

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ
Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

ГАВРОСЬ ІЛЛЯ ВАЛЕНТИНОВИЧ

РОЗРОБКА CRM-СИСТЕМИ МОВОЮ ПРОГРАМУВАННЯ PHP

Спеціальність: 122 Комп'ютерні науки
Освітньо–професійна програма: Комп'ютерні науки та інформаційні
технології
Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»

Науковий керівник:
ГОЛОВІН МИКОЛА
БОРИСОВИЧ,
Кандидат фізико-математичних наук,
доцент
кафедри комп'ютерних наук та
кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ
Протокол № _____
засідання кафедри комп'ютерних наук
та кібербезпеки від _____ 2025 р.
Завідувач кафедри

(_____) Гришанович Т. О.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1.....	5
CRM-СИСТЕМИ: ІСТОРІЯ ТА СУЧАСНИЙ СТАН.....	5
1.1 Визначення CRM-системи та її призначення.....	5
1.2 Історія виникнення та розвитку CRM-систем.....	5
1.3 Класифікація CRM-систем: операційні, аналітичні, колаборативні.....	5
1.4 Сучасні тенденції та популярні CRM-рішення.....	5
РОЗДІЛ 2.....	5
АНАЛІЗ І ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ	
2.1 Постановка задачі та вимоги до системи.....	5
2.2 Вибір мови програмування та середовища розробки.....	5
2.3 Інструменти та фреймворки що використовуються в проєкті.....	5
РОЗДІЛ 3.....	5
3.1 Архітектура системи.....	6
3.2 Структура проєкту (або опис структури проєкту).....	5
3.3 Етапи реалізації програмного засобу.....	5
3.3.1 Реєстрація та авторизація користувача.....	5
3.3.2 Панель адміністратора.....	5
3.3.3 Створення та обробка тікетів.....	5
3.3.4 Профіль користувача.....	5
РОЗДІЛ 4.....	5
4.1 Методи тестування.....	5
4.2 Результати тестування.....	5
ВИСНОВКИ.....	5
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	5
ДОДАТКИ.....	5

ВСТУП

В умовах динамічного розвитку ринку та жорсткої конкуренції для сучасного бізнесу стає критично важливим не лише залучення нових клієнтів, а й збереження та ефективне управління вже наявними. Задоволення потреб клієнта, швидке реагування на звернення, персоналізоване обслуговування та якісна підтримка — усе це вимагає систематизованого підходу, який виходить за межі звичайного обліку контактної інформації. Саме тому системи управління взаємовідносинами з клієнтами (CRM-системи — Customer Relationship Management) набули широкого поширення у різних галузях — від сфери обслуговування до державного управління.

CRM-система дозволяє централізувати й автоматизувати процеси взаємодії з клієнтами: зберігати історію контактів, аналізувати поведінку споживачів, здійснювати облік запитів та звернень, контролювати виконання завдань менеджерами тощо. У результаті підприємство отримує можливість не лише підвищити ефективність внутрішньої комунікації, а й покращити якість обслуговування клієнтів, що безпосередньо впливає на фінансові результати діяльності.

Останні роки показали значне зростання інтересу до впровадження CRM-систем навіть у невеликих компаніях. Це пов'язано із зростанням доступності веб-технологій, зниженням вартості розробки програмного забезпечення та необхідністю оперативного прийняття рішень на основі аналітичних даних. У зв'язку з цим особливої актуальності набуває розробка власних CRM-рішень, адаптованих до специфічних потреб конкретної організації, замість використання універсальних, але громіздких та дорогих комерційних продуктів.

Метою даної дипломної роботи є розробка прототипу CRM-системи для обробки клієнтських звернень із використанням мови програмування PHP.

Обрана мова дозволяє створювати функціональні, зручні у супроводі та розширенні веб-застосунки з невеликими витратами ресурсів. В основі проєкту — принципи простоти, мінімалізму та ефективності, що забезпечують можливість використання розробленої системи як у навчальних, так і в реальних умовах.

У процесі роботи були поставлені наступні завдання:

- провести аналіз існуючих CRM-систем, їх типів і функціоналу;
- визначити вимоги до майбутнього програмного забезпечення;
- обґрунтувати вибір мови програмування, технологій та архітектурного підходу;
- реалізувати основні модулі системи (реєстрація, авторизація, створення й обробка тікетів, рольова модель доступу, робота з базою даних);
- протестувати систему та оцінити її працездатність;
- визначити можливості подальшого розвитку та вдосконалення проєкту.

Об'єктом дослідження є процес організації взаємодії з клієнтами за допомогою цифрових інструментів. Предметом дослідження — програмна реалізація CRM-системи на базі веб-технологій із використанням PHP.

Практичне значення даної роботи полягає у створенні працюючого прототипу CRM-системи, який можна адаптувати під потреби малого чи середнього підприємства, а також використати як навчальний приклад для вивчення веб-розробки.

РОЗДІЛ 1

CRM-СИСТЕМИ: ІСТОРІЯ ТА СУЧАСНИЙ СТАН

1.1 Визначення CRM-системи та її призначення

У сучасних умовах одним із ключових факторів успіху будь-якої організації є ефективна взаємодія з клієнтами. Саме тому все більше компаній впроваджують CRM-системи (Customer Relationship Management) — комплексні інструменти, що забезпечують централізоване управління всіма аспектами стосунків з клієнтами.

CRM-система — це програмне забезпечення, яке дозволяє організувати, автоматизувати та синхронізувати бізнес-процеси, пов'язані з маркетингом, продажами, обслуговуванням клієнтів і технічною підтримкою. Вона виступає єдиним джерелом збереження та аналізу всієї інформації про клієнтів, що дозволяє покращити якість обслуговування, збільшити рівень задоволеності клієнтів і, як наслідок, підвищити прибутковість компанії.

Основна ідея CRM полягає в побудові довготривалої співпраці із клієнтами через глибоке розуміння їхніх потреб, історії покупок, вподобань і поведінки. Для цього CRM-системи надають такі функціональні можливості:

- ведення бази клієнтів із повною історією взаємодій;
- автоматизація процесів продажів і комунікацій;
- формування звітів і аналітики;
- планування зустрічей, дзвінків, розсилок;

CRM-системи дають можливість керівникам відслідковувати ефективність працівників, розподіляти навантаження між менеджерами, контролювати прогрес у виконанні планів. З іншого боку, для працівників це — зручний інструмент для щоденної роботи, який зменшує кількість ручної праці, зводить до мінімуму ризики втрати інформації та підвищує загальну продуктивність.

Таким чином, впровадження CRM-системи дає змогу забезпечити персоналізований підхід до кожного клієнта, побудувати стабільну комунікацію та реалізовувати ефективні маркетингові стратегії. CRM — це не просто технічне рішення, а філософія ведення бізнесу, орієнтована на клієнта.

1.2 Історія виникнення та розвитку CRM-систем

Історія CRM-систем починається задовго до їх цифрової епохи. Ще в середині XX століття підприємці та менеджери прагнули зберігати важливу інформацію про клієнтів — імена, попередні покупки, уподобання — щоб забезпечити якісніше обслуговування. Спершу це робилося в ручному режимі: у картотеках, блокнотах або на друкованих бланках. Такий підхід був обмеженим за обсягом і швидкістю обробки даних.

Зі зростанням обсягів бізнесу та появою персональних комп'ютерів у 1980-х роках, почали з'являтися перші системи автоматизації продажів (SFA) Sales Force Automation, контакт-менеджери, які дозволяли зберігати контактну інформацію та історію комунікацій. Одним із перших програмних рішень у цій сфері був АСТ! — система управління контактами, яка набула популярності серед малого бізнесу.

У 1990-х роках відбувся прорив: термін Customer Relationship Management почав активно використовуватися в корпоративному середовищі. Саме тоді на ринку з'явилися перші повноцінні CRM-системи, які об'єднували функціональність управління клієнтами, продажами, маркетингом і сервісом.

Визначальну роль у цьому відіграли компанії Siebel Systems, SAP, Oracle, PeopleSoft.

На цьому етапі CRM-системи були складними й дорогими — їх інсталяція потребувала окремих серверів, налаштування баз даних та тривалої технічної підтримки. Вони використовувалися переважно великим бізнесом.

Новий етап розвитку розпочався у 2000-х роках із поширенням інтернету та хмарних технологій. Першим великим SaaS-рішенням у сфері CRM стала Salesforce, яка запропонувала компаніям доступ до CRM як до хмарного сервісу без потреби в інсталяції. Це революціонізувало ринок, зробивши CRM-доступними для малого та середнього бізнесу.

З 2010-х років CRM-системи почали активно інтегруватися з соціальними мережами, мобільними застосунками, email-маркетингом і аналітикою. З'явилися функції штучного інтелекту: прогнозування поведінки клієнтів, автоматичне сегментування аудиторії, чат-боти.

На сучасному етапі CRM-системи — це вже не просто база даних клієнтів, а комплексні платформи, що підтримують повний цикл клієнтського досвіду (Customer Experience). Вони дозволяють працювати в реальному часі, адаптуються до потреб конкретного бізнесу, масштабуються та інтегруються з будь-якими сторонніми сервісами.

Таким чином, історія CRM-систем — це еволюція від паперових записів до інтелектуальних хмарних рішень, яка тісно пов'язана з розвитком інформаційних технологій, зміною підходів до ведення бізнесу та зростанням ролі клієнтського сервісу як ключової конкурентної переваги.

1.3 Класифікація CRM-систем: операційні, аналітичні, колаборативні

CRM-системи, як і більшість бізнес-інструментів, мають різні напрями спеціалізації залежно від потреб компаній. Відповідно до міжнародної

практики, CRM-рішення прийнято класифікувати за функціональним призначенням на три основні категорії: операційні, аналітичні та колаборативні (співпраця). У сучасних умовах також з'являються гібридні моделі, які поєднують риси кількох типів для комплексного покриття бізнес-процесів.

Операційна CRM (Operational CRM) — це найпоширеніший та найбільш базовий тип CRM, який автоматизує щоденні процеси взаємодії з клієнтами. Такі системи охоплюють:

- управління контактами — зберігання даних про клієнтів та історії їхньої активності;
- автоматизацію продажів — створення угод, формування лідів, контроль за етапами воронки продажів;
- обслуговування клієнтів — обробка запитів, підтримка, створення заявок;
- кампанії — організація розсилок, дзвінків, взаємодія через різні канали комунікації.

Цей тип CRM використовується безпосередньо менеджерами з продажу, маркетологами, працівниками кол-центрів. Основна мета — покращити взаємодію з клієнтом "тут і зараз", зробити процес обслуговування швидким, контрольованим і персоналізованим. Прикладом такої системи може бути Bitrix24, Zoho CRM або AmoCRM — платформи, які зосереджені на щоденній роботі з лідами та діючими клієнтами.

Аналітична CRM (Analytical CRM) орієнтована на глибокий аналіз інформації про клієнтів, їхню поведінку, історію покупок, реакцію на маркетингові кампанії. Завдяки використанню аналітики, штучного інтелекту та алгоритмів машинного навчання, такі системи допомагають:

- сегментувати клієнтів за різними критеріями;
- виявляти тренди й закономірності у продажах;
- прогнозувати майбутні потреби клієнтів;

- оцінювати рентабельність клієнтів (RFM-аналіз);
- підвищувати ефективність маркетингових стратегій.

На відміну від операційної CRM, аналітична не призначена для повсякденного використання менеджерами. Її основні користувачі — керівники відділів, аналітики, стратегічні планувальники.

Як приклад, Salesforce CRM або Microsoft Dynamics 365 мають потужні модулі аналітики, інтегровані з BI-платформами, які дозволяють формувати дашборди, індикатори KPI, робити багатофакторний аналіз.

Колаборативна CRM (Collaborative CRM) фокусується на узгодженій роботі між усіма підрозділами компанії, які взаємодіють із клієнтами. Це особливо важливо для середнього та великого бізнесу, де один клієнт може одночасно контактувати з кількома відділами: продажами, логістикою, техпідтримкою, бухгалтерією тощо.

- Функції такої CRM включають:
- централізовану базу даних;
- синхронізацію комунікацій через email, телефон, чат;
- єдину історію клієнта для всіх співробітників;
- планування задач та спільної роботи.

Колаборативні системи часто інтегруються з засобами корпоративного спілкування (Slack, Microsoft Teams), календарями, обміном документами, службами підтримки.

Як приклад — CRM-рішення HubSpot, яке дозволяє не тільки вести клієнтську базу, а й організовувати ефективну співпрацю між командами.

1.4 Сучасні тенденції та популярні CRM-рішення

У XXI столітті CRM-системи продовжують динамічно розвиватися відповідно до нових бізнес-викликів, цифрових трендів і зростаючих очікувань клієнтів. Сучасні CRM-рішення вже не обмежуються лише базовим збереженням контактів чи автоматизацією процесів — вони трансформуються у комплексні платформи управління клієнтським досвідом (Customer Experience Management), глибоко інтегровані з усіма бізнес-системами.

Головні тенденції розвитку CRM у сучасному світі:

Мобільність і доступність у хмарі (cloud-first):

Сьогодні більшість CRM-систем розміщуються в хмарних середовищах, що забезпечує доступ до інформації з будь-якої точки світу, цілодобово. Це особливо важливо для віддаленої роботи, яка стала нормою після пандемії COVID-19.

Інтеграція з соціальними мережами та омніканальність:

Сучасні CRM-платформи забезпечують єдину взаємодію з клієнтами через email, месенджери, телефонію, соціальні мережі (Facebook, Instagram, WhatsApp). Це дозволяє не втрачати запити та будувати безшовну комунікацію.

Штучний інтелект (AI) і машинне навчання (ML):

Більшість сучасних CRM мають вбудовані або інтегровані інструменти AI. Вони дозволяють прогнозувати поведінку клієнтів, автоматично розпізнавати "гарячих" лідів, оптимізувати комунікації, генерувати персоналізовані пропозиції.

Автоматизація бізнес-процесів (workflow automation):

CRM-системи не просто зберігають дані, а й запускають послідовності дій — наприклад, надсилають лист після запиту, призначають задачі менеджерам, змінюють статуси угод тощо.

Глибока аналітика та візуалізація даних:

Вбудовані дашборди, KPI, інтеграція з BI-сервісами (Power BI, Google Data Studio) — все це допомагає приймати обґрунтовані управлінські рішення на основі реальних даних.

Підвищення рівня безпеки:

З розвитком GDPR, ISO та інших стандартів зростає увага до зберігання персональних даних клієнтів. Сучасні CRM мають шифрування, контроль доступів, логування активностей користувачів.

РОЗДІЛ 2

АНАЛІЗ І ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ

2.1 Постановка задачі та вимоги до системи

CRM-система у сучасних умовах ведення бізнесу є дуже зручним та корисним інструментом для управління взаємодією з клієнтами. У межах даної **дипломної роботи** реалізовано вебзастосунок CRM-системи мовою програмування PHP для малого бізнесу, яка дозволяє ефективно керувати клієнтськими заявками, адмініструвати користувачів і вести історію звернень. Розроблена система має модульну структуру з поділом на користувацький та адміністративний інтерфейси, що забезпечує простоту використання і масштабованість проєкту.

Функціональні вимоги:

Реєстрація та авторизація користувачів: реалізовано окремі форми для реєстрації, входу в систему та зміни пароля. Передбачено логування дій користувачів.

Рольова модель: реалізовано поділ між звичайними користувачами (менеджерами) та адміністраторами. Адміністратор має доступ до керування користувачами, перегляду журналів доступу, управління заявками.

Створення, перегляд та редагування заявок (тікетів): користувачі можуть створювати звернення, переглядати історію заявок та змінювати їхній статус.

Робота з клієнтськими запитами (quotes): реалізовано функціонал створення та керування запитами від клієнтів.

Адміністративна панель: адміністратор має доступ до керування користувачами, заявками, перегляду історії взаємодії та статистики.

Графічне представлення даних: використано бібліотеку Highcharts для відображення статистики у вигляді графіків (за потреби).

Захист доступу: реалізовано перевірку доступу до сторінок, захист сесій, контроль логіна/логаута.

Визначення цільової аудиторії:

Розроблена CRM-система орієнтована на невеликі компанії, які надають послуги та мають потребу у веденні обліку клієнтських звернень. Це можуть бути ІТ-компанії, сервісні центри, ремонтні майстерні, юридичні або бухгалтерські фірми.

2.2 Вибір мови програмування та середовища розробки

Для розробки CRM-системи було обрано мову програмування PHP, яка є однією з найпоширеніших у сфері веброзробки. Основними критеріями вибору стали її простота у використанні, широке розповсюдження, активна спільнота розробників. Завдяки цьому PHP забезпечує швидку розробку вебдодатків і має вбудовані засоби для роботи з базами даних, зокрема з MySQL, яка використовується у проєкті для зберігання інформації про користувачів, заявки та клієнтські запити.

PHP дозволяє легко реалізувати динамічні вебсторінки, обробку форм, керування сесіями та автентифікацію користувачів. У даному проєкті він поєднується з HTML, CSS (Bootstrap) та JavaScript для створення зручного користувацького інтерфейсу. Упроваджено базову авторизацію та захист доступу до сторінок, а логіка обробки запитів користувачів винесена в окремі PHP-скрипти. Це дозволяє спростити підтримку та подальший розвиток системи.

Розробка велась у середовищі Visual Studio Code з використанням локального сервера XAMPP, що забезпечує доступ до повноцінного LAMP-стека (Apache, MySQL, PHP). Такий підхід дозволив створити гнучке, доступне для розгортання рішення, придатне для малого та середнього бізнесу. PHP також забезпечив хорошу інтеграцію з зовнішніми бібліотеками, зокрема для

побудови графіків статистики, що покращує візуалізацію аналітичних даних у системі.

2.3 Інструменти та фреймворки, що використовуються в проєкті

Під час розробки CRM-системи було використано низку інструментів, бібліотек і технологій, які забезпечили реалізацію функціональних можливостей, зручний інтерфейс та обробку даних. Комбінація серверних і клієнтських технологій дозволила створити повноцінний вебзастосунок з поділом прав доступу та інтерактивною взаємодією користувача з системою.

Основним середовищем розробки слугував XAMPP, який забезпечив локальний вебсервер з підтримкою Apache, MySQL і PHP, що дозволило ефективно тестувати систему в процесі створення.

Для реалізації серверної логіки застосовувалася мова PHP без використання повноцінного фреймворку. Логіка побудована у вигляді окремих модулів (реєстрація, авторизація, створення заявок тощо), які взаємодіють із базою даних MySQL через стандартні засоби PHP (розширення mysqli).

База даних реалізована у MySQL та містить кілька таблиць, що відповідають за зберігання даних користувачів, заявок, логів доступу та ролей. Уся взаємодія з базою організована через SQL-запити, вбудовані у PHP-скрипти.

На клієнтській стороні використовуються такі технології та бібліотеки:

- HTML5 і CSS3 — для створення структури та стилів сторінок;
- Bootstrap — для швидкої адаптивної верстки та зручного оформлення елементів інтерфейсу;
- JavaScript — для реалізації динамічної поведінки сторінок, зокрема валідації форм;

— Highcharts — бібліотека для побудови інтерактивних графіків і візуалізації статистики в адміністративному розділі (наприклад, графік кількості заявок).

Крім того, використовуються вбудовані механізми керування сесіями РНР для реалізації авторизації, контролю доступу та зберігання даних користувача між запитами.

Інтерфейс адміністратора та користувача реалізовано як окремі частини системи, кожна з яких має доступ до відповідного функціоналу згідно з роллю, що визначається у базі даних.

Таким чином, у проєкті не застосовуються складні MVC-фреймворки чи CMS-системи, що дає змогу краще контролювати логіку роботи і гнучко змінювати структуру системи залежно від вимог.

РОЗДІЛ 3

3.1 Архітектура системи

CRM-система, реалізована в межах дипломного проєкту, побудована на основі класичної клієнт-серверної архітектури, де веббраузер користувача виступає як клієнт, а сервер із встановленим PHP-інтерпретатором обробляє запити, працює з базою даних і формує відповіді.

Архітектура системи умовно поділена на декілька взаємодіючих рівнів:

Презентаційний рівень (Presentation Layer)

Це інтерфейс користувача, реалізований у вигляді HTML-сторінок з використанням CSS та JavaScript. Він містить окремі частини для адміністратора та звичайного користувача. Ролі визначаються через авторизацію, а доступ до сторінок — через перевірку поточного сеансу.

Основні компоненти:

- форма входу в систему (login.php);
- головна сторінка користувача (client/index.php);
- інтерфейс адміністратора (admin/index.php);
- шаблонні компоненти (меню, шапка, футер).
- Логічний рівень (Application Logic Layer)

Відповідає за обробку бізнес-логіки: створення заявок, їхній перегляд, зміна статусів, керування користувачами. Цей рівень частково реалізовано у вигляді скриптів PHP, які взаємодіють із базою даних напряду через db.php.

Наприклад:

- додавання тикета (create_ticket.php);
- перегляд детальної інформації по заявці;

- оновлення статусу (для адміністратора);
- авторизація та реєстрація користувача.

Рівень доступу до даних (Data Access Layer)

Представлений базою даних MySQL, яка містить таблиці для зберігання інформації про користувачів, тікети, статуси та ролі. Взаємодія з базою реалізована через розширення mysqli.

Типові операції:

- вибірка даних (SELECT),
- додавання нових записів (INSERT),
- оновлення статусів (UPDATE),
- фільтрація заявок за користувачами чи статусами.

Сервісна інфраструктура

Система використовує сесії PHP (\$_SESSION) для зберігання інформації про поточного користувача, а також механізми перенаправлення та захисту від несанкціонованого доступу. Всі адміністративні сторінки перевіряють наявність активної сесії та відповідну роль користувача.

Загальна схема побудови системи дозволяє легко масштабувати функціонал, додавати нові модулі (наприклад, повідомлення електронною поштою, аналітику, експортування звітів) або перенести систему на повноцінний MVC-фреймворк у майбутньому без суттєвих змін у структурі бази даних чи логіці.

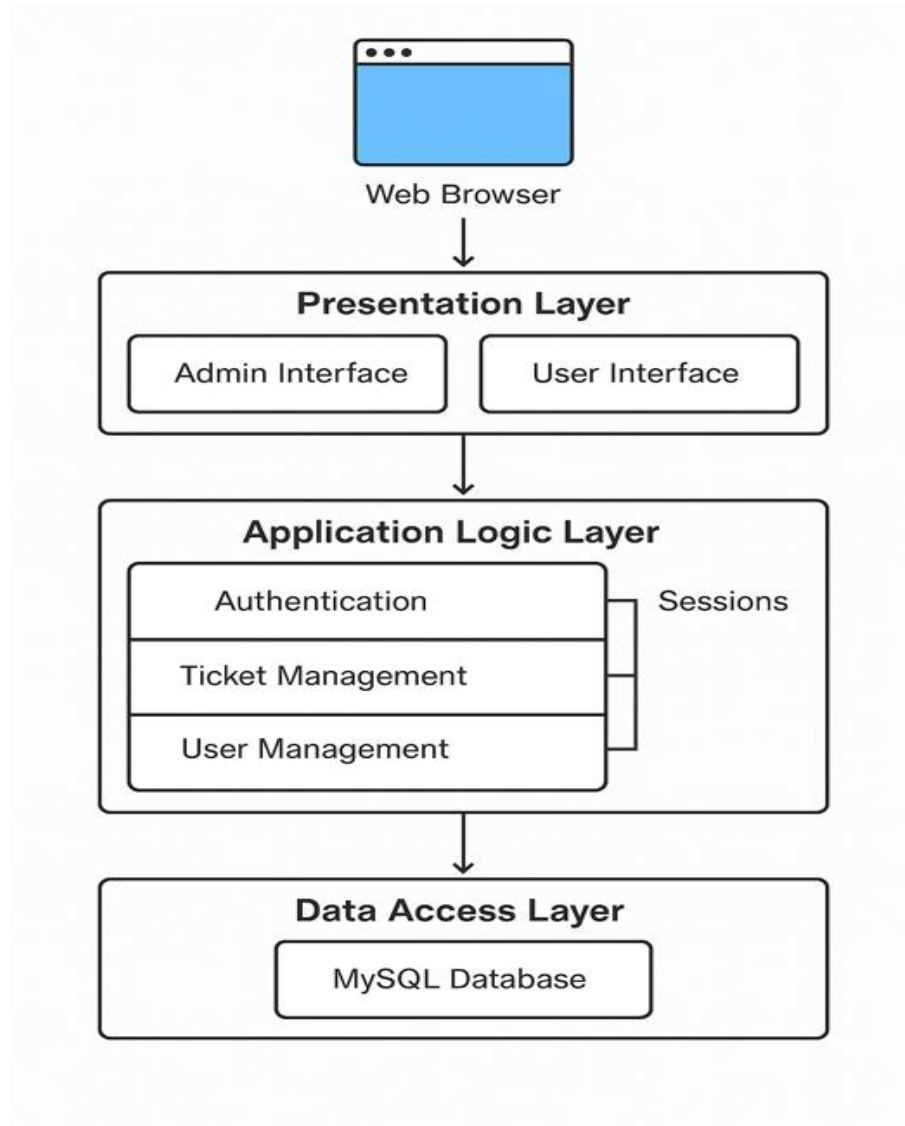


Рисунок 3.1 – Схема архітектури проєкту

3.2 Структура проєкту

CRM-система, реалізована в межах дипломного проєкту, має чітко організовану файлову структуру, що забезпечує розділення функціональності, полегшує підтримку й подальший розвиток проєкту. Структура організована за принципами модульності й логічного поділу обов’язків між окремими частинами застосунку.

На верхньому рівні проєкт містить дві основні функціональні підсистеми: клієнтську (користувацьку) та адміністративну, реалізовані у вигляді окремих директорій `client/` і `admin/`. Такий підхід дозволяє забезпечити рольовий доступ до різних розділів системи — звичайні користувачі мають змогу створювати та

переглядати заявки, тоді як адміністратори мають розширений функціонал для керування заявками, користувачами та системною статистикою.

У директорії `client/` зосереджено всі сторінки, пов'язані з діями звичайного користувача: створення заявки, перегляд статусу, редагування облікового запису. Кожна сторінка супроводжується шаблонними елементами, що підключаються через PHP, зокрема шапкою сайту, навігаційним меню та футером.

Директорія `admin/` містить окремий інтерфейс для адміністраторів. Тут реалізовано перегляд усіх наявних тікетів, можливість змінювати їхній статус, додавати коментарі, видаляти або редагувати інформацію. Також доступна функція перегляду зареєстрованих користувачів і базова аналітика.

Окрім цих двох частин, у кореневій директорії проєкту розташовано базові скрипти, які обробляють реєстрацію (`register.php`), вхід до системи (`login.php`) та завершення сеансу (`logout.php`). Файл `db.php` відповідає за встановлення з'єднання з базою даних MySQL, використовуючи розширення `mysqli`, що забезпечує стабільну та безпечну взаємодію з даними.

Також окремо винесено файли шаблонів (у папці `includes/`), які підключаються до кожної сторінки для забезпечення єдиного вигляду інтерфейсу. До таких елементів належать: шапка сайту (`header.php`), бічне меню (`sidebar.php`) та інші повторювані частини.

Папка `assets/` містить статичні ресурси: таблиці стилів CSS, скрипти JavaScript, а також сторонні бібліотеки й зображення. Це дозволяє розділити логіку відображення від логіки обробки даних, дотримуючись базових принципів розробки вебзастосунків.

Сесійне керування користувачами реалізоване через стандартні засоби PHP (`$_SESSION`). При завантаженні захищених сторінок система перевіряє наявність активної сесії та права доступу користувача до відповідного розділу

(адміністративного або користувацького). Це дозволяє забезпечити базову безпеку й обмеження доступу без використання сторонніх CMS чи фреймворків.

Отже, структура проєкту є логічно поділеною, з чітким розмежуванням за ролями й функціоналом, що полегшує її обслуговування, тестування й масштабування. Така організація забезпечує прозору архітектуру і дозволяє легко вносити зміни до окремих модулів без ризику порушити роботу всієї системи.

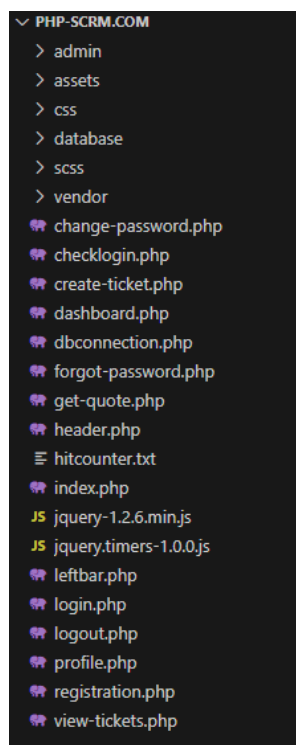


Рисунок 3.2 – Структура проєкту

3.3 Етапи реалізації програмного засобу

Процес створення CRM-системи розпочався з визначення основних функціональних можливостей, які мала забезпечувати система. До них належали: реєстрація та авторизація користувачів, розмежування доступу відповідно до ролей (звичайний користувач та адміністратор), створення та обробка звернень (тікетів), залишення відповідей, а також перегляд аналітичної інформації у вигляді графіків.

Після цього було спроектовано структуру бази даних. Створено таблиці для користувачів, ролей, звернень, повідомлень, відповідей та статусів. Була розроблена логічна модель зв'язків між сутностями, що забезпечує зручну обробку запитів та масштабованість у майбутньому.

Наступним етапом стало формування структури проєкту, організація файлів, підключення до бази даних та налаштування середовища розробки на основі ХАМРР. Було підключено стилізаційний фреймворк Bootstrap для швидшого створення адаптивного інтерфейсу, а також застосовано базові інструменти JavaScript для покращення динамічності роботи сторінок.

Розробка інтерфейсу передбачала створення головної сторінки, вікон реєстрації та авторизації, особистого кабінету користувача, а також адміністративної панелі. Усі сторінки адаптовано до різних типів пристроїв, реалізовано базові перевірки полів введення на клієнтському рівні.

Особливу увагу було приділено реалізації функціоналу звернень. Користувач мав змогу створювати нові тікети, вказуючи тему, опис, категорію звернення, переглядати свої поточні звернення та історію їх обробки. Адміністратор, у свою чергу, мав розширений функціонал: перегляд усіх звернень у системі, можливість залишати відповіді, змінювати статус тикета (наприклад, “новий”, “у роботі”, “закрито”).

Для підвищення зручності взаємодії адміністраторів із системою було додано блок зі статистикою, реалізований із використанням бібліотеки Chart.js. Графіки відображають розподіл звернень за статусами, а також кількість звернень за останні періоди.

Після реалізації основного функціоналу було проведено етап тестування. Перевірялися всі ключові функції системи: створення користувача, вхід у систему, створення та обробка звернень, робота з повідомленнями, відображення статистики. Виявлені помилки було усунуто, а також проведено базову оптимізацію SQL-запитів та очищення коду від надлишкових елементів.

Заключним етапом стала підготовка системи до розгортання: завершено верстку, оптимізовано розміщення файлів, реалізовано базовий захист від SQL-ін'єкцій, використано хешування паролів, а також збереження даних сесії на сервері для контролю доступу.

3.3.1 Реєстрація та авторизація користувача

Першим етапом реалізації системи стало створення механізмів реєстрації та авторизації. У базі даних створено таблицю users, у якій зберігаються всі необхідні дані: ім'я, email, пароль (захешований), роль користувача та інша супутня інформація (рис. 3.3).

id	name	email	alt_email	password	mobile	gender	address
4	????	salosilske@gmail.com	NULL	12345678	00953950008	male	NULL
5	Roni Col	havros.crypto@gmail.com		12345678	00953950008	female	
2	Mark Cooper	mcooper@mail.com	mcoop123@mail.com	mcooper1234	09123654789	m	Sample Address only
3	glalv1	gavrosbest@gmail.com		12345678	0953950007	male	
1	Admin Administrator	info@mail.com	admin@mail.com	admin123	1234567890	male	Block 6, Lot 14, 23 St., Here City, Down There, 23...

Рисунок 3.3 – Таблиця users

Реєстрація передбачає перевірку на унікальність email, валідацію введених даних, збереження у базу та перехід до системи після успішного входу. На рисунку 3.3.1 зображено фрагмент коду реєстрації користувача.

```

if($_SERVER['REQUEST_METHOD'] == 'POST')
{
    $name=$_POST['name'];
    $email=$_POST['email'];
    $password=$_POST['password'];
    $mobile=$_POST['phone'];
    $gender=$_POST['gender'];
    $query=mysqli_query($con,"select email from user where email='$email'");
    $num=mysqli_fetch_array($query);
    if($num>1)
    {
        echo "<script>alert('Email already register with us. Please try with diffrent email id.');

```

Рисунок 3.3.1 – Реєстрація користувача

3.3.2 Панель адміністратора

Модуль панелі адміністратора є важливою складовою CRM-системи, оскільки надає можливість керування ключовими елементами системи: користувачами, заявками, задачами, повідомленнями тощо. Доступ до цього інтерфейсу має лише авторизований користувач з роллю адміністратора, що зберігається в базі даних та перевіряється при кожному зверненні до адміністративної частини сайту.

Основні функції модуля адміністратора:

- Перегляд, редагування та видалення користувачів;
- Перегляд усіх заявок (з форм зворотного зв'язку або CRM-форми);
- Контроль за повідомленнями, які надходять від користувачів;
- Статистика та перегляд дій у системі;
- Можливість змінювати статуси задач або заявок.

Інтерфейс

На рисунку 3.3.2 можемо побачити інтерфейс адміністративної панелі реалізовану окремо від користувацької частини, з акцентом на зручність керування. Зовнішній вигляд побудований з використанням HTML, CSS та Bootstrap. Навігація організована через бокове меню, яке дозволяє швидко перемикатися між основними розділами: "Користувачі", "Заявки", "Повідомлення", "Налаштування".

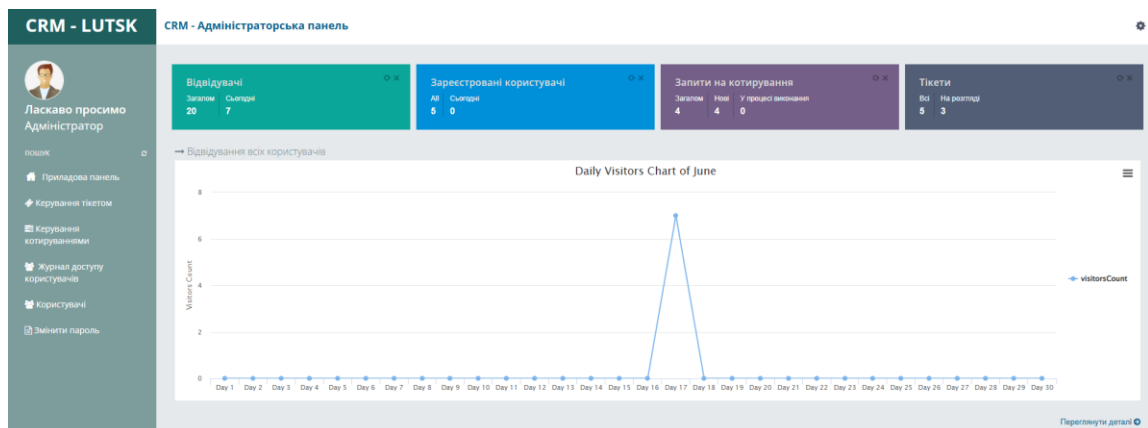


Рисунок 3.3.2 – Інтерфейс панелі адміністратора

3.3.3 Створення та обробка тікетів

Модуль тікетів у CRM-системі реалізує базову систему звернень користувачів, які можуть подавати заявки (тікети) на розв'язання певної проблеми або з питаннями щодо роботи сервісу. Цей функціонал дозволяє зручно комунікувати з адміністрацією сайту та вести облік запитів.

Основні можливості модуля:

- Створення нового тікета користувачем;
- Перегляд списку власних тікетів;
- Перегляд тікетів у панелі адміністратора;
- Зміна статусу тікета (новий, у процесі, завершено);

- Надсилання відповіді адміністратором;
- Відображення історії звернення (запит і відповідь).

Створення тікета

На стороні користувача реалізована форма для створення тікета (ticket_create.php), яка включає поля: тема звернення, текст повідомлення та, за потреби, додаткову контактну інформацію. Дані надсилаються через POST-запит і зберігаються у відповідній таблиці бази даних (tickets).


```

session_start();
//echo $_SESSION['id'];
//$_SESSION['msg'];
include("dbconnection.php");
include("checklogin.php");
check_login();
if(isset($_POST['send']))
{
$count_my_page = ("hitcounter.txt");
$hits = file($count_my_page);
$hits[0] ++;
$fp = fopen($count_my_page , "w");
fputs($fp , "$hits[0]");
fclose($fp);
$tid=$hits[0];
$email=$_SESSION['login'];
$subject=$_POST['subject'];
$tt=$_POST['tasktype'];
$priority=$_POST['priority'];
$ticket=$_POST['description'];
//$ticfile=$_FILES["tfile"]["name"];
$st="Open";
$pdate=date('Y-m-d');
$a=mysqli_query($con,"insert into ticket(ticket_id,email_id,subject,task_type,prioprity,ticket,status,posting_date)
values('$tid','$email','$subject','$tt','$priority','$ticket','$st','$pdate')");
if($a)
{
echo "<script>alert('Ticket Genrated'); location.replace(document.referrer)</script>";
}
}

```

Рисунок 3.3.3 – Код створення тикета

Перегляд та обробка тикетів

У кабінеті адміністратора реалізований інтерфейс для перегляду усіх надісланих тикетів (admin/tickets.php). Адміністратор може змінювати статуси тикетів та додавати відповіді. Відповіді зберігаються у пов'язаній таблиці (наприклад, ticket_responses), щоб вести історію комунікації. Окрім того, статус тикета змінюється на відповідний (in progress, closed) залежно від дій адміністратора.

Відображення у профілі користувача

На стороні користувача передбачено сторінку зі списком усіх створених тикетів, їх статусами та відповідями адміністратора. Це реалізовано у вигляді таблиці, яка витягує дані з бази за ідентифікатором користувача.

3.3.4 Профіль користувача

Профіль користувача є ключовим модулем у CRM-системі, що дозволяє кожному зареєстрованому користувачеві переглядати й редагувати особисту інформацію, змінювати пароль, а також керувати своїми тікетами.

Інтерфейс профілю реалізовано у вигляді окремої сторінки (наприклад, profile.php), доступ до якої мають лише авторизовані користувачі. При завантаженні сторінки система автоматично ідентифікує користувача за сесією та виводить відповідні дані з бази. Ці дані включають:

- Ім'я користувача;
- Email-адресу;
- Роль (user/admin);
- Дату реєстрації;
- Список надісланих тікетів.

Редагування профілю реалізовано через окрему форму, де користувач може оновити ім'я, змінити пароль (зі старим підтвердженням), а також у майбутньому — додати зображення профілю або інші персональні дані. Дані передаються через метод POST та обробляються на сервері для оновлення інформації у відповідній таблиці бази даних.

```
session_start();
include("dbconnection.php");
include("checklogin.php");
check_login();
if(isset($_POST['update']))
{
    $name=$_POST['name'];
    $aemail=$_POST['alt_email'];
    $mobile=$_POST['phone'];
    $gender=$_POST['gender'];
    $address=$_POST['address'];
    $a=mysqli_query($con,"update user set name='$name',mobile='$mobile',gender='$gender',alt_email='$aemail',
    address='$address' where email='".$_SESSION['login']."'");
    if($a)
    {
        echo "<script>alert('Your profile updated successfully.');

```

Рисунок 3.3.4 – Оновлення даних профілю

Для захисту доступу до профілю та зміни даних застосовано перевірку активної сесії, а також базову перевірку прав доступу. Якщо користувач не

авторизований або намагається змінити чужі дані — його буде автоматично перенаправлено на сторінку входу. На сторінці профілю також може бути розміщена кнопка «Вийти», яка завершує сесію користувача та переадресовує на головну або логін-сторінку.

Таким чином, модуль профілю користувача забезпечує персоналізований доступ до функціоналу CRM, зберігаючи при цьому гнучкість і безпеку взаємодії.

РОЗДІЛ 4

4.1 Методи тестування

Для забезпечення надійної роботи CRM-системи було проведено тестування ключових компонентів проєкту з використанням ручного (мануального) та частково модульного підходів. Основною метою тестування було виявлення помилок у логіці обробки запитів, перевірка коректності взаємодії з базою даних та оцінка зручності користувацького інтерфейсу.

У процесі тестування було використано наступні методи:

1. Функціональне тестування

Перевірялося, чи відповідає поведінка системи її специфікації. Основну увагу було приділено таким сценаріям:

- реєстрація та авторизація користувача;
- створення тікета з боку користувача;
- перегляд тікетів у кабінеті адміністратора;
- зміна статусу заявки;
- автоматичне обмеження доступу до функціоналу, не передбаченого роллю користувача.

2. Тестування інтерфейсу користувача (UI-тестування)

Оцінювалася коректність відображення елементів на сторінках, зручність навігації та візуальна відповідність кнопок, форм і таблиць:

- правильне завантаження сторінок у різних браузерях (Chrome, Firefox, Edge);
- перевірка адаптивності макету (оскільки використовується Bootstrap);
- відповідність кольорової гами та шрифтів елементам дизайну.

3. Тестування захисту та сесій

Було протестовано основні механізми безпеки, включно з:

- обмеженням доступу до адміністративного розділу без авторизації;
- збереженням та завершенням сесій після виходу користувача;
- перевіркою на SQL-ін'єкції шляхом ручного введення спеціальних символів у поля форми (тест пройдено завдяки базовій обробці даних).

4. Тестування взаємодії з базою даних

Перевірялась стабільність операцій читання, запису та оновлення інформації в базі даних:

- правильність збереження нових заявок;
- коректне оновлення статусів тікетів;
- видимість змін для відповідного користувача або адміністратора.

5. Кросбраузерне тестування

Система успішно протестована в сучасних браузерах, що забезпечує її універсальність для кінцевих користувачів. Повна функціональність доступна при мінімальній ширині вікна 768 пікселів.

4.2 Результати тестування

У процесі тестування CRM-системи було перевірено основні функціональні можливості, що реалізовані у проєкті. Тестування проводилося вручну, із залученням кількох користувачів для перевірки поведінки системи в різних ролях: звичайного користувача та адміністратора.

Результати показали, що система виконує основні задачі відповідно до поставлених вимог:

Процедури реєстрації та авторизації користувача працюють стабільно. У разі введення правильних облікових даних система надає доступ до відповідного функціоналу (залежно від ролі), при цьому некоректні дані викликають інформативне повідомлення про помилку. Створення тікетів від імені користувача реалізовано коректно. Дані успішно записуються у базу даних, і адміністратор має можливість переглядати нові звернення та змінювати їх статус. Контроль доступу працює належним чином. Неавторизовані користувачі не мають змоги перейти до захищених сторінок, відбувається автоматичне перенаправлення на сторінку входу. Адміністративна панель дозволяє змінювати статуси тікетів, що відображаються користувачам у їх персональному кабінеті. Всі зміни передаються через базу даних і коректно

відображаються в інтерфейсі. Сесії керуються коректно: після виходу користувача або адміністратора доступ до системи припиняється до наступного входу.

Окремі виявлені недоліки:

Інтерфейс частково втрачає адаптивність при перегляді на дуже малих екранах (наприклад, на мобільних пристроях нижче 768 px ширини). У формах не реалізовано повноцінний захист від SQL-ін'єкцій — частина введених вручну шкідливих SQL-рядків може бути оброблена системою. Це потребує доопрацювання.

Загальний висновок:

CRM-система функціонує стабільно в основних сценаріях використання. Виявлені дрібні недоліки не критичні, однак потребують виправлення для підвищення рівня безпеки та зручності користування. Система готова до подальшого розширення та впровадження у реальні робочі умови.

ВИСНОВКИ

У межах виконання дипломної роботи було реалізовано повний цикл створення CRM-системи для обробки клієнтських звернень, використовуючи мову програмування PHP та супутні веб-технології. Система дозволяє автоматизувати основні процеси взаємодії з клієнтами, зокрема реєстрацію звернень (тікетів), їх обробку адміністратором, контроль статусу запитів і збереження історії взаємодій.

На початковому етапі дослідження було розглянуто історію розвитку CRM-систем, визначено їх основні типи (операційні, аналітичні, колаборативні), а також проаналізовано сучасні рішення на ринку. Це дозволило визначити функціональні вимоги до розроблюваної системи та окреслити коло технологій, що найкраще підходять для реалізації поставлених завдань.

У ході реалізації було обґрунтовано вибір мови PHP як основного інструменту розробки. PHP надає широкі можливості для створення веб-застосунків, має низький поріг входження, гарну інтеграцію з базами даних і підтримку великої кількості бібліотек. Для збереження даних використано систему керування базами даних MySQL, інтерфейс побудовано з використанням HTML, CSS і JavaScript.

Створена система містить два рівні доступу — для користувачів і адміністраторів. Користувач має змогу зареєструватися, увійти до системи, створити тикет і переглядати статус поданих звернень. Адміністратор, у свою чергу, може переглядати всі звернення, змінювати їх статуси та вести облік звернень. Авторизація та контроль доступу реалізовані за допомогою вбудованих механізмів PHP-сесій.

Під час тестування системи було перевірено роботу основного функціоналу, зокрема:

- процес реєстрації та авторизації користувачів;
- створення та обробка тикетів;
- обмеження доступу до функцій згідно з роллю користувача;
- коректність збереження та відображення даних у базі.

Отримані результати свідчать про стабільну роботу системи, відповідність реалізованого функціоналу поставленим вимогам та можливість практичного використання у навчальному процесі або як основи для впровадження у малому бізнесі.

Таким чином, поставлену мету дипломної роботи досягнуто — розроблено дієву CRM-систему з базовими функціями, що дозволяє ефективно організувати процес обробки клієнтських звернень через веб-інтерфейс.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Simple Highcharts MySQL та PHP приклад від ItSolutionStuff. URL: <https://itsolutionstuff.com/post/simple-highcharts-chart-example-using-php-mysql-databaseexample.html> (дата звернення: 06.05.2025)
2. Highcharts: підключення даних із бази даних MySQL/PHP, офіційна документація. URL: <https://www.highcharts.com/docs/working-with-data/data-from-a-database> (дата звернення: 06.05.2025)

3. HighchartsPHP — PHP-обгортка для Highcharts. URL:
<https://goncaloqueiros.net/highcharts.php> (дата звернення: 06.05.2025)
4. Get started with Bootstrap 5.3 | Bootstrap. URL:
<https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 06.05.2025)
5. Bootstrap — світовий responsive CSS- фреймворк. URL:
<https://getbootstrap.com/> (дата звернення: 06.05.2025)
6. Bootstrap CSS 3.3 Reference. URL:
<https://getbootstrap.com/docs/3.3/css/> (дата звернення: 06.05.2025)
7. Вступ до MySQLi (PHP- розширення для MySQL). URL:
<https://www.tutorialspoint.com/mysqli/index.htm> (дата звернення: 06.05.2025)
8. MySQLi – PHP.net документація. URL:
<https://www.php.net/manual/en/book.mysqli.php> (дата звернення: 06.05.2025)
9. PHP + MySQL базовий приклад. URL:
https://www.w3schools.com/php/php_mysql_intro.asp (дата звернення: 06.05.2025)
10. PHP сесії — безпека і контроль (Cloudways). URL:
<https://www.cloudways.com/blog/php-session-security/> (дата звернення: 06.05.2025)
11. Enhancing PHP Session Security: Best Practices (Medium). URL:
<https://medium.com/@bazlyankov/enhancing-php-session-security-best-practices-and-solutions-c8d3ef22632d> (дата звернення: 06.05.2025)
12. OWASP Session Management Cheat Sheet. URL:
https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (дата звернення: 06.05.2025)
13. What is CRM? Salesforce official guide. URL:
<https://www.salesforce.com/crm/what-is-crm/> (дата звернення: 06.05.2025)
14. PHP Sessions: HTTP session management & security. URL:
<https://www.php.net/manual/en/session.security.php> (дата звернення: 06.05.2025)
15. PHP Sessions: документована підтримка сесій у PHP. URL:
<https://www.php.net/manual/en/features.sessions.php> (дата звернення: 06.05.2025)

16. Що таке CRM-система та як вона працює | Creatio. URL: <https://www.creatio.com/ua/crm/what-is-crm> (дата звернення: 05.05.2025)
17. CRM- системи: огляд та типи | IT- Enterprise. URL: <https://www.it.ua/knowledge-base/technology-innovation/customer-relationship-management-crm> (дата звернення: 05.05.2025)

ДОДАТКИ

ДОДАТОК А

Технічне завдання

1. Загальна характеристика

Розроблювана CRM-система призначена для автоматизації процесів обліку клієнтів, керування зверненнями (тікетами), а також організації внутрішньої взаємодії між співробітниками. Система реалізована у вигляді веб-застосунку на мові програмування PHP з використанням бази даних MySQL.

2. Основні функціональні можливості

Для звичайного користувача:

- Реєстрація облікового запису.
- Авторизація у системі.
- Створення нового тикету із зазначенням теми та опису.
- Перегляд історії поданих тикетів.
- Контроль статусу обробки (новий / у процесі / завершено / відхилено).
- Можливість змінити пароль.
- Персональний профіль (редагування e-mail, ПІБ).
- Повідомлення про зміну статусу тикета (на екрані або на e-mail).

Для адміністратора:

- Перегляд усіх тикетів системи.
- Зміна статусу будь-якого тикета.
- Перегляд детальної інформації про кожен тикет (текст, дата створення, автор).
- Видалення або редагування тикетів.
- Перегляд списку користувачів, зміна їхніх ролей або блокування.
- Доступ до адміністративної панелі зі статистикою (графіки, лічильники).
- Журнал дій користувачів (історія входів та взаємодій).

3. Технічні характеристики

- Мова програмування: PHP 8+
- База даних: MySQL 5.7 або вище
- Інтерфейс: HTML5, CSS3, Bootstrap 5
- Клієнтські скрипти: JavaScript (базовий), Highcharts (статистика)

- Сервер: Apache/Nginx
- ОС: Linux/Windows (локальний сервер XAMPP або продакшн LAMP)
- Браузер: Сумісність із останніми версіями Chrome, Firefox, Edge

ДОДАТКИ

ДОДАТОК Б

Інструкція користувача до CRM-системи

Розроблена CRM-система має веб-інтерфейс і дозволяє працювати як звичайним користувачам, так і адміністраторам. Для початку роботи з системою користувач повинен відкрити вебсторінку системи у браузері. На головній сторінці буде запропоновано ввести логін і пароль, або, якщо обліковий запис ще не створено — пройти реєстрацію. Процес реєстрації передбачає введення особистих даних: імені, електронної адреси та пароля. Після підтвердження — користувача буде перенаправлено до сторінки входу, де можна авторизуватись у системі.

The image shows a web form titled "Створити обліковий запис" (Create account) on a teal background. At the top, there is a link "Увійти!" (Log in!). Below it are several input fields: "Ім'я" (Name), "Електронна пошта" (Email), "Пароль" (Password), "Повторіть пароль" (Repeat password), and "Номер телефону" (Phone number). The last field is a dropdown menu currently showing "Чоловік" (Male). At the bottom is a green button labeled "Створити акаунт" (Create account).

Рисунок Б.1 – Веб-форма реєстрації нового користувача

Після успішної авторизації користувач потрапляє до особистого кабінету. На головній сторінці доступна інформація про поточні заявки (тікети), які були створені ним раніше. Якщо жодної заявки ще не подано — система запропонує створити нову.

Створення тікета здійснюється через спеціальну форму, в якій необхідно вказати тему звернення та детальний опис. За бажанням можна прикріпити файл або скріншот, якщо функціональність системи це підтримує. Після заповнення всіх полів користувач натискає кнопку "Надіслати", після чого заявка зберігається в базі даних, і стає доступною адміністратору для опрацювання.

Рисунок Б.2 – Форма створення нового тикета

У розділі «Мої заявки» користувач може переглядати історію всіх своїх звернень, а також статуси, що відображають етап обробки (наприклад: новий, в роботі, завершено, відхилено). Для зручності кожному статусу присвоєно свій колір. Якщо натиснути на окрему заявку, відкривається розширена інформація — опис, дата подання, поточний статус, а також додаткові дані (якщо такі додано адміністратором).

У вкладці «Профіль» користувач має можливість змінити своє ім'я, адресу електронної пошти або пароль. Після внесення змін потрібно натиснути кнопку «Зберегти», щоб оновити інформацію у базі.

Рисунок Б.3 – Редагування особистої інформації користувача

У разі, якщо користувач виконує роль адміністратора, після входу в систему він автоматично потрапляє до адміністративної панелі. У цьому розділі відображаються всі заявки, надіслані користувачами. Адміністратор має доступ

до функцій зміни статусу тікета, перегляду його деталей, а також — можливості видалити або редагувати запис. Крім цього, адміністратор може переглядати список усіх зареєстрованих користувачів, змінювати їхні ролі, блокувати облікові записи або переглядати історію їхньої активності.

Особливістю системи є наявність модуля статистики. У верхній частині панелі адміністратора розміщено графіки та лічильники, які показують кількість заявок за статусами, активність користувачів протягом певного періоду, середній час обробки тікетів тощо. Цей функціонал реалізовано за допомогою JavaScript-бібліотеки Highcharts.

Після завершення сеансу роботи користувач має натиснути кнопку «Вийти», щоб завершити сесію та уникнути несанкціонованого доступу до своїх даних у разі використання спільного комп'ютера.

АНОТАЦІЯ

Гаврось Ілля Валентинович — Розробка CRM-системи мовою програмування PHP — Рукопис. Кваліфікаційна робота за спеціальністю 122 «Комп'ютерні науки». — Волинський національний університет імені Лесі Українки, Луцьк. — 2025 р.

У цій роботі розглядається процес створення вебзастосунку для управління взаємодією з клієнтами (CRM), побудованого на основі мови програмування PHP. Основна увага зосереджена на побудові функціональної системи, яка забезпечує облік звернень користувачів, розмежування доступу за ролями, збереження даних у базі MySQL та керування тікетами через зручний

інтерфейс. У роботі проаналізовано вимоги до CRM-систем, обґрунтовано вибір технологій, розроблено архітектуру застосунку та проведено тестування його основних компонентів. Результатом дослідження є повноцінна система, яку можна адаптувати під потреби малого бізнесу або навчального середовища.

Ключові слова: CRM, CRM-система, PHP, MySQL, тікети, авторизація, веброзробка, рольовий доступ, клієнтська підтримка, інтерфейс, база даних, тестування, безпека.