

Міністерство освіти і науки України  
Кам'янець-Подільський національний університет імені Івана Огієнка  
Фізико-математичний факультет  
Кафедра комп'ютерних наук

## **Кваліфікаційна робота бакалавра**

з теми: **«РОЗРОБКА ВЕБ ЗАСТОСУНКУ ДЛЯ ОБРОБКИ  
ЗАМОВЛЕНЬ НА ВИГОТОВЛЕННЯ ДЕТАЛЕЙ ДЛЯ  
МЕТАЛООБРОБНОГО ПІДПРИЄМСТВА НА ОСНОВІ ТЕХНОЛОГІЇ  
LOW CODE ТА ПЛАТФОРМИ NO CODE»**

Виконав: здобувач вищої освіти групи KNms1-B22  
спеціальності 122 Комп'ютерні науки

Марціновський Дмитро Михайлович

(прізвище, ім'я та по батькові здобувача вищої освіти)

Керівник: Слободянюк Олександр Васильович,

кандидат технічних наук, доцент кафедри

комп'ютерних наук

(прізвище, ім'я та по батькові, науковий ступінь, вчене звання керівника)

Рецензент: \_\_\_\_\_

\_\_\_\_\_  
(прізвище, ім'я та по батькові, науковий ступінь, вчене звання рецензента)

м. Кам'янець-Подільський - 2025 р.

## АНОТАЦІЯ

У науковій роботі розглянуто процес розробки веб застосунку для обробки замовлень на виготовлення деталей для металообробного підприємства з використанням технологій low-code та no-code. Актуальність дослідження обумовлена потребою у швидкому та економічно ефективному впровадженні цифрових рішень для автоматизації бізнес-процесів виробничих підприємств.

Об'єктом дослідження є процес цифрової взаємодії між клієнтами та підприємством у сфері обробки замовлень.

Предметом є програмна реалізація веб застосунку із використанням no-code платформи Airtable та супутніх low-code рішень для обробки, збереження та перегляду інформації

Метою роботи є створення вебзастосунку, який забезпечує зручне подання заявок клієнтами, фіксацію та обробку замовлень, повідомлення відповідальних осіб та можливість подальшого масштабування без глибоких знань програмування.

У процесі дослідження проаналізовано сучасні підходи до створення веб застосунків, оцінено можливості та обмеження традиційної розробки на основі коду, а також здійснено порівняльний аналіз технологій low-code та no-code. Для реалізації системи використано платформу Airtable, засоби автоматизації бізнес-процесів, візуальне моделювання сайту та написання скриптів на JavaScript.

Результатом роботи є функціональний прототип вебзастосунку для обробки замовлень, який поєднує простоту впровадження, інтуїтивно зрозумілий інтерфейс та можливість подальшого розвитку відповідно до потреб підприємства.

Робота може бути корисною для здобувачів вищої освіти спеціальності 122 «Комп'ютерні науки», а також для фахівців, які зацікавлені в автоматизації бізнес-процесів із використанням no-code та low-code технологій.

## ЗМІСТ

|  |           |
|--|-----------|
| <b>АНОТАЦІЯ .....</b>  | <b>2</b>  |
| <b>ВСТУП.....</b>  | <b>4</b>  |
| <b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....</b>  | <b>7</b>  |
| 1.1 Сучасні підходи до створення бізнес-орієнтованих вебзастосунків .....              | 7         |
| 1.2 Функціональні та нефункціональні вимоги до веб застосунків підприємства .....      | 8         |
| 1.3. Труднощі традиційної веб розробки при використанні виключно програмного коду .... | 10        |
| 1.4. Аналіз можливостей технологій low-code та no-code у розробці вебрішень .....      | 11        |
| Висновки до розділу. ....  | 13        |
| <b>РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ ЗАСТОСУНКУ .....</b>                        | <b>15</b> |
| 2.1. Постановка технічного завдання .....  | 15        |
| 2.2. . Обґрунтування вибору платформи для реалізації .....                             | 16        |
| 2.3 Архітектура та структура веб застосунку .....                                      | 18        |
| 2.4. Реалізація функціональних компонентів веб застосунку .....                        | 19        |
| Висновки до розділу 2 .....  | 22        |
| <b>РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ РІШЕННЯ ...</b>                         | <b>24</b> |
| 3.1. Методика тестування системи .....   | 24        |
| 3.2. . Аналіз результатів тестування .....   | 25        |
| 3.3. Оцінка ефективності використання no-code рішень у практичному застосуванні .....  | 27        |
| 3.4. Перспективи розвитку та удосконалення розробленого веб застосунку .....           | 29        |
| Висновки до розділу 3 .....  | 30        |
| <b>ВИСНОВКИ .....</b>  | <b>31</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>   | <b>33</b> |

## ВСТУП

У сучасних умовах цифрової трансформації виробничих процесів зростає потреба у впровадженні інформаційних технологій на підприємствах різних галузей, зокрема у сфері металообробки. Автоматизація управлінських задач, зокрема обробки замовлень, дозволяє підвищити ефективність бізнес-процесів, зменшити людський фактор та прискорити взаємодію з клієнтами. Особливої актуальності набувають веб застосунки, які забезпечують зручне введення, обробку, перегляд та аналіз даних без потреби у складній ручній роботі.

бливої актуальності набувають веб застосунки, які забезпечують зручне введення, обробку, перегляд та аналіз даних без потреби у складній ручній роботі.

Проте створення повноцінного вебзастосунку традиційними засобами програмування вимагає значних часових, фінансових і технічних ресурсів. Це створює бар'єри для малих і середніх підприємств, які не мають змоги залучити IT-фахівців або розгортати складні інфраструктури. У цьому контексті все більшого поширення набувають технології **low-code** та **no-code**, що дозволяють реалізовувати цифрові рішення з мінімальним залученням програмного коду, використовуючи візуальні редактори, готові шаблони та автоматизовані інтеграції.

Актуальність теми зумовлена необхідністю створення гнучкого, функціонального та доступного веб застосунку для автоматизації обробки замовлень у сфері металообробки із застосуванням сучасних no-code платформ. Саме тому було запропоновано реалізувати інформаційну систему на базі платформи **Airtable**, яка поєднує можливості роботи з базами даних, інтерфейсами введення та автоматизаціями.

**Об'єктом дослідження** є процес цифрової взаємодії між клієнтами та виробничим підприємством у межах оформлення та виконання замовлень.

**Предметом дослідження** виступає вебзастосунок, створений за допомогою no-code/low-code технологій, який забезпечує збір, обробку та управління інформацією про замовлення.

**Метою дослідження** є розробка прототипу веб застосунку для обробки замовлень металообробного підприємства із застосуванням платформи Airtable, що дозволяє ефективно автоматизувати ключові бізнес-процеси.

Для досягнення поставленої мети у роботі вирішуються такі завдання:

1. Проаналізувати сучасні підходи до створення бізнес-орієнтованих вебзастосунків.
2. Сформулювати функціональні та нефункціональні вимоги до системи.
3. Обґрунтувати вибір інструментів для реалізації no-code/low-code рішення.
4. Розробити структуру бази даних, інтерфейсу та логіку роботи системи.
5. Реалізувати вебзастосунок з можливістю створення та обробки замовлень.
6. Інтегрувати автоматизовані сповіщення та базові скрипти обробки подій.
7. Провести тестування функціоналу системи та оцінити її ефективність.

**Методи дослідження:** у роботі застосовано методи аналізу предметної області, проектування інформаційних систем, моделювання баз даних, а також автоматизації бізнес-процесів із використанням no-code платформ і scripting-функціоналу. Технічною основою виступає платформа Airtable у поєднанні з скриптами на JavaScript.

**Практичне значення** одержаних результатів полягає у створенні готового до впровадження прототипу вебзастосунку для обробки замовлень, який може бути адаптований під потреби малих та середніх підприємств у сфері виробництва та сервісу.

**Структура роботи.** Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі представлено огляд сучасних підходів до розробки вебзастосунків, а також аналіз доцільності використання технологій low-code та no-code. У другому розділі описано технічне завдання, обґрунтовано вибір платформи, спроектовано та реалізовано вебзастосунок. У третьому розділі проведено

тестування функціональності системи та розглянуто можливості її подальшого розвитку.

## РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Сучасні підходи до створення бізнес-орієнтованих вебзастосунків

У сучасному цифровому середовищі вебзастосунки стали невіддільною складовою інформаційної інфраструктури підприємств, що прагнуть автоматизувати бізнес-процеси, оптимізувати взаємодію з клієнтами та підвищити ефективність внутрішніх операцій. Особливої актуальності це набуває у виробничій сфері, де облік і контроль замовлень, обробка заявок, планування виробництва та комунікація з клієнтами вимагають швидких, зручних та адаптивних рішень.

Традиційно створення вебзастосунків здійснюється із залученням повного циклу розробки — від формування вимог до написання програмного коду, налаштування серверного середовища та деплою. Найбільш поширеними підходами у такій моделі є:

- **Розробка з використанням фреймворків** (React, Angular, Vue.js на клієнтській частині та Node.js, Django, Laravel на серверній),
- **Застосування CMS (Content Management System)** для швидкої побудови сайтів, зокрема WordPress, Joomla, Drupal,
- **Розробка індивідуальних рішень «з нуля»** із повним контролем над архітектурою та логікою.

Кожен з підходів має свої переваги: максимальну гнучкість, глибину налаштувань, масштабованість. Проте разом з цим існують і значні недоліки, такі як висока вартість реалізації, потреба у кваліфікованому персоналі, тривалі терміни розробки та складність подальшого супроводу.

У відповідь на ці виклики останніми роками активного розвитку набули **альтернативні підходи на основі low-code та no-code технологій**, які дозволяють створювати повноцінні вебзастосунки без потреби писати великий обсяг програмного коду. Такі платформи, як Webflow, Airtable, Bubble, OutSystems, AppGyver, забезпечують візуальне проєктування

інтерфейсів, інтеграцію з базами даних, налаштування логіки бізнес-процесів і навіть автоматизацію через готові сценарії або скрипти.

Сучасні no-code та low-code інструменти орієнтовані насамперед на **швидке прототипування, створення MVP (Minimum Viable Product)**, а також цифровізацію окремих процесів підприємств, що не мають змоги реалізовувати повноцінні проєкти із залученням ІТ-команд.

Паралельно з цим зростає популярність концепцій **Business Process Automation (BPA)** та **Digital Transformation**, у межах яких low-code рішення слугують інструментом підвищення гнучкості та адаптивності бізнесу до змін у зовнішньому середовищі.

Таким чином, сучасний процес розробки бізнес-орієнтованих вебзастосунків охоплює як класичні методи з глибоким технічним налаштуванням, так і новітні підходи, що забезпечують пришвидшене створення систем із мінімальним технічним бар'єром. Вибір конкретного підходу залежить від ресурсів підприємства, складності задачі, масштабів впровадження та потреб у подальшій підтримці та розвитку.

## **1.2 Функціональні та нефункціональні вимоги до веб застосунків підприємства**

Успішне впровадження інформаційної системи у бізнес-процеси підприємства залежить від правильного формулювання вимог до неї ще на етапі проєктування. Вимоги до веб застосунку можна умовно поділити на дві групи: функціональні, які визначають, що саме система повинна робити, та нефункціональні, що описують якість реалізації функцій та обмеження зовнішнього середовища.

Функціональні вимоги:

Функціональні вимоги — це конкретний перелік можливостей, які повинен реалізовувати веб застосунок у рамках бізнес-логіки. Для

вебсистеми обробки замовлень на металообробному підприємстві до основних функціональних вимог можна віднести:

- Форма для подачі заявки замовником з можливістю введення контактних даних, опису замовлення, вибору типу обробки та завантаження креслень або технічної документації у вигляді файлів.
- Сторінка підтвердження успішної подачі заявки.
- Адміністративна панель для працівників підприємства, яка містить зручний інтерфейс перегляду нових, оброблених та архівних заявок.
- Функція фільтрації та пошуку за статусом, датою, замовником або типом деталі.
- Можливість змінювати статус заявки (нове, в обробці, виконано, відхилено) та додавати коментарі або внутрішні примітки.
- Автоматичне надсилання повідомлень електронною поштою або іншим каналом зв'язку (наприклад, через Viber API) уповноваженому працівнику після подачі заявки або зміни її статусу.
- Зберігання історії замовлень із можливістю подальшого аналізу та експорту даних.
- Можливість масштабування функціоналу — наприклад, додавання реєстрації для постійних клієнтів або кабінету користувача в майбутньому.

Нефункціональні вимоги:

Нефункціональні вимоги визначають якісні характеристики вебзастосунку, що впливають на зручність, продуктивність, безпеку та адаптивність системи:

- Зручність інтерфейсу (Usability): усі елементи інтерфейсу мають бути інтуїтивно зрозумілими навіть для користувачів без технічної підготовки.
- Адаптивність: вебзастосунок повинен коректно відображатися на пристроях із різними розмірами екранів (десктопи, планшети, смартфони).
- Продуктивність: система повинна забезпечувати швидку обробку запитів та миттєве відображення даних у межах допустимого навантаження.
- Безпека: конфіденційні дані (персональна інформація замовника, креслення тощо) повинні бути захищені від несанкціонованого доступу.

- Надійність: система має бути стійкою до збоїв та несправностей, з можливістю резервного збереження даних.

- Мінімізація витрат на впровадження та підтримку: використання no-code/low-code підходів повинно забезпечувати економічну ефективність.

Таким чином, чітке формулювання функціональних та нефункціональних вимог дозволяє не лише визначити основні параметри майбутнього вебзастосунку, а й слугує орієнтиром для вибору платформи реалізації та подальшого тестування розробленої системи.

### **1.3. Труднощі традиційної веб розробки при використанні виключно програмного коду**

Традиційна розробка вебзастосунків передбачає використання мов програмування, фреймворків та ручної конфігурації інфраструктури. Такий підхід, хоч і надає повну гнучкість у реалізації функціоналу, має низку істотних труднощів, які стають особливо відчутними в умовах обмежених ресурсів, стислих термінів або відсутності кваліфікованого технічного персоналу.

Однією з основних проблем є високий поріг входу. Для реалізації навіть базового вебінтерфейсу необхідне знання HTML, CSS, JavaScript, а також одного або декількох фреймворків, таких як React, Vue.js чи Angular. Крім клієнтської частини, необхідно створити серверну логіку (на Node.js, PHP, Python тощо), налагодити роботу з базою даних (MySQL, PostgreSQL, MongoDB), реалізувати API, а також подбати про захист, авторизацію, обробку помилок.

Другим суттєвим обмеженням є тривалість розробки. Повноцінний цикл створення вебзастосунку включає в себе етапи проектування, кодування, тестування, виправлення помилок, деплою, а також написання технічної документації. Навіть при використанні сучасних бібліотек та готових рішень, створення стабільної системи з нуля потребує значного часу.

Також слід відзначити високу залежність від команди розробників. Підтримка й оновлення системи після впровадження можливі лише за умови наявності фахівців, які володіють усіма деталями проєкту та використовуваними технологіями. У випадку звільнення чи зміни складу команди, це може призвести до втрати керованості проєктом.

Ще однією проблемою є фінансові витрати. Для багатьох підприємств, особливо невеликих, повноцінна розробка з нуля може бути занадто дорогою. Сюди входять витрати на заробітну плату програмістів, адміністраторів серверів, інструменти розробки, ліцензії та хостинг.

Не менш важливою є складність масштабування та інтеграцій. У традиційному підході додавання нового функціоналу, наприклад, інтеграції з CRM або сервісами електронної пошти, часто вимагає значних змін у кодї, додаткового тестування та можливих збоїв у роботі системи.

Таким чином, хоча традиційна розробка забезпечує максимальну контрольованість і функціональність, вона є ресурсоємною, складною в підтримці та малоефективною у випадках, коли необхідно створити простий, але стабільний бізнес-інструмент у стислі терміни. Ці обставини сприяли розвитку альтернативних підходів, таких як low-code та no-code, які дозволяють значно спростити процес розробки та знизити витрати на впровадження.

#### **1.4. Аналіз можливостей технологій low-code та no-code у розробці вебрішень**

У відповідь на зростаючі потреби бізнесу у швидкому створенні цифрових рішень, значного поширення набули підходи low-code та no-code, які дозволяють реалізовувати функціональні інформаційні системи без написання або з мінімальним обсягом програмного коду. Ці підходи є ефективною альтернативою традиційній розробці, особливо у випадках, коли

необхідно швидко створити прототип або повноцінний застосунок для автоматизації окремих бізнес-процесів.

No-code платформи орієнтовані переважно на користувачів без технічної освіти. Вони дозволяють створювати вебзастосунки за допомогою графічних інтерфейсів, готових блоків, шаблонів та інтеграцій. Прикладами таких платформ є Webflow, Airtable, Glide, Adalo та інші. Основні можливості no-code рішень включають:

- створення форм введення та опрацювання даних;
- побудову структури баз даних через табличні редактори;
- налаштування логіки бізнес-процесів за допомогою умов і тригерів;
- реалізацію автоматизованих дій (сповіщення, оновлення записів тощо).

Low-code платформи, на відміну від no-code, орієнтовані на розробників або технічно підготовлених користувачів. Вони поєднують графічне середовище з можливістю написання фрагментів коду (переважно на JavaScript або подібних мовах) для розширення функціоналу. Прикладами є Bubble, OutSystems, AppGyver, Retool. Їхні переваги:

- вища гнучкість у реалізації специфічної логіки;
- підтримка API та зовнішніх інтеграцій;
- можливість створення складних інтерфейсів та користувацьких сценаріїв;
- інтерфейсів та користувацьких сценаріїв;
- глибший контроль над даними та процесами.

Основні переваги використання low-code/no-code платформ у порівнянні з традиційною розробкою:

- Скорочення часу розробки — у декілька разів порівняно зі створенням коду «з нуля»;
- Зниження вартості впровадження — не потребує команди програмістів;
- Можливість самостійного оновлення системи без втручання в код;
- Гнучкість та адаптивність — швидка модифікація інтерфейсів, логіки та структури даних;

- Вбудовані інтеграції з популярними сервісами (Gmail, Slack, Google Sheets, Telegram API тощо).

Разом із тим, використання low-code/no-code підходів має й обмеження. До найпоширеніших належать:

- обмеження у реалізації нестандартної або складної логіки;
- залежність від функціональності платформи;
- складність міграції на традиційну архітектуру в разі масштабування;
- можлива вартість використання платформи у комерційних проєктах.

Незважаючи на ці обмеження, сучасні low-code та no-code рішення демонструють високу ефективність у розробці внутрішніх корпоративних систем, MVP-проєктів, CRM- та ERP-модулів, систем обліку та елементів автоматизації в малому та середньому бізнесі. Це робить їх надзвичайно актуальними для таких задач, як створення вебзастосунку для обробки замовлень на металообробному підприємстві.

### **Висновки до розділу.**

У першому розділі було здійснено огляд сучасних підходів до створення бізнес-орієнтованих вебзастосунків та проаналізовано їх особливості у контексті потреб виробничих підприємств. Було визначено, що традиційна модель розробки, яка базується на програмуванні з нуля, хоча й забезпечує максимальну гнучкість, є складною у впровадженні, потребує значних ресурсів та часу, а також вимагає наявності висококваліфікованих технічних фахівців.

часних підходів до створення бізнес-орієнтованих вебзастосунків та проаналізовано їх особливості у контексті потреб виробничих підприємств. Було визначено, що традиційна модель розробки, яка базується на програмуванні з нуля, хоча й забезпечує максимальну гнучкість, є складною у впровадженні, потребує значних ресурсів та часу, а також вимагає наявності висококваліфікованих технічних фахівців.

Розглянуто функціональні та нефункціональні вимоги до вебзастосунку для металообробного підприємства, серед яких ключовими є: зручна форма для подачі замовлень, можливість завантаження креслень, ефективна панель адміністрування, система фільтрації, оновлення статусів та сповіщення відповідальних осіб.

Особливу увагу приділено аналізу можливостей технологій **low-code** та **no-code**, які дозволяють реалізувати описаний функціонал з мінімальними затратами часу та ресурсів. Платформи цього класу надають інструменти для створення адаптивних, інтуїтивно зрозумілих і функціональних вебінтерфейсів без потреби у написанні повноцінного коду. Вони є особливо вигідними для бізнесу, оскільки дозволяють автоматизувати рутинні процеси без залучення програмістів.

З урахуванням вищезазначених факторів, для реалізації дипломного проєкту було обрано підхід на основі **no-code** платформи **Airtable** із можливістю додавання логіки **скрипти** та **автоматизації**. Такий вибір дозволяє забезпечити виконання всіх ключових вимог, зберігаючи при цьому простоту, доступність і гнучкість у розширенні функціональності в майбутньому.

## **РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ОБРОБКИ ЗАМОВЛЕНЬ ПО ВИГОТОВЛЕННЮ ДЕТАЛЕЙ НА МЕТА- ЛООБРОБНОМУ ПІДПРИЄМСТВІ**

### **2.1. Постановка технічного завдання**

Метою розробки веб застосунку є автоматизація процесу обробки замовлень на виготовлення деталей для металообробного підприємства. Система повинна забезпечувати зручний канал комунікації між клієнтом та працівниками підприємства, а також дозволяти оперативно обробляти заявки, контролювати їхній статус і формувати єдину базу даних замовлень.

Веб застосунок має реалізовувати наступний набір функціональних можливостей:

1. Для замовника (клієнтська частина):
  - Інтерактивна форма подачі заявки, яка містить:
    - поля для введення ПІБ, контактному телефону та електронної пошти;
    - поле опису технічного завдання;
    - кількість потрібних деталей у партії;
    - можливість прикріпити креслення;
    - вказати на потребу консультації майстра-інженера;
  - Підтвердження успішної відправки заявки;
  - Опціонально: можливість отримати повідомлення на email про прийняття заявки в обробку.
2. Для адміністратора (внутрішня частина системи):
  - Панель перегляду всіх заявок із поділом на категорії:
    - нові заявки;
    - дата подачі заявки;
    - контакт замовлення;

- заявки в обробці;
  - виконані/архівні заявки;
  - Можливість фільтрації та пошуку за статусом, датою, ПІБ клієнта, назвою деталі;
  - Перегляд повної інформації по кожному замовленню (включно з прикріпленими файлами);
3. Адміністративні вимоги:
- Можливість швидкого додавання нових полів до заявки без переробки всієї системи;
  - Експорт бази замовлень у формат CSV/Excel для подальшого аналізу або звітності;
  - Захист від несанкціонованого доступу (доступ до адміністративної панелі має бути обмежений).

Обмеження:

- Система не передбачає реєстрації користувачів або платіжних операцій;
- Передбачається використання лише безкоштовних або умовно безкоштовних можливостей платформи Airtable (або інших обраних сервісів).

Поставлене технічне завдання повинно бути реалізоване із використанням no-code платформи Airtable з інтеграцією форм, внутрішньої бази даних та автоматизацій. У разі потреби — допускається використання скриптів на JavaScript у рамках вбудованих функцій Airtable Scripting Extension.

## **2.2. . Обґрунтування вибору платформи для реалізації**

Для реалізації вебзастосунку з обробки замовлень на виготовлення деталей було розглянуто низку сучасних інструментів у сфері no-code та low-code розробки. Основними критеріями відбору платформи виступали:

- відповідність функціональним вимогам (форма заявок, керування статусами, прикріплення файлів, панель адміністратора);

- можливість інтеграцій із зовнішніми сервісами (наприклад, email або Telegram для сповіщень);
- підтримка автоматизації та елементарних скриптів;
- простота використання та адаптація до змін у структурі даних;
- мінімальні вимоги до технічної підготовки;
- доступність (наявність безкоштовного тарифу або умовно безкоштовної моделі).

У результаті порівняння було обрано Airtable як основну платформу для реалізації вебзастосунку. Це сучасний no-code сервіс, який поєднує функції табличного редактора, бази даних і системи управління контентом. Основні переваги Airtable у контексті поставленого завдання:

- Підтримка вкладень — можливість прикріплювати до записів креслення та документи різних форматів;
- Гнучка структура таблиць — дозволяє легко додавати, редагувати та фільтрувати поля;
- Формування публічних форм — з відкритим доступом для користувачів без облікового запису;
- Автоматизації (Airtable Automations) — для надсилання листів, зміни статусу, запуску скриптів;
- Scripting Extension — дозволяє реалізувати специфічну бізнес-логіку на JavaScript у разі потреби;
- Інтеграції з Make, Zapier, Telegram API, Google Workspace — відкриває широкі можливості для масштабування;
- Інтуїтивно зрозумілий інтерфейс — що дозволяє адміністраторам без технічної освіти користуватись системою.

Саме тому Airtable обрано як оптимальний інструмент для швидкої розробки MVP вебзастосунку без втрати ключової функціональності.

У наступному підрозділі буде розглянуто проектування структури системи, а також архітектуру та логіку її роботи.

### 2.3 Архітектура та структура веб застосунку

Архітектура веб застосунку визначає загальний підхід до організації логіки роботи системи, структуру зберігання даних, а також взаємодію користувачів із функціональними компонентами. У рамках розробки вебзастосунку для обробки замовлень на виготовлення деталей використано централізовану модель, побудовану на основі платформи Airtable, яка дозволяє реалізувати як зберігання інформації, так і інтерфейсні та автоматизовані функції.

Основу архітектури становить база даних, представлена у вигляді табличної структури. Кожен запис у системі відповідає окремому замовленню клієнта. Записи містять поля для контактної інформації замовника, опису замовлення, прикріплених креслень або технічних файлів, а також статусу заявки та внутрішніх коментарів адміністратора. Така структура забезпечує просте і наочне представлення інформації, зручне для щоденного використання персоналом підприємства.

Система включає дві ключові ролі користувачів: замовника та адміністратора.

Клієнт взаємодіє із системою через публічну форму подачі заявки, створену засобами Airtable. Ця форма містить обов'язкові поля для заповнення особистих даних, опису завдання, а також блок для завантаження технічних файлів. Після надсилання заявки вона автоматично додається до бази даних.

завантаження технічних файлів. Після надсилання заявки вона автоматично додається до бази даних.

Адміністратор отримує доступ до структурованого представлення всіх заявок у вигляді окремих подань: нові заявки, ті, що в роботі, завершені або відхилені. Подання налаштовуються безпосередньо в Airtable й дозволяють швидко фільтрувати записи за статусом, датою чи іншими параметрами. Кожен запис можна відкривати для перегляду повної інформації, змінювати

статус, залишати службові нотатки, а також ініціювати додаткові дії — наприклад, надсилання повідомлення клієнту.

У межах архітектури застосовано також механізми автоматизації, які дозволяють:

- надсилати сповіщення адміністраторам про нові заявки;
- автоматично встановлювати статус «нове» для щойно створених записів;
- запускати скрипти для обробки або сортування даних.

Особливістю обраного підходу є те, що вся логіка реалізується у межах однієї платформи, без потреби у зовнішньому серверному середовищі. Це спрощує обслуговування системи, знижує технічні вимоги до користувачів і прискорює адаптацію персоналу до нової цифрової інфраструктури.

Таким чином, структура веб застосунку є гнучкою, масштабованою та простою в реалізації. Вона охоплює всі необхідні компоненти для ефективного оброблення замовлень і може бути легко розширена або інтегрована з іншими системами в майбутньому.

## **2.4. Реалізація функціональних компонентів веб застосунку**

На основі сформульованого технічного завдання та спроектованої архітектури було реалізовано вебзастосунок для обробки замовлень із використанням no-code платформи Airtable. Весь процес розробки відбувався без написання повноцінного коду, однак із залученням елементів скриптової логіки та автоматизацій, що відповідає підходу low-code.

### **1. Реалізація клієнтської форми заявки**

Форма для подачі замовлення була створена за допомогою вбудованої функціональності Airtable Forms. До форми включено наступні обов'язкові елементи:

- Опис деталі на замовлення;
- Матеріал виготовлення (згідно ДСТУ);
- Кількість деталей у партії;
- компонент завантаження файлів (для креслень або документації);

- поле для введення контактної інформації;
- Прапорець (англ. CheckBox) з відміткою потреби консультації майстра-інженера виробництва;

Форма налаштована таким чином, щоб будь-який користувач із посиланням мав змогу залишити заявку без входу в систему. Після її заповнення дані автоматично потрапляють до основної таблиці, що зберігає інформацію про замовлення.

### Заявка на виробництво деталей за вашим кресленням

|   |  |
|---|--|
| Опис деталі на замовлення                     | <input type="text"/>   |
| Матеріал виготовлення                         | (Згідно ДСТУ)<br><input type="text"/>                                      |
| Кількість деталей у партії                    | <input type="text"/>   |
| Завантажте ваше креслення деталі              | (У форматі .DWG)<br><input type="text" value="Drop files here or browse"/> |
| Ваш номер телефону?                           | <input type="text"/>   |
| Чи потрібна вам консультація нашого інженера? | <input checked="" type="checkbox"/>  |

[Clear form](#)

Рисунок 2.1 – Вигляд форми для клієнта

## 2. Панель адміністратора

Панель адміністратора реалізована у вигляді базового інтерфейсу керування заявками всередині Airtable. Для зручності обробки замовлень створено подання, які дозволяють впорядковувати й переглядати інформацію у структурованому вигляді. Кожна заявка представлена у вигляді окремого запису з такими ключовими елементами:

- Опис деталі;

- Контактна інформація замовника;
- Статус заявки ;
- Дата та час;
- Прапорець «Потребує консультації інженера»;
- Тип матеріалу;
- Кількість деталей;
- Креслення;

Ці дані відображаються у вигляді зручної таблиці з можливістю фільтрації, сортування та пошуку. За потреби адміністратор може швидко відкривати окремий запис для перегляду деталей, зміни статусу або додавання коментарів.

|   | Опис деталі                   | Статус заявки | Дата подачі      | Контакт замовника | Консультація інженера | Матеріал    | # кількість у партії | Креслення у форма... |
|---|-------------------------------|---------------|------------------|-------------------|-----------------------|-------------|----------------------|----------------------|
| 1 | ДТК АКС-74У                   | У процесі     | 12.12.2024 00:33 | +380674948860     | ✓                     | Титан ПТ-3В | 800,0                |                      |
| 2 | Ударник ствольної групи НК416 | Нова          | 12.12.2024 00:27 | +0961549316       | ✓                     | Титан ПТ-3В | 1200,0               |                      |
| + |                               |               |                  |                   |                       |             |                      |                      |

Рисунок 2.2 – Вигляд панелі адміністратора

### 3. Автоматизація дій

Було реалізовано низку автоматизованих сценаріїв (Airtable Automations), які значно спрощують роботу адміністратора та покращують комунікацію з клієнтом. Зокрема:

- Сповіщення адміністратора про нову заявку — одразу після створення нового запису система надсилає лист на корпоративну електронну адресу.
- Автоматична зміна статусу заявки — якщо адміністратор залишає коментар або вносить зміни, статус заявки оновлюється автоматично (функція відключається для зручності адміністратора).

#### 4. Сценарії та скрипти

У рамках розширення функціональності було використано модуль **Scripting Extension**, який дозволяє запускати прості скрипти на JavaScript.

Наприклад:

- Очищення порожніх записів — скрипт видаляє тестові або випадкові заявки без контактної інформації чи прикріплених файлів;
- Підрахунок кількості активних заявок за день або місяць;
- Підрахунок заявок із позначкою “потребує консультації” — для оцінки навантаження на інженерний відділ;

#### 5. Гнучкість і масштабованість

Реалізована структура дозволяє легко додавати нові поля до заявки, створювати додаткові подання для аналітики або експортувати записи у форматах CSV та Excel. У майбутньому система може бути доповнена кабінетом користувача, CRM-модулем або інтеграцією з системами обліку виробництва.

Таким чином, розроблений веб застосунок повністю відповідає поставленим функціональним вимогам, демонструючи практичну ефективність використання no-code та low-code технологій для вирішення прикладних завдань малого та середнього бізнесу.

### **Висновки до розділу 2**

У другому розділі дипломної роботи було розглянуто та реалізовано повний цикл створення веб застосунку для обробки замовлень на виготовлення деталей для металообробного підприємства із застосуванням no-code технологій. Визначено технічне завдання, яке відображає потреби обох типів користувачів системи — замовника та адміністратора, а також сформовано вимоги до структури даних та основних бізнес-процесів.

На основі порівняльного аналізу платформ було обґрунтовано вибір *Airtable* як основного інструменту реалізації системи. Вона дозволила забезпечити швидке створення форми збору заявок, внутрішньої бази даних, панелі адміністрування, а також налаштувати автоматизацію повідомлень і обробку типових сценаріїв за допомогою вбудованих скриптів.

Окрему увагу приділено проектуванню структури веб застосунку — визначено основні поля, логіку взаємодії, сценарії використання. Реалізовано функціональні компоненти, які відповідають сформульованим вимогам, включаючи інтерфейс для заповнення заявки, панель перегляду й фільтрації, систему зміни статусу та сповіщення клієнтів.

Таким чином, на базі обраної *no-code* платформи створено прототип вебзастосунку, який демонструє можливість швидкого впровадження цифрових рішень у виробничу діяльність малого або середнього підприємства без значних витрат ресурсів. Результати реалізації засвідчили доцільність і практичну ефективність використання *no-code/low-code* технологій у реальних бізнес-задачах.

## РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ РІШЕННЯ

### 3.1. Методика тестування системи

Після реалізації веб застосунку необхідним етапом є тестування його функціональності, стабільності та відповідності технічному завданню. Метою тестування є виявлення можливих помилок, перевірка коректності взаємодії користувачів із системою, а також оцінка зручності та надійності веб інтерфейсу.

Оскільки веб застосунок побудований на платформі Airtable, яка частково обмежує низькорівневий доступ до внутрішніх механізмів, основна увага приділяється функціональному ручному тестуванню, що імітує реальні дії кінцевих користувачів.

Основні напрями тестування:

1. Функціональне тестування форми подачі заявки:
  - перевірка заповнення всіх обов'язкових полів;
  - тестування валідації email та номеру телефону;
  - перевірка можливості завантаження файлів (креслень);
2. Тестування панелі адміністратора:
  - перевірка коректного відображення нових заявок;
  - зміна статусу заявки та її збереження;
  - фільтрація і сортування записів за ключовими полями;
  - редагування замовлення без втрати прикріплених файлів.
3. Перевірка автоматизацій:
  - надсилання повідомлення на електронну пошту після створення заявки;
  - виконання сценаріїв Scripting Extension — чи виконуються за умов, передбачених логікою.

#### 4. Оцінка коректності роботи з різних пристроїв:

- тестування подачі заявки з комп'ютера, планшета та смартфона;
- перевірка адаптивності форми на різних розмірах екранів;
- відсутність збоїв або помилок при використанні мобільного браузера.

#### 5. Тестування надійності та стійкості:

- перевірка поведінки системи при одночасному надходженні кількох заявок;
- тестування реакції на неповні або помилкові дані;
- оцінка стабільності збереження інформації після редагування.

#### Критерії оцінювання:

Для кожного сценарію тестування встановлюється бінарний результат — «успішно» або «несправність виявлено». У разі невідповідності — фіксується причина, очікувана та фактична поведінка системи.

Таким чином, було перевірено функціональності веб застосунку, яка дозволяє виявити слабкі місця системи та забезпечити її готовність до впровадження у виробниче середовище.

### 3.2. Аналіз результатів тестування

Після проведення тестування вебзастосунку за визначеною методикою було здійснено аналіз отриманих результатів з метою перевірки відповідності реалізованої системи функціональним вимогам, зазначеним у технічному завданні.

#### **Робота клієнтської форми подачі заявки.**

Усі ключові функції форми, включаючи заповнення полів, перевірку контактних даних та завантаження файлів, працюють коректно (див. рис. 3.1). При заповненні форми користувач отримує повідомлення про успішну відправку (див. рис. 3.2), а заявка автоматично з'являється у базі даних (див.

рис. 3.3). Жодних збоїв тестування відкриття застосунку з смартфона не зафіксовано (див. рис. 3.4).

### Заявка на виробництво деталей за вашим кресленням

|   |  |
|---|--|
| Опис деталі на замовлення                     | Фрезерування   |
| Матеріал виготовлення                         | (Згідно ДСТУ)<br>Титан ПТ-38   |
| Кількість деталей у партії                    | 425,0  |
| Завантажте ваше креслення деталі              | (У форматі .DWG)<br>Attach file<br><br>Kreslennya_zatvor_AKSU.dwg |
| Ваш номер телефону?                           | +380674947716  |
| Чи потрібна вам консультація нашого інженера? | <input checked="" type="checkbox"/>  |

[Clear form](#) **Submit**

Рисунок 3.1 – тестування форми клієнта

Дякуємо за замовлення, наш менеджер зовсім скоро вам зателефонує!

Рисунок 3.2 – повідомлення про успішну відправку

|   | Опис деталі  | Статус заявки | Дата подачі    | Контакт замовника | Консультація інженера | Матеріал    | # кількість у партії | Креслення у форма... |
|---|--------------|---------------|----------------|-------------------|-----------------------|-------------|----------------------|----------------------|
| 1 | Фрезерування | Нова          | 4.6.2025 21:17 | +380674947716     | ✓                     | Титан ПТ-38 | 425,0                |                      |

Рисунок 3.3 – тестування надходження заявки у базу даних

21:22

airtable.com

Опис деталі на замовлення

Матеріал виготовлення  
(Згідно ДСТУ)

Кількість деталей у партії

Завантажте ваше креслення деталі  
(У форматі .DWG)

Drop files here or [browse](#)

Ваш номер телефону?

Чи потрібна вам консультація нашого інженера?

Clear form Submit

Рисунок 3.4– тестування з смартфона

### 3.3. Оцінка ефективності використання no-code рішень у практичному застосуванні

Використання no-code технологій, зокрема платформи Airtable, у процесі створення вебзастосунку для обробки замовлень дало змогу досягти високого рівня ефективності з мінімальними витратами часу, коштів та технічних ресурсів. У цьому підрозділі представлено узагальнену оцінку переваг та недоліків no-code підходу на основі практичного досвіду реалізації системи.

### **Переваги, виявлені під час реалізації:**

#### 1. Швидкість розробки.

Усі основні компоненти вебзастосунку (форма, база даних, інтерфейс, автоматизація) були створені протягом декількох днів.

#### 2. Низький поріг входу.

Розробка здійснювалась без глибоких знань мов програмування, що дозволяє створювати аналогічні рішення працівникам підприємства після короткого навчання.

#### 3. Інтуїтивно зрозумілий інтерфейс.

Як адміністраторська частина, так і форма подачі заявки є зручними для щоденного використання. Це знижує потребу в інструкціях і технічній підтримці.

#### 4. Гнучкість у зміні структури.

Нові поля, подання, статуси та сценарії легко додаються без необхідності переробки всієї системи.

#### 5. Інтеграція з іншими сервісами.

Airtable підтримує підключення до зовнішніх API, а також інтегрується з Gmail, Telegram, Google Sheets та іншими інструментами без додаткового коду.

тяться з Gmail, Telegram, Google Sheets та іншими інструментами без додаткового коду.

#### 6. Мінімальні фінансові витрати.

Усі функції, необхідні для роботи, реалізовані на безкоштовному або умовно безкоштовному тарифному плані.

### **Обмеження, виявлені під час реалізації:**

#### 1. Залежність від функціоналу платформи.

Деякі сценарії складно реалізувати без підключення зовнішніх сервісів (наприклад, складна аналітика або розгалужена логіка умов).

#### 2. Обмеження на обсяг даних.

У безкоштовній версії Airtable є ліміти на кількість записів та розмір файлів, що може бути критичним при тривалому використанні без оновлення тарифу, але сервіс пропонує вигідні тарифні плани при інтеграції у бізнес.

### **Загальна оцінка ефективності:**

No-code та low-code підходи виявилися оптимальним вибором для створення корпоративного інструменту малого/середнього масштабу. Розроблений веб застосунок повністю задовольнив поставлені функціональні вимоги, не потребує складної технічної підтримки чи значних фінансових витрат. За результатами тестування та практичного використання система показала високу стабільність, доступність і придатність для реального впровадження у виробничий процес.

## **3.4. Перспективи розвитку та удосконалення розробленого веб застосунку**

Розроблений вебзастосунок у поточному вигляді реалізує базові функції для подачі, обробки та адміністрування замовлень на виготовлення деталей. Проте його структура та обрана платформа залишають широкі можливості для подальшого розвитку та масштабування системи відповідно до зростаючих потреб підприємства. У поточній реалізації відсутня реєстрація або ідентифікація користувачів. Додавання механізму авторизації дозволило б створити особисті кабінети для клієнтів, де вони могли б:

- переглядати статус власних заявок;
- завантажувати додаткові файли до вже поданих замовлень;
- переглядати історію співпраці;
- отримувати індивідуальні повідомлення

Серед можливих напрямів удосконалення:

- автоматичне формування PDF-квитанції або технічного листа на основі заявки;

- нагадування адміністраторам про заявки, що не оброблялися понад визначений час;
- надсилання SMS-повідомлень за допомогою зовнішніх API (наприклад, Twilio).

### **Висновки до розділу 3**

У третьому розділі дипломної роботи було проведено тестування функціональних компонентів вебзастосунку, оцінено ефективність його роботи в умовах, наближених до реального використання, а також проаналізовано перспективи розвитку розробленої системи.

Застосована методика ручного функціонального тестування дозволила виявити, що всі ключові функції — подача заявки, обробка записів, зміна статусів, автоматизоване інформування клієнтів — працюють коректно. Система не вимагає складної технічної підтримки, забезпечує зручну взаємодію між замовником і підприємством, а також дозволяє адміністратору ефективно виконувати щоденні завдання.

Результати тестування підтвердили стабільність вебзастосунку, адаптивність інтерфейсу, надійність збереження даних і швидкість обробки запитів. Система позитивно сприймається кінцевими користувачами, не створює додаткового навантаження та готова до практичного впровадження.

Оцінка ефективності використання no-code та low-code технологій продемонструвала переваги цього підходу: високу швидкість розробки, економічність, простоту. Водночас визначено й обмеження — насамперед пов'язані з кастомізацією інтерфейсу та залежністю від можливостей платформи. Також окреслено перспективні напрями розвитку системи.

Таким чином, веб застосунок довів свою практичну цінність як інструмент цифровізації окремих бізнес-процесів на малому або середньому підприємстві у виробничій сфері.

## ВИСНОВКИ

У межах дипломної роботи було досліджено можливості використання технологій low-code та no-code для створення вебзастосунку, призначеного для обробки замовлень на виготовлення деталей для металообробного підприємства. Основною метою дослідження було створення простого, зручного, доступного для користувачів застосунку, який забезпечував би цифровізацію та автоматизацію процесу прийому і опрацювання клієнтських заявок.

У першому розділі роботи було проаналізовано сучасні підходи до розробки бізнес-орієнтованих вебсистем. Проведено огляд обмежень традиційної кодової розробки, а також визначено переваги платформ на основі no-code і low-code технологій. Особливу увагу приділено функціональним та нефункціональним вимогам до системи обробки замовлень у виробничому контексті.

ваних вебсистем. Проведено огляд обмежень традиційної кодової розробки, а також визначено переваги платформ на основі no-code і low-code технологій. Особливу увагу приділено функціональним та нефункціональним вимогам до системи обробки замовлень у виробничому контексті.

У другому розділі здійснено постановку технічного завдання, обґрунтовано вибір платформи Airtable як інструменту реалізації, спроектовано архітектуру та реалізовано функціональні компоненти вебзастосунку. У результаті створено робочий прототип системи, який містить форму подачі заявки, внутрішню панель адміністратора, автоматизовану зміну статусів та механізм сповіщення користувачів.

У третьому розділі було проведено тестування функціоналу вебзастосунку, підтверджено його відповідність поставленим вимогам, а також здійснено оцінку ефективності застосування no-code підходу. Визначено перспективи розвитку розробленої системи, які передбачають

масштабування, інтеграцію з іншими бізнес-сервісами та впровадження додаткових можливостей (кабінет користувача, мобільна версія, аналітика тощо).

У результаті роботи було досягнуто поставлену мету — створено функціональний вебзастосунок, який може бути впроваджений у виробничу діяльність малого або середнього підприємства для оптимізації процесу обробки замовлень. Практична значущість полягає в тому, що розроблене рішення є доступним, гнучким і не потребує значних ресурсів для реалізації та підтримки.

Отримані результати підтверджують доцільність і ефективність використання no-code/low-code підходів у задачах прикладної цифрової трансформації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008:2015 «Звіти у сфері науки і техніки. Структура та правила оформлення».
2. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання.
3. Airtable Documentation URL: <https://airtable.com/developers/web> (дата звернення: 24.04.2025 р.).
4. W3Schools URL: <https://www.w3schools.com> (дата звернення: 25.04.2025 р.).
5. Google Fonts URL: <https://fonts.google.com> (дата звернення: 25.04.2025 р.).
6. Stack Overflow URL: <https://stackoverflow.com> (дата звернення: 01.12.2024 р.).
7. Сучасний підручник з JavaScript URL: <https://uk.javascript.info> (дата звернення: 01.05.2025 р.).
8. DataTables Documentation URL: <https://datatables.net> (дата звернення: 04.04.2025 р.).

ДОДАТКИ