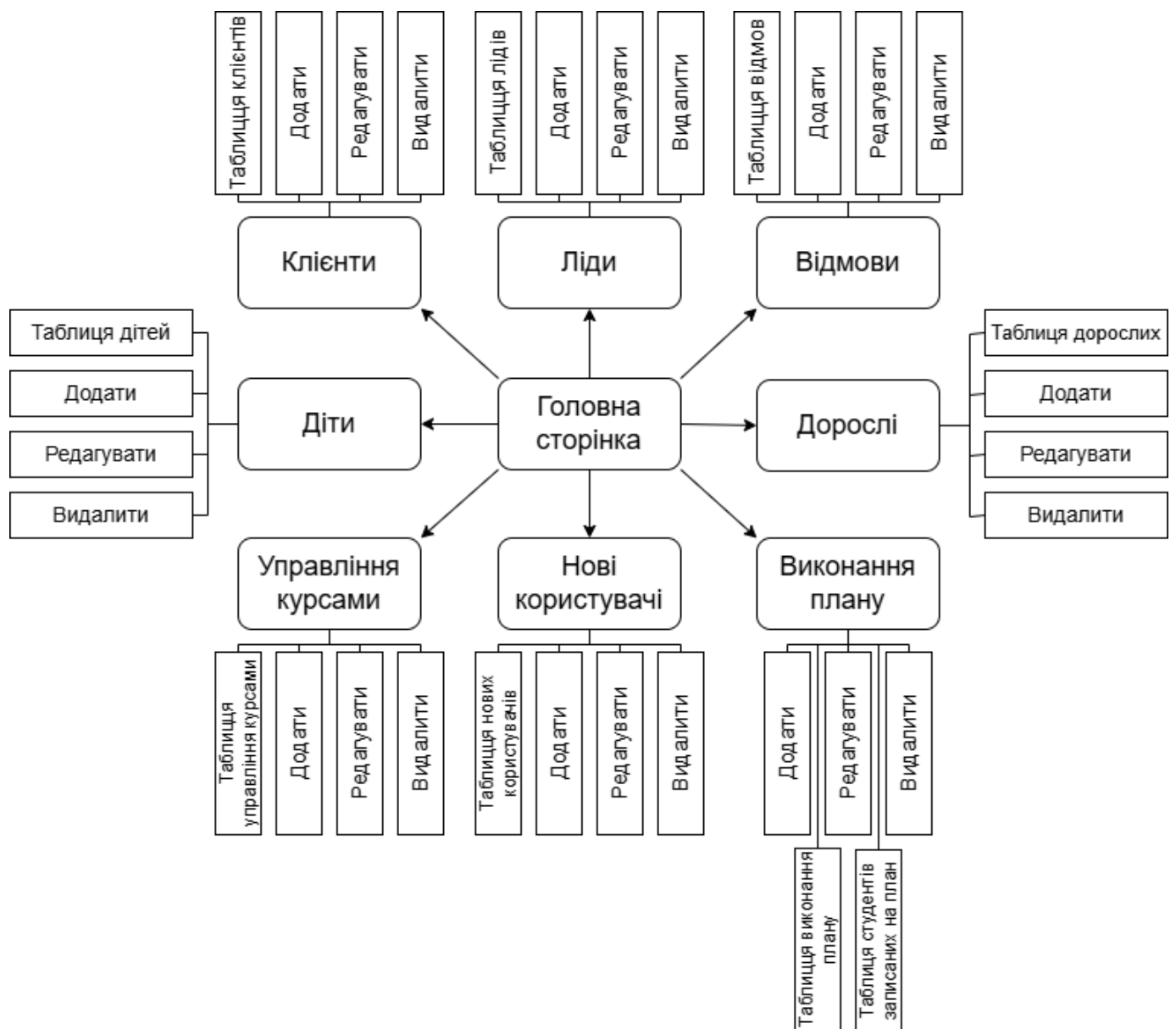


Рисунок 2.5 - Мапа сайту



## 2.5.1 Головна сторінка

Після входу у додаток, відбувається перехід на головну сторінку. На рис. 2.6 наведено скріншот головної сторінки, де виводиться інформація про лідів які були додані у базу даних сьогодні(нові ліди).

The screenshot shows a web application interface. On the left is a blue sidebar menu with the following items: Усі, Ліди, Клієнти, Відмовились, Управління курсами, Статистика, Виконання плану, Нові, Діти, Дорослі. At the top right, there is a yellow button labeled "Додати користувача". Below it is a table with columns: Ім'я, Прізвище, Вік, Статус, Email, Номер телефону, Дата дзвінка, Мова, Форма заняття, Група, Результат дзвінка, Дата повторного дзвінка, Реклама. The table contains six rows of lead data. The 'Статус' column has colored backgrounds: 'відмова' (grey), 'відмова' (grey), 'передзвонити' (purple), 'клієнт' (white), and 'клієнт' (white). Each row has a 'Зміни Вид' link in the final column.

Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	Зміни Вид
Назар	Назаров	20	відмова	пошта	97	08.03.2024 00:00:00	польська	офлайн	індивідуально	+	08.04.2023 00:00:00	Facebook	Зміни Вид
Николай	Сухар	15	відмова	@gmail	345	09.03.2024 11:40:00	німецька	онлайн	група	+	11.03.2024 11:40:00	Facebook	Зміни Вид
Николай	Тополя	30	передзвонити	@gmail	380	17.03.2024 00:36:00	польська	онлайн	індивідуально	Свободен вечером	30.03.2024 23:35:00	TikTok	Зміни Вид
Антончик	fdgfd	34	клієнт	@gmail	356	10.04.2024 18:49:00	англійська	онлайн	група	+	11.04.2024 18:49:00	YouTube	Зміни Вид
Тарас	Бурченко	27	клієнт	@	3	12.05.2024 22:41:00	польська	онлайн	група	+	17.05.2024 22:41:00	YouTube	Зміни Вид

Рисунок 2.6 - Вигляд головної сторінки

На кожній сторінці додатку розташоване головне меню, яке використовується по всіх сторінкам додатку. На головній сторінці розташована таблиця з повною інформацією про лідів та клієнтів. В останній колонці передбачені кнопки для видалення або редагування ліда. Для швидкого аналізу статусів лідів, у відповідність кожному статусу передбачено відповідний колір. Також на даній сторінці передбачено кнопку додавання нового ліда до бази даних.

Після натискання кнопки "Додати контакт" на головній сторінці відкривається вікно для створення нового ліда.

На рисунках 2.7 та 2.8 показана сторінка додавання нового ліда.

**Create**

**Lid**

Ім'я

Прізвище

Вік

Статус

Ел.Пошта

Номер телефону

Дата дзвінка

Дата дзвінка

Мова

Форма заняття

Група

Результат дзвінка

Дата повторного дзвінка

Реклама

Рисунок 2.7 - Додавання нового ліда

Рисунок 2.8 - Продовження сторінки додавання нового ліда

Це вікно складається з полів, які користувач має заповнити основними даними. У вікні потрібно заповнити поля для введення імені, прізвища, віку, статусу, електронної пошти, номера телефону та дати дзвінка, мову занять, форму занять, групу, результат дзвінка, дату повторного дзвінка та джерело реклами. Після заповнення всіх полів і натискання кнопки "Додати" новий лід додається до бази

даних із унікальним id. Це дозволяє надалі працювати з цим контактом, змінювати його інформацію при необхідності та зберігати дані актуальними.

Після натискання кнопки "Редагувати" у відповідній колонці таблиці на головній сторінці відкривається вікно редагування ліда.

**Edit**  
Lid

Ім'я: Назар

Прізвище: Назаров

Вік: 20

Статус: відмова

Ел.Пошта: пошта

Номер телефону: 97

Дата дзвінка: 08.03.2024 00:00

Рисунок 2.9 – Редагування ліда

Дата дзвінка: 08.03.2024 00:00

Мова: польська

Форма заняття: офлайн

Група: індивідуально

Результат дзвінка: +

Дата повторного дзвінка: 08.04.2023 00:00

Реклама: Facebook

**Зберегти**

[Назад до таблиці](#)

Рисунок 2.10 – Продовження сторінки редагування ліда

У цьому вікні можна побачити всі поточні дані користувача, який має унікальний id, що дозволяє легко знайти і вивести його актуальну інформацію з бази даних. Поля вікна редагування аналогічні полям вікна створення нового ліда: ім'я,

прізвище, вік, статус, електронна пошта, номер телефону, дата дзвінка, мова занять, форма занять, група, результат дзвінка, дата повторного дзвінка та джерело реклами.

Користувач може змінити будь-які необхідні поля, не обов'язково змінюючи всю інформацію. Після внесення змін і натискання кнопки "Зберегти" оновлені дані зберігаються в базі даних, і зміни набувають чинності. Це дозволяє зберігати інформацію про лідів актуальною та точною.

Після натискання кнопки "Видалити" відкривається вікно підтвердження видалення ліда.

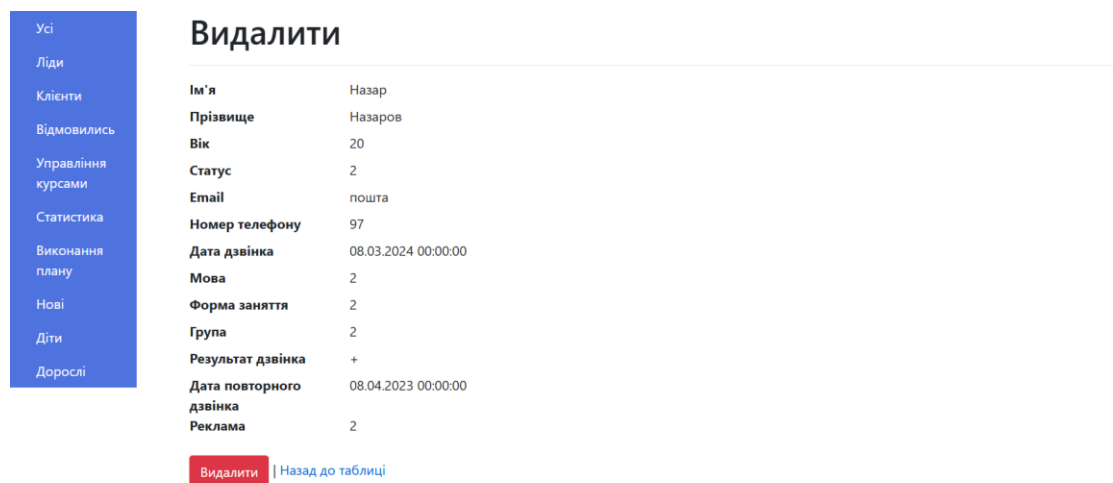


Рисунок 2.11 -Видалення ліда

У цьому вікні відображається вся інформація про ліда: ім'я, прізвище, вік, статус, електронна пошта, номер телефону, дата дзвінка, мова занять, форма занять, група, результат дзвінка, дата повторного дзвінка та джерело реклами.

Користувачу надаються дві кнопки: "Видалити" та "Назад до таблиці". Натискання кнопки "Видалити" призводить до остаточного видалення запису про ліда з бази даних та таблиці, таким чином очищуючи всі пов'язані з ним дані. Кнопка "Назад до таблиці" повертає користувача назад до таблиці без будь-яких змін у даних.

Цей процес забезпечує користувачам можливість перегляду та підтвердження своїх дій перед видаленням, запобігаючи випадковому видаленню важливої інформації.

## 2.5.2 Сторінка лідів

Ліди - це зацікавлені особи, які ще не прийняли остаточного рішення щодо користування послугами закладу, але розглядають таку можливість.

Додати ліда													
Сортувати за: ▾													
Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	
Николай	Тополя	30	передзвонити	@gmail	380	17.03.2024 00:36:00	польська	онлайн	індивідуально	Свободен вечером	30.03.2024 23:35:00	TikTok	<a href="#">Змінити</a> <a href="#">Видали</a>

Рисунок 2.12 - Вигляд сторінки лідів

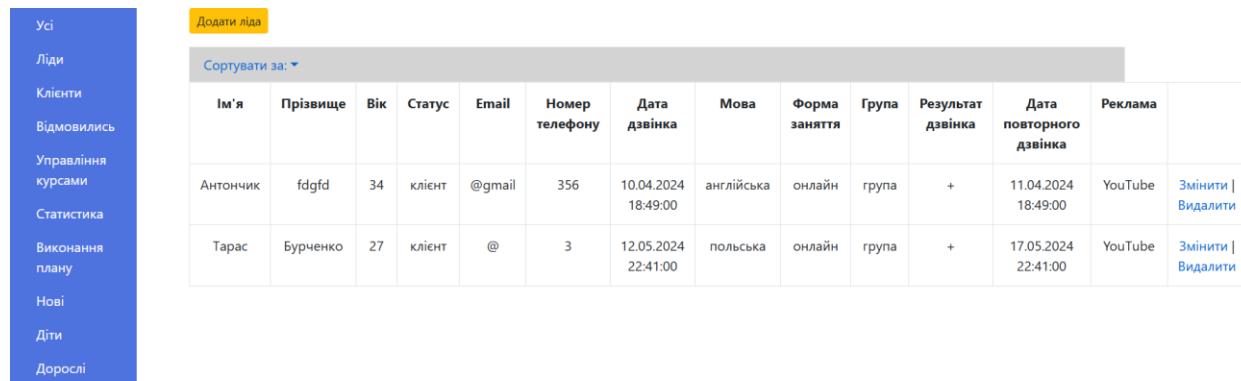
На сторінці лідів відображається детальна інформація про потенційних клієнтів (лідів) навчального закладу. Сторінка забезпечує зручний інтерфейс для перегляду, зміни та управління інформацією про лідів, допомагаючи відрізнити їх від постійних клієнтів.

Кожен лід у таблиці має свій статус, який визначає поточний стан взаємодії з ним. Є чотири основні статуси лідів: очікує, передзвонити, ігнор та провести співбесіду. Для зручності користувачів кожен статус відображається у таблиці різним кольором, що дозволяє швидко орієнтуватися у великій кількості записів. Ця сторінка є важливим інструментом для менеджерів навчального закладу, оскільки вона допомагає ефективно організувати роботу з потенційними клієнтами, надавати актуальну інформацію та вчасно реагувати на їхні потреби.

На сторінці також передбачені кнопки для редагування та видалення лідів, що дозволяє оперативно оновлювати інформацію та підтримувати базу даних у актуальному стані. Кнопка "Додати ліда" дає змогу швидко додавати нових потенційних клієнтів до системи.

### 2.5.3 Сторінка клієнтів

На сторінці клієнтів відображаються дані про осіб, які вже стали клієнтами навчального закладу.



The screenshot shows a web interface for managing clients. On the left is a blue sidebar menu with the following items: Усі, Ліди, Клієнти, Відмовились, Управління курсами, Статистика, Виконання плану, Нові, Діти, and Дорослі. The main content area has a yellow button labeled "Додати ліда" and a dropdown menu "Сортувати за:". Below this is a table with the following data:

Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	
Антончик	fdgfd	34	клієнт	@gmail	356	10.04.2024 18:49:00	англійська	онлайн	група	+	11.04.2024 18:49:00	YouTube	<a href="#">Змінити</a>   <a href="#">Видалити</a>
Тарас	Бурченко	27	клієнт	@	3	12.05.2024 22:41:00	польська	онлайн	група	+	17.05.2024 22:41:00	YouTube	<a href="#">Змінити</a>   <a href="#">Видалити</a>

Рисунок 2.13- Вигляд сторінки клієнтів

Ці клієнти визначилися зі своїм курсом, обрали зручну для них форму заняття та групу. Вони чітко знають, яку мову бажають вивчати, яка форма навчання (онлайн чи оффлайн) їм найбільше підходить, і чи віддають перевагу заняттям у групі або індивідуально.

Сторінка клієнтів забезпечує зручний доступ до актуальної інформації про кожного клієнта, що дозволяє адміністраторам та викладачам швидко отримати необхідні дані для організації навчального процесу. У разі, якщо клієнт передумає або змінить свої вподобання щодо курсу, форми занять чи групи, інформацію завжди можна актуалізувати, зробивши відповідні зміни у базі даних.

Ця сторінка є важливим інструментом для управління навчальним процесом, оскільки дозволяє ефективно координувати заняття, стежити за прогресом кожного клієнта та забезпечувати індивідуальний підхід до навчання.

#### 2.5.4 Сторінка відмов

Сторінка відмов містить інформацію про всіх осіб, які вирішили відмовитися від навчання у закладі.

Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	
Назар	Назаров	20	відмова	пошта	97	08.03.2024 00:00:00	польська	офлайн	індивідуально	+	08.04.2023 00:00:00	Facebook	<a href="#">Змінити</a>   <a href="#">Видалити</a>
Николай	Сухар	15	відмова	@gmail	345	09.03.2024 11:40:00	німецька	онлайн	група	+	11.03.2024 11:40:00	Facebook	<a href="#">Змінити</a>   <a href="#">Видалити</a>

Рисунок 2.14 - Вигляд сторінки відмов

Ця сторінка є важливим інструментом для аналізу причин відмов і виявлення можливих проблем чи недоліків у процесі надання освітніх послуг. На цій сторінці збираються дані про кожного, хто відмовився, з метою зрозуміти причини їх рішення та визначити, що саме їм не сподобалося.

Ця інформація є цінною для закладу, оскільки дозволяє виявити тенденції та повторювані проблеми, які можуть негативно впливати на задоволення потенційних клієнтів. Використання зібраних даних як статистики допоможе адміністрації приймати обґрунтовані рішення щодо покращення навчального процесу, сервісу та інших аспектів діяльності закладу.

Завдяки сторінці відмов, навчальний заклад має можливість постійно вдосконалювати свої послуги, адаптуватися до потреб і очікувань клієнтів, що, в свою чергу, може сприяти зниженню кількості відмов у майбутньому і підвищенню загального рівня задоволеності клієнтів.

### 2.5.5 Сторінка управління курсами

На сторінці управління курсами відображається інформація про всі курси, які доступні у навчальному закладі.

Сортувати за: ▾	
Мова	
англійська	<a href="#">Змінити курс</a>   <a href="#">Видалити курс</a>
польська	<a href="#">Змінити курс</a>   <a href="#">Видалити курс</a>
німецька	<a href="#">Змінити курс</a>   <a href="#">Видалити курс</a>
японська	<a href="#">Змінити курс</a>   <a href="#">Видалити курс</a>

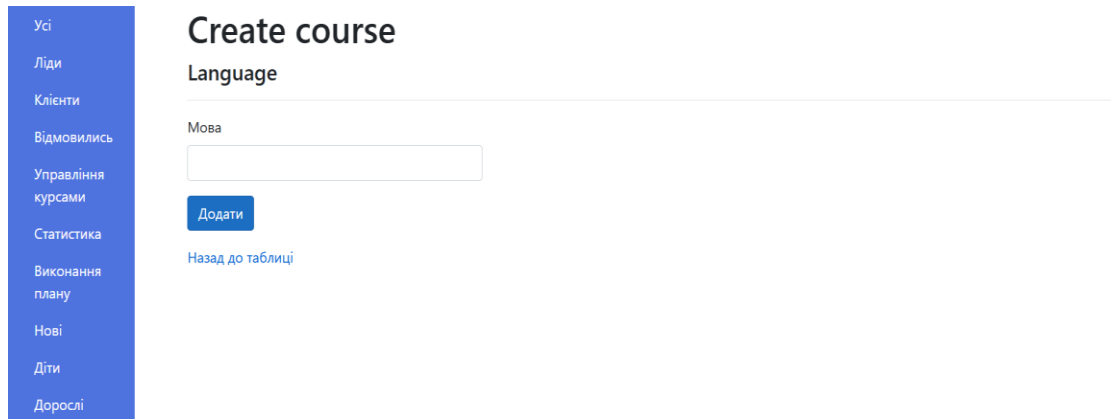
Рисунок 2.15 - Вигляд сторінки управління курсами

Ці курси використовуються при створенні та редагуванні записів про лідів. Адже навчальний заклад постійно, що означає збільшення кількості навчальних програм і курсів, які він пропонує.

Сторінка управління курсами забезпечує адміністраторам можливість легко додавати нові курси, редагувати наявні або видаляти ті, які більше не актуальні. Завдяки зручному інтерфейсу, адміністрація може підтримувати актуальність і якість навчальних програм, забезпечуючи гнучкість у їх управлінні.



У вікні додавання нового курсу можна створити новий навчальний курс, який буде внесений у базу даних та доступний для використання у всьому проекті. Це вікно містить поле, в яке треба ввести назву нового курсу.



The screenshot shows a web interface for creating a new course. On the left is a vertical blue sidebar with white text links: 'Усі', 'Ліди', 'Клієнти', 'Відмовились', 'Управління курсами', 'Статистика', 'Виконання плану', 'Нові', 'Діти', and 'Дорослі'. The main area is titled 'Create course' in bold. Below the title is the label 'Language' and a text input field. Under the input field is a blue button with the text 'Додати'. Below the button is a blue link that says 'Назад до таблиці'.

Рисунок 2.16 - Додавання нового курсу

Після заповнення всіх необхідних полів і підтвердження додавання, новий курс автоматично з'являється у списку доступних курсів на сторінці управління курсами. Він стає доступним для вибору при створенні та редагуванні записів про лідів, що дозволяє адміністраторам зручно інтегрувати нові освітні програми в роботу навчального закладу.

У вікні редагування курсу можна змінювати інформацію про існуючий навчальний курс, який вже внесений у базу даних та використовується у всьому проекті. Після внесення і підтвердження змін, оновлена інформація автоматично зберігається у базі даних і відображається у всіх відповідних розділах системи.

На рисунку нижче буде показана сторінка редагування курсу.

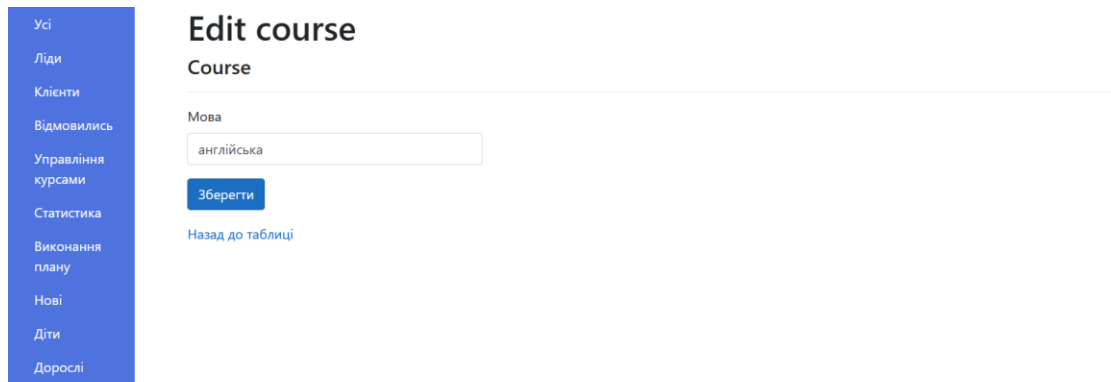


Рисунок 2.17 - Редагування курсу

У цьому вікні можна видалити існуючий курс, який прибирає його із бази даних і таблиці, якщо цей курс більше не використовується у навчальному закладі.

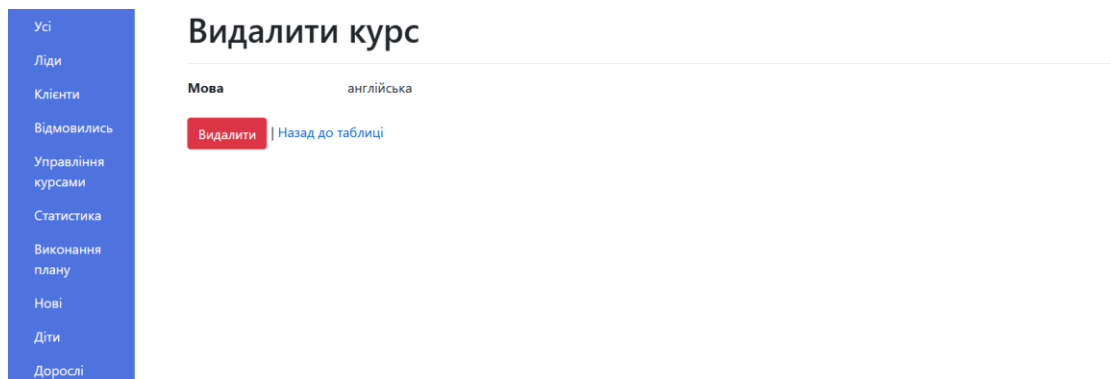


Рисунок 2.18 -Видалення курсу

У вікні є дві основні кнопки: "Видалити" та "Назад до таблиці". Натискання на кнопку "Видалити" підтверджує видалення курсу, що призводить до його повного видалення з бази даних і усіх пов'язаних таблиць. Кнопка "Назад до таблиці" дозволяє

адміністратору повернутися назад до таблиці курсів без здійснення будь-яких змін. Це необхідно для підтримання актуальності та організованості бази даних навчального закладу.

### 2.5.6 Сторінка виконання плану

На сторінці виконання плану відображається інформація про прогрес виконання плану набору студентів на різні курси в навчальному закладі.

Сортувати за: ▼			
Назва курсу	План	Фактичне виконання	
англійська	14	1	Змінити   Видалити
німецька	12	0	Змінити   Видалити
японська	3	0	Змінити   Видалити
польська	3	1	Змінити   Видалити

Додати план

Рисунок 2.19 - Вигляд сторінки виконання плану

Кожен курс має два види плану: теоретичний і дійсний.

Теоретичний план представляє собою кількість студентів, яку необхідно набрати на курс для його успішного запуску та проведення. Це встановлена мета, що відображає ідеальну кількість учасників, які повинні записатися на курс.

Дійсний план відображає фактичну кількість студентів, які вже записалися на курс і мають статус клієнта. Коли студент отримує статус клієнта і записується на курс, система автоматично зараховує його до дійсного плану. Таким чином, дійсний план відображає реальну кількість студентів, які активно беруть участь у курсі.

Ця сторінка надає адміністраторам навчального закладу чітке уявлення про те, наскільки успішно виконуються плани набору студентів. Вони можуть легко порівнювати теоретичні та дійсні показники.

У цьому вікні можна додати новий план, який вноситься у базу даних і використовується у всьому проекті.

Усі  
Ліди  
Клієнти  
Відмовились  
Управління курсами  
Статистика  
Виконання плану  
Нові  
Діти  
Дорослі

## Create Plan

Plan

Курс  
англійська

План

Додати

[Повернутися до таблиці](#)

Рисунок 2.20 - Додавання нового плану

Для створення плану необхідно обрати існуючий курс та вказати теоретичну кількість осіб на цей курс, фактична кількість автоматично зараховується, якщо людина зі статусом клієнта записана на цей курс.

У цьому вікні можна редагувати існуючий план, зміна якого вноситься у базу даних і використовується у всьому проекті. Після внесення і підтвердження змін, оновлена інформація автоматично зберігається у базі даних і відображається у всіх відповідних розділах системи.

На рисунку нижче показана робота сторінки редагування плану.

Усі  
Ліди  
Клієнти  
Відмовились  
Управління курсами  
Статистика  
Виконання плану  
Нові  
Діти  
Дорослі

## Edit Plan

### Plan

Курс  
англійська

План  
14

[Зберегти](#)

[Назад до таблиці](#)

Рисунок 2.21 - Редагування плану

У цьому вікні можна видалити план, записи якого прибираються із бази даних. При видаленні плану, курс залишається без змін. У вікні є дві основні кнопки: "Видалити" та "Назад до таблиці".

Усі  
Ліди  
Клієнти  
Відмовились  
Управління курсами  
Статистика  
Виконання плану  
Нові  
Діти  
Дорослі

## Видалити

Курс	1
План	14

[Видалити](#) | [Назад до таблиці](#)

Рисунок 2.22 -Видалення плану

При видаленні плану, курс залишається без змін. У вікні є дві основні кнопки: "Видалити" та "Назад до таблиці". Натискання на кнопку "Видалити" підтверджує

видалення курсу, що призводить до його повного видалення з бази даних і усіх пов'язаних таблиць. Кнопка "Назад до таблиці" дозволяє адміністратору повернутися назад до таблиці курсів без здійснення будь-яких змін.

### 2.5.7 Сторінка нових користувачів

На цій сторінці відображається інформація про нових лідів.

Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	
Олег	Прутько	18	клієнт	@gmail	38097	22.05.2024 01:35:00	польська	онлайн	група	+	23.05.2024 01:35:00	Telegram	<a href="#">Змінити</a>   <a href="#">Видалити</a>

Рисунок 2.23 - Вигляд сторінки нових користувачів

Це особи, які були додані у систему сьогодні. Сторінка допомагає адмінам зрозуміти, скільки осіб було зареєстровано на цей день і що їх треба обробити (редагувати).

### 2.5.8 Сторінка дітей

На цій сторінці відображається інформація про неповнолітніх лідів. Це особи, яким менше шістнадцяти років. Сторінка допомагає адмінам зорієнтуватися у великій кількості записів про лідів та знайти дітей. Вони мають іншу програму навчання та групи і не можуть самостійно записатися на курс.

Додати люда													
Сортувати за: ▼													
Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	
Николай	Сухар	15	відмова	@gmail	345	09.03.2024 11:40:00	німецька	онлайн	група	+	11.03.2024 11:40:00	Facebook	<a href="#">Змінити</a>   <a href="#">Видалити</a>

Рисунок 2.24 - Вигляд сторінки дітей

## 2.5.9 Сторінка дорослих

На цій сторінці відображається інформація про повнолітніх лідів.

Додати люда													
Сортувати за: ▼													
Ім'я	Прізвище	Вік	Статус	Email	Номер телефону	Дата дзвінка	Мова	Форма заняття	Група	Результат дзвінка	Дата повторного дзвінка	Реклама	
Назар	Назаров	20	відмова	пошта	97	08.03.2024 00:00:00	польська	офлайн	індивідуально	+	08.04.2023 00:00:00	Facebook	<a href="#">Змі</a> <a href="#">Вид</a>
Николай	Тополя	30	передзвонити	@gmail	380	17.03.2024 00:36:00	польська	онлайн	індивідуально	Свободен вечером	30.03.2024 23:35:00	TikTok	<a href="#">Змі</a> <a href="#">Вид</a>
Антончик	fdgfd	34	клієнт	@gmail	356	10.04.2024 18:49:00	англійська	онлайн	група	+	11.04.2024 18:49:00	YouTube	<a href="#">Змі</a> <a href="#">Вид</a>
Тарас	Бурченко	27	клієнт	@	3	12.05.2024 22:41:00	польська	онлайн	група	+	17.05.2024 22:41:00	YouTube	<a href="#">Змі</a> <a href="#">Вид</a>
Олег	Прутько	18	клієнт	@gmail	38097	22.05.2024 01:35:00	польська	онлайн	група	+	23.05.2024 01:35:00	Telegram	<a href="#">Змі</a> <a href="#">Вид</a>

Рисунок 2.25 - Вигляд сторінки дорослих

Це особи, яким більше шістнадцяти років, і вони можуть самостійно записуватися на курси. В останній колонці таблиці знаходяться кнопки для редагування та видалення користувача. Ця сторінка створена для зручного управління дорослими користувачами, які зацікавлені у курсах навчального закладу.

Висновки до розділу:

Цей розділ присвячений проектуванню CRM системи, який розглядає ключові аспекти реалізації проекту, від вибору технологій до детального опису інтерфейсів користувача.

Для реалізації проекту були обрані технології, які забезпечать ефективність, продуктивність та масштабованість системи. Було обрано сучасні та перевірені інструменти, такі як ASP.NET Core, EntityFramework, Bootstrap та інші технології, які дозволяють створити надійний і гнучкий продукт. Також для побудови бази даних проекту був використаний SQL Server Management Studio, який необхідний для проектування та розробки структури БД.

Були розглянуті варіанти використання системи, де окреслені основні сценарії, за якими користувачі взаємодіють із CRM системою. Розглянуті класи їх роль та функціонал у системі, включаючи взаємодію з базою даних та обробку запитів користувачів.

Також була створена інструкція користувач, яка надає докладний опис кожної сторінки системи, її інтерфейсу та функціональних елементів. Це допоможе користувачам швидко освоїти систему і ефективно використовувати її для управління лідами, клієнтами та курсами.



## ВИСНОВКИ

Розробка CRM-системи для навчального закладу є важливим етапом у вдосконаленні управління взаємодією зі студентами, викладачами та іншими зацікавленими сторонами. На основі аналізу потреб та вимог навчального закладу було визначено, що така система повинна включати в себе модулі для керування контактами, управління клієнтськими даними, ведення курсів та програм, аналітику та звітність, а також можливості автоматизації деяких рутинних процесів.

Розробка та впровадження CRM-системи для навчального закладу є важливим кроком у напрямку модернізації освітніх послуг. У майбутньому планується розширення функціоналу системи за рахунок інтеграції з іншими сервісами, використання штучного інтелекту для персоналізації навчального процесу та впровадження нових інструментів для аналізу даних.

У результаті розробки CRM-системи для навчального закладу очікується покращення ефективності управління студентською базою, підвищення задоволеності користувачів та оптимізація процесів взаємодії зі студентами та іншими зацікавленими сторонами. Така система допоможе навчальному закладу забезпечити більш якісне та персоналізоване обслуговування своїх клієнтів, а також відповідати вимогам сучасного освітнього середовища.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Judith W. Kincaid. Customer Relationship Management: Getting It Right!. Upper Saddle River, N.J.: Prentice Hall PTR, 2003. 480 pp.
2. CRM-технології як інструмент управління вищими навчальними закладами / Л.М. Папенко, О.В. Нищенко. - Київ: Київський національний університет технологій та дизайну, 2014. - 247-251 с.
3. Salesforce CRM. Веб-сайт. URL: <https://www.salesforce.com/crm/> (дата звернення 15.05.2024)
4. HubSpot CRM. Веб-сайт. URL: <https://www.hubspot.com/products/crm> (дата звернення 15.05.2024)
5. Microsoft Dynamics 365. Веб-сайт. URL: <https://learn.microsoft.com/uk-ua/dynamics365/> (дата звернення 15.05.2024)
6. Monday CRM. Веб-сайт. URL: <https://monday.com/> (дата звернення 16.05.2024)
7. Zoho CRM. Веб-сайт. URL: <https://www.zoho.com/ru/crm/lp/crm-software.html> (дата звернення 16.05.2024)
8. Key CRM. Веб-сайт. URL: <https://ua.keycrm.app/>(дата звернення 16.05.2024)
9. Nguyen, T. H., Sherif, J. S., & Newby, M. (2007). Strategies for successful CRM implementation. Information Management & Computer Security, 15(2), 102–115.
10. Короткевич. Д. SQL Server. Налаштування та оптимізація для фахівців. Київ: Лідер-Букс, 2017.24 - 30 с.
11. MVC Design Pattern in ASP.NET Core: веб-сайт. URL: <https://www.tektutorialshub.com/asp-net-core/asp-net-core-mvc-design-pattern/> (дата звернення 20.05.2024).
12. Get started with Bootstrap. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>(дата звернення 20.05.2024).
13. Entity Framework documentation. URL: <https://docs.microsoft.com/en-us/ef/> (дата звернення 20.05.2024).

14. Інструкція, як будувати UML-діаграми. URL: <https://dou.ua/forums/topic/40575/>(дата звернення 22.05.2024).
15. Діаграма класів. URL: <https://tinyurl.com/bddketu9> (дата звернення 22.05.2024).
16. Моркун Н. В., Маринич І. А. Методичні вказівки до виконання кваліфікаційної роботи бакалавру для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2019. 50 с.
17. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).
18. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).
19. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація).
20. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с. (Інформація та документація).

## ДОДАТОК А

## Лістинг коду контролера

На рисунках нижче буде показано код контролера HomeController, який відповідає за обробку запитів, що надходять до домашньої сторінки або інших сторінок, пов'язаних з головною частиною веб-додатку. Він забезпечує зв'язок між запитами користувачів та представленням.

```

4  using System;
5  using System.Collections.Generic;
6  using System.Diagnostics;
7  using System.Linq;
8  using System.Threading.Tasks;
9  using Microsoft.EntityFrameworkCore;
10 using ASP_TEST.Data;
11 using ASP_TEST.Views;
12 using Microsoft.AspNetCore.Mvc.Rendering;
13
14 namespace ASP_TEST.Controllers
15 {
16     Ссылка: 3
17     public class HomeController : Controller
18     {
19         private readonly ILogger<HomeController> _logger;
20         private readonly crmDbContext context;
21
22         Ссылка: 0
23         public HomeController(ILogger<HomeController> logger, crmDbContext context)
24         {
25             _logger = logger;
26             this.context = context;
27         }
28
29         Ссылка: 3
30         public IActionResult Table()
31         {
32             IEnumerable<Lid> newLid = context.Lid.Include("Status").Include("Group").Include("Language").Include("Ad_source").Include("Types").Where(p => EF.Functions.DateDif
33             return View(newLid);
34         }
35     }
36 }

```

Рисунок А.1 - Контекст БД та метод головної сторінки Table

```

60     Ссылка: 3
61     public IActionResult CourseMng()
62     {
63         IEnumerable<Language> course = context.Language.ToList();
64         return View("CourseTable", course);
65     }
66
67     Ссылка: 0
68     public IActionResult GetAllLids()
69     {
70         IEnumerable<Lid> allLid = context.Lid.Include("Status").Include("Group").Include("Language").Include("Ad_source").Include("Types").ToList();
71         return View("Table", allLid);
72     }
73
74     Ссылка: 0
75     public IActionResult GetLids()
76     {
77         IEnumerable<Lid> lid = context.Lid.Include("Status").Include("Group").Include("Language").Include("Ad_source").Include("Types").Where(x => x.Statusid != 2 && x.St
78         return View("Table", lid);
79     }
80
81     Ссылка: 0
82     public IActionResult GetNewLids()
83     {
84         IEnumerable<Lid> newLid = context.Lid.Include("Status").Include("Group").Include("Language").Include("Ad_source").Include("Types").Where(p => EF.Functions.DateDif
85         return View("Table", newLid);
86     }
87
88     Ссылка: 0
89     public IActionResult GetClientLids()
90     {
91         //ViewBag.client = clientId;
92         IEnumerable<Lid> clientLid=context.Lid.Include("Status").Include("Group").Include("Language").Include("Ad_source").Include("Types").Where(x => x.Statusid == 1).To
93         return View("Table", clientLid);
94     }
95
96     Ссылка: 0
97     public IActionResult GetRefuseLids()
98     {
99         IEnumerable<Lid> refuseLid = context.Lid.Include("Status").Include("Group").Include("Language").Include("Ad_source").Include("Types").Where(x => x.Statusid == 2).
100         return View("Table", refuseLid);
101     }

```

Рисунок А.2 - Методи, які повертають таблиці CRM системи

```

94      //CREATE
95      Ссылка 0
96      public IActionResult Create()
97      {
98          ViewData["ad_Sources"] = new SelectList(context.Ad_source, "id", "ad");
99          ViewData["Groups"] = new SelectList(context.Group, "id", "group_name");
100         ViewData["Languages"] = new SelectList(context.Language, "id", "title");
101         ViewData["Status"] = new SelectList(context.Status, "id", "status");
102         ViewData["Types"] = new SelectList(context.Types, "id", "type");
103         return View();
    }

```

Рисунок А.3 - Метод Create

```

104      [HttpPost]
105      [ValidateAntiForgeryToken]
106      Ссылка 0
107      public async Task<IActionResult> Create( Lid lid)
108      {
109          if (ModelState.IsValid)
110          {
111              context.Lid.Add(lid);
112              await context.SaveChangesAsync();
113              return RedirectToAction(nameof(Table));
114          }
115          return View(lid);
116      }

```

Рисунок А.4 – Продовження методу Create

```
140 //Edit
141 Ссылка: 0
142 public async Task<IActionResult> Edit(int? id)
143 {
144     ViewData["ad_Sources"] = new SelectList(context.Ad_source, "id", "ad");
145     ViewData["Groups"] = new SelectList(context.Group, "id", "group_name");
146     ViewData["Languages"] = new SelectList(context.Language, "id", "title");
147     ViewData["Status"] = new SelectList(context.Status, "id", "status");
148     ViewData["Types"] = new SelectList(context.Types, "id", "type");
149     if (id == null || context.Lid == null)
150     {
151         return NotFound();
152     }
153     var lid = await context.Lid.FindAsync(id);
154     if (lid == null)
155     {
156         return NotFound();
157     }
158     return View(lid);
159 }
160
```

Рисунок А.5 - Метод Edit

```
163 |
164 |
165 | <
166 | <
167 | <
168 | |
169 | |
170 | |
171 | |
172 | <
173 | <
174 | <
175 | |
176 | |
177 | |
178 | |
179 | |
180 | |
181 | <
182 | <
183 | |
184 | |
185 | <
186 | |
187 | |
188 | |
189 | |
190 | |
191 | |
192 | |
193 | |
```

```
[HttpPost]
[ValidateAntiForgeryToken]
Ссылка 0
public async Task<IActionResult> Edit(int id, Lid lid)
{
    if (id != lid.id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            context.Update(lid);
            await context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!LidExists(lid.id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Table));
    }
    return View(lid);
}
```

Рисунок А.6 – продолжения метода Edit

```

245 //Delete
246 Ссылка: 0
247 public async Task<IActionResult> Delete(int? id)
248 {
249     if (id == null || context.Lid == null)
250     {
251         return NotFound();
252     }
253     var lid = await context.Lid
254         .FirstOrDefaultAsync(m => m.id == id);
255     if (lid == null)
256     {
257         return NotFound();
258     }
259     return View(lid);
260 }
261

```

Рисунок А.7 - Метод Delete

```

263 [HttpPost, ActionName("Delete")]
264 [ValidateAntiForgeryToken]
265 Ссылка: 0
266 public async Task<IActionResult> DeleteConfirmed(int id)
267 {
268     if (context.Lid == null)
269     {
270         return Problem("Entity set 'CRM_context.User' is null.");
271     }
272     var lid = await context.Lid.FindAsync(id);
273     if (lid != null)
274     {
275         context.Lid.Remove(lid);
276     }
277     await context.SaveChangesAsync();
278     return RedirectToAction(nameof(Table));
279 }

```

Рисунок А.8 – продолжения методу Delete



## ДОДАТОК Б

## Лістинг коду сторінок

Нижче буде показаний код представлення або сторінки моєї CRM системи. Як приклад я наведу такі сторінки: Table, Create, Edit, Delete. Це базові сторінки, які необхідні для управління даними користувачів.

```

1  @model IEnumerable<Lid>
2  @{
3
4  }
5  <p>
6  <a asp-action="Create"></a>
7  @Html.ActionLink("Додати ліда", "Create", routeValues: null, htmlAttributes: new { @class = "btn btn-sm btn-warning", id = "createBtn" })
8  </p>
9
10 <View>
11 <ul class="nav nav-tabs" style="background-color: lightgrey">
12 <li class="nav-item dropdown">
13 <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">Сортувати за:</a>
14 <div class="dropdown-menu">
15 <a class="dropdown-item" href="#">Нові(за сьогодні)</a>
16 <a class="dropdown-item" href="#">Невідмічені</a>
17 </div>
18 </li>
19 </ul>
20 <table border="1" class="table table-bordered">
21 <tr class="text-center">
22 <th class="col-4">Ім'я</th>
23 <th class="col-4">Прізвище</th>
24 <th class="col-4">Вік</th>
25 <th class="col-4">Статус</th>
26 <th class="col-4">Email</th>
27 <th class="col-4">Номер телефону</th>
28 <th class="col-4">Дата дзвінка</th>
29 <th class="col-4">Нова</th>
30 <th class="col-4">Форма заняття</th>
31 <th class="col-4">Група</th>
32 <th class="col-4">Результат дзвінка</th>
33 <th class="col-4">Дата повторного дзвінка</th>

```

Рисунок Б.1 - Лістинг сторінки таблиці Table

```

34     <th class="col-4">Реклама</th>
35 </tr>
36 @foreach (var item in Model)
37 {
38     <tr class="text-center">
39         <td>
40             @Html.DisplayFor(modelItem => item.firstname)
41         </td>
42         <td>
43             @Html.DisplayFor(modelItem => item.surname)
44         </td>
45         <td>
46             @Html.DisplayFor(modelItem => item.age)
47         </td>
48         <td class="@GetBackgroundColor(item.Statusid.ToString())">
49             @Html.DisplayFor(modelItem => item.Status.status)
50         </td>
51         <td>
52             @Html.DisplayFor(modelItem => item.email)
53         </td>
54         <td>
55             @Html.DisplayFor(modelItem => item.phone_number)
56         </td>
57         <td>
58             @Html.DisplayFor(modelItem => item.date_call)
59         </td>
60         <td>
61             @Html.DisplayFor(modelItem => item.Language.title)
62         </td>
63         <td>
64             @Html.DisplayFor(modelItem => item.Types.type)
65         </td>
66         <td>
67             @Html.DisplayFor(modelItem => item.Group.group_name)
68         </td>
69         <td>
70             @Html.DisplayFor(modelItem => item.result)
71         </td>
72         <td>
73             @Html.DisplayFor(modelItem => item.date_recall)
74         </td>
75         <td>
76             @Html.DisplayFor(modelItem => item.Ad_source.ad)
77         </td>
78         <td>
79             <a asp-action="Edit" asp-route-id="@item.id">Змінити</a> |
80             <a asp-action="Delete" asp-route-id="@item.id">Видалити</a>
81         </td>
82     </tr>
83 }
84 </table>
85 </View>

```

Рисунок Б.2 – Продовження лістингу сторінки таблиці Table

```

1  @model ASP_TEST.Models.Lid
2
3  @{
4      ViewData["Title"] = "Create";
5  }
6
7  <h1>Create</h1>
8
9  <h4>Lid</h4>
10 <hr />
11 <div class="row">
12     <div class="col-md-4">
13         <form asp-action="Create">
14             <div asp-validation-summary="ModelOnly" class="text-danger"></div>
15             <div class="form-group">
16                 <label class="control-label">Ім'я</label>
17                 <input asp-for="firstname" class="form-control" />
18                 <span asp-validation-for="firstname" class="text-danger"></span>
19             </div>
20             <div class="form-group">
21                 <label class="control-label">Прізвище</label>
22                 <input asp-for="surname" class="form-control" />
23                 <span asp-validation-for="surname" class="text-danger"></span>
24             </div>
25             <div class="form-group">
26                 <label class="control-label">Вік</label>
27                 <input asp-for="age" class="form-control" />
28                 <span asp-validation-for="age" class="text-danger"></span>
29             </div>
30             <div class="form-group">
31                 <label class="control-label">Статус</label>
32                 <select asp-for="Statusid" class="form-control" asp-items="ViewBag.Status"></select>
33                 <span asp-validation-for="Statusid" class="text-danger"></span>

```

Рисунок Б.3 - Лістинг сторінки створення даних Create

```

34 </div>
35 <div class="form-group">
36   <label class="control-label">Эл.Пошта</label>
37   <input asp-for="email" class="form-control" />
38   <span asp-validation-for="email" class="text-danger"></span>
39 </div>
40
41 <div class="form-group">
42   <label class="control-label">Номер телефону</label>
43   <input asp-for="phone_number" class="form-control" />
44   <span asp-validation-for="phone_number" class="text-danger"></span>
45 </div>
46
47 <div class="form-group">
48   <label class="control-label">Дата дзвінка</label>
49   <input asp-for="date_call" class="form-control" />
50   <span asp-validation-for="date_call" class="text-danger"></span>
51 </div>
52
53 <div class="form-group">
54   <label class="control-label">Мова</label>
55   <select asp-for="Languageid" class="form-control" asp-items="ViewBag.Languages"></select>
56   <span asp-validation-for="Languageid" class="text-danger"></span>
57 </div>
58
59 <div class="form-group">
60   <label class="control-label">Форма заняття</label>
61   <select asp-for="Typesid" class="form-control" asp-items="ViewBag.Types"></select>
62   <span asp-validation-for="Typesid" class="text-danger"></span>
63 </div>
64
65 <div class="form-group">
66   <label class="control-label">Група</label>

```

Рисунок Б.4 – Продовження лістингу сторінки створення даних Create

```

67     <select asp-for="Groupid" class="form-control" asp-items="ViewBag.Groups"></select>
68     <span asp-validation-for="Groupid" class="text-danger"></span>
69 </div>
70
71 <div class="form-group">
72     <label class="control-label">Результат дзвінка</label>
73     <input asp-for="result" class="form-control" />
74     <span asp-validation-for="result" class="text-danger"></span>
75 </div>
76
77 <div class="form-group">
78     <label class="control-label">Дата повторного дзвінка</label>
79     <input asp-for="date_recall" class="form-control" />
80     <span asp-validation-for="date_recall" class="text-danger"></span>
81 </div>
82
83 <div class="form-group">
84     <label class="control-label">Реклама</label>
85     <select asp-for="Ad_sourceid" class="form-control" asp-items="ViewBag.ad_Sources"></select>
86     <span asp-validation-for="Ad_sourceid" class="text-danger"></span>
87 </div>
88 <div class="form-group">
89     <input type="submit" value="Додати" class="btn btn-primary" />
90 </div>
91 </form>
92 </div>
93 </div>
94
95 <div>
96     <a asp-action="Table">Назад до таблиці</a>
97 </div>

```

Рисунок Б.5 – Завершення лістингу сторінки створення даних Create

```

1  @model ASP_TEST.Models.Lid
2
3  @{
4      ViewData["Title"] = "Edit";
5  }
6
7  <h1>Edit</h1>
8
9  <h4>Lid</h4>
10 <hr />
11 <div class="row">
12     <div class="col-md-4">
13         <form asp-action="Edit">
14             <div asp-validation-summary="ModelOnly" class="text-danger"></div>
15             <input type="hidden" asp-for="id" />
16             <div class="form-group">
17                 <label class="control-label">Имя</label>
18                 <input asp-for="firstname" class="form-control" />
19                 <span asp-validation-for="firstname" class="text-danger"></span>
20             </div>
21             <div class="form-group">
22                 <label class="control-label">Прізвище</label>
23                 <input asp-for="surname" class="form-control" />
24                 <span asp-validation-for="surname" class="text-danger"></span>
25             </div>
26             <div class="form-group">
27                 <label class="control-label">Вік</label>
28                 <input asp-for="age" class="form-control" />
29                 <span asp-validation-for="age" class="text-danger"></span>
30             </div>
31             <div class="form-group">
32                 <label class="control-label">Статус</label>
33                 <select asp-for="Statusid" class="form-control" asp-items="ViewBag.Status"></select>

```

Рисунок Б.6 - Лістинг сторінки редагування даних Edit

```

34     <span asp-validation-for="Statusid" class="text-danger"></span>
35 </div>
36 <div class="form-group">
37     <label class="control-label">Эл.Пошта</label>
38     <input asp-for="email" class="form-control" />
39     <span asp-validation-for="email" class="text-danger"></span>
40 </div>
41
42 <div class="form-group">
43     <label class="control-label">Номер телефону</label>
44     <input asp-for="phone_number" class="form-control" />
45     <span asp-validation-for="phone_number" class="text-danger"></span>
46 </div>
47
48 <div class="form-group">
49     <label class="control-label">Дата дзвінка</label>
50     <input asp-for="date_call" class="form-control" />
51     <span asp-validation-for="date_call" class="text-danger"></span>
52 </div>
53
54 <div class="form-group">
55     <label class="control-label">Мова</label>
56     <select asp-for="Languageid" class="form-control" asp-items="ViewBag.Languages"></select>
57     <span asp-validation-for="Languageid" class="text-danger"></span>
58 </div>
59
60 <div class="form-group">
61     <label class="control-label">Форма заняття</label>
62     <select asp-for="Typesid" class="form-control" asp-items="ViewBag.Types"></select>
63     <span asp-validation-for="Typesid" class="text-danger"></span>
64 </div>
65
66 <div class="form-group">

```

Рисунок Б.7 – Продовження лістингу сторінки редагування даних Edit

```

67     <label class="control-label">Група</label>
68     <select asp-for="Groupid" class="form-control" asp-items="ViewBag.Groups"></select>
69     <span asp-validation-for="Groupid" class="text-danger"></span>
70 </div>
71
72 <div class="form-group">
73     <label class="control-label">Результат дзвінка</label>
74     <input asp-for="result" class="form-control" />
75     <span asp-validation-for="result" class="text-danger"></span>
76 </div>
77
78 <div class="form-group">
79     <label class="control-label">Дата повторного дзвінка</label>
80     <input asp-for="date_recall" class="form-control" />
81     <span asp-validation-for="date_recall" class="text-danger"></span>
82 </div>
83
84 <div class="form-group">
85     <label class="control-label">Реклама</label>
86     <select asp-for="Ad_sourceid" class="form-control" asp-items="ViewBag.ad_Sources"></select>
87     <span asp-validation-for="Ad_sourceid" class="text-danger"></span>
88 </div>
89 <div class="form-group">
90     <input type="submit" value="Зберегти" class="btn btn-primary" />
91 </div>
92 </form>
93 </div>
94 </div>
95
96 <div>
97     <a asp-action="Table">Назад до таблиці</a>
98 </div>
99

```

Рисунок Б.8 – Завершення лістингу сторінки редагування даних Edit



```

1  @model ASP_TEST.Models.Lid
2
3  @{
4      ViewData["Title"] = "Delete";
5  }
6
7  <h1>Видалити</h1>
8  <div>
9      <hr />
10     <dl class="row">
11         <dt class="col-sm-2">
12             Ім'я
13         </dt>
14         <dd class="col-sm-10">
15             @Html.DisplayFor(model => model.firstname)
16         </dd>
17         <dt class="col-sm-2">
18             Прізвище
19         </dt>
20         <dd class="col-sm-10">
21             @Html.DisplayFor(model => model.surname)
22         </dd>
23         <dt class="col-sm-2">
24             Вік
25         </dt>
26         <dd class="col-sm-10">
27             @Html.DisplayFor(model => model.age)
28         </dd>
29         <dt class="col-sm-2">
30             Статус
31         </dt>
32         <dd class="col-sm-10">
33             @Html.DisplayFor(model => model.Statusid)

```

Рисунок Б.9 - Лістинг сторінки видалення даних Delete

```
34 </dd>
35 <dt class="col-sm-2">
36     Email
37 </dt>
38 <dd class="col-sm-10">
39     @Html.DisplayFor(model => model.email)
40 </dd>
41 <dt class="col-sm-2">
42     Номер телефону
43 </dt>
44 <dd class="col-sm-10">
45     @Html.DisplayFor(model => model.phone_number)
46 </dd>
47 <dt class="col-sm-2">
48     Дата дзвінка
49 </dt>
50 <dd class="col-sm-10">
51     @Html.DisplayFor(model => model.date_call)
52 </dd>
53 <dt class="col-sm-2">
54     Мова
55 </dt>
56 <dd class="col-sm-10">
57     @Html.DisplayFor(model => model.Languageid)
58 </dd>
59 <dt class="col-sm-2">
60     Форма заняття
61 </dt>
62 <dd class="col-sm-10">
63     @Html.DisplayFor(model => model.Typesid)
64 </dd>
65 <dt class="col-sm-2">
66     Група
```

Рисунок Б.10 – Продовження лістингу сторінки видалення даних Delete

```

67     </dt>
68     <dd class="col-sm-10">
69         @Html.DisplayFor(model => model.Groupid)
70     </dd>
71     <dt class="col-sm-2">
72         Результат дзвінка
73     </dt>
74     <dd class="col-sm-10">
75         @Html.DisplayFor(model => model.result)
76     </dd>
77     <dt class="col-sm-2">
78         Дата повторного дзвінка
79     </dt>
80     <dd class="col-sm-10">
81         @Html.DisplayFor(model => model.date_recall)
82     </dd>
83     <dt class="col-sm-2">
84         Реклама
85     </dt>
86     <dd class="col-sm-10">
87         @Html.DisplayFor(model => model.Ad_sourceid)
88     </dd>
89 </dl>
90
91 <form asp-action="Delete">
92     <input type="hidden" asp-for="id" />
93     <input type="submit" value="Видалити" class="btn btn-danger" /> |
94     <a asp-action="Table">Назад до таблиці</a>
95 </form>
96 </div>

```

Рисунок Б.11 – Завершення лістингу сторінки видалення даних Delete