

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет менеджменту

Кафедра економічної кібернетики, комп'ютерних наук
та інформаційних технологій

Кваліфікаційна наукова
праця на правах рукопису

СРЕЩЕНКО Олександр Олександрович

УДК 004.4:005.7:658.8

КВАЛІФІКАЦІЙНА РОБОТА

**ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КАСТОМІЗОВАНОЇ CRM-
СИСТЕМИ ДЛЯ УПРАВЛІННЯ ВЗАЄМОВІДНОСИНАМИ З
КЛІЄНТАМИ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ**

Спеціальність 122 «Комп'ютерні науки»
Галузь знань – 12 «Інформаційні технології»

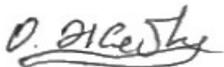
Подається на здобуття освітнього ступеня «Бакалавр»

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.



О.О. Єрещенко

Науковий керівник: Волосюк Юрій Вікторович, кандидат технічних наук,
доцент 

Науковий керівник: Жебко Олександр Олегович, асистент 

Завідувач кафедри: Тищенко Світлана Іванівна, кандидат педагогічних
наук, доцент 

Миколаїв–2025

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ	9
1.1 Опис предметної сфери.....	9
1.2 Огляд та аналіз наявних аналогів	10
1.3 Постановка задачі.....	19
Висновки до розділу 1.....	25
РОЗДІЛ 2 ТЕХНОЛОГІЇ СТВОРЕННЯ CRM-СИСТЕМИ.....	27
2.1 Вибір середовища розробки.....	27
2.2 Система управління базами даних SQL.....	30
2.3 Цілі впровадження CRM-системи	34
Висновки до розділу 2.....	35
РОЗДІЛ 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ CRM-СИСТЕМИ	36
3.1 Загальні вимоги до інформаційної системи.....	36
3.2 Структура й функціонал CRM-системи	40
3.3 Тестування додатку	49
Висновки до розділу 3.....	55
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

с. – сторінка

р. – рік

рр. – роки

рис. – рисунок

АНОТАЦІЯ

Єрещенко О.О. *Проектування та реалізація кастомізованої сrm-системи для управління взаємовідносинами з клієнтами малого та середнього бізнесу* – Кваліфікаційна наукова праця на правах рукопису. Робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 «Комп'ютерні науки». – Миколаївський національний аграрний університет, Миколаїв, 2025.

У кваліфікаційній роботі розглядається процес розробки CRM-системи (Customer Relationship Management) для онлайн-школи.

Метою дослідження є створення ефективного інструменту для автоматизації взаємодії з клієнтами, управління навчальними процесами, обліку успішності учнів, а також організації роботи викладачів.

У роботі проаналізовано вимоги до функціональності CRM-системи в контексті освітніх онлайн-платформ, розглянуто сучасні підходи до проектування архітектури програмного забезпечення та обґрунтовано вибір технологій для реалізації системи. Основну увагу приділено розробці вебінтерфейсу, модулів реєстрації та розкладу занять.

У результаті виконано повний цикл розробки програмного забезпечення: від аналізу вимог і проектування до тестування та впровадження прототипу CRM-системи. Результати роботи можуть бути використані онлайн-школами для покращення організації освітнього процесу та підвищення якості взаємодії з клієнтами.

Ключові слова: CRM, онлайн-школа, веброзробка, клієнтські дані, автоматизація, програмне забезпечення, управління взаємовідносинами.

ABSTRACT

Yereshchenko O.O. *Design and implementation a customized CRM system for managing relationships with small and medium-sized business clients* – Qualification scientific work in the form of a manuscript. Work for obtaining a bachelor's degree in specialty 122 “Computer Science”. – Mykolaiv National Agrarian University, Mykolaiv, 2025.

The qualification work considers the process of developing a CRM system (Customer Relationship Management) for an online school.

The purpose of the research is to create an effective tool for automating interaction with clients, managing educational processes, recording student performance, and organizing the work of teachers.

The work analyzes the requirements for the functionality of the CRM system in the context of online educational platforms, considers modern approaches to designing software architecture, and justifies the choice of technologies for implementing the system. The main attention is paid to the development of the web interface, registration modules, and class schedules.

As a result, a full software development cycle was completed: from requirements analysis and design to testing and implementation of a CRM system prototype. The results of the work can be used by online schools to improve the organization of the educational process and improve the quality of interaction with customers.

Keywords: CRM, online school, web development, customer data, automation, software, relationship management.

ВСТУП

Сучасний розвиток інформаційних технологій має значний вплив на сферу освіти, зокрема на її цифрову трансформацію. Онлайн-навчання стало невід'ємною частиною освітнього простору, що особливо помітно після активізації дистанційного навчання внаслідок глобальних подій останніх років. Онлайн-школи, що пропонують широкий спектр курсів та програм, потребують ефективних інструментів для управління взаємодією з учнями, викладачами та адміністрацією.

CRM-системи (Customer Relationship Management) набули широкого поширення в бізнес-середовищі, однак їхнє застосування в освітній сфері досі не є достатньо розвиненим. Використання CRM у контексті онлайн-школи дає можливість автоматизувати процеси реєстрації, комунікації з учнями, обліку успішності, керування розкладом та зберігання навчальної інформації.

Існуючі рішення для управління освітнім процесом часто не враховують специфіки онлайн-навчання, мають обмежену функціональність або є надто складними для впровадження в невеликих навчальних закладах. Брак адаптивних CRM-систем, орієнтованих саме на онлайн-школи, створює потребу в розробці гнучких, зручних та безпечних рішень, які відповідали б сучасним вимогам до цифрової освіти.

Актуальність розробки CRM-системи для онлайн-школи також обумовлена зростанням конкуренції серед освітніх платформ і зростаючими очікуваннями користувачів щодо якості сервісу, персоналізованого підходу та доступності освітніх матеріалів. Ефективна CRM-система дозволяє онлайн-школі не лише оптимізувати внутрішні процеси, а й покращити якість взаємодії з клієнтами, забезпечуючи стійкий розвиток та конкурентоспроможність на ринку освітніх послуг.

Метою кваліфікаційної роботи є розробка CRM-системи, адаптованої до потреб онлайн-школи, яка дозволить автоматизувати процеси взаємодії з клієнтами, організувати ефективне управління навчальним процесом, забезпечити зручний інтерфейс для адміністраторів, викладачів та учнів, а також підвищити загальну ефективність роботи освітньої платформи.

Для досягнення поставленої мети були визначені **наступні завдання**:

1. Проаналізувати існуючі підходи до розробки CRM-систем, з урахуванням сучасних технологічних тенденцій.
2. Визначити основні вимоги до функціональності та дизайну сайту CRM-системи, які забезпечать її зручність та ефективність.
3. Розробити концепцію сайту, включаючи архітектуру, навігацію та інтерфейс користувача, враховуючи сучасні стандарти та потреби користувачів.
4. Виконати технічну реалізацію сайту з використанням актуальних технологій веб-розробки, таких як JavaScript(React), HTML5, CSS3, C#.
5. Провести тестування розробленого сайту з метою виявлення та усунення можливих недоліків, а також оцінити його ефективність за допомогою ключових показників продуктивності.

Об'єктом дослідження є процес розробки сайту CRM-системи для онлайн-школи, що включає різні етапи від планування та дизайну до технічної реалізації і тестування. Особливу увагу приділено вивченню процесів взаємодії користувачів із сайтом, забезпечення його зручності та безпеки.

Предметом дослідження є конкретні методи та технології, що застосовуються при розробці сайту CRM-системи, включаючи програмне забезпечення для створення інтерфейсу користувача, інструменти для управління базами даних, а також оптимізації роботи сайту.

Для досягнення поставленої мети були використані як специфічні **методи дослідження**, такі як спостереження, вимірювання, експеримент, так і загальнонаукові методи, включаючи аналіз і синтез, індукцію і дедукцію, абстрагування, узагальнення та моделювання. Використання цих методів

дозволило отримати достовірні наукові знання та практичні навички, необхідні для розробки високоякісного веб-сайту.

Інформаційна база дослідження включає нормативно-правові акти, наукові та навчально-методичні документи, матеріали з вітчизняних та закордонних наукових видань, а також з періодичних видань, довідкової літератури та матеріалів спеціалізованих веб-сайтів і порталів. Статистичні дані також використовувались для аналізу ринку та визначення основних тенденцій в області CRM-систем та структур онлайн-шкіл.

Практичне значення дослідження полягає у створенні ефективного інструменту для інтернет-маркетингу, який може бути використаний як приклад для подібних проектів у майбутньому.

Структура та обсяг роботи. Логіка дослідження зумовила **структуру** кваліфікаційної роботи: вступ, 3 розділи, висновки, список використаних джерел із 30 найменувань. Загальний обсяг 60 сторінок.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної сфери

Сфера онлайн-освіти є однією з найдинамічніших у контексті впровадження інформаційних технологій. З переходом багатьох навчальних процесів у дистанційний формат, суттєво зросла потреба в автоматизації управлінських та комунікаційних процесів у межах освітніх платформ. Онлайн-школи, які працюють у сфері неформальної або додаткової освіти, особливо активно впроваджують цифрові рішення для забезпечення якісної взаємодії між адміністрацією, викладачами та студентами.

На практиці організація роботи онлайн-школи включає декілька ключових напрямів: управління базою клієнтів, адміністрування навчального процесу, контроль розкладу занять, облік фінансових операцій, а також аналітика ефективності освітніх послуг. У звичайних умовах для виконання цих завдань використовуються різноманітні інструменти, зокрема електронні таблиці, календарі, месенджери, хмарні сховища та сторонні CRM-рішення, не адаптовані до специфіки освітнього середовища.

Використання такої фрагментованої інфраструктури ускладнює обробку інформації, підвищує ризик виникнення помилок, а також створює труднощі у масштабуванні освітнього продукту. Крім того, більшість універсальних CRM-систем орієнтовані на комерційні бізнес-процеси і не враховують особливості онлайн-навчання, такі як динамічний розклад, зміна тривалості курсів, індивідуальні навчальні траєкторії чи специфічні ролі користувачів.

Предметна сфера дослідження охоплює розробку прикладного програмного забезпечення, орієнтованого на автоматизацію процесів управління онлайн-школою. Основною задачею є створення CRM-системи, яка забезпечить централізоване зберігання даних, ефективну взаємодію між усіма учасниками освітнього процесу та гнучкість у налаштуванні згідно з потребами конкретної організації. У результаті впровадження такої системи очікується підвищення керованості навчальним процесом, зменшення навантаження на

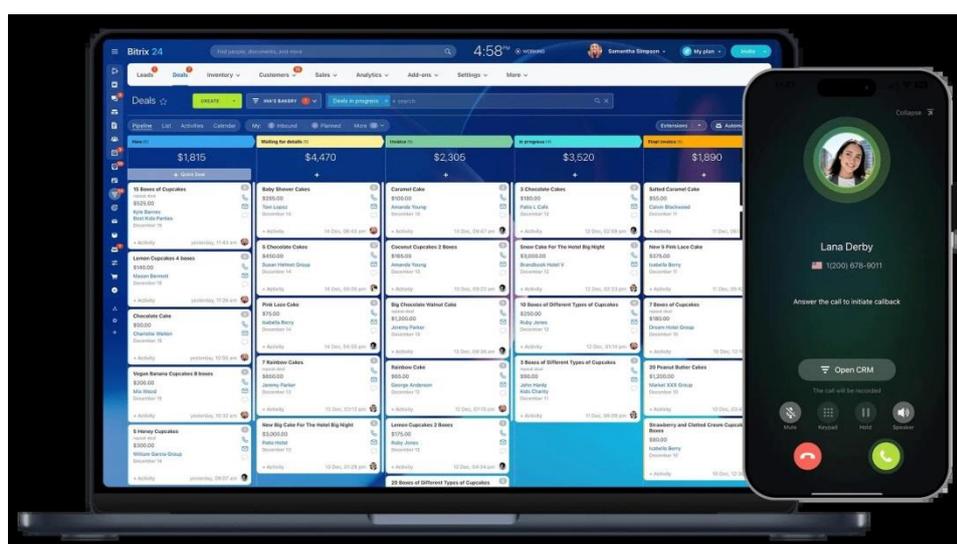
адміністративний персонал та покращення сервісу для кінцевого користувача [5].

1.2 Огляд та аналіз наявних аналогів

На сучасному ринку програмного забезпечення представлено велику кількість CRM-систем, однак переважна більшість із них орієнтована на бізнес-сектор і не враховує специфіку освітньої діяльності. Серед найбільш відомих універсальних рішень варто виділити такі продукти, як Bitrix24, HubSpot, Zoho CRM, amoCRM та Salesforce. Ці системи мають розвинену функціональність у сфері керування продажами, маркетинговими кампаніями та клієнтською підтримкою, проте їхнє застосування в онлайн-освіті є обмеженим через відсутність модулів, що безпосередньо підтримують навчальний процес.

Розглянемо детально ці системи:

1. **Bitrix24** — це комплексна CRM-система, яка окрім класичних функцій роботи з клієнтами включає можливості побудови внутрішньої корпоративної комунікації, управління завданнями, проектами та автоматизації бізнес-процесів. Вона підтримує ведення клієнтської бази, налаштування воронки продажів, роботу із заявками та інтеграцію з популярними каналами



комунікації, такими як електронна пошта, телефони та месенджери.

Рисунок 1.1 – Головна сторінка Bitrix24

Джерело: сформовано автором на основі офіційного сайту

Особливістю системи є те, що вона поєднує робочі інструменти в єдиному інтерфейсі, що дозволяє уникати необхідності використання декількох окремих програм. Наприклад, менеджери можуть одночасно працювати із завданнями, відповідати на повідомлення клієнтів, переглядати воронку продажів, оновлювати базу знань і спілкуватися з командою — і все це без виходу з системи. Vitrix24 також пропонує вбудовані можливості для телефонії, електронної пошти, інтеграції з популярними месенджерами та соціальними мережами, що значно полегшує ведення комунікації з клієнтами.

Інтерфейс платформи адаптивний та сучасний, підтримує мобільні пристрої та надає широкі можливості для кастомізації. Завдяки відкритому API та підтримці вебхуків, Vitrix24 можна інтегрувати з іншими сервісами або доповнити власними рішеннями. Система підтримує як хмарне використання, так і локальну інсталяцію на сервері компанії, що є важливою перевагою для підприємств з підвищеними вимогами до безпеки або автономності.

Завдяки широкому функціоналу Vitrix24 використовується у найрізноманітніших сферах — від торгівлі та логістики до освіти й державного управління. Це рішення дозволяє суттєво підвищити прозорість бізнес-процесів, ефективність роботи з клієнтами та загальний рівень організації в компанії.

Проте Vitrix24 має суттєві обмеження у сфері освіти: немає вбудованої підтримки навчальних модулів, розкладу занять або обліку успішності учнів. Для онлайн-шкіл використання Vitrix24 вимагає розробки додаткових кастомних модулів або складної інтеграції через API, що збільшує витрати на впровадження та супровід системи.

2. **HubSpot** — це потужна платформа для автоматизації маркетингу, управління взаємовідносинами з клієнтами (CRM), продажів і обслуговування, яка стала одним із найпопулярніших рішень у сфері цифрового бізнесу. Вона орієнтована на допомогу компаніям у залученні потенційних клієнтів, їх

конвертації та подальшій підтримці, що робить її універсальним інструментом для зростання бізнесу в умовах сучасної конкуренції.



Рисунок 1.2 – Головна сторінка HubSpot

Джерело: сформовано автором на основі офіційного сайту

Основна ідея HubSpot — це підхід Inbound Marketing, тобто залучення клієнтів за рахунок створення цінного контенту, оптимізації веб-сайтів, SEO, електронних розсилок і активної присутності в соціальних мережах. Усі ці функції об'єднані у зручному інтерфейсі, де всі відділи компанії — маркетинг, продажі та служба підтримки — можуть працювати у синергії, маючи доступ до єдиної клієнтської бази та аналітики.

HubSpot CRM надає повноцінну картину кожної взаємодії з клієнтом: від першого контакту до укладання угоди й післяпродажного обслуговування. Система автоматично збирає інформацію з форм на сайті, листування, дзвінків, соціальних мереж і створює хронологію взаємодій з кожним контактом. Завдяки цьому менеджери з продажів можуть краще розуміти потреби клієнтів і будувати персоналізовану комунікацію.

Маркетинговий модуль дозволяє створювати складні кампанії, керувати контентом, створювати лендинги та автоматизувати розсилки. Усе це інтегрується з аналітикою, що надає можливість точно вимірювати

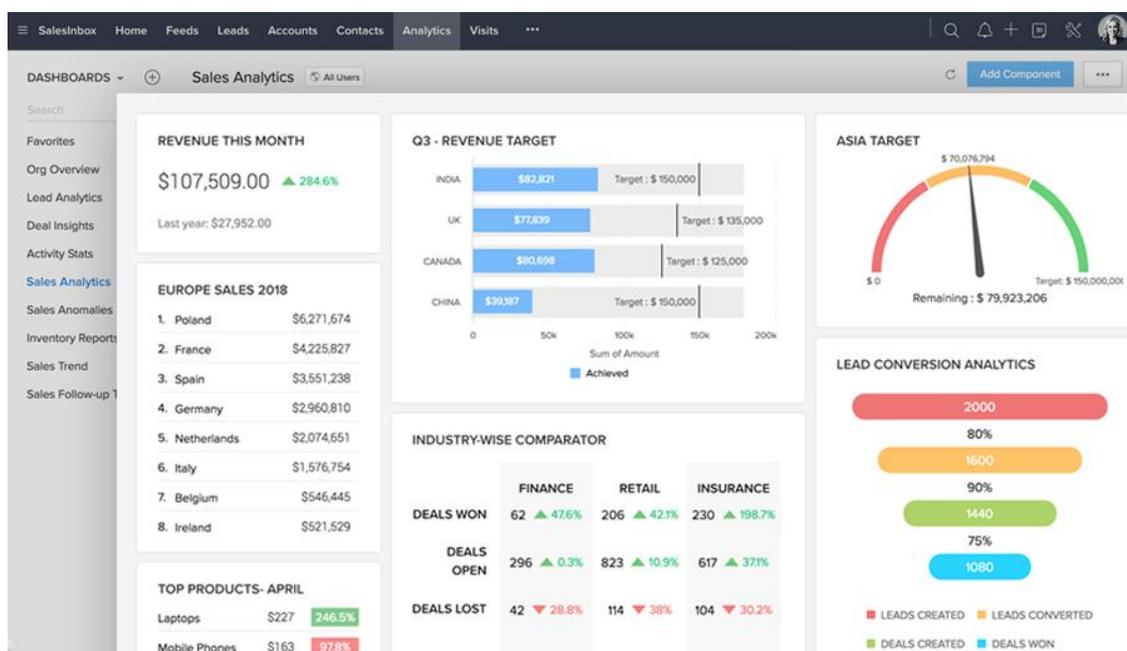
ефективність кожного каналу та кожної активності. Це особливо важливо для оптимізації маркетингового бюджету та стратегічного планування.

Також HubSpot включає інструменти для обслуговування клієнтів: база знань, чат-боти, тікетингова система й автоматизовані відповіді. Це дозволяє забезпечити високий рівень підтримки клієнтів і зменшити навантаження на команду.

Платформа має модульну структуру, тому компанії можуть вибрати тільки ті функції, які їм потрібні: CRM, маркетинг, продажі або підтримка. Доступна як безкоштовна базова версія, так і розширені платні тарифи з широким функціоналом. Завдяки API, HubSpot легко інтегрується з іншими сервісами, такими як Gmail, Outlook, Slack, Shopify, Zapier та багато інших.

Завдяки простому інтерфейсу, гнучкості у використанні та великій кількості навчальних матеріалів, HubSpot підходить як для малого бізнесу, так і для великих компаній, які прагнуть централізувати свої процеси та побудувати ефективну комунікацію з клієнтами.

3. **Zoho CRM** — це гнучка, масштабована хмарна платформа для управління взаємовідносинами з клієнтами, яка орієнтована на автоматизацію процесів продажів, маркетингу, підтримки клієнтів і аналітики. Система розроблена компанією Zoho Corporation та активно використовується бізнесами



різного масштабу — від стартапів до великих корпорацій у всьому світі.

Рисунок 1.3 – Головна сторінка Zoho

Джерело: сформовано автором на основі офіційного сайту

На відміну від багатьох традиційних CRM-систем, Zoho CRM має комплексний підхід до автоматизації бізнес-процесів. Вона дозволяє компаніям зберігати всі дані про клієнтів у єдиному середовищі, включаючи контактну інформацію, історію взаємодій, статуси угод, електронну переписку, дзвінки та активність у соціальних мережах. Це забезпечує прозорість на всіх етапах воронки продажів і дозволяє менеджерам приймати обґрунтовані рішення в режимі реального часу.

Суттєвою перевагою Zoho CRM є висока ступінь автоматизації. Система дозволяє створювати правила обробки лідів, запускати автоматизовані дії залежно від поведінки клієнта, призначати завдання та сповіщення відповідальним співробітникам. Це зменшує ручну працю та прискорює процеси закриття угод.

Zoho CRM також активно використовує можливості штучного інтелекту через свого вбудованого помічника Zia, який аналізує дані, прогнозує ймовірність укладання угод, виявляє відхилення у поведінці клієнтів і пропонує рекомендації для підвищення ефективності роботи менеджерів.

Система інтегрується з понад 300 зовнішніми сервісами, серед яких Zoho Mail, Google Workspace, Microsoft Office 365, Slack, WhatsApp, Zoom, Shopify, Facebook, Twitter та інші. Це дозволяє гнучко адаптувати CRM до вже існуючої цифрової інфраструктури компанії.

Zoho CRM надає розгорнуту аналітику — користувачі можуть створювати кастомні звіти, будувати дашборди, аналізувати KPI та відстежувати ефективність кожного менеджера або каналу комунікації. Також підтримується сегментація клієнтів, що дає змогу проводити більш персоналізовані кампанії.

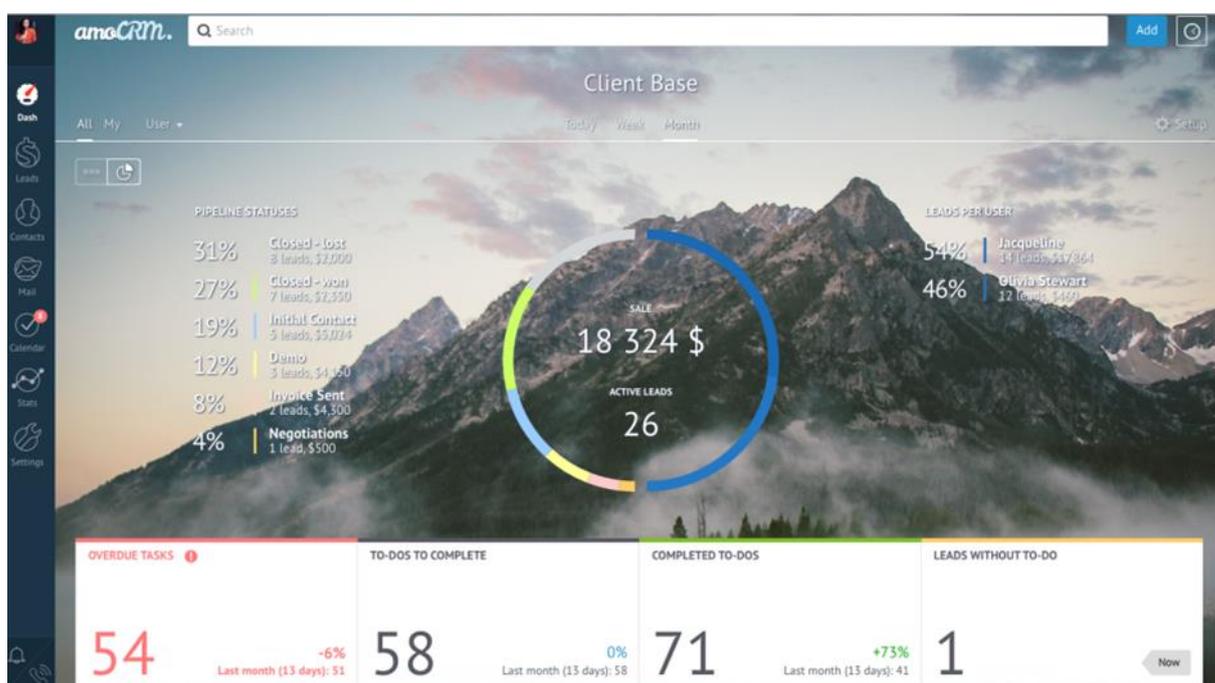
Інтерфейс платформи інтуїтивно зрозумілий і налаштовується відповідно до потреб конкретної команди. Крім того, Zoho CRM підтримує мобільні

додатки, що дозволяє менеджерам працювати з системою поза офісом. Гнучка система ролей і доступів забезпечує контроль над конфіденційністю даних.

Система доступна у кількох тарифних планах: від безкоштовної базової версії до розширених корпоративних рішень із підтримкою кастомної автоматизації, API-доступу, розробки модулів та використання AI-функцій.

Zoho CRM — це повнофункціональне рішення для організацій, які прагнуть централізувати взаємодію з клієнтами, підвищити продуктивність команди, автоматизувати рутинні процеси та отримувати аналітичну підтримку в реальному часі.

4. **AmoCRM** — це популярна хмарна CRM-система, розроблена для малого та середнього бізнесу з фокусом на автоматизацію процесів продажів і підвищення ефективності взаємодії з клієнтами. Вона широко застосовується у різних сферах, таких як послуги, торгівля, IT, маркетинг та інші галузі, де



важлива швидка та прозора робота з лідами та клієнтами.

Рисунок 1.4 – Головна сторінка AmoCRM

Джерело: сформовано автором на основі офіційного сайту

Основна особливість amoCRM полягає у простоті та зручності використання. Система побудована на візуальній воронці продажів, де

менеджери можуть легко відстежувати стадії угод, переміщати їх між етапами, додавати нотатки та ставити завдання. Такий підхід дозволяє чітко контролювати весь цикл взаємодії — від першого контакту до укладання угоди.

AmoCRM активно інтегрується з різними комунікаційними каналами: телефоном, електронною поштою, чатами на сайті, месенджерами (Telegram, Viber, WhatsApp), соціальними мережами. Завдяки цьому всі повідомлення, дзвінки та листування автоматично потрапляють у систему, що знижує ризик втрати важливої інформації і дозволяє менеджерам оперативно відповідати клієнтам.

Система має розвинені можливості автоматизації. Можна налаштувати тригери та робочі процеси, які запускають певні дії в залежності від подій, наприклад, автоматичне призначення лідів конкретним менеджерам, відправка нагадувань, повідомлень або пропозицій. Це допомагає оптимізувати час співробітників та мінімізувати людські помилки.

AmoCRM також надає аналітичні інструменти для оцінки ефективності роботи команди та бізнес-процесів. За допомогою дашбордів і звітів можна відстежувати конверсії, середній час закриття угод, продуктивність менеджерів і інші ключові показники, що сприяє прийняттю обґрунтованих управлінських рішень.

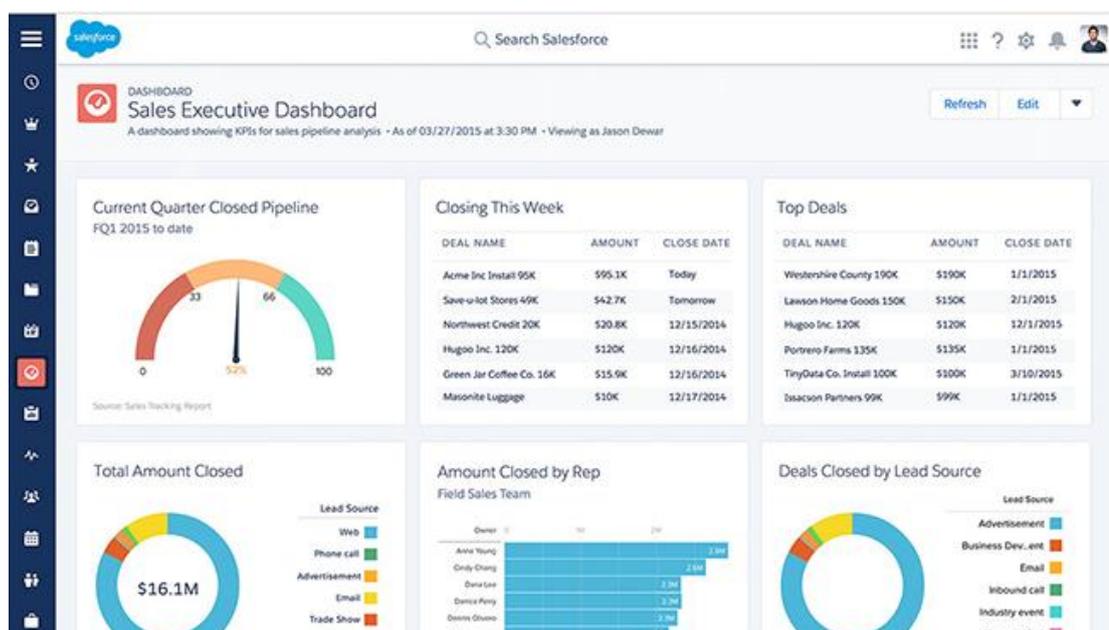
Важливою перевагою є гнучкість налаштувань і можливість масштабування системи. amoCRM дозволяє підлаштовувати поля, етапи воронки, шаблони комунікацій під специфіку конкретного бізнесу. Для компаній із нестандартними процесами доступні можливості розробки власних інтеграцій через API.

Інтерфейс amoCRM відзначається інтуїтивністю та зручністю, що робить процес навчання нових користувачів швидким і простим. Крім веб-версії, доступні мобільні додатки, які дозволяють менеджерам працювати з клієнтами на ходу, зберігаючи актуальність даних.

AmoCRM — це ефективний інструмент для бізнесів, які прагнуть підвищити продуктивність продажів, забезпечити якісну взаємодію з клієнтами

і оптимізувати роботу команди за допомогою автоматизації та зручних інструментів управління.

Salesforce — це провідна світова платформа управління взаємовідносинами з клієнтами (CRM), яка надає комплексні хмарні рішення для автоматизації бізнес-процесів, підтримки продажів, маркетингу, обслуговування клієнтів та аналітики. Ця система стала однією з найпопулярніших у світі завдяки своїй масштабованості, гнучкості та



широкому спектру можливостей, що дозволяють адаптувати її під потреби компаній будь-якого розміру та галузі.

Рисунок 1.5 – Головна сторінка Salesforce

Джерело: сформовано автором на основі офіційного сайту

Однією з ключових переваг Salesforce є її архітектура на основі хмарних технологій, що забезпечує доступ до системи з будь-якої точки світу через інтернет без необхідності локального встановлення складного програмного забезпечення. Це дозволяє компаніям швидко впроваджувати рішення, масштабувати їх у міру зростання бізнесу і зменшувати витрати на IT-інфраструктуру.

Salesforce пропонує широкий набір модулів і продуктів, включаючи Sales Cloud (для автоматизації продажів), Service Cloud (для обслуговування

клієнтів), Marketing Cloud (для управління маркетинговими кампаніями), Commerce Cloud (для електронної комерції), а також платформу для розробки власних додатків — Salesforce Platform. Завдяки цьому компанії можуть вибрати саме ті інструменти, які відповідають їхнім бізнес-завданням, або створювати унікальні рішення на базі готової інфраструктури.

Інтерфейс Salesforce орієнтований на користувача і дозволяє легко налаштовувати робочі процеси, створювати кастомні об'єкти, автоматизувати рутинні завдання та інтегруватися з великою кількістю сторонніх систем через API. Система підтримує гнучкі налаштування бізнес-логіки за допомогою вбудованих інструментів, таких як Workflow Rules, Process Builder та Apex — мови програмування Salesforce.

Важливою особливістю Salesforce є потужна аналітика і можливості звітності. Інструменти, такі як Salesforce Einstein, використовують штучний інтелект для прогнозування трендів, рекомендацій та автоматичного виявлення аномалій у даних, що дозволяє приймати більш обґрунтовані рішення.

Salesforce також має велику екосистему партнерів і розробників, які створюють додаткові додатки та розширення у Salesforce AppExchange — своєрідному маркетплейсі, що надає сотні тисяч готових рішень для різних бізнес-задач.

Безпека даних і відповідність міжнародним стандартам є пріоритетом для Salesforce. Система підтримує багаторівневий захист, включно з багатофакторною автентифікацією, шифруванням даних, аудитом дій користувачів і суворими політиками доступу, що забезпечує надійний захист інформації клієнтів.

Таким чином, Salesforce є комплексною CRM-платформою, яка дозволяє компаніям підвищувати ефективність продажів, покращувати якість обслуговування клієнтів, оптимізувати маркетингові кампанії і автоматизувати внутрішні бізнес-процеси, забезпечуючи при цьому масштабованість, безпеку і можливість глибокої кастомізації під індивідуальні потреби [4].

Окремі освітні платформи, такі як Moodle або Google Classroom, частково реалізують функції управління навчальним процесом, але не є повноцінними CRM-системами. Вони орієнтовані переважно на взаємодію між викладачем і студентом, мають обмежену інтеграцію з платіжними системами, клієнтською підтримкою або модулями аналітики продажів. Крім того, їхній інтерфейс часто вимагає високої технічної грамотності з боку користувачів.

Також існують вузькоспеціалізовані рішення, розроблені під конкретні навчальні заклади або франшизні освітні мережі. Ці системи зазвичай недоступні для широкого використання та не мають відкритого коду, що унеможлиблює їх адаптацію під потреби інших онлайн-шкіл. Деякі стартапи у сфері EdTech пропонують SaaS-рішення з інтегрованими CRM-функціями, однак вони часто мають обмежену функціональність, відсутність підтримки кастомізації або працюють за підписковою моделлю з високою вартістю [7].

1.3 Постановка задачі

У конкурентному бізнес-середовищі ефективне управління взаємовідносинами з клієнтами є одним із ключових чинників довгострокового успіху компанії. Підвищення рівня обслуговування, персоналізація підходу до кожного клієнта та своєчасна комунікація напряду впливають на рівень продажів, утримання клієнтів та розвиток бізнесу.

Традиційні методи ведення обліку клієнтів (наприклад, використання таблиць Excel, розрізнених нотаток або електронної пошти) є неефективними в умовах масштабування компанії та зростання кількості клієнтів. Відсутність єдиної централізованої системи ускладнює процес взаємодії між відділами, призводить до втрати важливої інформації, зниження якості обслуговування та, відповідно, до втрат потенційного прибутку.

CRM-система (Customer Relationship Management) дозволяє вирішити ці проблеми, об'єднавши усю інформацію про клієнтів, угоди, комунікації та завдання в одному інструменті. Вона забезпечує автоматизацію основних

бізнес-процесів, сприяє підвищенню прозорості роботи відділів продажу та підтримки, допомагає швидко аналізувати результати діяльності та виявляти можливості для покращення [6].

Крім того, сучасна CRM-система має стати не лише інструментом обліку, а й потужною платформою для:

- планування та прогнозування продажів;
- управління маркетинговими кампаніями;
- контролю за виконанням завдань співробітників;
- аналізу даних для прийняття стратегічних рішень;
- підвищення загального рівня цифрової зрілості компанії.

Враховуючи вищезазначене, розробка власної CRM-системи стане важливим стратегічним кроком для оптимізації внутрішніх процесів, підвищення якості обслуговування клієнтів та забезпечення сталого розвитку бізнесу.

CRM-система (Customer Relationship Management) є комплексним інструментом, який об'єднує, автоматизує та оптимізує процеси взаємодії з клієнтами на всіх етапах їх життєвого циклу. Основні функціональні можливості CRM включають:

1. Управління базою клієнтів

Централізоване збереження даних: єдина база даних усіх клієнтів, контактних осіб, історії взаємодій та документів.

Профілі клієнтів: детальна інформація про кожного клієнта (контактні дані, історія покупок, особливості співпраці, вподобання).

Сегментація клієнтів: групування за різними критеріями (географія, тип клієнта, обсяги покупок тощо) для таргетованої роботи.

2. Управління продажами

Автоматизація воронки продажів: ведення угод на різних етапах, автоматичне оновлення статусів, встановлення пріоритетів.

Фіксація комунікацій: збереження історії дзвінків, листування, зустрічей із клієнтами.

Планування завдань: створення нагадувань для менеджерів про дзвінки, листи, зустрічі, контроль за термінами виконання.

Прогнозування продажів: аналітика щодо поточних угод, ймовірності їхнього закриття та прогнозованого доходу.

3. Автоматизація маркетингових кампаній

Розсилки: автоматичні та ручні розсилки електронною поштою, SMS, месенджерами.

Персоналізація пропозицій: створення індивідуальних маркетингових пропозицій для різних сегментів клієнтів.

Аналітика кампаній: відстеження результатів розсилок та кампаній (відкриття листів, переходи за посиланнями, конверсії).

4. Підтримка клієнтів

Тікетинг: реєстрація звернень клієнтів, контроль за їх обробкою та закриттям.

База знань: інтеграція з системою допомоги для швидкого вирішення типових питань клієнтів.

Оцінка якості обслуговування: збори фідбеку від клієнтів, аналіз рівня задоволеності.

5. Аналітика та звітність

Індивідуальні та загальні звіти: автоматичне формування звітів по продажах, активності клієнтів, ефективності співробітників.

Дашборди в реальному часі: візуалізація ключових показників (KPI), наприклад, кількість нових клієнтів, рівень конверсії, обсяг продажів.

Аналіз тенденцій: виявлення патернів у поведінці клієнтів, визначення найефективніших каналів комунікації.

6. Управління завданнями та проектами

Постановка задач: можливість ставити задачі співробітникам, контролювати строки виконання.

Координація між відділами: спільна робота над проектами, обмін файлами та коментарями всередині системи.

7. Інтеграції з іншими системами

Інтеграція з електронною поштою та календарями: автоматична синхронізація листування та планування зустрічей.

Інтеграція з телефонією: здійснення та реєстрація дзвінків напряду через CRM.

Інтеграція з фінансовими системами та ERP: обмін даними про оплату, рахунки, залишки товару.

8. Безпека та права доступу

Ролі та доступи: налаштування рівнів доступу до інформації для різних співробітників.

Контроль змін: ведення історії змін даних для запобігання втраті важливої інформації.

Захист даних: відповідність вимогам законодавства щодо персональних даних (GDPR, локальні регулювання).

Інтерфейс користувача має бути зручним, інтуїтивно зрозумілим і підтримувати ефективну взаємодію користувачів із системою. Основні вимоги:

1. Зрозуміла та логічна навігація
2. Інтуїтивна взаємодія
3. Адаптивність та кросплатформеність
4. Персоналізація інтерфейсу
5. Висока швидкість роботи
6. Візуальна ясність і сучасний дизайн

При розробці та впровадженні CRM-системи важливо враховувати низку технічних обмежень, які можуть впливати на продуктивність, безпеку, масштабованість та інтеграційні можливості рішення. Їхнє своєчасне виявлення дозволяє мінімізувати ризики критичних збоїв у майбутньому.

Щодо продуктивності, система має підтримувати обробку великого обсягу даних, що включає записи про клієнтів, угоди та активності. Недоліки архітектури або обмеження бази даних можуть призводити до зниження швидкості доступу до інформації. При збільшенні кількості користувачів або

транзакцій можуть спостерігатися затримки завантаження сторінок, оновлення даних або виконання звітів. Обчислювальні ресурси серверів (процесор, пам'ять, дискова система) також можуть ставати вузьким місцем, особливо за умов пікових навантажень.

У сфері зберігання даних можуть виникати обмеження щодо максимального обсягу документів, історій активностей, листувань або файлів залежно від обраної інфраструктури — локальної чи хмарної. Часте створення резервних копій може впливати на доступність системи або потребувати збільшення обсягів сховища. Також виникає необхідність у процедурі архівування старих даних для запобігання перевантаженню активної бази.

Технологічні обмеження проявляються у залежності від стабільності корпоративної інфраструктури, зокрема внутрішньої мережі, електропостачання та серверної підтримки. CRM-система може бути сумісною лише з певними операційними системами чи браузерами, що потребує оптимізації роботи під конкретні платформи. Оновлення системи також може ускладнюватися через обмеження, пов'язані з використовуваними фреймворками, базами даних або версіями API інтегрованих сервісів.

Безпекові аспекти визначаються рівнем шифрування даних, наприклад, обов'язковим використанням протоколів SSL/TLS для передавання інформації, а також внутрішнім шифруванням даних у базі. Упровадження багаторівневої аутентифікації та гнучке управління правами доступу на рівні окремих модулів або полів може ускладнювати налаштування системи. Крім того, відповідність таким нормативам, як GDPR або CCPA, накладає додаткові обмеження на обробку і використання персональних даних.

Інтеграційні обмеження пов'язані з квотами на кількість API-запитів до сторонніх сервісів, залежністю від їхніх політик оновлення або підтримки, що може вимагати частих адаптацій інтеграційних механізмів. Помилки або затримки синхронізації даних між CRM та іншими системами, такими як бухгалтерські або маркетингові платформи, можуть впливати на достовірність і цілісність інформації.

Стосовно масштабованості та гнучкості, окремі рішення можуть мати обмеження щодо кількості одночасних користувачів або обсягу оброблюваних даних без необхідності модернізації інфраструктури. Адаптація CRM під специфічні бізнес-процеси також може бути ускладнена або вимагати залучення розробників та придбання додаткових модулів.

Терміни виконання та оцінка ресурсів проєкту визначаються відповідно до складності завдань і обсягу робіт на кожному етапі. На першій стадії відбувається планування та формування вимог, що триває приблизно два тижні. У цьому процесі задіяні Project Manager та Business Analyst, які збирають бізнес-потреби та розробляють технічне завдання.

Далі слідує етап проєктування архітектури та UI/UX дизайну. Створення прототипів користувацького інтерфейсу та розробка загальної архітектури системи триває близько трьох тижнів. Для цього залучаються фахівці з UI/UX дизайну, а також backend і frontend розробники.

Основна частина проєкту – розробка функціоналу – є найбільш тривалою. Вона займає від восьми до десяти тижнів залежно від обсягу функціональних вимог. На цьому етапі працюють двоє backend-розробників та один frontend-розробник.

Після завершення програмування починається тестування, яке триває приблизно від двох до трьох тижнів. В роботі беруть участь один або два QA-інженери, які перевіряють систему на відповідність технічному завданню, виявляють помилки та забезпечують якість продукту.

Наступним етапом є впровадження рішення та навчання користувачів. Протягом двох тижнів відбувається інсталяція системи, проведення тренінгів для персоналу та підготовка супровідної документації. У цій фазі знову залучаються Project Manager та Business Analyst.

Після впровадження передбачена стадія підтримки та оптимізації CRM-системи, яка триває на постійній основі. До неї можуть бути залучені розробники, тестувальники та менеджери залежно від потреб бізнесу та користувачів.

Очікувана загальна тривалість реалізації всього проєкту становить близько чотирьох-п'яти місяців. Для успішної реалізації потрібна відповідна інфраструктура, що включає хмарні сервіси або локальні сервери. Окрім основних ресурсів, потрібно врахувати додаткові витрати на придбання ліцензійного програмного забезпечення, організацію навчання співробітників і технічну підтримку після запуску системи [4].

Тож, **основними етапами розробки є:**

1. Планування та формування вимог **(2 тижні):**
Аналіз бізнес-процесів компанії, визначення основних функціональних модулів системи, складання технічного завдання.

2. Проєктування архітектури та UI/UX дизайну **(3 тижні):**
Створення прототипів інтерфейсу, визначення структурної логіки, розробка загальної архітектури рішення.

3. Розробка основного функціоналу **(8–10 тижнів):**
Програмування серверної частини, реалізація клієнтського інтерфейсу, налаштування баз даних.

4. Тестування системи **(2–3 тижні):**
Проведення функціонального, інтеграційного та навантажувального тестування.

5. Впровадження та навчання користувачів **(2 тижні):**
Інсталяція системи, проведення тренінгів для співробітників, підготовка документації.

6. Підтримка та оптимізація **(постійно):**
Усунення можливих недоліків, оновлення системи відповідно до потреб бізнесу.

Висновки до розділу 1

У першому розділі було проаналізовано предметну сферу онлайн-освіти з фокусом на управління діяльністю онлайн-шкіл, визначено ключові проблеми,

які виникають при використанні розрізнених інструментів для організації освітнього процесу, та сформульовано основні вимоги до програмного забезпечення, здатного забезпечити ефективне функціонування освітньої платформи.

Огляд існуючих CRM-рішень показав, що більшість із них орієнтовані на комерційний сектор і не адаптовані до специфіки онлайн-навчання. Вони потребують значних доопрацювань або додаткових інтеграцій для забезпечення функцій, пов'язаних із навчальним процесом, таких як контроль розкладу, облік успішності, адаптивні освітні траєкторії тощо. Також було виявлено, що на ринку існує суттєвий дефіцит CRM-систем, які б поєднували в собі як інструменти управління клієнтськими відносинами, так і функціонал повноцінної освітньої платформи.

У результаті проведеного аналізу було поставлено задачу створення спеціалізованої CRM-системи для онлайн-шкіл, яка відповідатиме специфічним вимогам освітньої сфери та дозволить автоматизувати ключові процеси, підвищити якість взаємодії між учасниками освітнього процесу, а також забезпечити гнучкість, масштабованість і простоту використання. Запропонована система має потенціал стати ефективним інструментом цифрової трансформації онлайн-освіти.

РОЗДІЛ 2 ТЕХНОЛОГІЇ СТВОРЕННЯ CRM-СИСТЕМИ

2.1 Вибір середовища розробки

Веб-додаток було розроблено в середовищі Visual Studio.

Microsoft Visual Studio — один з продуктів компанії Майкрософт, який представляє з себе інтегровану середу розробки програмного забезпечення і ряд інших інструментальних засобів. Даний продукт дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в не керованому, так і керованому кодах для всіх платформ, підтримуваних Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework і Microsoft Silverlight. Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторингу коду.

Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і як відладчик машинного рівня. Решта вбудовуваних інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду, додавання нових наборів інструментів, наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів циклу розробки програмного забезпечення.

Загальний вигляд вікна, яке відображається при завантаженні Visual Studio, представлено на рисунку 2.1

```

1 import React, { useEffect, useState } from 'react';
2 import { useParams } from 'react-router-dom';
3 import { Modal, Button, Form, Row, Col } from 'react-bootstrap';
4 import 'bootstrap/dist/css/bootstrap.min.css';
5 import { format, parseISO } from 'date-fns';
6 function Client() {
7   const { id } = useParams();
8   const [client, setClient] = useState(null);
9   const [lessons, setLessons] = useState([]);
10  const [abonements, setAbonements] = useState([]);
11  const [loading, setLoading] = useState(true);
12  const [error, setError] = useState(null);
13  const [currentPage, setCurrentPage] = useState(1);
14  const [showModal, setShowModal] = useState(false);
15  const handleClose = () => setShowModal(false);
16  const [selectedAbonement, setSelectedAbonement] = useState(null);
17  const [abonementToAdd, setAbonementToAdd] = useState(null);
18  const [showAddAbonement, setShowAddAbonement] = useState(false);
19  const [abonementOptions, setAbonementOptions] = useState([]);
20  const [isEditing, setIsEditing] = useState(false);
21  const [editedClient, setEditedClient] = useState(client);
22
23  useEffect(() => {
24    if (client) setEditedClient(client);
25  }, [client]);
26
27  const handleEditClick = () => setIsEditing(true);
28  const handleCancelEdit = () => {
29    setEditedClient(client);
30    setIsEditing(false);
31  };
32
33  const handleChange = (e) => {
34    const { name, value } = e.target;
35    setEditedClient(prev => ({
36      ...prev,
37      [name]: name === 'dob' ? new Date(value).toISOString() : value
38    }));
39  };
40  const handleSaveClick = async () => {
41    try {
42      console.log(JSON.stringify(editedClient));
43      const response = await fetch('http://localhost:8000/client/update', {
44        method: 'PUT',
45        headers: { 'Content-Type': 'application/json' },
46        body: JSON.stringify(editedClient),
47      });
48    }

```

Рисунок 2.1 - Вікно з проектом у Visual Studio

Джерело: сформовано автором

Frontend

Для написання інтерфейсу користувача я обрав React JS — це відкритий JavaScript-фреймворк, а точніше, бібліотекою JavaScript, яка використовується для розробки інтерфейсів користувача. Він був створений компанією Facebook і швидко набув популярності серед розробників з усього світу. React дозволяє ефективно створювати застосунки з високою продуктивністю і масштабованістю. Одним з ключових концепцій у React JS є компоненти. Вони представляють собою незалежні блоки коду, які відповідають за рендеринг певної частини користувацького інтерфейсу.

React використовується для розробки користувацького інтерфейсу веб-додатків. Основною метою React Frontend є створення інтерактивних, динамічних та відзивчивих інтерфейсів для користувачів. React Front end дозволяє створювати багатофункціональні та інтерактивні застосунки зі швидким рендерингом і переходом між сторінками. Також він має такі переваги:

– використання віртуального DOM та ефективного алгоритму оновлення дозволяє робити мінімальні зміни в реальному DOM, що покращує продуктивність застосунків;

– JS React базується на компонентній архітектурі, що дозволяє розбивати інтерфейс на незалежні компоненти. Це спрощує розробку, тестування та підтримку коду, оскільки компоненти можуть бути повторно використані та легко змінюються;

– React пропагує односторонній потік даних, що сприяє простоті та передбачуваності управління станом додатків, полегшує відлагодження та тестування застосунків;

– спільнота розробників, що використовує React велика та активна. Є безліч ресурсів, бібліотек та інструментів для розробки. Також існує багато сторонніх бібліотек, які підтримують React і допомагають розширити його функціональність;

React добре підходить для розробки проектів будь-якого масштабу. Він надає можливості для легкого розширення та перевикористання компонентів, інтеграції з іншими бібліотеками та фреймворками. Також підтримує серверний рендеринг, що дозволяє поліпшити швидкість завантаження сторінок та оптимізувати пошукову оптимізацію.

Загалом використання бібліотеки дає можливість React developers ефективно будувати потужні та швидкі інтерфейси, полегшує роботу з компонентами та станом додатків, і має широкую підтримку спільноти розробників. Він відмінно підходить для командної розробки завдяки дотриманню UI та шаблону робочого процесу [3].

Backend

Для написання внутрішньої робочої частини було обрано програмування Golang. Синтаксис Go базується на звичних елементах мови Сі з окремими запозиченнями з мови Python. Мова досить лаконічна, але при цьому код залишається легким для читання і сприйняття. Проект від початку розробляється з оглядкою на багатониткове програмування та ефективну

роботу на багатоядерних системах, в тому числі надаючи реалізовані на рівні операторів засоби для організації паралельних обчислень та взаємодії між паралельно виконуваними методами.

Мова надає вбудовані засоби захисту від виходу за допустимі області виділених блоків пам'яті і забезпечує можливість використання збирача сміття. При цьому код мовою Go компілюється у відокремлені бінарні виконувачі файли, що виконуються нативно без використання віртуальної машини (модулі профілювання, зневадження та інші підсистеми виявлення проблем на етапі виконання інтегруються у вигляді runtime-компонентів), що дозволяє домогтися продуктивності порівнянної з програмами на мові C.

Синтаксис Go дуже схожий на синтаксис C: блоки коду в фігурних дужках; загальна структура керування виконанням програми включає `for`, `switch` та `if`. На відміну від C, крапка з комою в кінці рядка не є обов'язковою; оголошення змінних відбувається по-іншому; перетворення типів строге; нові інструкції `go` та `select` було введено для підтримки паралельного програмування. Нові вбудовані типи включають хеш-таблиці, рядки Unicode, `slices`-масиви та канали для передачі даних між нитками.

Go спроектований для швидкого компілювання, навіть на застарілому обладнанні. Ця мова підтримує прибирання сміття. Чітка паралельно-орієнтована структура Go (канали є альтернативою каналам вводу даних) запозичені з *Communicating Sequential Processes* Тоні Гоара. На відміну попередніх паралельно-орієнтованих мов програмування, таким як *occam* чи *Limbo*, Go не надає будь-якої вбудованої підтримки попереджень безпеки чи перевірки паралельності. На сьогодні Go не має ніякої вбудованої підтримки шаблонів, але це може бути додано в майбутньому [2].

2.2 Система управління базами даних SQL

PostgreSQL — це одна з найстаріших і найнадійніших систем управління базами даних з відкритим кодом, яка поєднує класичний реляційний підхід із можливостями об'єктного моделювання даних. Вона відома своєю

стабільністю, широким спектром налаштувань і безкомпромісною відповідністю стандартам SQL, що робить її основним вибором для розробників складних корпоративних систем.

Однією з ключових особливостей PostgreSQL є архітектура, заснована на процесах, а не потоках. Кожен користувачський запит обробляється окремим процесом операційної системи, що спрощує управління пам'яттю і забезпечує високу стабільність навіть при великій кількості підключень. При цьому дані захищаються через механізм Write-Ahead Logging (WAL), де будь-які зміни спочатку записуються в журнал, що гарантує повне відновлення бази після збоїв.

Система використовує MVCC (Multi-Version Concurrency Control) для забезпечення паралельного доступу до даних без блокувань, що особливо важливо для високонавантажених додатків. Завдяки цій технології транзакції бачать консистентний стан даних на момент початку виконання, що усуває затримки через очікування звільнення ресурсів [7].

У плані індексації PostgreSQL підтримує декілька типів індексів, включаючи традиційні B-tree для швидкого пошуку, GIN для повнотекстового пошуку та JSON-полів, BRIN для великих, але слабо змінюваних таблиць. Ця гнучкість дозволяє ефективно налаштовувати продуктивність під специфіку кожного окремого випадку.

PostgreSQL не обмежується роботою лише з традиційними реляційними структурами. Завдяки вбудованій підтримці JSON і спеціальному типу даних JSONB база даних стала ідеальним рішенням для зберігання та обробки напівструктурованих даних. При цьому, на відміну від класичних NoSQL-рішень, вона дозволяє комбінувати переваги гнучкої структури з потужністю транзакційної обробки.

Ще однією важливою складовою є підтримка горизонтального масштабування через механізми реплікації і шардінгу. Стрімінгова реплікація дозволяє створювати фізичні копії бази для резервування і розвантаження

запитів на читання, а логічна реплікація забезпечує більш гнучке копіювання окремих таблиць або баз на інші сервери.

PostgreSQL активно підтримується великим міжнародним співтовариством розробників і користувачів, що забезпечує постійні оновлення, високий рівень безпеки та швидку адаптацію до нових вимог ринку. Офіційні релізи виходять регулярно, і кожна нова версія приносить не лише виправлення помилок, але й суттєві покращення продуктивності та нові функціональні можливості.

Під час розробки систем на базі PostgreSQL потрібно враховувати кілька важливих технічних особливостей. Наприклад, оптимальна робота потребує коректної настройки параметрів роботи кешу пам'яті, розміру WAL-сегментів, планувальника запитів. Також правильне індексування та використання сучасних типів зберігання даних мають критичне значення для забезпечення стабільної роботи при великих навантаженнях.

У питаннях безпеки PostgreSQL пропонує гнучкі механізми аутентифікації та авторизації. Адміністратор може налаштовувати рівні доступу до окремих таблиць, колонок або функцій, використовуючи різні механізми шифрування з'єднань та зовнішні системи аутентифікації, такі як LDAP або Kerberos.

Основними перевагами PostgreSQL є:

- Реляційна модель даних
- Об'єктно-орієнтовані можливості
- Відповідність стандартам
- Розширюваність
- ACID-сумісність
- Механізми реплікації

Однією з важливих переваг є розвинені механізми розширюваності. У PostgreSQL можна створювати власні типи даних, оператори, агрегатні функції, а також використовувати сторонні розширення. Відомі розширення, такі як

PostGIS, забезпечують роботу з просторовими даними, перетворюючи PostgreSQL на потужну геоінформаційну платформу, що здатна обробляти великі обсяги картографічної інформації.

Система активно розвиває підтримку розподілених баз даних. Такі рішення, як Foreign Data Wrappers (FDW), дають змогу підключати зовнішні джерела даних: інші бази даних, API, файли. Це дозволяє інтегрувати дані з різних систем у єдиний запит без дублювання або попереднього імпорту. Завдяки функціоналу Row-Level Security (RLS), адміністратор може встановлювати правила доступу до даних на рівні окремих рядків таблиці, що значно підвищує захист конфіденційної інформації у багатокористувацьких середовищах.

PostgreSQL також вирізняється своїми можливостями роботи з аналітичними запитами. Завдяки таким функціям, як віконні функції, Common Table Expressions (CTE) та рекурсивні запити, система дозволяє створювати складні обчислювальні звіти без значних витрат продуктивності. Інструменти планування запитів і можливість вручну керувати використанням індексів дають можливість досягти оптимальної швидкодії навіть у найбільших базах даних. PostgreSQL прекрасно інтегрується із контейнерними технологіями на зразок Docker та Kubernetes, що дозволяє автоматизувати розгортання та масштабування інфраструктури. Крім того, завдяки стабільному API та підтримці популярних мов програмування, таких як Python, Java, Go, Node.js та інші, розробники можуть швидко та безпечно створювати гнучкі додатки на основі PostgreSQL.

Підтримка мультиверсійної обробки транзакцій (MVCC) без блокування читачів і записувачів робить PostgreSQL особливо ефективним для високонавантажених систем. У той час як деякі інші бази використовують блокування записів або таблиць, PostgreSQL дозволяє виконувати одночасно тисячі транзакцій без суттєвого впливу на продуктивність.

Серед нових можливостей останніх версій варто відзначити покращення автошардінгу через Foreign Data Wrappers, введення асинхронних запитів і прискорення роботи з великими наборами даних [3].

2.3 Цілі впровадження CRM-системи

Впровадження CRM-системи дозволяє організації комплексно вирішувати завдання, пов'язані з обслуговуванням, аналітикою та збереженням клієнтської бази. Головною метою є забезпечення системного підходу до збору, аналізу та використання інформації про клієнтів для побудови довготривалих і взаємовигідних відносин. Значну увагу приділяють цілісному баченню кожного клієнта. CRM-система надає можливість фіксувати всі точки дотику — звернення через сайт, дзвінки, електронні листи, історію покупок та інші дані. Це створює так званий 360-градусний огляд клієнта, що допомагає приймати персоналізовані рішення, прогнозувати поведінку покупців та вчасно реагувати на їхні потреби.

Також важливою ціллю є підвищення ефективності маркетингових та продажних кампаній. Завдяки впровадженню CRM, компанія отримує змогу сегментувати базу клієнтів за різними критеріями (географія, демографія, історія покупок, поведінка тощо) і надсилати цільові пропозиції. Це підвищує відгук на маркетингові активності, знижує витрати на рекламу та сприяє зростанню конверсії. Управлінський аспект також є важливою складовою: CRM-система дозволяє формувати детальні звіти, графіки та прогнози, які дають змогу керівництву оперативно оцінювати результати діяльності відділів, виявляти «вузькі місця» в обслуговуванні, а також коригувати стратегії ведення бізнесу.

Ще однією метою є забезпечення безперервного обслуговування клієнтів. У багатьох компаніях взаємодія з клієнтами залежить від конкретного менеджера. У разі звільнення працівника або його відсутності, без CRM-системи втрачається важлива інформація. Централізоване зберігання даних усуває цю проблему та забезпечує послідовність обслуговування. Крім цього, CRM-система підтримує інтеграцію з іншими бізнес-інструментами, такими як

бухгалтерські системи, ERP, e-mail-маркетинг, аналітика тощо. Це сприяє створенню єдиного інформаційного простору, де дані не дублюються, а доповнюють одне одного, утворюючи цілісну екосистему управління клієнтськими відносинами.

Впровадження CRM-системи — це стратегічний крок, спрямований на оптимізацію процесів взаємодії з клієнтами, підвищення ефективності персоналу, поліпшення якості обслуговування, збільшення продажів і формування конкурентної переваги на ринку. Це інструмент, який зменшує хаотичність у роботі з клієнтською базою, забезпечує прозорість бізнес-процесів та дозволяє компанії діяти проактивно, а не реактивно.

Висновки до розділу 2

У розділі 2 було розглянуто три основні технології, які були використані при розробці веб-додатку CRM-системи: React, Go (Golang) та PostgreSQL. Кожна з цих технологій має свої унікальні особливості та сильні сторони, що робить їх відмінним вибором для різних частин стека.

React забезпечує зручну платформу для розробки інтерфейсів користувача завдяки компонентному підходу, що дозволяє легко масштабувати і повторно використовувати код. Його здатність до ефективного оновлення DOM за допомогою віртуального DOM дозволяє створювати швидкі й інтерактивні користувацькі інтерфейси, що є важливим аспектом для сучасних веб-додатків.

Go (Golang) виділяється серед інших мов програмування своєю простотою та високою продуктивністю, особливо коли мова йде про створення серверних додатків і обробку великої кількості одночасних запитів. Завдяки своїй конкурентоспроможній моделі паралелізму та надійній підтримці мульти-тредингу, Go ідеально підходить для розробки високопродуктивних API, мікросервісів і розподілених систем.

PostgreSQL є однією з найпотужніших реляційних баз даних, яка пропонує високий рівень стабільності, гнучкості та масштабованості. Її

здатність підтримувати транзакції з рівнем ACID, а також численні можливості для розширення та оптимізації запитів, роблять її ідеальним вибором для зберігання великих обсягів даних у складних системах.

У сукупності ці три технології утворюють потужний стік для створення надійних, масштабованих та високопродуктивних веб-додатків, де кожна з них виконує свою роль: React відповідає за взаємодію з користувачем, Go забезпечує ефективну обробку запитів на сервері, а PostgreSQL забезпечує надійне та ефективне зберігання даних.

РОЗДІЛ 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ CRM-СИСТЕМИ

3.1 Загальні вимоги до інформаційної системи

Процес проектування будь-якого програмного продукту завжди починається з опису вимог до нього. За допомогою математичної моделі та формального опису задач можна виявити необхідний для програмного продукту набір вимог. Перш ніж перейти до проектування системи, важливо встановити завдання досліджуваної системи і способи їх взаємодії. Необхідно визначити основні функціональні вимоги, які пропоновані до системи, тобто визначити діапазон завдань системи та програми бази даних, складу її користувачів і областей застосування.

В результаті розробки CRM-системи створена система з доступом і інтерфейсом для різних категорій користувачів: користувач-відвідувач; зареєстрований-користувач. Набір функцій для цих категорій може бути повністю різний, а може і перетинатися. Розмежування прав доступу в межах інформаційної системи зазвичай здійснюється шляхом дозволу/заборони на перегляд певної інформації для певної групи користувачів. У кожного з користувачів системи є свої права. Залежно від прав, користувачі мають або не мають можливості перегляду тієї чи іншої інформації в системі, або можливість додавання, видалення та редагування інформації. Зареєстрований користувач має доступ до інформації, яка відповідає цьому користувачу, та має право керувати цією інформацією.

Гість має право переглядати всю існуючу інформацію щодо CRM-системи та можливість зареєструватись у системі. Веб-система потребує реєстрації користувачів. Реєстрація потрібна для можливості використання та працездатності системи. Головну перевагу від CRM-системи можна отримати, якщо розмістити на її сторінках необхідну користувачам інформацію. Тому потрібно приділити особливу увагу розробці змісту сторінок. Інформація, розміщена на веб-сторінці, повинна задовольняти наступним критеріям:

- повинна відповідати цілям створення веб-системи;
- повинна враховувати особливості цільового сегмента споживачів;
- повинна бути певною мірою унікальною, щоб привернути увагу відвідувачів;
- повинна бути об'єктивною та достовірною, тому що відвідувачі, які виявлять помилкову або неточну інформацію, навряд чи виникне бажання вдруге звернутися до цієї сторінки;
- не повинна носити надмірно рекламний, нав'язливий характер.

Вся найважливіша інформація, а також посилання і кнопки повинні бути розташовані по периметру одного розміру екрану, так як тільки 10% користувачів прокручують інформацію, що не вміщається на екран. Кожне посилання має супроводжуватися відповідним описом. Якщо посиланням є слово з тексту, то навколишній текст повинен описувати об'єкт, на який вона вказує.

Якщо посилання вказує на файл, який може бути отриманий користувачем, то слід вказати біля неї розмір цього файлу. Також важливу роль у формуванні змісту Web-сторінок грають графічні і мультимедійні матеріали. Вони підвищують ефективність сприйняття повідомлень. Однак тут різко постає питання про кількість та якість графіки. Чи слід розміщувати красиві фотографії і малюнки, які довго завантажуються, або можна обмежитися невеликими низькоякісними зображеннями, збільшивши швидкість завантаження? Обирати відповідний варіант потрібно з урахуванням особливостей цільової аудиторії.

Також потрібно розробити зручну систему навігації. На комп'ютері користувача для коректної роботи веб-системи повинні бути присутні такі програмні продукти:

- веб-оглядач (Mozilla Firefox, Google Chrome, Opera тощо);
- будь-який з браузерів повинен бути оновлений до останніх версій, так як у веб-системі екологічної організації використовуються сучасні технології створення сторінок.

Коли користувач заходить на головну сторінку інформаційної системи, то отримує доступ до переглянути коротку інформацію о системі, та можливість зареєструватися у системі. Головним входом до системи є головна сторінка. Саме від неї належатиме перше враження, що отримується відвідувачами. Текст на головній сторінці повинен знайомити з актуальною інформацією.

Для представлення логічної структури CRM-системи обрана модель представлення даних «сутність-зв'язок». Модель «сутність-зв'язок» ґрунтується на важливій семантичній інформації про реальний світ і призначена для логічного представлення даних. Вона визначає значення даних в контексті їх взаємозв'язку з іншими даними. Визначивши функціональні вимоги до веб-системи, спроектована база даних, що реалізує модель «Сутність – Зв'язок».

При розробці системи була створена база даних, за допомогою СКБД MySQL. Визначено основні сутності проєктованої системи. Сутність (entity) – це об'єкт, який може бути ідентифікований якимсь способом, що відрізняє його від інших об'єктів. Сутність фактично являє собою безліч атрибутів, які описують властивості всіх членів даного набору сутностей. Безліч значень (область визначення) атрибуту називається доменом. Зв'язок – це асоціація, встановлена між кількома сутностями.

Перший етап процесу проєктування бази даних називається концептуальним проєктуванням бази даних. Він полягає у створенні концептуальної моделі даних предметної області. Ця модель даних створюється на основі функціональних вимог користувачів. Концептуальне проєктування бази даних абсолютно не залежить від таких подробиць її реалізації, як тип

обраної СКБД, набір створюваних прикладних програм, використовувани мови програмування, тип обраної обчислювальної платформи, а також від будь-яких інших особливостей фізичної реалізації. Концептуальне проектування – створення концептуального уявлення бази даних, що включає визначення типів найважливіших сутностей та існуючих між ними зв'язків і атрибутів. Послідовність етапів проектування концептуальної моделі даних: визначення сутностей; визначення взаємозв'язків між сутностями; визначення атрибутів сутностей; завдання первинних і альтернативних ключів.

Для бази даних системи виділимо наступні сутності: користувач, замовлення, нотатки, та форум. Кожна сутність повинна містити атрибут або групу атрибутів, які будуть однозначно ідентифікувати кожен екземпляр сутності. Такий атрибут називають первинним ключем. Тому необхідно зробити відбір атрибутів для включення до складу первинного ключа. Первинний ключ повинен бути підібраний таким чином, щоб за значеннями атрибутів, в нього включених, можна було точно ідентифікувати примірник сутності, крім того ніякий з атрибутів первинного ключа не повинен мати нульове значення. Значення атрибутів первинного ключа не повинні змінюватися. Якщо значення змінилося, значить, це вже інший примірник сутності. При виборі первинного ключа можна внести в сутність додатковий атрибут і зробити його ключем.

Так, для визначення первинного ключа часто використовують унікальні номери, які можуть автоматично генеруватися системою при додаванні екземпляра сутності в БД. Застосування унікальних номерів полегшує процес індексації та пошуку в БД. Первинний ключ, обраний при створенні логічної моделі, може бути невдалим для здійснення ефективного доступу до БД і повинен бути змінений при проектуванні фізичної моделі.

Тепер визначимо атрибути для кожної сутності, а також поставимо для всіх сутностей первинні ключі. Далі розставимо зв'язки між сутностями. Представимо базу даних CRM-системи у вигляді моделі «Сутність –Зв'язок» Отримані в результаті проектування об'єкти були приведені до третьої

нормальної форми, так як відсутні транзитивні функціональні залежності неключових атрибутів від ключових. Відповідно об'єкти знаходяться в першій і другій нормальній формі. Перша нормальна форма забороняє повторювання стовпців, забороняє множинні стовпці і вимагає визначити первинний ключ для таблиці. Друга нормальна форма вимагає, щоб не ключові стовпці таблиць зависили від первинного ключа в цілому, але не від його частини. Основним компонентом реляційних БД є таблиця. Таблиця використовується для структуризації та зберігання інформації.

У реляційних БД кожна клітинка таблиці містить одне значення. Крім того, всередині однієї БД існують взаємозв'язки між таблицями, кожна з яких задає спільне користування даними таблиці. Діаграма «Сутність – Зв'язок» графічно представляє структуру даних проектованої інформаційної веб-системи. При проектуванні системи були розглянуті сутності та їх атрибути, які використовуються у роботі CRM-системи. Сутність користувач містить наступні атрибути:

- номер користувача (первинний ключ);
- Ім'я користувача;
- електрона пошта;
- пароль.

Між таблицями Користувач і Заовлення існує зв'язок типу «один до багатьох»: у одного користувача може бути декілька замовлень, у одного користувача може бути декілька нотаток. Так як система має більш інформативний характер і створена с метою просування у мережі Інтернет, то не має сенсу створювати базу даних з великою кількістю сутностей. Далі розглянемо типи даних кожного атрибута сутності.

3.2 Структура й функціонал CRM-системи

Головна сторінка сайту CRM-системи є одночасно сторінкою списку учнів й відображає список клієнтів за: ім'ям, прізвищем, класом, номером телефону датою народження й датою створення акаунту. Також, на сторінці

відображене вікно пошуку. Пошук здійснюється за ім'ям, прізвищем або номером телефону, див рис. 3.1

СПИСОК КЛІЄНТІВ

Ім'я	Прізвище	Клас	Номер телефону	Дата
------	----------	------	----------------	------

Рисунок 3.1 – Меню пошуку

Джерело: сформовано автором

Крім пошуку, на сторінці є функція перегортання бази даних клієнтів, тобто натискаючи кнопки з цифрами під списком клієнтів можна переходити далі по списку, або використовувати стрілки, див. рис. 3.2.

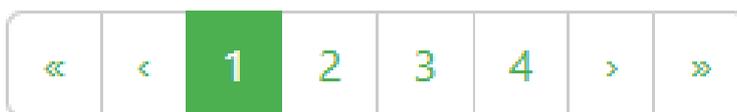


Рисунок 3.2 – Функція перелистування сторінок

Джерело: сформовано автором

На головній сторінці є 4 посилання на інші сторінки CRM-системи. Щоб запустити CRM-систему потрібно увімкнути базу даних, за допомогою відкриття застосунку «school-crm-be» й відкривши папку «school-crm» запустити консоль розробника й написати команду «npm start». Вигляд головної сторінки див. рис. 3.3

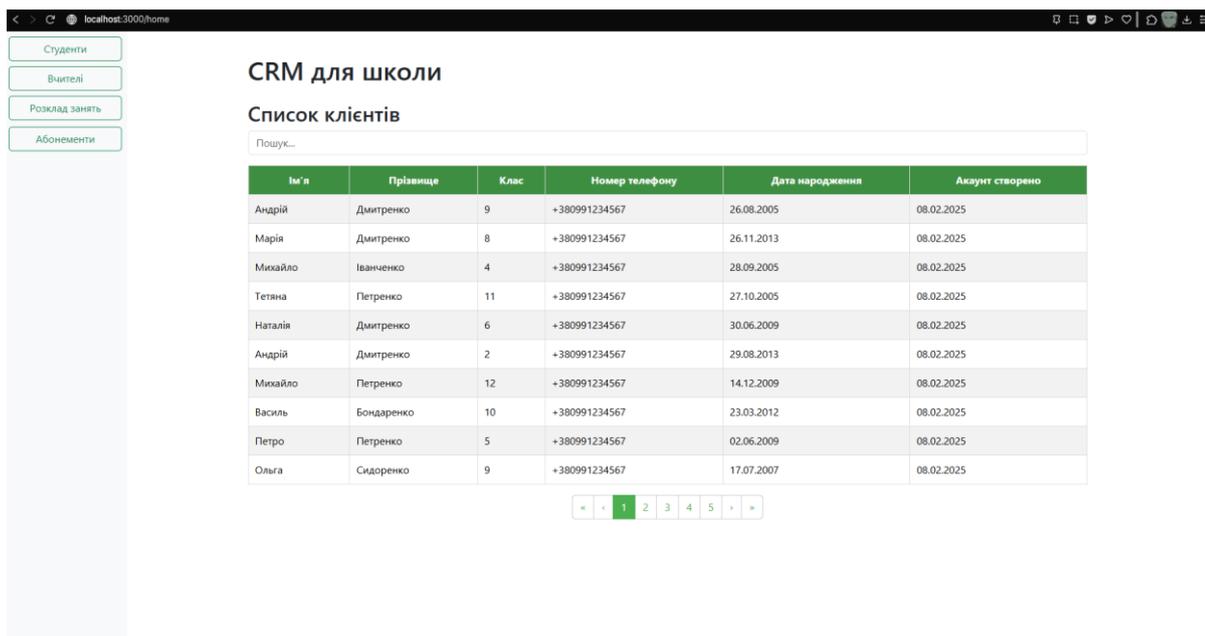


Рисунок 3.3 – Головна сторінка

Джерело: сформовано автором

Також, на кожного «учня» можна натиснути й перейти на його персональну сторінку, де буде більше інформації про конкретного клієнта, див. рис. 3.4

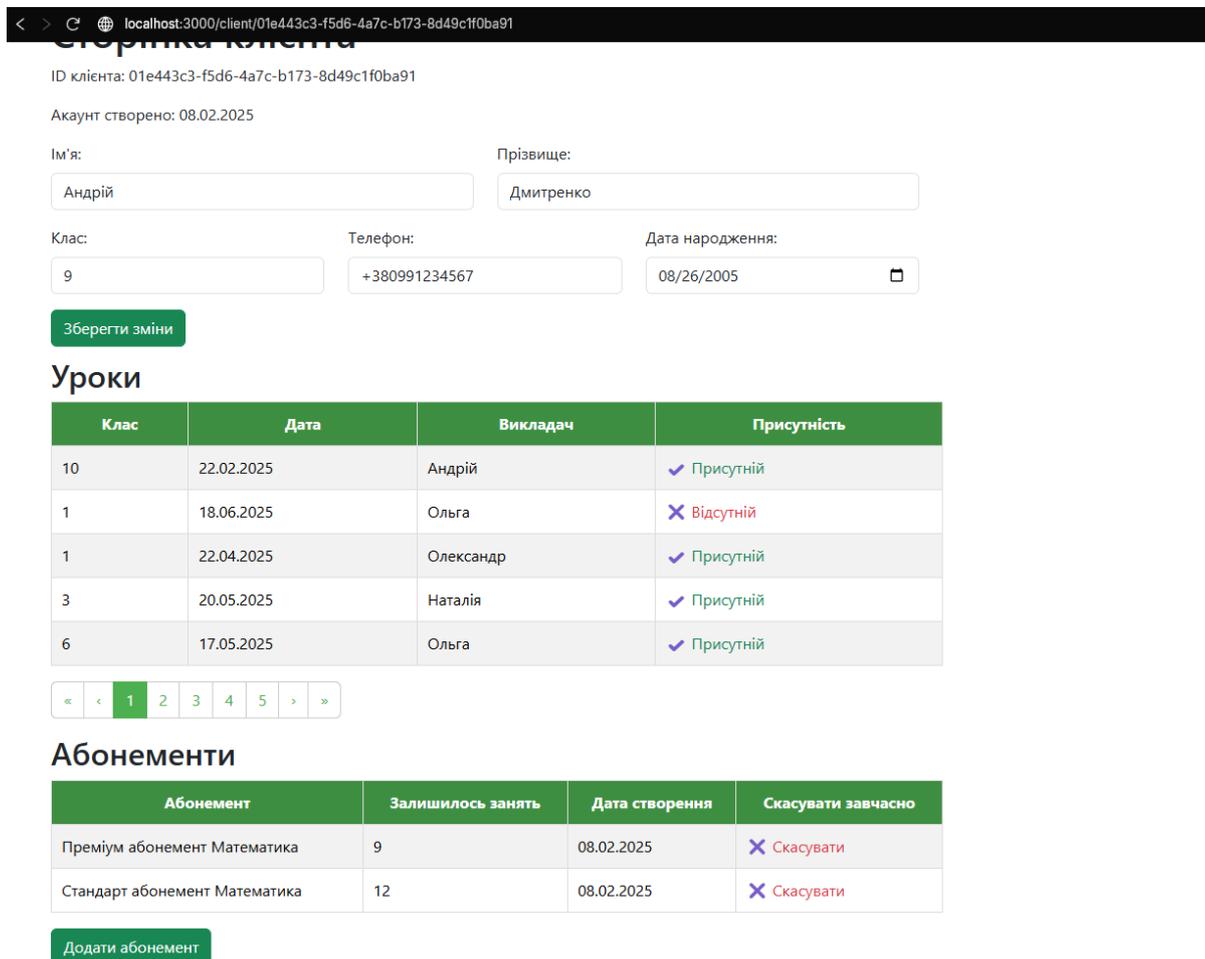
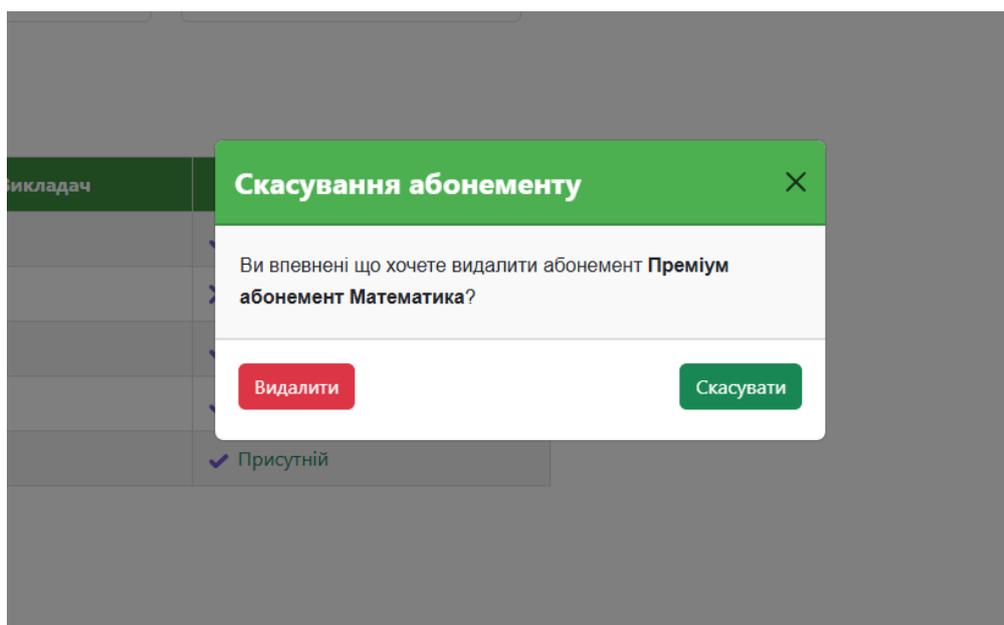


Рисунок 3.4 –Сторінка клієнта

Джерело: сформовано автором

Сторінка клієнта представляє багато інформації про конкретного ученика, таку як: унікальний ID, дату створення акаунту, ім'я, прізвище, клас, телефон, дату народження, уроки й абонементи. Крім того, на сторінці присутня можливість редагування персональної інформації клієнта з можливістю збереження даних, можливість переглядати всі уроки цього ученика (з функціоналом перелистування як на головній сторінці) й функціонал перегляду/редагування абонементів.

Меню «Абонементи» є таблицею, де показано назву абонементу, кількість занять, дату створення абонементу й можливість його скасування. При натисканні на кнопку «Скасувати» у таблиці «Абонементи» ми отримаємо



підтвердження наших дій, див. рис. 3.5.

Рисунок 3.5 – Меню підтвердження видалення абонементу

Джерело: сформовано автором

При натисканні на кнопку «Додати абонемент» буде отримано аналогічне підтвердження зі списком вибору абонементу, див. рис 3.6. Після вибору абонементу й його додавання, на персональній сторінці учня новий абонемент буде відображений у таблиці «Абонементи».

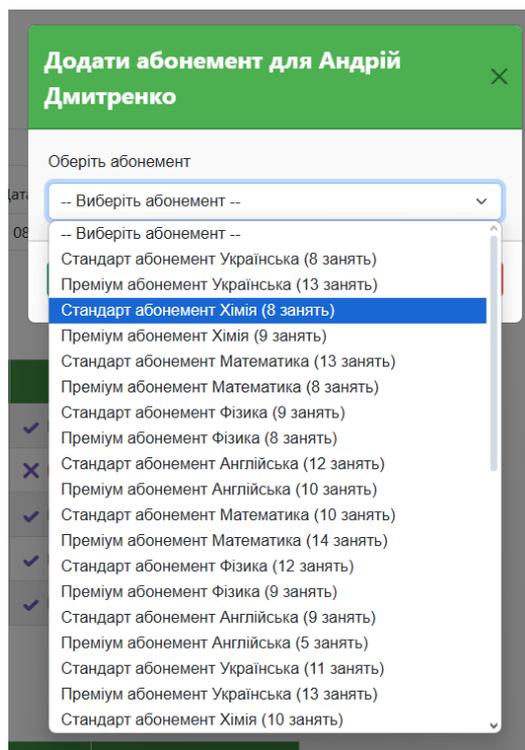


Рисунок 3.6 – Меню вибору додавання абонементу

Джерело: сформовано автором

Сторінка «Список вчителів» є аналогічною до сторінки «Список

Студенти

Вчителі

Розклад занять

Абонементи

CRM для школи

Список вчителів

Пошук...

Ім'я	Прізвище	Номер телефону	Дата народження	Акаунт створено
Андрій	Іванченко	+380991234567	25.11.1992	08.02.2025
Андрій	Коваленко	+380991234567	05.09.1995	08.02.2025
Михайло	Петренко	+380991234567	20.05.1990	08.02.2025
Андрій	Сидоренко	+380991234567	08.02.1989	08.02.2025
Наталія	Петренко	+380991234567	27.12.1990	08.02.2025
Тетяна	Коваленко	+380991234567	18.11.1980	08.02.2025
Марія	Дмитренко	+380991234567	02.07.1997	08.02.2025
Олександр	Іванченко	+380991234567	13.06.1992	08.02.2025
Олександр	Мельник	+380991234567	22.03.1983	08.02.2025
Ольга	Іванченко	+380991234567	24.03.1996	08.02.2025

« < 1 2 3 4 5 > »

клієнтів», з різницею у колонці «Клас», див. рис. 3.7

Рисунок 3.7 – Сторінка «Список вчителів»

Джерело: сформовано автором

При переході на сторінку викладача представлена така інформація: унікальний ID вчителя, дату створення акаунта, ім'я, прізвище, телефон, дата

Сторінка викладача

ID вчителя: 43434c66-d177-42a9-9d6e-38691eb80846

Акаунт створено 08.02.2025

Ім'я:

Андрій

Прізвище:

Іванченко

Телефон:

+380991234567

Дата народження:

11/25/1992

Предмети

Хімія +

Зберегти зміни

Уроки

Клас	Дата
2	02.02.2025
9	13.02.2025
8	06.02.2025
3	24.06.2025
8	09.03.2025

<< < 1 2 3 4 5 > >>

народження, предмети, що веде вчитель й уроки, див. рис. 3.8.

Рисунок 3.8 – Сторінка викладача

Джерело: сформовано автором

Меню «Предмети» має в собі функціонал додавання й видалення предметів, що веде вчитель. Видалення відбувається миттєво, додавання потребує вибору типу предмету й підтвердження додавання, див. рис. 3.9.

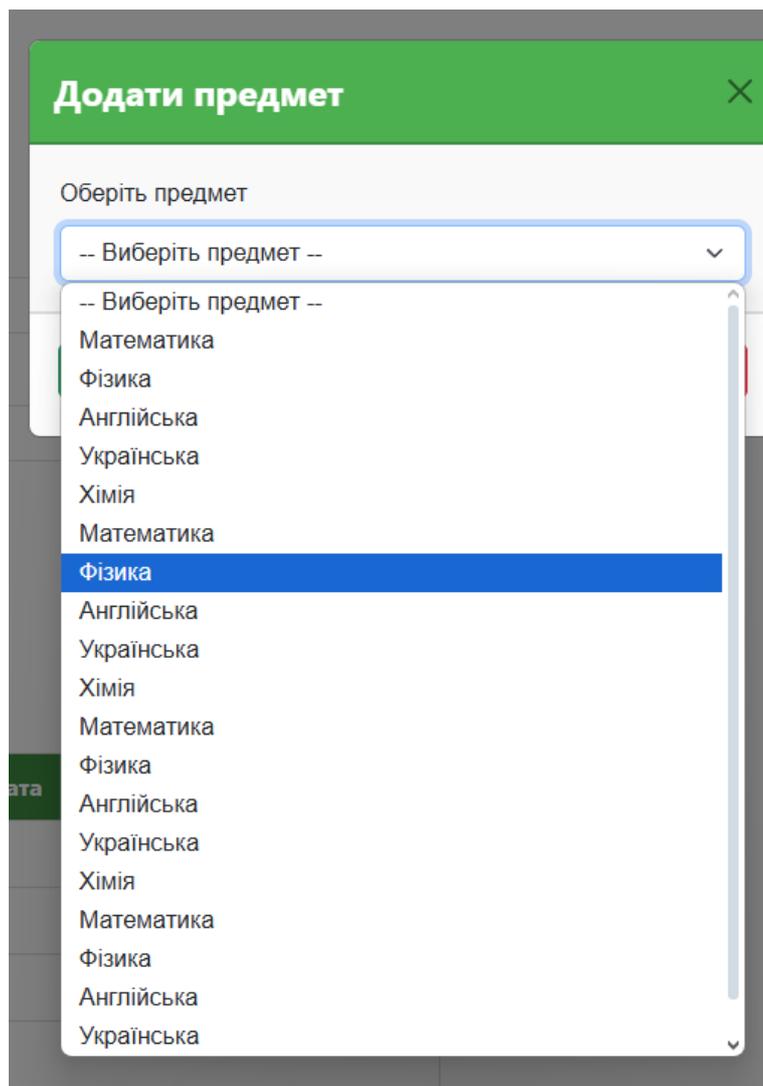


Рисунок 3.9 – Меню вибору додавання предмету викладачу

Джерело: сформовано автором

Сторінка «Розклад занять» являє собою календар з розкладом занять усіх клієнтів/вчителів з функціями перегляду кожного заняття окремо, кнопкою перегляду занять за сьогодні, перегортання розкладу на місяць вперед та редагування присутності на уроці, див. рис. 3.10.

Розклад занять

червень 2025 р.

Сьогодні



нд	пн	вт	ср	чт	пт	сб
1	2	3	4	5	6	7
14:00 практика	14:00 лекція	14:00 практика	14:00 лекція	14:00 практика	14:00 лекція	14:00 лекція
14:00 практика	14:00 лекція	14:00 практика	15:00 лекція	15:00 лекція	14:00 практика	14:00 лекція
14:00 практика	14:00 лекція	15:00 практика	15:00 лекція	15:00 практика	14:00 практика	14:00 практика
+26 more	+20 more	+22 more	+18 more	+26 more	+25 more	+31 more
8	9	10	11	12	13	14
15:00 лекція	14:00 практика					
16:00 лекція	14:00 лекція	14:00 практика	14:00 практика	14:00 практика	14:00 лекція	14:00 практика
17:00 практика	14:00 практика	16:00 лекція	14:00 практика	15:00 лекція	14:00 лекція	15:00 лекція
+17 more	+24 more	+22 more	+27 more	+29 more	+19 more	+24 more
15	16	17	18	19	20	21
14:00 лекція	14:00 лекція	14:00 лекція	14:00 лекція	14:00 практика	14:00 лекція	14:00 практика
14:00 лекція	14:00 лекція	14:00 практика				
14:00 практика	14:00 практика	15:00 лекція	14:00 практика	15:00 лекція	14:00 практика	15:00 лекція
+29 more	+20 more	+16 more	+31 more	+19 more	+29 more	+27 more
22	23	24	25	26	27	28
14:00 практика	14:00 лекція	14:00 лекція	14:00 практика	14:00 лекція	14:00 лекція	14:00 лекція
15:00 лекція	14:00 лекція	14:00 практика	14:00 практика	14:00 лекція	14:00 лекція	14:00 практика
15:00 лекція	14:00 лекція	15:00 лекція	14:00 практика	14:00 лекція	14:00 лекція	14:00 практика
+22 more	+32 more	+22 more	+20 more	+31 more	+17 more	+23 more

Рисунок 3.10 – Сторінка розкладу занять

Джерело: сформовано автором

При натисканні на конкретне заняття відкривається вікно з переглядом й редагуванням присутності, яке відбувається моментально за натискання на кнопку «Присутній/неприсутній», також є основна інформація по заняттю

Заняття 01.06.2025 о 14:00
✕

Інформація про заняття:

Назва: практика
Предмет: Фізика
Викладач: Бондаренко Тетяна

Список учнів:

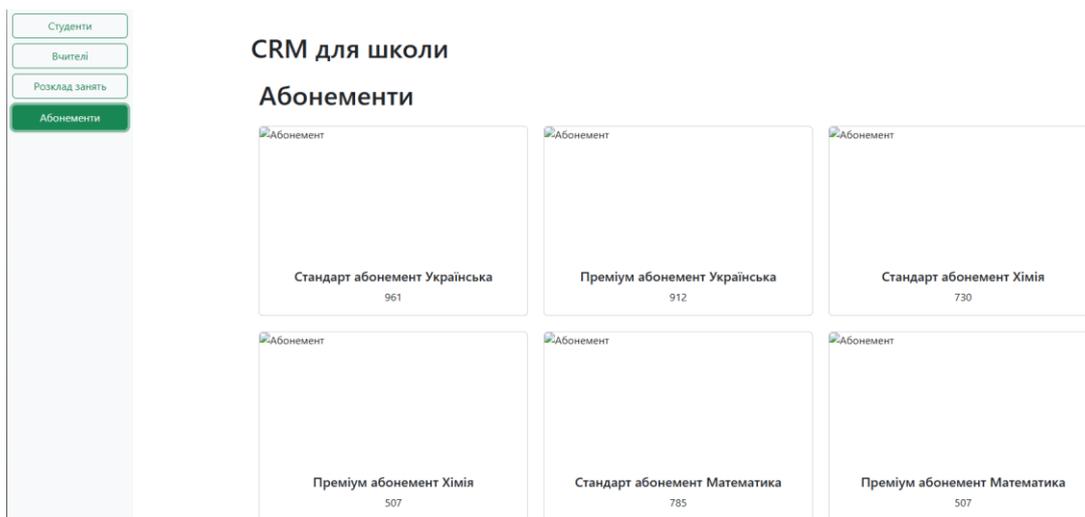
Мельник Тетяна	Присутній
Дмитренко Марія	Присутній
Петренко Олександр	Присутній
Шевченко Андрій	Присутній
Іванченко Василь	Присутній
Бондаренко Петро	Неприсутній
Петренко Тетяна	Присутній
Сидоренко Наталія	Неприсутній
Поліщук Марія	Присутній

(предмет, дата, час, ученики, вчитель), див рис. 3.11.

Рисунок 3.11 – Меню заняття

Джерело: сформовано автором

Сторінка «Абонементи» відображає наявність абонементів різних типів для покупки клієнтами у режимі реального часу. Відображення реалізовано у вигляді прямокутників (які пізніше можна буде замінити на зображення), де написаний рівень абонементу, з якого предмету цей абонемент й його

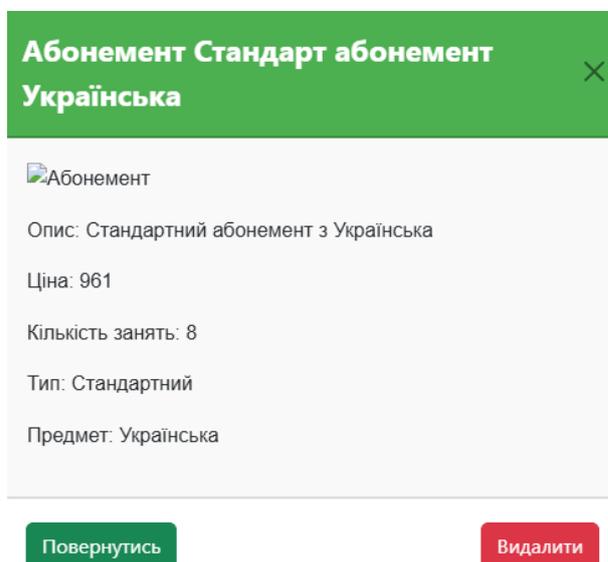


унікальний номер, див. рис. 3.12.

Рисунок 3.12 – Сторінка абонементів

Джерело: сформовано автором

При відкритті певного абонементу з'являється вікно з інформацією за такими параметрами: опис, ціна, кількість занять, тип й предмет. Також, є кнопки «Повернутись» для закриття інформаційного вікна й кнопка



«Видалити», що безпосередньо видаляє абонемент із загального списку, див Рисунок 3.13.

Рисунок 3.13 – Меню певного абонемента

Джерело: сформовано автором

Внизу сторінки знаходиться кнопка реалізації функціоналу «Додавання абонемента», натискаючи яку можна створити новий абонемент. Для створення нового абонемента потрібні такі данні: назва, опис, ціна, кількість занять,

предмет, тип й вибір файлу (якщо потрібно додати зображення), див. рис. 3.14.

Рисунок 3.14 – Меню додавання абонемента

Джерело: сформовано автором

3.3 Тестування додатку

При ручному тестуванні (manual testing) тестувальники *вручну* виконують тести, не використовуючи ніяких засобів автоматизації. Ручне тестування – самий низькорівневий та простий тип тестування, що не вимагає великої кількості додаткових знань.

Тим не менш, перед тим як автоматизувати тестування будь-якого додатку, необхідно спочатку виконати серію тестів вручну. Мануальне тестування вимагає більших зусиль, але без нього ми не зможемо переконатися в тому, чи можлива автоматизація взагалі. Один із фундаментальних принципів тестування свідчить: 100% автоматизація неможлива. Тому, ручне тестування – це необхідність.

Автоматизоване тестування є цікавим та матеріально вигідним заняттям, це важко не помітити. Новачку у сфері тестування завжди буде здаватись, що це найвигідніша альтернатива, адже вона передбачає використання спеціального інструменту для виконання набору тестів продукту. Ручне тестування ж виконується людиною, що сидить перед комп'ютером, ретельно виконує всі етапи тестування із мінімальною кількістю інструментів.

Зазвичай тестувальників-початківців, так само як і свічерів, приваблює концепт того, що можна запуснути код, а потім вже смакувати ранковою кавою, допоки робота «сама себе робить». У реальному житті, не все так просто і на проєктах часто, навіть досвідчені автоматизатори можуть бути спантеличені продуктом, через його природу та особливості. Тож іноді мануальний тестувальник може мати перевагу над автоматизатором.

Отже, можна зробити висновок, що автоматизоване тестування – це тестування за допомогою спеціальних інструментів тестування, які можуть охоплювати великі об'єми та інтенсивність тестових випадків. При створенні цих тестових випадків або тестів, щойно набір тестів автоматизовано, втручання людини є зайвим. Це, власне, і є метою автоматизації – скорочення кількості тестів, які потрібно запускати вручну.

Автоматизоване тестування припускає використання спеціального програмного забезпечення (окрім того, що тестується) для контролю виконання тестів та порівняння очікуваного і фактичного результату роботи програми. Цей тип тестування допомагає автоматизувати дії, що часто повторюються, але які, в той же час, необхідні для максимального тестового покриття завдання.

Плюсом автоматизації ПЗ є те, що вона може вводити тестові дані в систему, здійснювати порівняння очікуваного та фактичного результатів та генерувати детальні звіти про тестування і для цього спеціаліст має володіти мовою програмування. Однак саме тому автоматизація вимагає значного вкладення ресурсів тестувальника у свої знання. Вибір та вивчення мови програмування є одним із таких вкладень в себе, як у майбутнього автоматизатора. Хоча, важливо розуміти, що іноді і тестувальники або розробники вже з досвідом вивчають нові мови для роботи над новими проєктами.

Деякі завдання тестування, такі як – низькорівневе регресійне тестування, можуть бути трудомісними та вимагати багато часу, якщо їх виконувати вручну. Окрім того, мануальне тестування може недостатньо ефективно знаходити деякі класи помилок. У таких випадках автоматизація може допомогти заощадити час і зусилля проектної команди.

Після створення автоматизованих тестів, їх можна в будь-який момент запустити знову, причому запускаються і виконуються вони швидко та точно. Таким чином, якщо є необхідність частого повторного прогону тестів, значення автоматизації для спрощення супроводу проєкту і зниження його вартості важко переоцінити. Адже навіть мінімальні патчі та зміни коду можуть стати причиною появи нових багів.

Існує кілька основних видів автоматизованого тестування:

– *Автоматизація тестування коду (Code-driven testing)* – тестування на рівні програмних модулів, класів та бібліотек (фактично, автоматичні юніт-тести);

– *Автоматизація тестування графічного інтерфейсу користувача (Graphical user interface testing)* – спеціальна програма (фреймворк автоматизації тестування) дозволяє генерувати користувальницькі дії – натиснення кнопок, кліки мишкою, та відслідковувати реакцію програми на ці дії – чи відповідає вона специфікації.

– *Автоматизація тестування API* (Application Programming Interface) – програмного інтерфейсу програми. Тестуються інтерфейси, призначені для взаємодії, наприклад, з іншими програмами або з користувачем. Тут, знову ж таки, як правило, використовуються спеціальні фреймворки.

Для складання автоматизованих тестів, QA-фахівець повинен вміти програмувати. Автоматичні тести – це повноцінні програми, просто призначені для тестування.

Коли, що і як автоматизувати і чи автоматизувати взагалі – дуже важливі питання, відповіді на які повинна дати команда розробки. Вибір правильних елементів програми для автоматизації більшою мірою визначатиме успіх автоматизації тестування в принципі. Потрібно уникати автоматизації тестування ділянок коду, які можуть часто змінюватися.

Як ручне, так і автоматизоване тестування можуть використовуватися на різних рівнях тестування, а також бути частиною інших типів і видів тестування.

Автоматизація зберігає час, сили та гроші. Одного разу автоматизований тест можна запускати знову і знову, докладаючи мінімум зусиль.

Вручну можна протестувати практично будь-який додаток, в той час як автоматизувати варто тільки стабільні системи. Автоматизоване тестування використовується головним чином для регресії. Крім того, деякі види тестування, наприклад, ad-hoc або дослідницьке тестування можуть бути виконані тільки вручну.

Мануальне тестування може бути повторюваним та нудним. У той же час, автоматизація може допомогти цього уникнути – за вас все зробить комп'ютер.

Таким чином, на реальних проектах найчастіше використовується комбінація ручного і автоматизованого тестування, причому рівень автоматизації буде залежати як від типу проекту, так і від особливостей постановки виробничих процесів в компанії.

Після завершення основного етапу розробки CRM-системи було проведено комплексне тестування з метою виявлення помилок, перевірки

відповідності функціоналу вимогам, а також оцінки стабільності та продуктивності системи. Тестування охоплювало як фронтенд, реалізований на React.js, так і серверну частину, створену за допомогою мови програмування Go та бази даних PostgreSQL.

Першочергово було проведено модульне (unit) тестування, яке дозволило перевірити правильність роботи окремих компонентів backend-сервісу. За допомогою вбудованого пакету testing у Go були протестовані основні функції обробки запитів, логіки авторизації та взаємодії з базою даних. Тестові сценарії охоплювали як позитивні, так і негативні кейси, що дозволило переконатися в коректному обробленні виняткових ситуацій.

Інтеграційне тестування здійснювалося для перевірки взаємодії між окремими модулями системи. Було змодельовано типові користувацькі сценарії: реєстрація та авторизація користувача, створення нового клієнта, перегляд історії контактів, генерація звітів. Для цього використовувалися інструменти Postman та Swagger, які дозволяли тестувати REST API запити до сервера.

Фронтенд також пройшов серію тестів. Особлива увага приділялася перевірці коректної роботи форм, відображення повідомлень про помилки, реакції інтерфейсу на запити з сервера, адаптивності дизайну. У процесі ручного тестування були виявлені та усунені помилки валідації форм, недоліки інтерфейсу на мобільних пристроях та несумісність деяких елементів з певними версіями браузерів.

Окремо було проведено навантажувальне тестування бази даних PostgreSQL, щоб переконатися в її здатності ефективно обробляти велику кількість запитів. Було протестовано сценарії з одночасним створенням великої кількості записів та вибірками даних для масивних звітів. Результати показали задовільну продуктивність за умови оптимізації SQL-запитів та використання індексів.

Результатом проведеного тестування стало підтвердження працездатності CRM-системи, її відповідності технічному завданню та готовності до подальшого розгортання й використання в реальних умовах.

Крім традиційного тестування, до процесу перевірки функціональності було залучено автоматизовані підходи, що значно підвищили ефективність виявлення регресійних помилок та забезпечили контроль за стабільністю змін у коді. Для цього були налаштовані скрипти автоматизованого запуску тестів через системи CI/CD (наприклад, GitHub Actions або GitLab CI), що дозволяло виконувати тести щоразу при оновленні репозиторію з вихідним кодом.

Для фронтенду використовувалися бібліотеки Jest і React Testing Library, які дозволили емулювати поведінку користувача, тестувати візуальні компоненти в ізольованому середовищі, а також перевіряти логіку обробки введених даних. Це дало змогу переконатися у правильності відображення інформації, змін станів компонентів і відповідності очікуваному UI-поведінню.

Щодо тестування REST API, було також застосовано інструменти автоматичного тестування типу Newman (CLI для Postman) і Swagger Codegen. Це дозволяло запускати сотні сценаріїв перевірки API одночасно та отримувати структуровані звіти з результатами. Завдяки цьому можна було не тільки пришвидшити тестування, але й значно зменшити ймовірність пропуску критичних багів.

Важливою частиною стала перевірка безпеки системи. Було проведено базовий аудит типових вразливостей: SQL-ін'єкцій, XSS (міжсайтового скриптингу), CSRF (міжсайтових запитів). Було підтверджено, що система належно фільтрує вхідні дані, використовує параметризовані запити до бази даних і застосовує токенізовану авторизацію для захисту сесій.

Також проводилося тестування на відповідність нефункціональним вимогам, зокрема:

- Час відгуку інтерфейсу при типових сценаріях користування не перевищував 300 мс;

– Система залишалася стабільною при навантаженні понад 100 одночасних користувачів;

– При відмові однієї з компонент (наприклад, бази даних або сервера) система коректно повідомляла про помилку, що підтвердило правильну реалізацію обробки збоїв.

Висновки до розділу 3

У третьому розділі було здійснено розробку, структурування та тестування CRM-системи, орієнтованої на потреби онлайн-шкіл. Визначено загальні вимоги до системи, її функціональні характеристики, логічну структуру бази даних, а також ключові принципи формування інтерфейсу з урахуванням категорій користувачів. Запропонована модель передбачає розмежування прав доступу, гнучке налаштування вмісту веб-сторінок, а також побудову структури бази даних відповідно до концепції «сутність–зв’язок» і нормалізації.

Завдяки застосуванню сучасних підходів до архітектури системи, було забезпечено логічну узгодженість між функціональними вимогами, структурою бази даних та інтерфейсними рішеннями. Реалізовано повноцінний функціонал CRM-системи, зокрема: відображення списків учнів і викладачів, управління абонементом, створення розкладу занять, перегляд та редагування персональних даних користувачів, що підтверджує здатність системи підтримувати повний цикл управління освітнім процесом.

У процесі тестування системи були використані як ручні, так і автоматизовані підходи, що дозволило комплексно перевірити функціональність усіх компонентів, виявити та усунути помилки, оцінити продуктивність і стабільність роботи під навантаженням. Особлива увага приділялася перевірці безпеки, сумісності з різними браузерами, а також ефективності взаємодії між frontend та backend частинами.

Таким чином, розроблена CRM-система відповідає встановленим технічним та функціональним вимогам і готова до впровадження в середовище

онлайн-освіти з метою автоматизації ключових процесів і підвищення якості управління навчальною діяльністю.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було здійснено комплексне дослідження предметної області управління взаємовідносинами з клієнтами (CRM) та реалізовано сучасне технічне рішення, спрямоване на автоматизацію ключових бізнес-процесів, що мають вирішальне значення для підвищення ефективності діяльності підприємств малого та середнього бізнесу.

У першому розділі проведено ґрунтовний аналіз потреб сучасних підприємств у впровадженні CRM-систем. Визначено основні проблеми, які постають перед організаціями в умовах динамічного розвитку цифрової економіки, зокрема розрізненість клієнтських даних, відсутність системного підходу до супроводу клієнтів та обмеженість аналітичних можливостей. Обґрунтовано доцільність впровадження CRM-системи як засобу централізованого управління відносинами з клієнтами, що дозволяє підвищити рівень персоналізації обслуговування, ефективність маркетингових і сервісних стратегій, а також якість управлінських рішень.

У другому розділі обґрунтовано вибір технологічного стеку для реалізації проєкту. Зокрема, React було обрано як фреймворк для побудови інтерфейсу користувача завдяки його масштабованості, повторному використанню компонентів і підтримці сучасних веб-стандартів. Golang, у свою чергу, забезпечив ефективну роботу серверної логіки, завдяки високій швидкодії, одночасній обробці запитів і простоті обслуговування. Для побудови реляційної

бази даних застосовано PostgreSQL, що гарантує надійність зберігання інформації, підтримку транзакцій та складних запитів, а також гнучкість у структуризації даних.

У третьому розділі було реалізовано повний цикл розробки CRM-системи — від формування вимог до тестування та перевірки працездатності. Розроблено ключові функціональні модулі системи: управління контактами, взаємодіями з клієнтами, замовленнями, абонементом, аналітичними даними. Інтерфейс системи оптимізовано для зручності користувача з урахуванням принципів UX/UI-дизайну. Здійснено тестування працездатності, навантаження, безпеки та адаптивності системи, що підтвердило її відповідність функціональним і нефункціональним вимогам. Особливу увагу було приділено реалізації ролей користувачів, прав доступу та засобів захисту інформації.

Узагальнюючи, можна зробити висновок, що всі завдання бакалаврської кваліфікаційної роботи були виконані повністю й у повному обсязі досягнуто поставленої мети. Розроблена CRM-система має високий ступінь практичної цінності, є гнучкою, масштабованою та може бути адаптована до потреб різних галузей малого та середнього бізнесу. Її впровадження сприятиме підвищенню ефективності комунікації з клієнтами, зростанню рівня їх задоволеності, оперативному прийняттю рішень і загальному підвищенню конкурентоспроможності підприємства у сучасному цифровому середовищі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PostgreSQL: The World's Most Advanced Open Source Relational Database [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org> – Назва з екрана. – Дата звернення: 27.05.2025.
2. Golang – Go Programming Language [Електронний ресурс]. – Режим доступу: <https://go.dev> – Назва з екрана. – Дата звернення: 27.05.2025.
3. React – A JavaScript library for building user interfaces [Електронний ресурс]. – Режим доступу: <https://reactjs.org> – Назва з екрана. – Дата звернення: 27.05.2025.
4. CRM-системи: що це, для чого потрібні та як обрати [Електронний ресурс]. – Режим доступу: <https://blog.prom.ua/ua/crm-sistemy/> – Назва з екрана. – Дата звернення: 27.05.2025.
5. CRM-система: що це таке і як вона працює [Електронний ресурс]. – Режим доступу: <https://www.microsoft.com/uk-ua/microsoft-365/business-insights-ideas/resources/what-is-a-crm-system> – Назва з екрана. – Дата звернення: 27.05.2025.
6. Порівняння CRM-систем: які бувають і як обрати [Електронний ресурс]. – Режим доступу: https://biz.ligakon.net/analytics/210991_poryvnyannya-crm-sistem – Назва з екрана. – Дата звернення: 27.05.2025.

7. Вступ до PostgreSQL [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/502950/> – Назва з екрана. – Дата звернення: 27.05.2025.
8. Erturk, E., He, S. Study on A High-integrated Cloud-Based Customer Relationship Management System [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1812.09005> – Назва з екрана. – Дата звернення: 27.05.2025.
9. Mithas, S., Almirall, D., Krishnan, M. S. Do CRM Systems Cause One-to-One Marketing Effectiveness? [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/math/0609199> – Назва з екрана. – Дата звернення: 27.05.2025.
10. Claudelin, M. Agile CRM System Implementation Project [Электронный ресурс]. – Режим доступа: <https://lutpub.lut.fi/handle/10024/59847> – Назва з екрана. – Дата звернення: 27.05.2025.
11. Laakso, T. Customer Relationship Management Software Implementation in SMEs [Электронный ресурс]. – Режим доступа: <https://aaltodoc.aalto.fi/items/c19783a1-9190-4cbc-a68a-289396f0aa85> – Назва з екрана. – Дата звернення: 27.05.2025.
12. Mishra, A., Maheswarappa, S. S., Maity, M., Samu, S. Assessing the Implementation of AI Integrated CRM System for B2C Relationship Management [Электронный ресурс]. – Режим доступа: <https://link.springer.com/article/10.1007/s10796-022-10261-w> – Назва з екрана. – Дата звернення: 27.05.2025.
13. Implementation effects in the relationship between CRM and its performance [Электронный ресурс]. – Режим доступа: <https://www.sciencedirect.com/science/article/abs/pii/S0148296318300808> – Назва з екрана. – Дата звернення: 27.05.2025.
14. Implementing customer relationship management successfully: A configurational perspective [Электронный ресурс]. – Режим доступа: <https://www.sciencedirect.com/science/article/abs/pii/S0040162523007680> – Назва з екрана. – Дата звернення: 27.05.2025.

15. Mapping Customer Relationship Management Research in Higher Education: Trends and Future Directions [Электронный ресурс]. – Режим доступа: <https://www.mdpi.com/2076-3387/15/2/68> – Назва з екрана. – Дата звернення: 27.05.2025.

16. ReAct: Synergizing Reasoning and Acting in Language Models [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2210.03629> – Назва з екрана. – Дата звернення: 27.05.2025.

17. Hero: On the Chaos When PATH Meets Modules [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2102.12105> – Назва з екрана. – Дата звернення: 27.05.2025.

18. Ports, D. R. K., Grittner, K. Serializable snapshot isolation in PostgreSQL [Электронный ресурс]. – Режим доступа: <http://dx.doi.org/10.14778/2367502.2367523> – Назва з екрана. – Дата звернення: 27.05.2025.

19. Kawada, A., Nishimura, G., Teramoto, J., Fujita, E., Yamamuro, T., Wakamori, T. PostgreSQL: Leader in Cost Reduction [Электронный ресурс]. – Режим доступа: <http://dx.doi.org/10.53829/ntr201406fa6> – Назва з екрана. – Дата звернення: 27.05.2025.

20. Stonebraker, M., Rowe, L. A. The design of POSTGRES [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/current/biblio.html> – Назва з екрана. – Дата звернення: 27.05.2025.

21. Stonebraker, M., Rowe, L. A., Hirohama, M. The implementation of POSTGRES [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/current/biblio.html> – Назва з екрана. – Дата звернення: 27.05.2025.

22. Kolbe, L. M. Processes and systems for integrated customer relationship management [Электронный ресурс]. – Режим доступа: <http://aleph.unisg.ch/hsgscan/hm00132039.pdf> – Назва з екрана. – Дата звернення: 27.05.2025.

23. Thejaswarup, S. Strategies for Improving the Effectiveness of Customer Relationship Management Systems [Электронный ресурс]. – Режим доступа: <https://scholarworks.waldenu.edu/dissertations/4334> – Назва з екрана. – Дата звернення: 27.05.2025.

24. CRM Implementation Case Studies [Электронный ресурс]. – Режим доступа: <https://crmside.com/crm-case-studies/> – Назва з екрана. – Дата звернення: 02.06.2025.

25. Building a Go, PostgreSQL, Redis REST API with React.js Frontend [Электронный ресурс]. – Режим доступа: <https://medium.com/@ferdousazad12/building-a-go-postgresql-redis-rest-api-with-react-js-frontend-192432380f81> – Назва з екрана. – Дата звернення: 02.06.2025.

26. A Case Study on IKEA's Successful CRM Strategy [Электронный ресурс]. – Режим доступа: <https://medium.com/@loyaltyconsulting0400/a-case-study-on-ikeas-successful-crm-strategy-fa1be8b462ff> – Назва з екрана. – Дата звернення: 02.06.2025.

27. Create a Full-Stack App With Go and React [Электронный ресурс]. – Режим доступа: <https://blog.jetbrains.com/go/2024/11/04/create-a-full-stack-app-with-go-and-react/> – Назва з екрана. – Дата звернення: 02.06.2025.

28. Zurich Insurance's new CRM is like Spotify for insurance agents [Электронный ресурс]. – Режим доступа: <https://www.businessinsider.com/zurich-insurance-ai-customer-relationship-management-crm-system-helps-agents-2025-5> – Назва з екрана. – Дата звернення: 02.06.2025.

29. CRM Adoption in Small and Medium Enterprises: A Literature Review [Электронный ресурс]. – Режим доступа: <https://www.sciencedirect.com/science/article/pii/S1877050918319122> – Назва з екрана. – Дата звернення: 02.06.2025.

30. Modern Full-Stack Development: Using Go, React, and PostgreSQL [Электронный ресурс]. – Режим доступа: <https://www.oreilly.com/library/view/modern-full-stack-development/9781492076327/> – Дата звернення: 02.06.2025.

