

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА
БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО

ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: **“ Розробка додатку з графічним інтерфейсом користувача
для системи CRM в Monday work platform”**

Виконав: ст. гр. ІТ-41

Спеціальності 126 – «Інформаційні системи та
технології»

(шифр і назва)

Браташук Володимир Володимирович

(Прізвище та ініціали)

Керівник: к.т.н., доцент Луб Павло Миронович

(Прізвище та ініціали)

Рецензент: _____

(Прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ

ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти
126 – «Інформаційні системи та технології»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу студентці

Братащук Володимир Володимирович

1. Тема роботи: «Розробка додатку з графічним інтерфейсом користувача для системи CRM в Monday work platform»

Керівник роботи: Луб Павло Миронович

Затверджені наказом університету 123/к-с від 25.02.2025.

2. Строк подання студентом роботи 16.06.2025 р.

3. Початкові дані до роботи: 1. Довідкова література. 2. Система CRM.

3. Фреймворки, мови програмування, інструменти, класи. 4. Методика тестування ПЗ.

4. Зміст розрахунково-пояснювальної записки:

1. Аналіз ІТ-сервісів для CRM.

2. Технології для створення додатку із графічним інтерфейсом користувача.

3. Проектування додатку з графічним інтерфейсом користувача для системи CRM.

4. Охорона праці та безпека в надзвичайних ситуаціях.

Висновки та пропозиції.

Бібліографічний список.

Додатки.

5. Перелік презентаційного матеріалу 1–2 – Тема, мета, завдання дипломної роботи; 3 – Загальна характеристика платформи monday.com; 4 – Особливості використання CRM-систем у сучасному бізнесі; 5 – Аналіз рішень для автоматизації роботи з бізнес-даними; 6 – Вибір технологічного стеку та архітектурного підходу; 7 – Архітектура програмного рішення та взаємодія компонентів; 8 – Реалізація модулів обліку та збереження даних; 9 – Розробка користувацького інтерфейсу для взаємодії з Crunchbase; 10 – Серверна частина: логіка інтеграції з Monday API та Crunchbase API; 11 – Оптимізація черг і обробка великих обсягів даних; 12 – Тестування та оптимізація додатку

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	Луб П.М., доцент кафедри інформаційних технологій		
4	Городецький І.М., доцент кафедри інженерної механіки		

Дата видачі завдання 25.02.2025.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи виконання кваліфікаційної роботи	Строки виконання роботи	Примітка
1.	Написання першого розділу та означення головних завдань роботи	25.02.2025 – 01.04.2025	
2.	Виконання другого розділу, опис інструментів та формування головних показників для розробки	01.04.2025 – 01.05.2025	
3.	Виконання третього розділу, реалізація мети роботи	01.04.2025 – 01.05.2025	
4.	Написання розділу: «Охорона праці та безпека в надзвичайних ситуаціях»	01.05.2025 – 01.06.2025	
5.	Завершення оформлення пояснювальної записки та презентаційного матеріалу	01.05.2025 – 01.06.2025	
6.	Завершення роботи в цілому	01.06.2025 – 09.06.2025	

Студент _____ Браташук В.В.
(підпис)

Керівник роботи _____ Луб П.М.
(підпис)

Кваліфікаційна робота: 47 с. текст. част., 16 рис., 5 табл., 12 слайдів, 23 джерела.

Розробка додатку з графічним інтерфейсом користувача для системи CRM в Monday work platform. Браташук В.В. Кафедра ІТ. – Дубляни, Львівський НУВМБ, 2025.

У роботі проведено аналіз програмних додатків, зосереджено увагу на CRM-системах, їхньому функціональному призначенні та ролі у веденні бізнесу. Розглянуто сучасні приклади реалізації таких рішень, проаналізовано технології веб-розробки, які найчастіше використовуються для створення CRM-додатків. Детально описано інтерфейс, функціональні блоки та можливості Monday work platform як бази для реалізації власного програмного продукту.

Сформовано технічне завдання, структуру веб-додатку та визначено основні модулі системи. Реалізовано інтерактивний інтерфейс, передбачено можливості редагування та збереження даних, а також протестовано додаток на предмет працездатності та відповідності поставленим вимогам. Результатом стала стабільно працююча система з базовим функціоналом CRM.

Робота демонструє комплексний підхід до створення програмного забезпечення з урахуванням сучасних вимог до функціональності та безпеки.

Проведено розрахунок параметрів освітлення для приміщення комп'ютерного класу, описано організацію безпечного робочого місця та заходи, що мають бути вжиті у разі виникнення надзвичайних ситуацій.

Ключові слова: CRM-система, веб-додаток, Monday platform, розробка інтерфейсу, програмне забезпечення.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1.	
АНАЛІЗ ІТ-СЕРВІСІВ ДЛЯ CRM.....	7
1.1 Поняття та класифікація додатків.....	7
1.2 Призначення та аналіз головних завдань CRM.....	9
1.3 Аналіз веб-сайтів із сервісами CRM.....	12
РОЗДІЛ 2. ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ ДОДАТКУ ІЗ ГРАФІЧНИМ ІНТЕРФЕЙСОМ КОРИСТУВАЧА.....	19
2.1 Загальний огляд Monday work platform.....	19
2.2 Головні інструменти для створення додатку.....	22
2.3 Технічне завдання розробки.....	26
РОЗДІЛ 3. ПРОЕКТУВАННЯ ДОДАТКУ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ КОРИСТУВАЧА ДЛЯ СИСТЕМИ CRM	28
3.1 Розробка структури та створення макету.....	28
3.2 Розробка функціональності	32
3.3 Тестування та оптимізація додатку.....	34
3.4 Інструкція відвідувача та приклад функціонування.....	36
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	40
4.1 Структурно-функціональний аналіз технологічного процесу...	40
4.2. Розрахунок освітлення приміщення комп'ютерного кабінету...	41
4.3. Безпека в надзвичайних ситуаціях.....	44
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46

ВСТУП

У сучасних умовах цифровізації бізнесу CRM-системи стають критично важливими інструментами для управління клієнтськими взаємовідносинами, автоматизації робочих процесів та підвищення продуктивності компаній. Одним із прикладів таких систем є платформа Monday.com, яка надає широкі можливості для кастомізації робочих просторів і процесів за допомогою інтегрованих додатків та рецептів автоматизації;

Хоча Monday.com вже забезпечує графічний інтерфейс користувача, у багатьох випадках виникає необхідність у створенні спеціалізованої серверної логіки для автоматичної обробки даних, інтеграції з зовнішніми сервісами та дотримання обмежень самої платформи. Одним із таких прикладів є завдання збагачення CRM-даних шляхом щоденної синхронізації з базою компаній Crunchbase;

Мета роботи - розробити кастомний додаток для Monday work platform, який забезпечує автоматичне збагачення CRM-даних компаній на основі інформації з Crunchbase, з урахуванням обмежень платформи, використовуючи можливості черги повідомлень та рецептів Monday Code.

Завдання роботи:

- проаналізувати існуючі CRM-рішення та сервіси, що надають доступ до даних компаній;
- дослідити можливості Monday work platform та інструменти для створення додатків з UI і бекенд-логікою;
- спроектувати структуру додатку та реалізувати основну бізнес-логіку;
- протестувати та оптимізувати додаток, забезпечити його стабільну роботу в умовах обмежень Monday Code.

РОЗДІЛ 1. АНАЛІЗ ІТ-СЕРВІСІВ ДЛЯ CRM

1.1 Поняття та класифікація додатків

У сучасному програмному забезпеченні додаток – це окрема програмна одиниця, яка виконує конкретну функцію або набір функцій у межах певної платформи або операційної системи. Додатки можуть бути автономними або інтегрованими в інші системи, забезпечуючи взаємодію користувача з інформаційною інфраструктурою підприємства.

У контексті хмарних CRM-систем, зокрема платформи Monday.com, додаток – це розширення (іноді званий "app" або "integration feature"), яке додає до платформи нову функціональність, дозволяє підключення до зовнішніх API або реалізацію бізнес-логіки за допомогою коду.

Залежно від функціонального призначення та способу реалізації, додатки можна класифікувати наступним чином:

- інтеграційні додатки – дозволяють з'єднувати CRM із зовнішніми сервісами, API або базами даних (наприклад, з Crunchbase, Google Sheets, Slack)
- автоматизаційні додатки – реалізують певну бізнес-логіку або автоматичну реакцію на події (наприклад, надсилання повідомлення після зміни статусу елемента)
- інтерфейсні додатки – додають візуальні компоненти або користувацький інтерфейс на основі даних з Monday.com або інших джерел
- системні додатки – розширюють можливості самої платформи, як-от облік ресурсів, логування чи безпечна авторизація

З технічної точки зору, додаток може бути реалізований за допомогою веб-технологій (JavaScript, HTML, CSS), серверного коду (Node.js у Monday Code), або через готові інструменти з SDK-платформами, що полегшують доступ до API Monday та реалізацію бізнес-логіки.

У контексті CRM-систем додатки виступають як інструменти розширення стандартного функціоналу платформи, надаючи можливість адаптувати її під специфічні бізнес-потреби. Додатки можуть бути внутрішніми (розробленими власною командою компанії) або сторонніми (створеними зовнішніми розробниками чи партнерами платформи).

Класифікація додатків за функціональним призначенням дає змогу структурувати підхід до їх розробки та інтеграції:

- Інтеграційні додатки дозволяють поєднувати CRM із зовнішніми джерелами даних. Наприклад, синхронізація з Crunchbase для отримання даних про компанії чи з Google Sheets для роботи з табличними структурами.
- Автоматизаційні додатки допомагають реалізувати правила, які запускаються при настанні певних умов (зміна статусу, оновлення поля тощо), автоматизуючи рутинні процеси.
- Інтерфейсні додатки створюють візуальні елементи взаємодії з користувачем -дашборди, форми, інтерактивні таблиці.
- Системні додатки призначені для внутрішніх потреб платформи, наприклад, контролю доступу, логування або реалізації кастомної авторизації.

Знання типів додатків допомагає ефективніше планувати архітектуру майбутньої CRM-системи та обирати відповідні технології для реалізації.

Таблиця 1.1 – Порівняльна таблиця типів додатків у CRM-платформах

Тип додатку	Призначення	Приклади використання
1	2	3
Інтеграційний	З'єднання з зовнішніми системами або базами даних	Підключення до Crunchbase, Google Sheets, Salesforce
Автоматизаційний	Реалізація реакцій на події або умовні сценарії	Авто-зміна статусу, надсилання листів, оновлення полів
Інтерфейсний	Надання користувачеві візуальних елементів для взаємодії	Дашборди, форми, таблиці з виводом додаткових даних
Системний	Розширення внутрішніх можливостей платформи	Авторизація, логування, контроль доступу

За типом конструкції склади поділяють на закриті, напівзакриті (навіси) і складські площадки, які відрізняються технологічними режимами зберігання

матеріальних запасів (температурою та відотною вологістю робочого приміщення, доступністю до товарів та ін.). Основні види складів за типом конструкції наведені в табл. 1.1.

Складське приміщення має певні робочі зони:

- для зберігання матеріальних запасів;
- для сортування запасів (формування партій товарів на відвантаження);
- для прийомки і відпуску запасів;
- для розміщення внутрішньо складської підйомно-транспортної техніки.

Технологічний процес на складі включає комплекс робіт (логістичних операцій), що пов'язані із підготовкою до прийомки і самої прийомки товарів, розміщенням їх для зберігання, організації зберігання, підготовки до відпуску і відпуску/доставки товарів споживачам.

1.2 Призначення та аналіз головних завдань CRM

Система управління взаємовідносинами з клієнтами (CRM – *Customer Relationship Management*) – це сукупність інструментів, стратегій і технологій, призначених для керування взаємодією компанії з наявними та потенційними клієнтами.

Головне призначення CRM-системи -централізація даних про клієнтів, автоматизація процесів продажу, маркетингу та обслуговування, а також підвищення ефективності взаємодії між співробітниками компанії.

CRM-системи (Customer Relationship Management) є ключовим інструментом для організацій, що прагнуть налагодити ефективну взаємодію з клієнтами. Вони дозволяють збирати, структурувати та аналізувати інформацію про клієнтів, що сприяє прийняттю обґрунтованих рішень у сфері маркетингу, продажів і обслуговування.

Серед основних завдань CRM-систем можна виділити (рис. 1.1):

- централізація клієнтських даних. Усі контакти, історія взаємодії, покупки та звернення зберігаються в єдиній базі даних;
- підвищення ефективності продажів. CRM допомагає менеджерам відстежувати етапи угод, автоматизувати нагадування та своєчасно реагувати на запити;
- автоматизація рутинних процесів. Це знижує навантаження на персонал та дозволяє зосередитись на ключових завданнях;
- аналітика та звітність. Система надає детальні звіти, що допомагають виявити слабкі місця у процесах та оптимізувати бізнес-стратегії;
- покращення обслуговування. Завдяки швидкому доступу до історії клієнта, компанія здатна оперативно вирішувати проблеми та покращувати якість сервісу.

Головні завдання CRM-системи

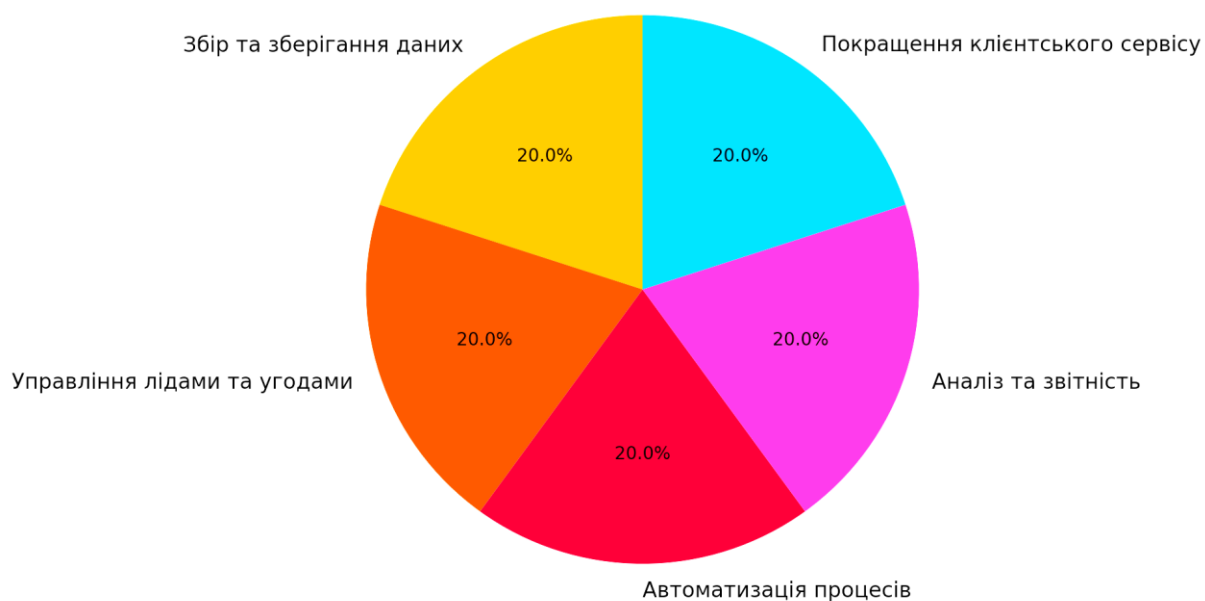


Рисунок 1.1 – Головні Завдання CRM-Системи

Таким чином, CRM-системи є не просто інструментом зберігання даних, а повноцінною платформою для стратегічного управління відносинами з клієнтами. Вони відіграють ключову роль у підвищенні конкурентоспроможності підприємства та забезпеченні довгострокового зростання.

У сучасних умовах цифрової трансформації CRM-системи дедалі частіше інтегруються з іншими корпоративними платформами - такими як ERP, маркетингові платформи, сервіси підтримки клієнтів, системи електронної комерції тощо. Це дозволяє створити єдиний інформаційний простір, де всі бізнес-процеси взаємопов'язані та працюють синхронно.

Особливу роль у розвитку CRM-технологій відіграє поява хмарних рішень, які забезпечують зручний доступ до системи з будь-якого пристрою та в будь-який час. Завдяки SaaS-моделі (Software as a Service) впровадження CRM стало можливим не лише для великих корпорацій, а й для малих та середніх підприємств. Платформи, подібні до Monday Work OS, HubSpot, Zoho CRM або Pipedrive, надають широкий спектр функцій без потреби у складній локальній інфраструктурі.

Крім того, сучасні CRM-системи активно використовують штучний інтелект та машинне навчання для прогнозування поведінки клієнтів, виявлення трендів і автоматизації рутинних дій. Інтелектуальні підказки, скоринг лідів, автоматичне сегментування аудиторії - усе це сприяє підвищенню точності рішень та персоналізації обслуговування.

У контексті розробки власного додатку для CRM-системи важливим стає орієнтир на зручний графічний інтерфейс користувача (GUI). Візуальна простота, інтуїтивна навігація, кастомізація під конкретні бізнес-процеси - усе це значною мірою впливає на ефективність використання системи та рівень прийняття її співробітниками компанії.

Таким чином, розвиток CRM-систем відбувається не лише в напрямку розширення функціоналу, а й у площині покращення користувацького досвіду. Це створює актуальну потребу в розробці спеціалізованих інтерфейсів, адаптованих під конкретну CRM-платформу та типову модель використання, що і є предметом даної дипломної роботи.

1.3 Аналіз веб-сайтів із сервісами CRM

На сучасному ринку існує велика кількість CRM-систем, які призначені для автоматизації процесів управління взаємовідносинами з клієнтами. Серед них вирізняються такі платформи як Monday CRM, HubSpot CRM, Zoho CRM, Salesforce, а також Pipedrive. Для зазначених платформ в даному розділі буде зніснено порівняльний аналіз наступних характеристик:

- функціональності;
- інтерфейсу;
- можливостей кастомізації;
- орієнтації на користувача.

У таблиці 1.2 представлено порівняльна таблиця функціональності CRM платформ.

Таблиця 1.2 – Порівняльна таблиця функціональності CRM платформ

№ п/п	Платформа	Основні модулі	Автоматизація без коду	Аналітика та звіти	AI / ML-можливості
1	Monday CRM	Продажі, маркетинг, підтримка, управління проектами	Правила «якщо-те», шаблони, інтеграції	Візуальні дашборди, кастомні KPI	Обмежені (рекомендації на дошках)
2	HubSpot CRM	Контакти, угоди, email-маркетинг	Робочі процеси у платних «Hubs»	Вбудована вирва продажів, маркетингові метрики	AI-інструменти для контенту (Marketing Hub)
3	Zoho CRM	Продажі, маркетинг, сервіс	Сценарії, макроси, правила	Гнучкі звіти, Canvas-дашборди	Zia AI-помічник (прогнози, чати)
4	Salesforce	Sales, Service, Marketing, Commerce	Flow Builder, Process Builder	Розширена аналітика, Tableau CRM	Einstein AI (прогнози, боти)
5	Pipedrive	Продажі	Простенькі тригери та шаблони email	Базові звіти, воронка	AI-прогноз імовірності угоди

Аналіз функціональних можливостей сучасних CRM-систем показує, що більшість із них пропонують широкий набір інструментів для управління лідами, автоматизації процесів продажів та аналітики. Проте саме Monday CRM вирізняється серед конкурентів завдяки гнучкості адаптації під потреби різних команд, тісній інтеграції з таск-менеджментом та зручності візуального налаштування бізнес-процесів.

Такі платформи як Salesforce і Zoho CRM мають значно більший функціональний потенціал, однак часто вимагають технічних навичок або залучення розробників, що може ускладнити впровадження в невеликих компаніях. Натомість HubSpot CRM та Pipedrive орієнтовані на простоту, але поступаються в можливостях кастомізації та розширення.

У контексті розробки додатку для платформи Monday Work OS, її функціональна архітектура надає достатню гнучкість та стабільний API, що дозволяє інтегрувати власні модулі або інтерфейси без шкоди для загальної логіки системи. Це робить її оптимальним середовищем для реалізації CRM-рішень із кастомним графічним інтерфейсом.

Окрім функціональності, одним з ключових чинників ефективності CRM-систем є зручність взаємодії кінцевого користувача з інтерфейсом платформи. Простий, інтуїтивно зрозумілий та візуально гнучкий інтерфейс суттєво підвищує продуктивність персоналу та швидкість впровадження рішень в бізнес-процеси.

З метою візуального порівняння інтерфейсів було проаналізовано п'ять популярних CRM-платформ: Monday CRM, HubSpot, Zoho CRM, Salesforce та Pipedrive.

На рис. 1.2 представлено порівняння візуальної гнучкості та простоти навчання обраних для аналізу платформ шляхом виставлення оцінок від 1 до 5.

1. Візуальна гнучкість - наскільки легко змінювати вигляд полів, дошки, дашбордів та налаштувань без коду.

2. Простота навчання - яка кількість часу та зусиль потрібна новому користувачу, щоби впевнено працювати в системі (вища оцінка = легше).

Monday CRM - Поєднує максимальну гнучкість (5) із відносно легкою кривою навчання (4). Завдяки drag-and-drop та яскравому UI нові користувачі швидко адаптуються.

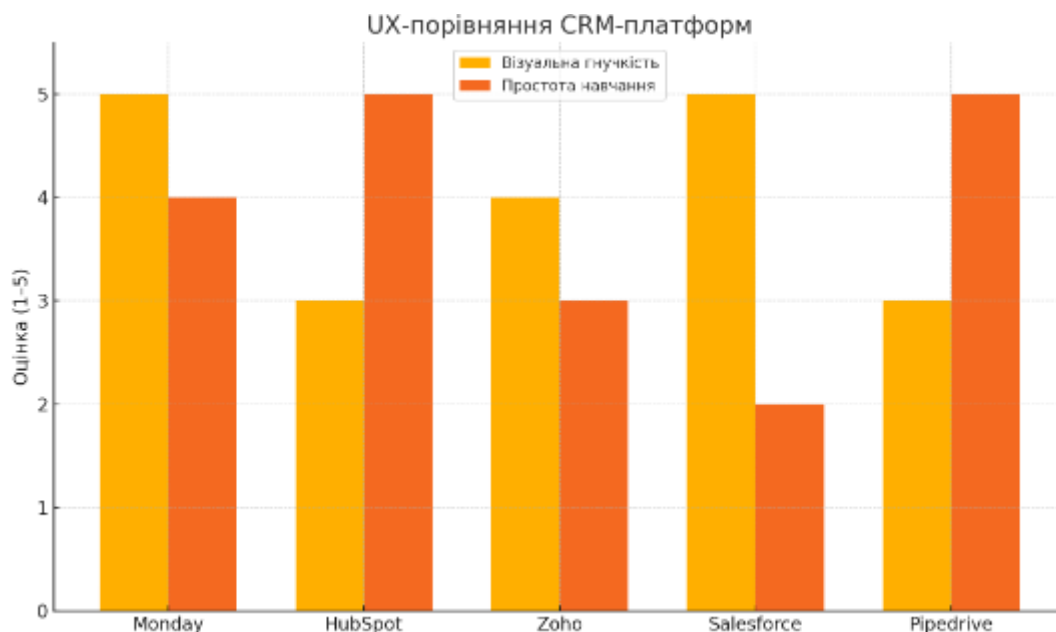


Рисунок 1.2 – «UX-порівняння CRM-платформ»

HubSpot CRM - Найдружніша до новачків (5), але менш «налаштовувана» за зовнішнім виглядом (3). Добре підходить для команд, які хочуть «увімкнути й працювати».

Zoho CRM - Баланс між гнучкістю (4) та середньою складністю освоєння (3). Інтерфейс потужний, проте більш «щільний» за елементами.

Salesforce - Абсолютний лідер за можливістю кастомізації (5), але найвищий поріг входу (2) через велику кількість меню та технічні нюанси (Apex, LWC).

Pipedrive - Дуже проста у навчанні (5) завдяки фокусу на воронці продажів, проте пропонує базову візуальну адаптацію (3).

Результати порівняння вказують на доцільність вибору Monday CRM як базової платформи для подальшої розробки GUI-додатку.

Завдяки поєднанню високої адаптивності інтерфейсу та дружності до користувача, розроблюваний інтерфейс повинен:

- підтримувати швидке освоєння користувачем без технічної підготовки;
- базуватись на знайомих елементах взаємодії (таблиці, статуси, тощо);

- бути гнучким до зміни логіки, етапів, полів, та візуального представлення даних;

- дотримуватись стилістики, прийнятої у Monday Work OS (кольорові статуси, drag-and-drop, інтерактивні віджети).

Таким чином, результати аналізу інтерфейсів CRM-систем закладають фундамент для проектування власного графічного інтерфейсу в межах розробки CRM-додатку під Monday Work OS, що є предметом цієї дипломної роботи.

Для повноцінного впровадження CRM-системи в існуючу бізнес-інфраструктуру важливо, щоб платформа не лише підтримувала базові функції управління клієнтами, але й дозволяла гнучке налаштування (кастомізацію) та інтеграцію з іншими сервісами -наприклад, електронною поштою, системами обліку, календарями, платформами аналітики тощо.

У таблиці нижче представлено порівняння популярних CRM-рішень за наступними критеріями:

Кастомізація - можливість змінювати логіку роботи системи, структуру елементів, поля, статуси та бізнес-процеси без програмування або з мінімальним залученням розробника.

Інтеграції - кількість доступних сторонніх сервісів для підключення, підтримка API, доступ до внутрішніх автоматизацій та маркетплейсу додатків.

Відкритість API - наявність документації, REST API, підтримка вебхуків та SDK для розробників.

У таблиці 1.3 представлена порівняльна таблиця для аналізу можливостей кастомізації та інтеграції обраних для дослідження CRM платформ. Цей аналіз дозволяє обґрунтовано оцінити потенціал кожної системи для адаптації під потреби бізнесу або створення власних рішень на її основі - зокрема, розробки графічного інтерфейсу, що є предметом цієї дипломної роботи.

Порівняльний аналіз CRM-систем за рівнем кастомізації та інтеграційних можливостей показує, що Monday CRM є одним із найбільш гнучких та відкритих до зовнішніх інтеграцій рішень. Платформа забезпечує широкий спектр налаштувань без потреби в програмуванні, включаючи модифікацію полів,

статусів, форм подання інформації та автоматизацію робочих процесів за допомогою графічного конструктора.

Таблиця 1.3 – Порівняльний аналіз можливостей кастомізації та інтеграції

Платформа	No-code налаштування	Кастомізація з кодом	Маркетплейс / API
Monday CRM	Поля, стани, автоматизації, табло	Немає потреби в коді (складні сценарії через API)	5000+ інтеграцій, відкритий REST API
HubSpot CRM	Поля, pipeline, email-шаблони	JavaScript у «CMS Hub»	1 000+ додатків в App Marketplace
Zoho CRM	Layout-редактор, Canvas, правила	Deluge Script, Functions	500+ інтеграцій, OpenAPI
Salesforce	Flow, Page Layouts, Lightning	Apex, LWC, Heroku	3 500+ компонентів AppExchange
Pipedrive	Поля, етапи, email-шаблони	Webhooks, простий API	~400 інтеграцій у Marketplace

У той час як Salesforce пропонує ще вищий рівень кастомізації, він передбачає використання власної мови програмування (Apex) та значні технічні ресурси, що ускладнює її впровадження для невеликих або нетехнічних команд. HubSpot і Pipedrive мають обмежену кастомізацію, але натомість вирізняються простотою використання. Zoho CRM знаходиться посередині, пропонуючи помірний рівень налаштувань та доступ до великої кількості інтеграцій.

Таким чином, Monday CRM доцільно розглядати як оптимальну платформу для створення власного графічного інтерфейсу CRM-додатку. Вона забезпечує баланс між технічною доступністю, розширюваністю та простотою реалізації інтегрованих рішень через API та внутрішній маркетплейс додатків. Це повністю відповідає цілям дипломної роботи, орієнтованої на розробку кастомного графічного інтерфейсу в рамках Monday Work OS.

Під час вибору CRM-рішення важливо враховувати не лише технічні характеристики платформи, але й ринкову орієнтацію продукту та вартість його використання. Різні системи позиціонуються для компаній різного масштабу - від фрілансерів і малих бізнесів до корпорацій з тисячами користувачів.

Також важливо оцінити гнучкість тарифної моделі, наявність безкоштовних

планів, вартість повного доступу до функцій, а також надання додаткових можливостей у вищих пакетах (автоматизація, API, підтримка, аналітика).

У таблиці 1.4 наведено порівняльний аналіз п'яти популярних CRM-систем - Monday CRM, HubSpot, Zoho CRM, Salesforce та Pipedrive - з урахуванням:

- орієнтованості платформи на тип бізнесу (B2B, малі команди, великі корпорації тощо);
- наявності безкоштовного тарифного плану;
- стартової вартості платного доступу;
- особливостей ліцензування на користувача.

Ці фактори впливають на вибір технологічної платформи для створення власного рішення, а також визначають потенційні обмеження для цільових користувачів майбутнього додатку.

Таблиця 1.4 – Порівняльний аналіз цільової аудиторії та ціноутворення обраних для дослідження сервісів

Платформа	Цільові користувачі	Безкоштовний тариф	Базова ціна, \$/кор./міс	Масштабованість
Monday CRM	Стартапи, креативні та IT-команди (5-500 осіб)	Тест 14 днів	10–24	До ~1 000 користувачів (Enterprise)
HubSpot CRM	Малий + середній бізнес, маркетологи	Так	0 (Free), → 18+ (Starter)	Без обмежень користувачів
Zoho CRM	SMB → великі компанії	Ні	14–45	Висока, модульні адд-они
Salesforce	Середній + корпоративний сегмент	Ні	25–300+	Практично необмежена
Pipedrive	Команди продажів 3-100 осіб	Ні	14–99	Обмежена (менше 500 користувачів)

Аналіз моделей ціноутворення та цільової аудиторії показує, що різні CRM-системи орієнтовані на різні сегменти бізнесу - як за розміром команди, так і за бюджетними можливостями. Наприклад, HubSpot CRM приваблює користувачів безкоштовним базовим функціоналом, проте значно обмежує доступ до

розширених функцій без підписки на дорогі тарифи. Salesforce, навпаки, орієнтований переважно на великі корпоративні структури з відповідними ресурсами, а його тарифна модель малопридатна для малого та середнього бізнесу.

Monday CRM у цьому контексті вирізняється гнучким позиціонуванням: пропонує як доступні тарифи для малих команд, так і потужні можливості масштабування для середнього та великого бізнесу. Це робить платформу зручною для використання на різних етапах розвитку компанії. Крім того, ціна за користувача в Monday CRM є конкурентною відносно обсягу функцій, що надаються вже у стартових планах.

У контексті дипломної роботи, що передбачає розробку власного додатку з графічним інтерфейсом на базі Monday Work OS, така тарифна модель і цільова гнучкість дозволяють орієнтуватися на ширший сегмент потенційних користувачів: від стартапів до стабільних середніх компаній. Це підвищує практичну цінність створеного рішення.

РОЗДІЛ 2.

ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ ДОДАТКУ ІЗ ГРАФІЧНИМ ІНТЕРФЕЙСОМ КОРИСТУВАЧА

2.1 Загальний огляд Monday work platform

Monday Work Platform – це хмарна платформа для управління робочими процесами, що надає користувачам можливість створювати адаптовані рішення для планування, комунікації та взаємодії в командах. В основі платформи лежить концепція Work OS – операційної системи для роботи, яка дозволяє будувати індивідуальні цифрові робочі простори.

Платформа monday стала популярною завдяки своїй візуальній простоті, гнучкості в налаштуваннях та розвинутим інтеграціям із зовнішніми сервісами, зокрема Slack, Google Workspace, Microsoft Teams, Zoom, HubSpot, Salesforce, Zapier та багатьма іншими. Вона активно використовується компаніями для управління проєктами, завданнями, маркетинговими кампаніями, клієнтськими взаємодіями (CRM) та навіть внутрішніми HR-процесами.



Серед ключових характеристик платформи можна виділити:

- базова структура у вигляді дощок (boards), які можуть бути налаштовані під будь-який робочий процес (від простих списків до повноцінних CRM-систем);
- гнучка система статусів, стовпців і форм вводу, яку можна адаптувати під специфіку бізнесу;
- автоматизація дій за допомогою вбудованого конструктора сценаріїв (автоматичне призначення задач, надсилання повідомлень, зміна статусів тощо);
- підтримка додатків і розширень (Apps), що дозволяє створювати власні віджети, інтеграції та автоматизації;

– доступ до відкритого API, що дає змогу розробникам створювати кастомні рішення, підключати зовнішні сервіси або будувати повноцінні інтерфейси взаємодії.

Особливістю Monday Work Platform є її орієнтація на no-code та low-code підхід, що дозволяє бізнес-користувачам самостійно модифікувати робочі середовища без глибоких технічних знань (рис. 1.1.). Водночас, для складніших задач розробники можуть використовувати Monday Apps Framework на базі React.js, GraphQL або будь які інші веб технології [16].

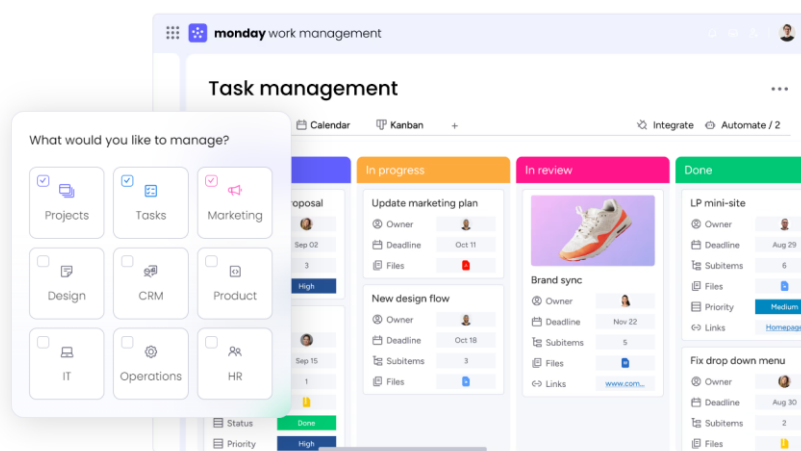


Рисунок 1.1 – Task management в Monday Work Platform

У контексті дипломної роботи саме Monday Work Platform обрана як база для розробки CRM-додатку, завдяки її технічній відкритості, підтримці кастомних елементів інтерфейсу та широким можливостям для інтеграції з сторонніми ресурсами.

Платформа Monday Work OS позиціонується не лише як універсальний інструмент для управління завданнями, а як гнучка операційна система для будь-яких бізнес-процесів. В межах платформи доступна низка спеціалізованих продуктів (Work OS Products), які базуються на єдиному функціональному ядрі, але орієнтовані на різні сфери діяльності. Основні з них – Monday Work Management, Monday CRM, Monday Dev і Monday Service.

Monday Work Management – це базовий і найпоширеніший продукт у системі Monday, який дозволяє структурувати, контролювати та автоматизувати

будь-які командні завдання. Його головна мета – надати користувачам єдиний візуальний простір для планування, координації та відстеження проєктів у режимі реального часу.

Основні можливості:

- побудова кастомних дошок (boards) для відображення процесів: від особистих to-do списків до багаторівневих проєктних планів;
- інструменти колаборації: коментарі, теги, обговорення під завданнями;
- гнучка система подань: канбан, календар, часові шкали (Gantt), таблиці, workload (навантаження на співробітників);
- вбудовані шаблони для швидкого старту в різних галузях (маркетинг, фінанси, продажі, HR тощо);
- потужна автоматизація процесів (через правила if/then) та інтеграція з поширеними сервісами (Slack, Outlook, Gmail, Google Drive тощо).

Цей модуль є надзвичайно гнучким, що робить його ідеальним фундаментом для побудови кастомних рішень, зокрема CRM-додатків.

Monday CRM – спеціалізований продукт, створений на базі Work Management, але вже орієнтований на управління взаємовідносинами з клієнтами. Це рішення дає змогу бізнесу централізовано управляти потенційними клієнтами (лідами), етапами продажу, контактами, угодами, комунікаціями та післяпродажним обслуговуванням.

Ключові функції:

- побудова воронки продажів з візуальним поданням;
- управління лідами, клієнтськими базами, нотатками та історією контактів;
- автоматизація follow-up листів, нагадувань, оновлення статусів лідів;
- інтеграція з поштовими сервісами та календарями;
- звіти про продуктивність команди продажів, тривалість угод, коефіцієнти конверсії.

Monday CRM дозволяє налаштовувати поля, структуру та логіку роботи під потреби конкретної компанії без залучення розробників. Усе це робить продукт

придатним як для малого бізнесу, так і для складніших сценаріїв B2B-продажів [8].

У межах дипломної роботи саме цей продукт буде слугувати прикладом для розробки графічного інтерфейсу CRM-додатку, з урахуванням реальної структури та логіки обробки клієнтських даних.

Monday Dev – це продукт, орієнтований на команди розробників програмного забезпечення. Він забезпечує процеси agile-менеджменту: спринти, беклоги, планування релізів, відстеження багів. Завдяки підтримці GitHub/GitLab інтеграцій, а також інструментів автоматизації, платформа дає змогу поєднувати менеджмент і розробку в єдиному просторі.

Monday Service (або Monday IT/Helpdesk) – це рішення для організації служби підтримки користувачів та управління запитами (тикетами). Система дозволяє фіксувати звернення клієнтів, автоматизувати призначення відповідальних, слідкувати за SLA та аналізувати якість обслуговування.

Усі продукти Monday Work OS побудовані на єдиному функціональному ядрі, що дозволяє легко переходити від загального управління задачами до спеціалізованих рішень у сфері продажів, обслуговування чи розробки. У фокусі даної дипломної роботи знаходяться Monday Work Management і Monday CRM, оскільки саме вони надають необхідний інструментарій для реалізації кастомного додатку з графічним інтерфейсом для керування клієнтськими даними, статусами угод та аналітикою. Їх гнучка структура, API-доступність та підтримка кастомних інтеграцій створюють відповідне середовище для реалізації проєкту.

2.2 Головні інструменти для створення додатку

Для розробки CRM-додатку в екосистемі Monday Work Platform було використано низку сучасних технологій, які забезпечили швидку розробку, зручне масштабування, стабільну інтеграцію з платформою Monday та гнучке адміністрування даних. Обраний стек технологій поєднує інструменти для

фронтенду, бекенду, баз даних, інфраструктури та тестування, що дозволило створити повноцінну інтеграцію стороннього сервісу з Monday CRM.

Для ефективної реалізації CRM-додатку важливо забезпечити стабільне та зручне середовище розробки, що підтримує кросплатформені інструменти та командну взаємодію. У цьому проєкті було використано операційне середовище з підтримкою Linux-функціоналу всередині Windows, а також систему контролю версій, що дозволила налагодити зберігати історію змін та забезпечити безперервність розробки.

WSL2 (Windows Subsystem for Linux 2) – забезпечив повноцінне середовище Linux всередині Windows-системи, що дозволило використовувати всі переваги Linux-командного рядка та спрощену роботу з пакетами, при цьому не змінюючи основну ОС розробника [4].



Git – основний інструмент для контролю версій. Використовувався для відстеження змін, управління гілками та синхронізації командної роботи. Завдяки інтеграції з GitHub було організовано зручний pull request-цикл для перевірки змін [5].

Під час розробки CRM-додатку виникла потреба створити невеликий графічний інтерфейс, який би дозволяв користувачу взаємодіяти з додатком та передавати свій особистий Crunchbase API key. Для цього було використано сучасні frontend-технології, які забезпечили швидку розробку, зручну стилізацію, реактивність інтерфейсу та ефективне керування станом. Простота реалізації і водночас гнучкість обраного підходу дозволили легко інтегрувати UI у загальну архітектуру додатку, не перевантажуючи його зайвим функціоналом.



React

React – обраний як основна бібліотека для побудови компонентного, модульного інтерфейсу. Його екосистема

дозволила швидко створити інтерактивні UI-компоненти, зручні у масштабуванні та повторному використанні[4, 23].

Vite – сучасний збирач фронтенду, який забезпечив миттєву перезбірку, гаряче оновлення модулів (HMR) та дуже швидкий старт проєкту.

Sass – препроцесор для CSS, що забезпечив більшу гнучкість стилізації, підтримку змінних, вкладеності та міксинів, що полегшило масштабування стилів у великому UI [4, 13].

MobX – бібліотека для керування станом у фронтенд-застосунку. Завдяки реактивній природі MobX оновлення інтерфейсу були автоматизовані при зміні стану, що покращило продуктивність та спростило код [15].



Щоб обробляти запити користувача, взаємодіяти з базою даних і зовнішніми сервісами, зокрема Crunchbase API, було реалізовано серверну частину додатку. Основний акцент зроблено на продуктивності, масштабованості та підтримці суворої типізації, що дозволило знизити кількість помилок і спростити підтримку коду. Архітектура бекенду побудована на сучасному стеку JavaScript/TypeScript із використанням легкого HTTP-фреймворку та інструментів для роботи з SQL, що забезпечило швидке створення надійного API для взаємодії між усіма складовими системи [20].

Axios – використано для реалізації HTTP-запитів між frontend та backend, а також для взаємодії з Crunchbase API.

Node.js у поєднанні з TypeScript – дозволили створити надійну backend-архітектуру з чіткою типізацією, що зменшило кількість помилок і підвищило читабельність коду [5, 6, 13].



Fastify – обрано як легкий, швидкий і продуктивний фреймворк для побудови RESTful API. Завдяки своїй плагінній архітектурі забезпечив швидке підключення логів, валідації, CORS та інших сервісів.

Knex.js – SQL query builder, який спростив побудову запитів до бази даних, забезпечив міграції, сидування та підвищив безпечність при роботі з різними СУБД.

Для зручності та ефективності розробки були використані сучасні

інструменти, що забезпечили швидке написання, тестування та налагодження коду. Visual Studio Code – основне середовище розробки з підтримкою численних розширень для форматування, перевірки типів, автодоповнення та інтеграції з Git.

Postman – незамінний інструмент для ручного тестування API-запитів, перевірки відповіді від Crunchbase API, дебагу авторизації та побудови колекцій запитів для різних середовищ.

Додаток було розгорнуто безпосередньо в екосистемі monday.com із використанням її внутрішніх інструментів, що забезпечили стабільну роботу, інтеграцію з платформою та фонову обробку даних.

Monday Code – дозволив безпосередньо хостити серверну частину додатку всередині Monday Work Platform. Це забезпечило кращу інтеграцію з внутрішнім API Monday, підвищену безпеку та зниження залежності від зовнішнього хостингу.

Monday Queue – застосовано для реалізації оновлення дощечки порціями щоб обійти ранінг-тайм обмеження мандей коду.

Monday Storage – застосовано для зберігання необхідних для збагачення дощечки даних про саму дощечку такий як: список стовпців дощечки; відповідь між полями з Crunchbase та стовпцями тощо.

Integration for sentence builder – дозволило безпосередньо інтегрувати функціонал додатку в середовище Monday.

Для зберігання даних про обробку айтемів та інтеграцію з Crunchbase було використано сучасні хмарні та локальні рішення, що забезпечили надійність, масштабованість і зручність розгортання як на етапі розробки, так і в продакшн-середовищі.

PostgreSQL – реляційна СУБД, яка виступає ядром для збереження інформації про айтеми, статуси обробки, інтеграції та логів взаємодії. Її використання обумовлене стабільністю, гнучкою структурою та підтримкою складних транзакцій.

Docker – використано для локального розгортання бази даних[19, 23].

AWS RDS – на етапі тестування (staging) база розміщувалась у хмарному

середовищі Amazon, що забезпечило доступність і масштабованість під час демонстрацій.

Neon – хмарне рішення, обране для production-версії бази даних. Його ключова перевага – гнучке масштабування, безперервне збереження змін і легка інтеграція з Vercel, Railway та іншими хостингами.

Застосування сучасного технологічного стеку дало змогу реалізувати CRM-додаток, що: 1) має зручний інтерфейс, орієнтований на кінцевого користувача; 2) інтегрується напрямку з Monday Work OS, використовуючи її власні інструменти та API; 3) підтримує масштабовану архітектуру, яка дозволяє розширювати функціонал; 4) забезпечує стійкість і надійність при роботі з великими обсягами даних; 5) дозволяє легко обслуговувати, оновлювати та розгортати нові версії у production-середовищі.

2.3 Технічне завдання розробки

Для досягнення поставленої мети – створення інтегрованого CRM-додатку для платформи Monday.com з автоматичним отриманням і синхронізацією інформації з Crunchbase – було сформульовано технічне завдання, що охоплює функціональні, нефункціональні вимоги, технологічні обмеження та типові сценарії використання.

1. Загальна мета: Основна мета розробки полягає у створенні додатку, який дозволяє автоматизувати процес оновлення даних про компанії на дошці Monday.com шляхом інтеграції з Crunchbase API. Такий інструмент покликаний спростити аналітику для користувачів, які відстежують зміни у фінансовому, організаційному чи ринковому статусі обраних компаній.

2. Функціональні вимоги: Додаток повинен реалізовувати низку ключових функцій:

- авторизація користувача через введення Crunchbase API key;
- валідація ключа при початковому налаштуванні;

- динамічний мапінг полів із Crunchbase до відповідних колонок;
- реакція на події: створення айтема, зміна назви або вебсайту компанії;
- автоматична щоденна синхронізація всіх записів із Crunchbase;
- обробка помилок (наприклад, некоректний ключ або компанія не знайдена) з відображенням повідомлень для користувача;
- обхід ліміту запитів до Crunchbase API шляхом реалізації черги з поступовою обробкою записів.

3. Нефункціональні вимоги: Для стабільної роботи система має відповідати наступним критеріям:

- висока надійність і відмовостійкість;
- можливість масштабування до 4000 айтемів на акаунт;
- час обробки черги не перевищує 300 секунд, що відповідає обмеженням Monday Code.

4. Технологічні обмеження: Реалізація додатку обмежується такими умовами:

- використання Monday Code для хостингу логіки функцій;
- застосування Monday Queue для контрольованої обробки подій та уникнення перевищення лімітів API;
- взаємодія з Crunchbase API через бібліотеку Axios (на платформі Node.js);
- використання бази даних PostgreSQL з різними середовищами для розробки, тестування та продакшену (Docker, AWS RDS, Neon);
- UI-застосунок створено на базі React, MobX і Vite – виключно для передачі Crunchbase API key.

5. Сценарії використання: Додаток передбачає наступні типові ситуації взаємодії:

- користувач додає або змінює інформацію про компанію в Monday, після чого додаток автоматично завантажує актуальні дані з Crunchbase;
- щоденна фонові синхронізація всіх компаній із зовнішнім сервісом;
- у разі помилки дані про неї записуються в спеціальну колонку на дошці та користувач отримує повідомлення.

РОЗДІЛ 3.

ПРОЕКТУВАННЯ ДОДАТКУ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ КОРИСТУВАЧА ДЛЯ СИСТЕМИ CRM

3.1 Розробка структури та створення макету

На етапі проектування структури додатку було приділено особливу увагу архітектурі взаємодії між його складовими компонентами, побудові внутрішніх процесів синхронізації даних, а також загальній логіці обробки подій у рамках платформи Monday.com. Зважаючи на обмеження середовища виконання Monday Code та потребу в масштабуванні, додаток був реалізований як набір взаємопов'язаних фаз обробки з використанням черги повідомлень (Monday Queue) та фазового підходу до оновлення даних.

Для кращого розуміння структури та логіки роботи системи було створено низку UML-діаграм послідовностей (sequence diagrams). Вони відображають ключові сценарії взаємодії компонентів додатку:

- Recipe Install + Snapshot Phase – початкове встановлення рецепту та перше зчитування даних з дошки (snapshot);
- Sync Phase – основна фаза збагачення даних з використанням Crunchbase API;
- Daily Sync – щоденне оновлення даних айтемів на борді;
- Finalize Phase – завершальну фаза синку, що перевіряє чи всі необхідні рядки були оновлені і якщо все гаразд робить відповідні помітки про завершення циклу оновлення.

Нижче кожна з діаграм буде детально розглянута з поясненням її функціонального змісту та ролі у загальній логіці системи.

На рис. 3.1 продемонстровано діаграму що демонструє логіку що планувалася для початкового встановлення рецепту та першого зчитування даних з дошки (snapshot)

Дана діаграма послідовностей демонструє повний цикл дій, який виконується при першому встановленні інтеграційного рецепту користувачем у Monday work platform. Цей сценарій охоплює логіку з моменту взаємодії користувача з інтерфейсом до запуску першої фази синхронізації (Snapshot Phase), яка фіксує поточний стан дошки для подальшої обробки.

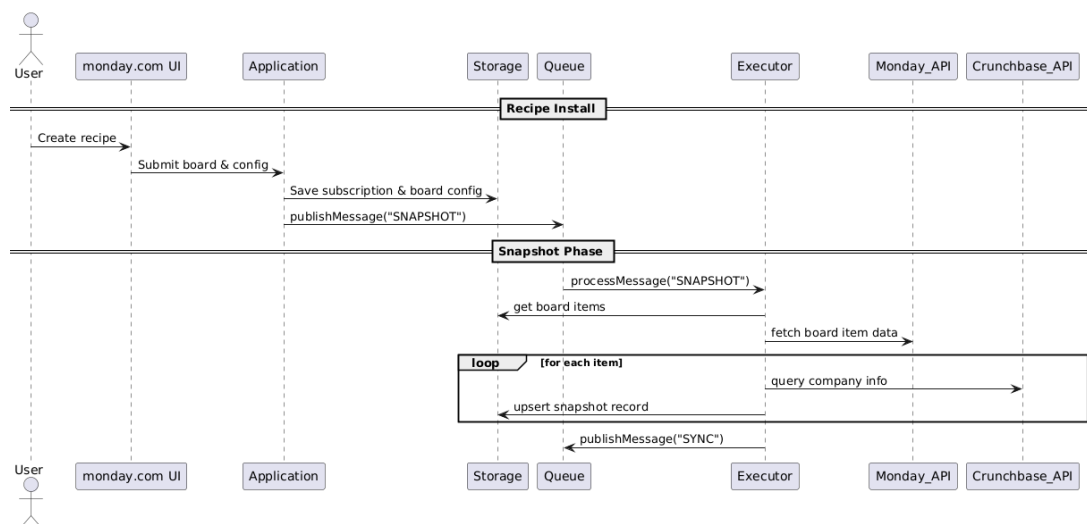


Рисунок 3.1 – Recipe Install + Snapshot Phase Sequence Diagram

На рис. 3.2 продемонстровано діаграму що демонструє щоденне оновлення даних айтемів на борді. Ця діаграма демонструє щоденний процес ініціалізації синхронізації, який відбувається автоматично раз на 24 години для кожного акаунта, у якого активовано інтеграційний рецепт. Основна мета цього сценарію – забезпечити регулярне збагачення CRM-даних, навіть якщо користувач не вносив змін вручну.

Процес включає такі ключові етапи:

- Запуск планувальника. Синхронізація запускається зовнішнім тригером (наприклад, AWS Lambda або іншим scheduler-сервісом), який надсилає POST-запит на спеціальний endpoint додатку;
- Визначення цільових підписок. Додаток отримує всі активні підписки (Subscriptions), які ще не були синхронізовані за останні 24 години, і для кожної з них формує чергове повідомлення типу SNAPSHOT;

- Обробка SNAPSHOT-повідомлення. QueueProcessor приймає повідомлення з черги, визначає boardId, accountId, отримує список актуальних айтемів з monday.com, та порівнює їх зі збереженим snapshot;
- Оновлення структури Snapshot. Виявляються зміни (нові, змінені, або видалені айтеми), і база оновлюється відповідно. Також записується інформація про крок виконання та статус обробки (SyncState);
- Публікація SYNC-повідомлень. Якщо в результаті snapshot знайдено айтеми, що підлягають оновленню, то вони розбивається на «порції» і для кожної з них додається окреме повідомлення типу SYNC;
- Контроль обсягу та часу. Під час обробки враховується обмеження у 4000 айтемів на акаунт та загальний час виконання (до 300 секунд). У разі перевищення – частина айтемів буде оброблена в наступному циклі;
- Оновлення Subscription. Після завершення або переривання циклу обробки оновлюється поле lastCycleCompletedAt, яке слугує міткою останньої синхронізації.

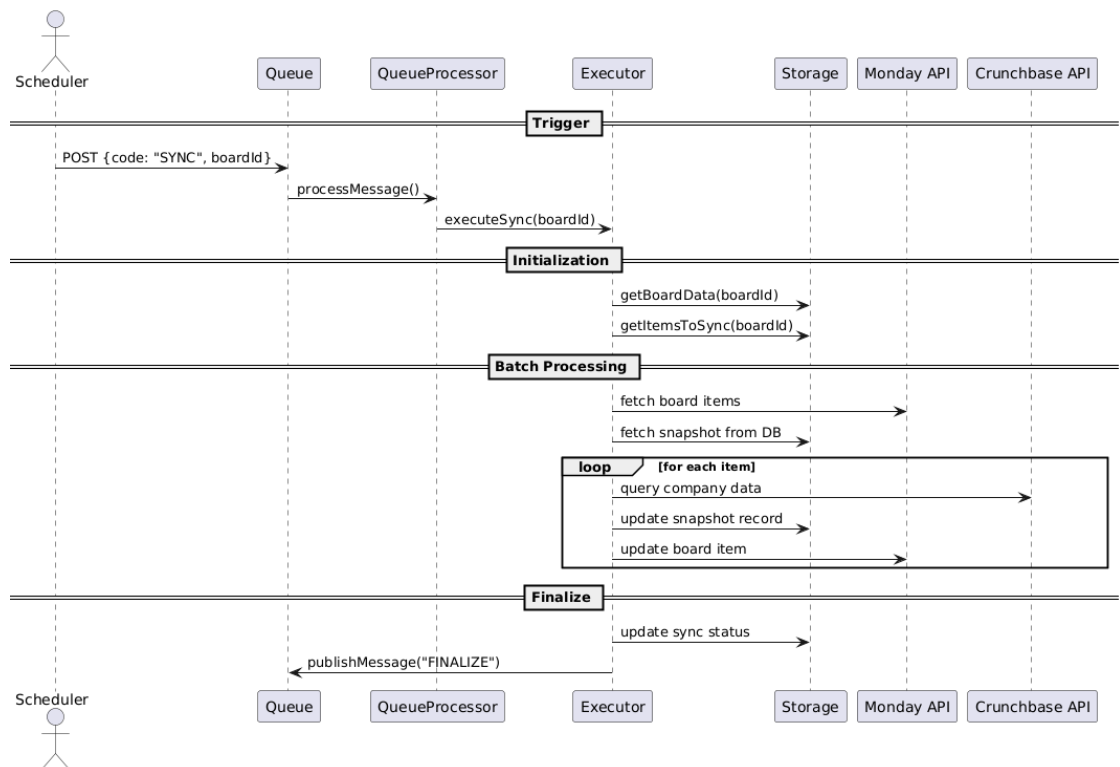


Рисунок 3.2 – Daily Sync Sequence Diagram

Ця щоденна процедура гарантує, що навіть без змін із боку користувача дані на борді залишаються актуальними, а інтеграція із Crunchbase працює стабільно у фоновому режимі.

На рис. 3.3 продемонстровано діаграму що відображає основну фазу збагачення даних з використанням Crunchbase API. Дана діаграма відображає логіку обробки одного з ключових сценаріїв — фази синхронізації (SYNC Phase). Вона є частиною загального циклу оновлення даних компаній і відповідає за збагачення item-ів на дошці користувача актуальною інформацією з Crunchbase. Запуск цієї фази здійснюється через чергу (monday queue), яка надсилає повідомлення типу SYNC.

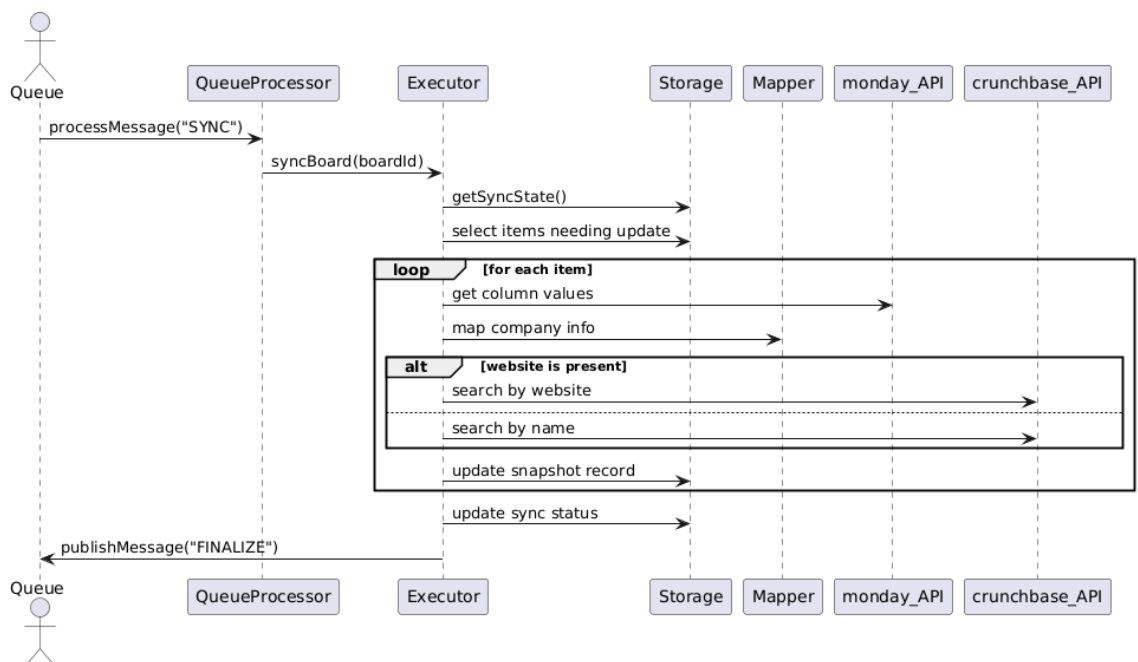


Рисунок 3.3 – Sync Phase Sequence Diagram

На рис. 3.4 зображено діаграму що демонструє завершальну фазу синку, що перевіряє чи всі необхідні рядки були оновлені і якщо все гаразд робить відповідні помітки про завершення циклу оновлення. Дана діаграма демонструє логіку виконання завершальної фази синхронізації, яка запускається після обробки всіх запланованих повідомлень SYNC для певної дощечки. Основною метою цієї фази є фіксація завершення циклу, перевірка наявності залишкових айтемів та, у разі потреби, ініціація нового етапу обробки.

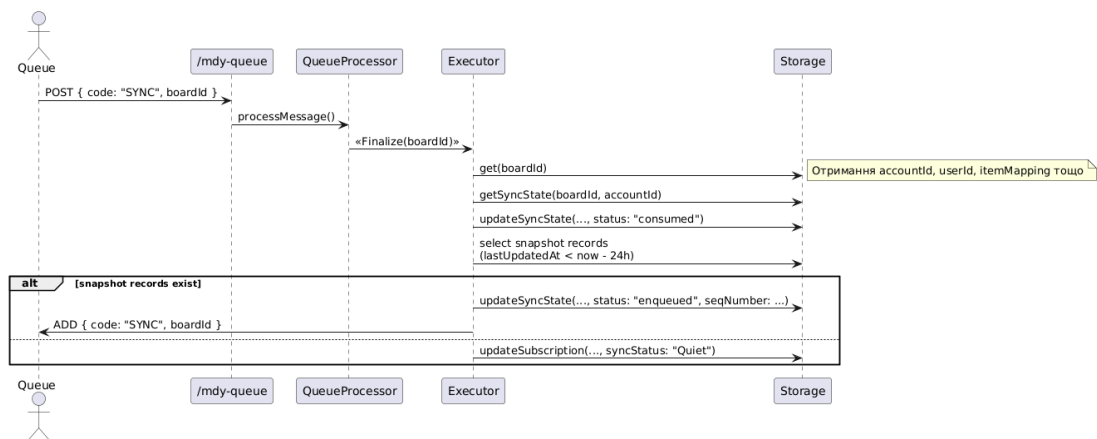
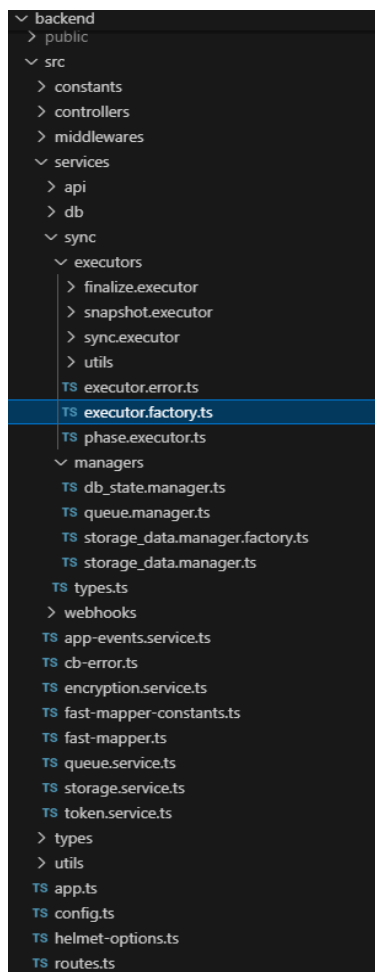


Рисунок 3.4 – Finalize Phase Sequence Diagram

3.2 Розробка функціональності

Розробка функціональності додатку базується на модульній архітектурі з чітким розподілом відповідальності між компонентами.



Основна логіка реалізована на мові програмування TypeScript з використанням фреймворку Fastify для серверної частини та бібліотеки Knex для роботи з базою даних PostgreSQL. Основна комунікація з monday.com та Crunchbase API здійснюється через HTTP-запити з використанням Axios.

Рисунок 3.5 – Структура файлів проєкту

Кожен із ключових процесів — встановлення рецепту, створення snapshot, синхронізація даних, обробка черги — винесений у окремий клас або сервіс, що дозволяє легко масштабувати і супроводжувати проєкт. У цьому підрозділі буде описано основні частини коду, з яких складається система, а також надано

фрагменти найбільш показового функціоналу.

На рис. 3.5 представлено знімок екрану що демонструє структуру файлів проєкту. У корені присутній конфігураційний файл `app.ts`, який ініціалізує сервер `Fastify`, реєструє всі плагіни, маршрути та `middlewares`. Ця структура дозволяє розмежувати логіку на рівні API, синхронізації, сховища та інтерфейсу зовнішніх систем. Такий підхід забезпечує не лише масштабованість, але й зручність при тестуванні та супроводі додатку.

Папка `public/` містить статичні файли, які видаються бекендом (`Fastify`) під час запиту користувача. Це дозволяє уникнути потреби в окремому фронтенд-хостингу і спростити розгортання. Хоча основна логіка додатку реалізована на бекенді, для забезпечення зручності взаємодії з користувачем було розроблено невеликий клієнтський інтерфейс.

Його єдине призначення – дати можливість користувачу безпосередньо ввести та передати свій API-ключ від `Crunchbase`.

Окремо розгляньмо реалізацію “`ExecutorFactory`” (рис. 3.6). Метою даного класу є організація обробки вхідних повідомлень, що надходять до додатку через чергу повідомлень “`monday queue`”. Він відповідає за динамічне створення об'єктів-виконавців (`executors`) відповідно до типу задачі, зазначеного в повідомленні.

Цей підхід базується на патерні `Factory Method` і дозволяє інкапсулювати логіку створення об'єктів, розділяючи ініціалізацію обробників (`SnapshotExecutor`, `SyncExecutor`, `FinalizeExecutor`) від решти системи. Завдяки використанню `ExecutorFactory`, додаток легко масштабується – додавання нових фаз або обробників потребує лише розширення переліку у `switch` без змін основної логіки черги.

`DbStateManager` – інкапсулює логіку доступу до бази даних під час виконання задачі (читання/оновлення стану синку, `item-ів` тощо).

```

export class ExecutorFactory {
  private dbStateManager: DbStateManager;
  private queueManager: QueueManager;

  constructor() {
    this.dbStateManager = new DbStateManager();
    this.queueManager = new QueueManager(queueService);
  }

  /**
   * Creates an executor based on the provided code.
   * @param code The message code that determines which executor to create
   * @returns An instance of the appropriate executor
   * @throws Error if the code is not recognized
   */
  createExecutor(code: SyncStateCode): any {
    switch (code) {
      case SyncStateCode.SNAPSHOT:
        return new SnapshotExecutor(this.dbStateManager, this.queueManager, EXECUTOR_TIME_LIMIT_MS);
      case SyncStateCode.SYNC:
        return new SyncExecutor(this.dbStateManager, this.queueManager, EXECUTOR_TIME_LIMIT_MS);
      case SyncStateCode.FINALIZE:
        return new FinalizeExecutor(this.dbStateManager, this.queueManager, EXECUTOR_TIME_LIMIT_MS);
      default:
        logger.error(`Unknown executor code`, { code });
        throw new Error(`Unknown executor code: ${code}`);
    }
  }
}

```

Рисунок 3.6 – Реалізація класу ExecutorFactory

QueueManager — відповідає за постановку нових задач у чергу після завершення поточної (наприклад, додавання SYNC або FINALIZE подій після SNAPSHOT). EXECUTOR_TIME_LIMIT_MS — константа, яка передається у кожен executor і вказує граничний час виконання задачі в мілісекундах, щоб дотриматися обмежень середовища Monday Code.

3.3 Тестування та оптимізація додатку

Після реалізації основної функціональності було проведено комплексне тестування та низку оптимізацій, спрямованих на забезпечення стабільності, масштабованості та відповідності вимогам середовища виконання Monday Code. Для ручного тестування API-запитів активно використовувався Postman, а також було створено тестові дошки в monday.com з різними сценаріями використання (заповнені/порожні поля, некоректні ключі, відсутність компаній у Crunchbase тощо).

Для перевірки коректності виконання всіх фаз циклу збагачення борди було реалізовано власний симулятор черги, який імітує поведінку Monday Queue, дозволяючи запускати обробку повідомлень у контрольованих умовах. Ключовий фрагмент коду зображено на рис. 3.7

```
// Enqueue endpoint
app.post('/enqueue', async (req, res) => {
  const message = req.body;
  console.log('\nReceived message:', JSON.stringify(message, null, 2));

  const shouldDispatch = process.env.AUTO_DISPATCH === 'true' ? true : await promptForDispatch(message);

  res.json({ status: 'dispatched', message });

  if (shouldDispatch) {
    await dispatchMessage(message);
  } else {
    console.log('Message dispatch cancelled');
  }
});
```

Рисунок 3.7 – Ключовий фрагмент коду власного симулятора черги

```
it('should not get subscriptions with sync errors', async () => {
  const twoDaysAgo = new Date(Date.now() - 2 * 24 * 60 * 60 * 1000);

  // Create test subscriptions
  const subscriptions = [
    // Account 1: QUIET with error
    {
      id: '1',
      accountId: 7860841,
      boardId: 1,
      userId: 1,
      websiteColumnId: 'col1',
      webhookUrl: 'http://test1.com',
      lastCycleCompletedAt: twoDaysAgo,
      syncStatus: SyncStatus.QUIET,
      syncError: 'API error occurred',
    },
    // Account 2: QUIET without error
    {
      id: '2',
      accountId: 7860842,
      boardId: 2,
      userId: 2,
      websiteColumnId: 'col2',
      webhookUrl: 'http://test2.com',
      lastCycleCompletedAt: twoDaysAgo,
      syncStatus: SyncStatus.QUIET,
      syncError: null,
    },
  ];

  await db(TABLE_NAMES.Subscriptions).insert(subscriptions);

  // Test with limit 2
  const result = await mondaySubscriptionsDao.getSubscriptionsToSync(2);

  // Should only get Account 2 (no error) and not Account 1 (has error)
  expect(result).toHaveLength(1);
  expect(+result[0].accountId).toBe(7860842);
});
```

Рисунок 3.8 – Модульний тест одного з методів Subscriptions DAO

Цей фрагмент ілюструє реалізацію кастомного симулятора черги, що був використаний для інтеграційного тестування додатку. Завдяки цьому підходу розробник мав змогу вручну або автоматично контролювати надсилання повідомлень, які в реальних умовах надсилає черга monday.com. Це дозволило поетапно верифікувати правильність виконання кожної з фаз обробки - SNAPSHOT, SYNC та FINALIZE — у контрольованому середовищі, що значно спростило виявлення та виправлення помилок на ранніх етапах розробки.

Велику увагу було приділено логіці відбору підписок (subscriptions) для запуску синхронізації. Зокрема, розроблено модульний тест, який перевіряє, що система коректно фільтрує записи з помилками (рис. 3.8).

3.4 Інструкція відвідувача та приклад функціонування

Розроблений додаток дозволяє автоматично збагачувати CRM-дані на дошці платформи monday.com актуальною інформацією з Crunchbase, що особливо корисно для бізнес-аналітики та моніторингу змін у компаніях-партнерах чи клієнтах.

Після встановлення додатку та налаштування інтеграції користувач отримує змогу налаштувати так званий «рецепт» — набір правил для автоматичного заповнення полів на дошці. Для цього необхідно:

- вказати колонку з веб-сайтом компанії;
- обрати, які саме поля з Crunchbase мають бути відображені на дошці;
- визначити колонку для відображення повідомлень про помилки.

Інтеграція запускається:

- під час створення нового рядка (item) на дошці;
- при оновленні назви або веб-сайту компанії;
- один раз на добу — щоденний синхронізуючий цикл з метою актуалізації даних.

Під час першого налаштування користувач повинен ввести валідний API ключ Crunchbase. Додаток перевіряє його справжність. У разі використання тестового (trial) або базового (basic) ключа користувач отримає повідомлення про помилку. Зміна ключа після первинної установки рецепту можлива лише шляхом повного перевстановлення додатку.

Нижче продемонстровано повний цикл взаємодії користувача з додатком Crunchbase CRM Enrichment у середовищі monday.com від початкової інсталяції до збагачення дошки актуальними бізнес-даними.

- користувач обирає робочий простір та конкретну дошку, де буде активовано додаток;

- із доступних інтеграцій користувач обирає шаблон, що дозволяє знаходити компанії в Crunchbase та автоматично переносити їхні дані до дошки (рис. 3.9);

- система запитує дозволу на доступ до необхідних даних (рис. 3.10).

Користувач підтверджує доступ до дошок, вебхуків та прав на зміну даних:

- після підтвердження доступу користувач повинен ввести особистий API-ключ Crunchbase якщо ключ недійсний або є пробним, система не дозволить завершити налаштування (рис. 3.11);

- користувач задає відповідність між полями в Crunchbase (наприклад, *City*, *Acquired Date*) та відповідними колонками на дошці (рис. 3.12);

- після налаштування система автоматично виконує початкову синхронізацію.

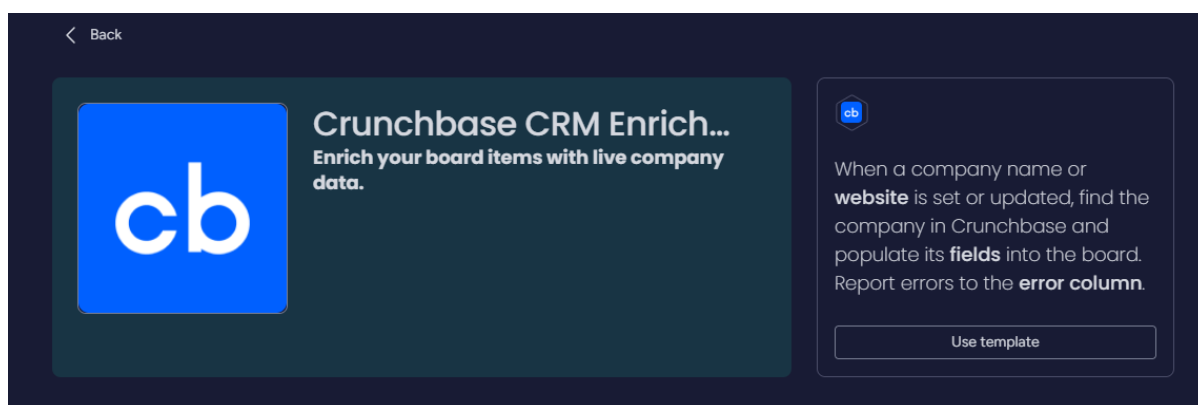


Рисунок 3.9 – Вибір шаблону автоматизації

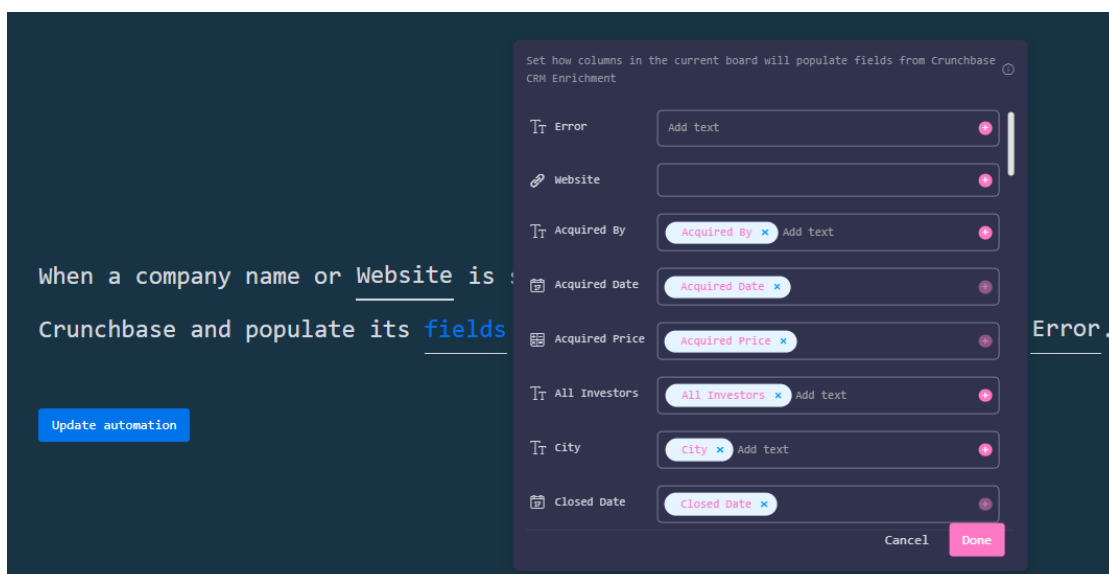


Рисунок 3.12 – Задання відповідності між полями Crunchbase та відповідними колонками на дошці

Як видно з рисунку, після заповнення полів, зазначених користувачем у налаштуваннях інтеграції, додаток автоматично підставляє відповідні значення, отримані з API Crunchbase.

Item	City	Closed Date	Contact Email	Country	Crunchbase Cat...	Crunchbase Rank	Crunchbase Ran...	Crunchbase Ran...	Crunchbase Ran...	Crunchbase URL	Diversity Spot...	Focus
google	Mountain View		google@google...	United States		151	-0.2	-0.4	1.2	www.crunchbase...		See
						151 sum	-0.2 sum	-0.4 sum	1.2 sum			See

Рисунок 3.13 – Результат збагачення рядку на дошці

У даному випадку для компанії “google” були успішно отримані та відображені такі атрибути, як місто, країна, контактний email, категорії, рейтинг, URL профілю на Crunchbase та інші. Це дозволяє учасникам команди, що працюють у monday.com, отримати актуальну бізнес-аналітику без потреби самостійного пошуку зовнішньої інформації.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Структурно-функціональний аналіз технологічного процесу

Розробка та вживання ефективних заходів запобігання аварійним і травмонебезпечним ситуаціям можливі лише при завчасному виявленні тих небезпек, з яких починаються процеси їх формування. Оскільки небезпечні умови не завжди завчасно можна виявити, а для вивчення небезпечних дій іноді потрібно багато часу, щоб зібрати статичний матеріал, то і методи виявлення цих небезпек повинні бути відповідно диференційовані (табл. 4.1) [1].

Таблиця 4.1. Моделі формування та виникнення травмонебезпечних і аварійних ситуацій

Вид робіт, виробничий підрозділ, робоче місце, виробниче обладнання, склад агрегату	Виробнича безпека			Можливі наслідки	Заходи запобігання небезпечним ситуаціям
	Небезпечна умова (НУ)	Небезпечна дія (НД)	Небезпечна ситуація (НС)		
Виконання робіт із електрообладнанням	Не вимкнено живлення. Відсутність заземлення.	Нехтування правилами ТБ	Ураження струмом	Травма (Т)	Проведення повторного інструктажу з ТБ. Розробка нових способів захисту. Встановлення заземлення.
НД ↓ НУ → НС → Т					

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи [1]:

- характеризують стан або рівень небезпеки обладнання, які використовуються.
- сприяють виникненню технологічних помилок обслуговуючого персоналу впродовж виробничого процесу;
- створювати умови та можливість проникнення працівника в небезпечну зону;
- приводять до виникнення небезпечних дій (внаслідок низького рівня професійної підготовки працівників та організації навчання з охорони праці).

Моделі формування та виникнення травмонебезпечних і аварійних ситуацій в комп'ютерному кабінеті представлено у вигляді моделі формування та виникнення травмонебезпечних і аварійних ситуацій – табл. 5.1.

4.2. Розрахунок освітлення приміщення комп'ютерного кабінету

Освітленість виробничих приміщень може бути штучною і природною. Природне освітлення при правильному обладнанні найбільш сприятливе для людини. Основні вимоги для освітлення наступні:

- освітлення повинне бути достатнім для швидкого і легкого розпізнання об'єктів роботи;
- освітлення повинно бути рівномірне без різких тіней;
- джерело світла не повинно осліплювати працівника;
- рівень освітленості не повинен обмежуватись часом.

Природне освітлення забезпечується обладнанням вікон (бокове освітлення) фонарів і світильних покриттів приміщень (верхнє освітлення). Природне освітлення нормується коефіцієнтом природної освітленості. Коефіцієнт природної освітленості – це процентне відношення фактичної освітленості F_v в будь-якій точці приміщення до освітленості F_n розсіяної світлом небозводу точки, яка лежить на відкритій місцевості. Розрахунок природного освітлення через бокові вікна по нормам освітленості ведеться для

самої дальньої від вікон точки, тобто знаходять мінімальне значення стик коефіцієнта природної освітленості [1]:

$$e_{\min} = \frac{F_b}{F_n} \cdot 100. \quad (4.1)$$

Значення коефіцієнта природної освітленості визначається не менше чим в п'яти точках. Значення коефіцієнта природної освітленості для сільськогосподарських виробничих приміщень в даному випадку ремонтній майстерні, беремо $e_{\min} = 5\%$.

Розрахунок природного освітлення зводиться до визначення площі світлових променів.

Сумарну площу світлових променів $\sum F_o (m^2)$ по коефіцієнту природної освітленості для бокових променів визначаємо по формулі:

$$\sum F_o = \frac{F_n \cdot e_{\min} \cdot r_o \cdot K}{100 \cdot \tau \cdot \Gamma_1}, \quad (4.2)$$

де F_n – площа підлоги, m^2 ; $e_{\min}$ – величина мінімального коефіцієнта природного освітленості; τ – загальний коефіцієнт світловикористання віконного отвору із врахуванням його забруднення, $\tau = 0,25$; r_o – світлова характеристика вікна, $r_o = 9,5$; Γ_1 – коефіцієнт, який враховує підвищення освітленості за рахунок світла, яке відбивається від стін і стелі, $\Gamma_1 = 1,2$; K – коефіцієнт, який враховує затінення вікон сусідніми приміщеннями і загорожею, $K = 1$.

$$\sum F_o = \frac{36 \cdot 0,5 \cdot 9,5 \cdot 1}{100 \cdot 0,25 \cdot 1,2} = 5,7 m^2.$$

Кількість світлових променів визначимо:

$$N = \frac{\sum F_o}{F_o}, \quad (4.3)$$

де F_o – площа вікна згідно стандарту, m^2 .

$$N = \frac{5,7}{6} = 0,95.$$

Приймаємо кількість вікон – одне вікно.

При розрахунку природного освітлення найбільш поширеним і простим є метод світлового потоку. При цьому методі розраховуємо світловий потік $F_{\text{л}}$ (Лк), який повинна випромінювати кожна лампа (при заданій кількості ламп).

$$F_{\text{л}} = \frac{k \cdot S_{\text{п}} \cdot E}{n_{\text{л}} \cdot \eta \cdot r^2}, \quad (4.4)$$

де k – коефіцієнт запасу, $k=1,3$; $S_{\text{п}}$ – площа підлоги, м^2 ; $S_{\text{п}}=36 \text{ м}^2$. E – нормативна освітленість, $E=300 \text{ Лк}$; $n_{\text{л}}$ – кількість встановлених ламп, $n_{\text{л}}=6 \text{ од}$; η – коефіцієнт використання світлового потоку, $\eta=0,25$; r – коефіцієнт нерівномірності освітленості, $r=0,545$.

Коефіцієнт запасу (K) враховує можливість забруднення світильників пилом, що залежить від характеру виробництва.

Розрахунок штучного освітлення починаємо із визначення висоти розташування світильника і їх кількості. Висоту $h_{\text{н}}$ (м) розташування світильників над робочим місцем знаходимо за формулою:

$$h_{\text{н}} = H - (h_1 + h_2), \quad (4.5)$$

де H – висота приміщення, м; h_1 – віддаль від підлоги до освітлювальної поверхні, м; h_2 – віддаль від стелі до світильника, м.

$$h_{\text{н}} = 4,5 - (2,2 + 1,5) = 0,8 \text{ м.}$$

При симетричному розміщенні світильників по вершинах квадратів їх кількість визначається за формулою:

$$n_{\text{с}} = \frac{S_{\text{п}}}{l^2}, \quad (4.6)$$

де l – віддаль між світильниками, м.

Підставивши значення отримаємо:

$$n_{\text{с}} = \frac{36}{9} = 4 \text{ од.}$$

Тоді світловий потік буде становити

$$F_{\text{л}} = \frac{1,3 \cdot 36 \cdot 300}{4 \cdot 0,25 \cdot 0,545} = 2576,2 \text{ Лк.}$$

З урахуванням світлового потоку 2576,2 Лк, ми обираємо тип лампи – LED-трубка T8 (G13) довжиною 1200 мм, яка замінює люмінесцентну 40 Вт.

Як варіант для вибору може бути – Philips CorePro LEDtube 1200mm 16.5W 840, або Osram SubstiTUBE Advanced T8 1200mm 16.8W.

4.3. Безпека в надзвичайних ситуаціях

Забезпечення захисту населення і території у разі загрози і виникнення надзвичайних ситуацій є одним з най важливіших завдань держави.

Захист населення є системою загально державних заходів, які реалізуються центральними і місцевими органами виконавчої влади, виконавчими органами влад, органами управління з питань надзвичайних ситуацій та цивільного захисту населення, підпорядкованими їм системами, та підприємств, що забезпечують виконання організаційних, інженерно – технічних, санітарно – гігієнічних, проти епідемічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій.

Загрози життєво важливих інтересів громадян, держави, суспільства поділяють на зовнішні та внутрішні, виконують під час надзвичайних ситуацій техногенного та природного характеру та воєнних конфліктах.

Принципи захисту впливають з основних положень Женевської конвенції щодо захисту жертв війни та додаткових протоколів до неї, можливого характеру воєнних дій, реальних можливостей держави щодо створення матеріальної бази захисту.

Оповіщення та інформування, яке досягається завчасним створенням і підтримкою в постійній готовності загально державної, територіальних та об'єктивних систем оповіщення населення.

ВИСНОВКИ

У межах кваліфікаційної роботи було успішно реалізовано додаток Crunchbase CRM Enrichment, який інтегрується з платформою Monday.com та автоматизує процес збагачення CRM-даних за допомогою API сервісу Crunchbase. Розробка охоплювала всі ключові етапи створення повноцінного SaaS-рішення: від аналізу предметної області та вибору технологій до проектування структури, реалізації бізнес-логіки, тестування та розгортання.

У ході роботи було досягнуто таких результатів:

- проаналізовано особливості CRM-систем та існуючих сервісів інтеграції;
- вибрано сучасний технологічний стек (Node.js, TypeScript, Fastify, PostgreSQL, React, Monday SDK), що забезпечив масштабованість та модульність рішення;
- розроблено зручний інтерфейс авторизації, який дозволяє користувачеві інтегрувати власний Crunchbase API ключ та налаштувати відповідність полів;
- реалізовано багатофазну архітектуру синхронізації, що складається з фаз Snapshot, Sync та Finalize, і дозволяє надійно обробляти великі обсяги даних у межах обмежень Monday Code;
- забезпечено fault-tolerant логіку чергування задач через Monday Queue, а також контроль лімітів по часу й кількості записів;
- проведено модульне та інтеграційне тестування всіх основних компонентів додатку;
- розроблено та оформлено супровідну документацію для користувача та адміністрування додатку.

Результатом є гнучкий, надійний інструмент для збагачення CRM-даних, який може використовуватись у реальному бізнес-середовищі. Його використання дозволяє значно зменшити ручну працю та підвищити точність аналітики щодо контрагентів або потенційних клієнтів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Довідник нормативних документів у сфері охорони праці, пожежної безпеки, гігієни праці та соціального страхування від нещасних випадків / К.: Фонд соціального страхування від нещасних випадків на виробництві та професійних захворювань України, 2009. — 244 с.
2. Винокурова Л.Є., Васильчук М.В., Гаман М.В. Основи охорони праці: Підручник для проф.-техн. навч. закладів. — 2001. — 192 с.
3. Вікіпедія. JavaScript. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення: 22.04.2025).
4. Ерік Фрімен, Елізабет Робсон. Head First. Програмування JavaScript. — 2022. — 672 с.
5. Програмування на JavaScript для веб-розробників: практичні аспекти та приклади / за ред. В. П. Кравченка. — Львів: 2022. — 220 с.
6. Робін Н. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS та HTML5. — 6-те вид. — Прес, 2019. — 816 с. Crunchbase Enterprise API Documentation. SwaggerHub. URL: https://app.swaggerhub.com/apis-docs/Crunchbase/crunchbase-enterprise_api/1.0.3
7. Axios GitHub Repository. URL: <https://github.com/axios/axios>
8. Crunchbase Enterprise API Documentation. SwaggerHub. URL: https://app.swaggerhub.com/apis-docs/Crunchbase/crunchbase-enterprise_api/1.0.3
9. Docker Documentation. URL: <https://docs.docker.com/>
10. Fastify Documentation. URL: <https://www.fastify.io/docs/latest/>
11. Johnson, E. Розробка веб-сайтів з використанням Node.js, Express та MongoDB. — O'Reilly Media, 2014. — 352 с.
12. Knex.js Documentation. URL: <https://knexjs.org/>
13. MacFarland, D. CSS: Практичний посібник. — O'Reilly Media, 2017. — 840 с.

- 14.MDN Web Docs. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (дата звернення: 22.04.2025).
- 15.MobX Documentation. URL: <https://mobx.js.org/README.html>
- 16.Monday Apps SDK Documentation. URL: <https://developer.monday.com/apps/docs>
- 17.Neon Database Documentation. URL: <https://neon.tech/docs>
- 18.PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/>
- 19.Postman Learning Center. URL: <https://learning.postman.com/docs/getting-started/introduction/>
- 20.Vite Documentation. URL: <https://vitejs.dev/>
- 21.Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs>
- 22.DOU. Рейтинг мов програмування 2023. URL: <https://dou.ua/lenta/articles/language-rating-2023/> (дата звернення: 22.04.2025).
- 23.Sass Language Documentation. URL: <https://sass-lang.com/documentation>