

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методики швидкого створення веб-застосунків для малого та середнього бізнесу»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

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

_____ Віталій ХАЛЕЦЬКИЙ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-62

_____ Віталій ХАЛЕЦЬКИЙ

Керівник: _____ Тимур ДОВЖЕНКО

к.т.н.

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Халецькому Віталію Сергійовичу _____

1. Тема кваліфікаційної роботи: Розробка методики швидкого створення веб-застосунків для малого та середнього бізнесу

керівник кваліфікаційної роботи Тимур ДОВЖЕНКО к.т.н.,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методології розробки, методологія no-code.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1.Дослідження методології no-code.

2.Аналіз технологій впливу no-code на малий і середній бізнес.

3.Розробка платформи створення застосунків no-code для малого та середнього бізнесу.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз існуючих ІТ-рішень.
2. Порівняння підходів до розробки застосунків.
3. Алгоритм створення застосунку на основі платформи.
4. Структура no-code платформи.
5. Структура no-code платформи.
6. Приклад процесу створення веб-застосунку.
7. Результат роботи з no-code платформою.
8. Результати впровадження платформи і вплив на показники компанії.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вивчення матеріалів для аналізу методології розробки, no-code	06.11-12.11.23	
3	Дослідження no-code технологій	13.11-19.11.23	
4	Аналіз особливостей впливу платформ no-code на малий і середній бізнес	20.11-26.11.23	
5	Дослідження технологій no-code платформ	27.11-03.12.23	
6	Розробка методики та платформи для швидкого створення застосунків за допомогою технології no-code та набору готових шаблонів і компонентів для малого і середнього бізнесу	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

_____ (підпис)

Віталій ХАЛЕЦЬКИЙ

Керівник
кваліфікаційної роботи

_____ (підпис)

Тимур ДОВЖЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 64 стор., 2 табл., 7 рис., 28 джерел.

Мета роботи: пришвидшення процесу створення веб-застосунків для малого та середнього бізнесу за рахунок використання технології no-code розробки.

Об'єкт дослідження: процес створення веб-застосунків.

Предмет дослідження: методи та технології створення веб-застосунків.

Короткий зміст роботи: У роботі проведено дослідження методології розробки застосунків за допомогою no-code. Проаналізовано основні принципи роботи з no-code платформами. Проаналізовано як за допомогою no-code технологій пришвидшити процес створення веб-застосунків для малого та середнього бізнесу. Було розроблено методику для швидкого створення застосунків за допомогою технології no-code та набору готових шаблонів і компонентів для малого і середнього бізнесу. На основі розробленої методики було написано програмну реалізацію no-code платформи. Було проведено дослідження впливу використання платформи на фінансові показники бізнесу.

КЛЮЧОВІ СЛОВА: NO-CODE, МАЛИЙ ТА СЕРЕДНІЙ БІЗНЕС, МЕТОДОЛОГІЇ РОЗРОБКИ, СТВОРЕННЯ ЗАСТОСУНКІВ.

ABSTRACT

Text part of the master's qualification work: 64 pages, 7 pictures, 2 table, 28 sources.

The purpose of the work – Accelerate the process of creating web applications for small and medium-sized businesses through the use of no-code development technology.

Object of research – The process of creating web applications

Subject of research – Methods and technologies for creating web applications

Summary of the work: The master thesis explores no-code technology, the fundamental principles of no-code development, analyzes the impact of no-code technology on business processes, investigates challenges and limitations of no-code development, and provides a rationale for choosing no-code for small and medium-sized enterprises.

The traditional approach to web application development and the no-code methodology have been analyzed and compared. The advantages and disadvantages of both approaches, including development speed, resource expenditures, editing capabilities, and flexibility, have been identified. Further investigation includes an examination of the impact of no-code technology on the financial metrics of businesses, particularly in reducing costs and increasing the efficiency of the application development process.

The analysis includes an examination of the impact of no-code technology on the speed of the development process, its influence on business resource costs in application development, and the effect of leveraging ready-made functional blocks on the speed of product development.

A developed methodology for the rapid creation of applications is described, based on no-code technology and the utilization of ready-made templates and functional blocks.

Ready-made templates and functional blocks developed for use in the no-code platform are described.

The master thesis also outlines the developed platform and its architecture based on the methodology for the rapid creation of applications. The main components of the developed platform are analyzed, including its architecture, interface, and various functional capabilities. A detailed overview of the tools for creating, editing, and customizing web applications through the no-code methodology is provided.

The utilization of the developed methodology and platform is discussed, exemplified by the creation of a web application using no-code technologies. An evaluation of the results of applying the developed methodology, especially in terms of reducing time and costs for creating web applications, is conducted.

As a result of the conducted research, a reduction in business resource costs in application development is demonstrated, specifically a decrease in time and financial expenditures by more than 45% after implementing the platform into the business.

KEYWORDS: NO-CODE, SMALL AND MEDIUM-SIZED BUSINESSES, DEVELOPMENT METHODOLOGIES, APPLICATION DEVELOPMENT.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. ОГЛЯД МЕТОДОЛОГІЇ NO-CODE.....	13
1.1 Вступ до no-code розробки.....	13
1.2 Основні принципи no-code розробки.....	14
1.3 Вплив no-code на бізнес-процеси	17
1.4 Виклики та обмеження no-code розробки	19
1.5 Обґрунтування вибору no-code для малих та середніх підприємств.....	21
РОЗДІЛ 2. РОЗРОБКА МЕТОДИКИ ШВИДКОГО СТВОРЕННЯ ПРОДУКТІВ ДЛЯ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ NO-CODE	26
2.1 Аналіз впливу no-code на швидкість процесу розробки.....	26
2.2 Зменшення затрат ресурсів при розробці.....	27
2.3 Готові функціональні блоки і їх перевикористання як ключ до швидкої розробки продуктів	28
2.4 Аналіз і порівняння існуючих no-code платформ.....	30
РОЗДІЛ 3. РОЗРОБКА ПЛАТФОРМИ ДЛЯ ШВИДКОГО СТВОРЕННЯ ПРОДУКТІВ ДЛЯ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ НА ОСНОВІ ТЕХНОЛОГІЙ NO-CODE	36
3.1 Опис використаних програмних засобів	36
3.2 Опис структури проекту.....	39
3.3 Опис інтерфейсу.....	43
ВИСНОВКИ.....	58
ПЕРЕЛІК ПОСИЛАНЬ	59
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	62

ВСТУП

В сучасному інформаційному суспільстві розвиток технологій визначає нові тенденції у функціонуванні бізнесу, зокрема для малих та середніх підприємств. Завдяки стрімкому зростанню використання Інтернету та переходу до цифрової економіки, стає очевидною необхідність у використанні сучасних веб-технологій для підтримки та просування бізнес-процесів малого та середнього бізнесу.

Однією з ключових складових цього технологічного та економічного розвитку є розробка ефективних методик для швидкого створення веб-застосунків, спеціально адаптованих до потреб малого та середнього бізнесу. Дослідження в цій області відіграє критичну роль у забезпеченні доступу малого та середнього бізнесу до інструментів, які сприяють автоматизації бізнес-процесів, підвищенню ефективності та забезпеченню конкурентоспроможності на ринку.

Мета роботи: оптимізація процесу створення веб-застосунків для малого та середнього бізнесу за рахунок використання технології no-code розробки.

Для досягнення поставленої мети в роботі вирішено наступні завдання:

1. Дослідити технологію no-code, її вплив на швидкість розробки застосунків, та можливість використання технології бізнесом задля оптимізації використання ресурсів.

2. Проаналізувати існуючі рішення що надають можливість використовувати технологію no-code для створення застосунків.

3. Розробити методику швидкого створення застосунків за допомогою використання технології no-code та набору готових функціональних блоків і шаблонів.

4. Провести дослідження впливу на фінансові показники бізнесу після впровадження методики і використання платформи no-code для створення застосунків.

Об'єкт дослідження: процес створення веб-застосунків.

Предмет дослідження: методи та технології створення веб-застосунків.

Методи дослідження: методи формалізації; методи емпіричного дослідження (спостереження, порівняння, вимірювання, експеримент); аналіз та синтез; методи аналізу вимог до програмного забезпечення; методи розробки веб-застосунків.

Дипломна робота присвячена розробці методики швидкого створення веб-застосунків для малого та середнього бізнесу з використанням сучасних технологій програмування та веб-дизайну. У процесі дослідження було приділено увагу аналізу потреб та вимог бізнес-середовища малого та середнього бізнесу, а також визначено оптимальні інструменти та підходи до розробки веб-застосунків, що враховують специфіку та обмеження цього сегменту ринку. Результати даного дослідження сприяли покращенню доступності та використання інформаційних технологій в секторі малого та середнього бізнесу, сприяючи його подальшому розвитку та утвердженню на ринку.

Нова генерація методик розробки веб-застосунків включає в себе сучасний підхід, що став широко відомий як "no-code" чи "low-code" розробка. Цей підхід базується на використанні графічних інтерфейсів та конфігураційних налаштувань, що дозволяють створювати веб-застосунки без програмування або з мінімальним рівнем кодування.

У контексті розробки веб-застосунків для малого та середнього бізнесу, використання no-code підходу є важливою складовою нової методики. Це дозволяє підприємствам швидко створювати та налаштовувати свої власні веб-застосунки без значних витрат на програмістське вирішення завдань. No-code розробка робить технології доступними для більшого кола користувачів, навіть тих, хто не має глибоких знань у сфері програмування.

В дипломній роботі особлива увага була приділена вивченню та аналізу no-code рішень, їх можливостей та обмежень у контексті розробки веб-застосунків для малого та середнього бізнесу. Впровадження цих інструментів дозволить бізнес-користувачам самостійно створювати та модифікувати веб-застосунки, прискорюючи процес впровадження та знижуючи витрати на розробку, що, в свою чергу, сприятиме підвищенню конкурентоспроможності та розвитку малих та середніх підприємств.

Апробація результатів та публікації. Робота пройшла апробацію на науково-практичних конференціях:

1. Телекомунікаційні та інформаційні технології. «Оптимізація створення веб-застосунків для малого та середнього бізнесу на основі технології no-code розробки». №1, 2024.

2. Проблеми комп'ютерної інженерії. «Розробка методики швидкого створення веб-застосунків для малого та середнього бізнесу». Київ, ДУІКТ, 2022.

За результатами апробації опубліковано тези доповідей [27] та [28].

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 ОГЛЯД МЕТОДОЛОГІЇ NO-CODE

1.1 Вступ до no-code розробки

В останні роки спостерігається радикальна трансформація у сфері розробки програмного забезпечення завдяки широкому впровадженню концепції "No-code"[14]. Ця методологія відзначається відсутністю необхідності в глибоких знаннях програмування, дозволяючи користувачам створювати веб-застосунки за допомогою графічних інтерфейсів та конфігураційних налаштувань. Огляд цієї методології відіграє важливу роль у розумінні та оцінці її потенціалу для використання в сучасному бізнес-середовищі [8]. В контексті стрімкого розвитку технологій та поширення цифрової трансформації, no-code розробка представляє собою інноваційний підхід до створення програмного забезпечення, який перевертає звичайні уявлення про розробку за допомогою коду. Основним принципом no-code є відмова від потреби в глибоких знаннях програмування, відкриваючи двері для тих, хто раніше відчував бар'єри у власних можливостях створення власних веб-застосунків [14].

Одним із ключових елементів в no-code розробці є використання графічних інтерфейсів та інтуїтивно зрозумілих інструментів, які дозволяють користувачам створювати дизайн та налаштовувати функціонал веб-застосунків без написання коду. Цей перехід від традиційного програмування до no-code розробки великою мірою диктується необхідністю швидкого впровадження інновацій та відповіді на зростаючі вимоги бізнес-середовища [16].

No-code інструменти створюють унікальну можливість для бізнес-користувачів брати активну участь у процесі розробки та модифікації веб-застосунків. Це важливо не лише для зниження трудовитрат на розробку, але і для створення більш гнучких та індивідуалізованих рішень, відповідних унікальним потребам кожного конкретного підприємства [16].

Зокрема, no-code розробка розвивається в напрямку "візуального програмування", де користувачі мають можливість створювати логіку та функціонал за допомогою перетягування та з'єднання графічних елементів. Це надає більше простору для творчості та інновацій, зменшуючи одночасно бар'єри для тих, хто раніше схильний був відмовлятися від розробки через відсутність технічних навичок.[18]

У цьому контексті, огляд no-code розробки впирається в ідею демократизації технологій, роблячи їх доступними для широкого кола користувачів, що може значно вплинути на підвищення конкурентоспроможності та інноваційності бізнесу в умовах сучасного цифрового середовища.

1.2 Основні принципи no-code розробки

В основі no-code розробки лежать ключові принципи, які забезпечують високий рівень доступності та простоти у створенні веб-застосунків без програмування. Розглянемо основні принципи, що визначають цей інноваційний підхід:[14]

- Графічний інтерфейс: Однією з головних характеристик no-code розробки є зосередження на графічному інтерфейсі. Користувачі працюють з візуальними елементами, такими як блоки, кнопки та поля, і перетягують їх, щоб створити логіку свого веб-застосунку. Це робить процес розробки більш інтуїтивним та легким для розуміння, що зменшує бар'єри для тих, хто не має глибоких технічних знань.[14]

- Конфігураційні налаштування: No-code розробка дозволяє користувачам налаштовувати параметри та функції за допомогою конфігураційних налаштувань, які визначаються через інтерфейс. Замість кодування, користувачі можуть визначити, які дії повинні виконуватися при різних умовах та взаємодіяти зі своїм застосунком, просто конфігуруючи його параметри.[14]

- Відсутність програмування: Одним з ключових принципів no-code розробки є відмова від необхідності в програмуванні. Це означає, що користувачі не повинні

знати або використовувати кодові рядки для створення та модифікації своїх веб-застосунків. Замість цього, їм пропонуються інструменти та опції, які дозволяють виконувати необхідні завдання за допомогою графічного взаємодії та конфігурації.[14]

- Візуальне програмування: Сучасні no-code платформи нерідко нагадують «візуальне програмування», де користувачі можуть створювати логіку та функціонал, перетягуючи та з'єднуючи візуальні блоки. Це робить розробку більш доступною та інтерактивною, дозволяючи відобразити логіку програми в інтуїтивно зрозумілій формі. Оцінюючи ці принципи, важливо враховувати, як вони спільно працюють, створюючи ефективний інструмент для швидкої та простої розробки веб-застосунків для бізнесу. Це особливо актуально в умовах, коли швидкість впровадження та гнучкість реагування на зміни стають ключовими факторами успіху в сучасному бізнесі.[16]

- Широкий спектр готових компонентів: В no-code розробці важливою особливістю є наявність різноманітних готових компонентів, які можна використовувати для швидкої побудови функціональності. Ці компоненти включають в себе готові модулі для роботи з базами даних, обробки форм, відправлення електронних листів та інші. Вони є невід'ємною частиною екосистеми no-code і дозволяють розробникам зосередитися на бізнес-логіці та потребах клієнта, замість того, щоб витратити час на написання коду з нуля.[18]

- Інтеграція зі сторонніми сервісами: No-code платформи активно розвивають можливості інтеграції з різноманітними сторонніми сервісами та API. Це дає змогу створювати веб-застосунки, які взаємодіють із зовнішніми системами, використовуючи готові засоби з'єднання та обміну даними. За допомогою інтуїтивних інтерфейсів та конфігурацій, користувачі можуть легко налаштувати взаємодію свого застосунку з різними сервісами без необхідності глибоких знань в області програмування.[18]

- Підтримка ефективного тестування: Принципи no-code розробки сприяють також ефективному тестуванню створених застосунків. Завдяки графічному інтерфейсу та конфігураційним налаштуванням, тестувальники та розробники

можуть легко створювати тестові сценарії, перевіряти різні варіанти взаємодії та переконатися у правильності роботи застосунку. Це сприяє виявленню та усуненню помилок на ранніх етапах розробки, що веде до підвищення якості продукту.[18]

- Швидкість впровадження та адаптації: Завдяки вищезазначеним принципам, no-code розробка забезпечує швидке впровадження та гнучкість в реагуванні на зміни. Користувачі можуть швидко модифікувати свої веб-застосунки, вносячи зміни в конфігурації та додаючи нові функціональності без потреби в складних програмістських перетвореннях. Це особливо важливо в умовах швидкозмінюючогося ринку, де важливо оперативно адаптуватися до нових умов та вимог клієнтів.[14]

- Сприяння творчості та інновацій: No-code розробка створює умови для більшого прояву творчості серед розробників та бізнес-аналітиків. Графічний інтерфейс та візуальне програмування дозволяють втілювати ідеї в життя без глибоких знань програмування. Це сприяє появі новаторських рішень та швидкому експериментуванню з концепціями без значних витрат часу на написання коду.[16]

- Спільна робота та командна розробка: No-code платформи активно підтримують можливості спільної роботи та командної розробки. Кілька користувачів можуть одночасно працювати над одним проектом, вносячи зміни та спільно розвиваючи функціональність. Це полегшує комунікацію в команді та сприяє швидкому об'єднанню зусиль для досягнення спільних цілей.[16]

- Підтримка масштабованості: No-code розробка може стати ефективним інструментом для масштабування бізнес-процесів та розширення функціональності веб-застосунків. Завдяки швидкій реакції на зміни та легкій модифікації, no-code дозволяє адаптувати систему до зростаючих потреб бізнесу без необхідності переписування коду з нуля.[16]

- Зниження витрат на розробку та підтримку: No-code розробка може значно знизити витрати на розробку та підтримку веб-застосунків. Відсутність необхідності в програмуванні та швидкість розвитку дозволяють економити час та гроші. Крім того, легкість тестування та підтримки сприяє зниженню витрат на усунення помилок та підтримання функціональності.

Зазначені характеристики роблять no-code розробку не тільки ефективним інструментом для швидкої реалізації ідей, але і ключовим компонентом в стратегії цифрової трансформації та підтримки інновацій у сучасному бізнесі, важливо врахувати, що no-code розробка не лише спрощує та прискорює процес створення веб-застосунків, але й робить його більш доступним для широкого кола користувачів з різним рівнем технічної підготовки.

1.3 Вплив no-code на бізнес-процеси

No-code розробка впливає на бізнес-процеси у багатогранний спосіб, революціонізуючи підхід до створення веб-застосунків та відкриваючи нові можливості для підприємств:

- Швидкість розробки: Швидкість, з якою можна створювати та впроваджувати веб-застосунки за допомогою no-code підходу, є великою перевагою для бізнесу. Замість проведення довгих циклів розробки, команди можуть оперативно реалізовувати нові ідеї та відгукуватися на зміни в ринковому середовищі. Це особливо важливо в умовах швидкої динаміки сучасного бізнесу.

- Ефективність витрат: При використанні no-code розробки відпадає потреба у великому штаті програмістів та ІТ-фахівців. Бізнес-користувачі можуть використовувати інтерфейси no-code для створення та модифікації власних веб-застосунків, що водночас знижує витрати на технічний персонал і робить розробку більш доступною для широкого кола співробітників.[14]

- Гнучкість та адаптивність: No-code дозволяє бізнес-користувачам легко змінювати та адаптувати свої веб-застосунки без глибоких знань в програмуванні. Це дозволяє компаніям швидко реагувати на зміни у власних процесах або в зовнішньому середовищі, підтримуючи гнучкість та ефективність у вирішенні завдань.[16]

- Легкість використання та доступність: Завдяки простим інтерфейсам та інтуїтивно зрозумілим інструментам, no-code розробка робить технології доступнішими для всіх користувачів, навіть тих, хто не має технічної освіти. Це

важливо для забезпечення широкого застосування no-code внутрішньо в організації, залучаючи різні підрозділи та команди до процесу розробки.

- Сприяння інноваціям: No-code розробка відкриває двері для внутрішньої інноваційної активності. Коли працівники можуть самостійно експериментувати з власними ідеями та втілювати їх у веб-застосунках, це створює синергію інновацій на всіх рівнях організації. Легка доступність no-code інструментів забезпечує розвиток ідей в реальні проекти без значних затримок.[16]

- Забезпечення документацією: Використання no-code дозволяє підприємствам створювати власні інноваційні рішення, не обтяжуючи IT-відділ лише рутинними та повсякденними завданнями. Це дає IT-спеціалістам можливість фокусуватися на більш стратегічних та складних завданнях, сприяючи підвищенню ефективності використання технічних ресурсів та розкриттю їхнього потенціалу в області інновацій.

- Підтримка цифрової трансформації: No-code розробка виступає ключовим інструментом для компаній, які прагнуть до цифрової трансформації своїх бізнес-процесів. Швидкість впровадження нових технологій та можливість оперативно адаптувати рішення до змін у бізнес-середовищі сприяє покращенню ефективності, оптимізації внутрішніх процесів та забезпеченню конкурентних переваг.

- Залучення департаментів без IT-фахівців: No-code розробка вирішує проблему залежності від IT-відділу для реалізації технічних завдань. Різні відділи та команди в організації можуть самостійно розробляти та впроваджувати веб-застосунки без необхідності очікування ресурсів та підтримки від програмістів. Це збільшує самостійність та відповідальність різних функціональних підрозділів.

- Підтримка внутрішнього підприємництва: No-code надає можливість внутрішнім підприємцям та інноваторам прискорювати власні проекти та експерименти, не очікуючи на допомогу IT-відділу. Це сприяє розвитку культури внутрішнього підприємництва та дозволяє компанії ефективно використовувати потенціал талановитих співробітників.

Загалом, no-code розробка не тільки революціонує процес розробки веб-застосунків, але і впливає на всі бізнес-процеси, створюючи нові можливості для

швидкого реагування на зміни, забезпечення ефективності та стимулювання інновацій. Підприємства, які використовують no-code підходи, можуть отримати конкурентну перевагу в умовах швидкозмінюючогося бізнес-середовища. Результатом є покращена адаптивність, зниження витрат та збільшення швидкості реакції на внутрішні та зовнішні виклики.

1.4 Виклики та обмеження no-code розробки

Незважаючи на величезний потенціал та переваги, які no-code розробка приносить у світ бізнесу, важливо розглянути й виклики, що можуть виникнути при використанні цього підходу, а також обмеження, які необхідно враховувати при плануванні та розробці проектів.

- Обмежені можливості для складних алгоритмів: Однією з основних обмежень no-code розробки є відносно обмежені можливості для реалізації складних алгоритмів. Основні no-code інструменти зазвичай створюються для вирішення стандартних завдань, і вони можуть не вистачати для проектів, де потрібно високорівневе програмування та математичні операції.

- Заборона доступу до низькорівневих налаштувань: No-code розробка зазвичай ґрунтується на високому рівні абстракції, що означає відсутність доступу до низькорівневих кодових налаштувань. Це може стати викликом для досвідчених розробників, які звикли максимально контролювати кожний аспект своїх проектів та вибирати оптимальні рішення.

- Потреба у вдосконаленні засобів контролю версій: Засоби контролю версій для no-code проектів не завжди так розвинені, як у традиційних програмних продуктах. Це може становити виклик у великих командних проектах, де необхідно ефективно відстежувати та керувати змінами.

- Спрощені засоби тестування: No-code інструменти можуть мати обмежені засоби для валідації та тестування, що може ускладнити процес забезпечення якості програмного продукту. Це особливо актуально в проектах з великою кількістю складних взаємодій та логічних переходів.

- Віддалена підтримка та розвиток: Застосунки, розроблені за допомогою no-code підходу, можуть стикатися з викликами в плані підтримки та розвитку, оскільки їх створення часто пов'язане з конкретними no-code платформами. Це може виникнути проблеми, якщо платформа зазнає змін або стає неактуальною в майбутньому.

- Підготовка та навчання користувачів: Хоча і no-code розробка спрощує процес створення веб-застосунків, вона також вимагає від користувачів вивчення нових інтерфейсів та парадигм роботи. Це може вимагати додаткових зусиль у підготовці персоналу та забезпеченні ефективного використання no-code інструментів.

- Відсутність повноцінного програмування: Однією з важливих обмежень no-code розробки є відсутність можливості використання повноцінного програмування. Деякі складні завдання та специфічні функціональності можуть вимагати розширеного програмування, яке no-code інструменти не завжди можуть надати. Це може обмежувати можливості проектів, де потрібно враховувати унікальні бізнес-вимоги.

- Питання безпеки: Використання no-code інструментів може викликати питання безпеки, оскільки деякі з них можуть не надавати високого рівня захисту даних. Розробники повинні уважно вивчати можливості безпеки no-code платформ та вживати додаткові заходи для захисту конфіденційної інформації.

- Залежність від постачальників: Багато no-code рішень пов'язані з конкретними постачальниками платформ. Це створює ризик залежності від одного постачальника та може ускладнити перехід до іншого рішення в майбутньому.

- Проблеми інтеграції: Інтеграція no-code розроблених застосунків з іншими системами може бути викликом, оскільки не всі платформи мають широкі можливості для взаємодії зі сторонніми сервісами. Це може стати проблемою для компаній, які потребують синхронізації даних або інтеграції з існуючими ІТ-інфраструктурами.

- Можливість "тіньового ІТ": No-code розробка може призводити до ситуації, коли різні команди та відділи створюють свої власні застосунки без відома та

контролю IT-відділу. Це може призвести до виникнення "тіньового IT" та ускладнити управління та підтримку цих застосунків.

- Проблеми масштабованості: No-code розробка може бути менше ефективною для великих та складних проектів. Відсутність докладної архітектури та контролю може призвести до проблем з масштабованістю проектів у разі їхнього зростання.

Хоча врахування цих викликів є важливим етапом в плануванні проектів з використанням no-code розробки, вони не зменшують загальний потенціал цього підходу для прискорення та оптимізації процесів створення веб-застосунків у бізнес-середовищі.

1.5 Обґрунтування вибору no-code для малих та середніх підприємств

No-code розробка стає ключовим аспектом сучасної технологічної еволюції, вносячи значний вклад у спосіб, яким бізнес-середовище створює та впроваджує програмне забезпечення. Розглянемо перспективи розвитку no-code розробки та її вплив на бізнес-середовище:

- Зростання функціональності: Однією з основних перспектив розвитку no-code розробки є постійне зростання функціональності інструментів. Розробники активно працюють над додаванням нових можливостей, що дозволить більш детально та глибоко розробляти складні веб-застосунки, а також впроваджувати більше розширень та інтеграцій.

- Розширення екосистеми no-code: Зростання популярності no-code розробки сприятиме розширенню її екосистеми. З'являться нові інструменти, бібліотеки та сервіси, які спростять і розширять можливості розробників та бізнес-користувачів. Це також може включати розвиток спеціалізованих рішень для різних галузей та видів діяльності.

- Ефективність витрат ресурсів: Малим та середнім підприємствам часто важко конкурувати з великими корпораціями у витраті ресурсів на IT-проекти. Використання no-code дозволяє малому та середньому бізнесу знижувати витрати

на програмістів та скорочувати час розробки, що особливо важливо в умовах обмежених бюджетів.

- Легкість впровадження та масштабованість: No-code розробка дозволяє малому та середньому бізнесу швидко створювати та впроваджувати веб-застосунки без значних зусиль. Це особливо важливо для невеликих команд та підприємств, які можуть вирішувати конкретні завдання та швидко адаптуватися до змін в умовах ринку.

- Відсутність потреби у великій ІТ-команді: Багато малих та середніх бізнесів можуть не мати можливості залучати велику кількість ІТ-спеціалістів для розробки та підтримки веб-застосунків. No-code дозволяє бізнес-користувачам самостійно створювати та підтримувати свої проекти, зменшуючи залежність від зовнішніх ІТ-ресурсів.

- Швидка адаптація до змін: Для малого та середнього бізнесу, які часто працюють в динамічних умовах, важливо мати можливість швидко реагувати на зміни в ринкових умовах. No-code розробка дозволяє здійснювати швидкі зміни та впроваджувати нові функції без значного затримання.

- Візуальна модель розробки для непрограмістів: Малим та середнім підприємствам може бути важко знайти та зберегти висококваліфікованих програмістів. No-code розробка, яка базується на візуальних елементах та конфігурації, робить розробку доступною для тих, хто не має глибоких технічних знань.

- Зростання популярності no-code В Галузях: Перспективи no-code розробки визначаються не лише її технічними можливостями, але й розумінням та прийняттям її в бізнес-середовищі. З часом, ймовірно, no-code розробка стане широко використовуваною в галузях, де традиційно високий рівень технічних знань не є обов'язковим.

- Інтеграція з іншими технологіями: Перспективи no-code розробки також пов'язані з інтеграцією з іншими технологіями, такими як штучний інтелект (ШІ), блокчейн та Інтернет речей (IoT). Використання no-code для створення розумних

та інтегрованих систем може сприяти ще ширшому застосуванню цього підходу в різних галузях.

- Навчання та підтримка в організаціях: Передбачається, що компанії будуть звертати більше уваги на навчання свого персоналу щодо no-code розробки. Розширення використання цього підходу вимагатиме ефективної системи підтримки та навчання для працівників різних рівнів компетентності.

- Розвиток спільноти користувачів: Створення активної спільноти користувачів no-code розробки сприятиме обміну досвідом та розробкою кращих практик. Це може також призвести до створення стандартів та конвенцій для no-code розробки, що полегшить взаємодію між різними інструментами.

- Експансія на міжнародні ринки: No-code розробка відкриває можливості для малих та середніх підприємств не лише на внутрішньому ринку, але й для експорту продуктів та послуг. Легкість створення та модифікації веб-застосунків дозволяє адаптувати їх під потреби різних аудиторій та культур.

- Підвищення ефективності бізнес-процесів: No-code розробка може сприяти оптимізації бізнес-процесів внутрішньої діяльності підприємства. Автоматизація завдань та створення індивідуальних рішень допомагають підвищити ефективність та знизити часові та фінансові витрати.

- Створення діджитал-продуктів: No-code розробка може послужити стимулом для створення нових цифрових продуктів та послуг, що відповідають потребам сучасного споживача. Можливість експериментування та швидкого внесення змін дозволяє бізнесу активно реагувати на вимоги ринку.

- Залучення та ретенція талановитих співробітників: Сучасні фахівці цінують можливість брати участь у розробці без глибоких технічних знань. No-code стає додатковим інструментом для команд різного складу, що сприяє розширенню кола талановитих працівників та підвищенню рівня командної співпраці.

- Поширення використання в освіті: No-code розробка може знайти своє застосування у сфері освіти, де вона стане інструментом для навчання основам програмування та розвитку творчих навичок. Застосування no-code в освіті може сприяти створенню інноваційних методик та форматів навчання.

При порівнянні використання no-code та традиційної розробки, можемо виділити наступні ключові властивості для порівняння (табл.1.1).

Таблиця 1.1

Порівняльна таблиця підходів

Критерій	No-Code	Традиційна Розробка
Швидкість та простота	Швидко створює прототипи та застосунки без великого обсягу коду.	Вимагає докладної кодування та тестування, що часто призводить до тривалих термінів розробки.
Вартість та ефективність	Знижує витрати на програмістів, оскільки не вимагає глибоких технічних навичок.	Високі витрати на програмістів та інфраструктуру для розробки.
Доступність для нетехнічних користувачів	Дозволяє нетехнічним фахівцям та бізнес-аналітикам створювати застосунки без кодування.	Вимагає глибокого розуміння програмування, обмежуючи доступність для нетехнічних користувачів.
Гнучкість та розширені можливості	Годиться для швидкого розгортання простих застосунків, але може обмежувати гнучкість у складних проектах.	Надає повну гнучкість та контроль, ідеально підходить для складних та масштабних проектів.
Масштабованість	Може стикатися з обмеженнями масштабування для великих та складних систем.	Забезпечує можливість масштабування та розвитку великих корпоративних систем.

Порівняльна таблиця підходів

Критерій	No-Code	Традиційна Розробка
Керованість та контроль	Забезпечує простий інтерфейс, але може обмежувати контроль над деякими аспектами розробки.	Надає повний контроль та керованість, але вимагає глибоких технічних знань.
Інтеграція та зовнішні системи	Забезпечує інтеграцію з деякими сторонніми сервісами, але може бути обмеженим у випадках складних інтеграцій.	Забезпечує глибоку інтеграцію з будь-якими зовнішніми системами та сервісами.

2 РОЗРОБКА МЕТОДИКИ ШВИДКОГО СТВОРЕННЯ ПРОДУКТІВ ДЛЯ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ NO-CODE

Впровадження no-code підходу у розробку продуктів для малих і середніх бізнесів суттєво змінює динаміку цієї галузі. За допомогою математичних формул можна визначити, як no-code впливає на швидкість розробки, затрати на ІТ-персонал, інноваційність продуктів.

2.1 Аналіз впливу no-code на швидкість процесу розробки

Швидкість розробки в сфері інформаційних технологій визначається як час, необхідний для створення та впровадження нового програмного продукту. Введення no-code підходу в розробку для малих і середніх бізнесів вносить значні зміни у цей аспект.

При класичному сценарію без No-Code наприклад для розробки повноцінного програмного продукту необхідно витратити 100 робочих днів. Кожен модуль або функціональність в середньому потребує 10 робочих днів на написання та тестування коду, тобто:

Час розробки одного модуля (без No-Code) = 10 днів

Загальний час розробки (без No-Code) = 10 модулів × 10 днів/модуль = 100 днів

Введення no-code платформи дозволяє зменшити час розробки кожного модуля до 7 робочих днів, так як відповідає потреба у кодуванні вручну.

Час розробки одного модуля (без No-Code) = 7 днів

Загальний час розробки (без No-Code) = 10 модулів × 7 днів/модуль = 70 днів

Приріст в ефективності ми можемо визначити за допомогою формули (2.1):

$$\text{Приріст в ефективності} = \frac{\text{Час розробки без no-code}}{\text{Час розробки з no-code}} = \frac{100 \text{ днів}}{70 \text{ днів}} \sim 1.43 \quad (2.1)$$

Отже, використання no-code підходу дозволяє зменшити час розробки на приблизно 43%. Це значить, що компанії можуть швидше виводити продукти на ринок, реагувати на зміни в умовах бізнесу та вигравати в гонці конкурентів. Важливо зауважити, що саме цей приріст в ефективності може бути ключовим чинником для малого та середнього бізнесу, де обмежені ресурси та потреба в оперативному введенні нових продуктів на ринок.

2.2 Зменшення затрат ресурсів при розробці

Зниження витрат на IT-персонал стає ключовою перевагою використання no-code платформ у розробці продуктів для малих і середніх бізнесів.

У традиційному підході до розробки програмного забезпечення для МСБ, часто необхідно мати експертний IT-персонал, який відповідає за написання, тестування та підтримку коду. Наприклад, на розробку програмного продукту витрачається \$50,000, з яких витрати ресурсів компанії на написання і тестування коду є 60% цієї суми.

$$\text{Затрати компанії на написання коду} = 0.6 \times 50,000 = 30,000\$$$

Використання no-code платформ дозволяє представникам бізнесу, які не є програмістами, створювати та модифікувати програми. Вартість підтримки та розширення функціоналу за допомогою no-code може бути меншою, оскільки це може робитися без залучення великої кількості ресурсів.

В такому разі затрати компанії на написання коду дорівнюють 0, а витрати на налаштування і створення застосунку в платформі можуть не перевищувати 20%

$$\text{Орієнтовні затрати ресурсів компанії на створення продукту в платформі} = 0.2 \times 50,000 = 10,000\$$$

$$\text{Орієнтовна вартість доступу до платформи з налаштування} = 1500\$$$

Заощадження = Затрати на написання коду – (Затрати на створення продукту в No-Code платформі + Затрати на No-Code платформу (підписка, або одноразова купівля))

$$\text{Заощадження} = 30000 - (10000 + 1500) = 18500\$$$

Отже, використання no-code підходу дозволяє заощадити \$18500 на витратах на написання коду і розширенні функціоналу. Ці гроші можуть бути перерозподілені на інші бізнес-потреби, такі як маркетинг, розвиток, та інше.

2.3 Готові функціональні блоки і їх перевикористання як ключ до швидкої розробки продуктів

В сучасному світі, де технології стрімко розвиваються, використання готових функціональних блоків в розробці продуктів стає важливим чинником для прискорення процесу та зменшення витрат часу та ресурсів. Готові функціональні блоки - це передбачені компоненти або модулі, які можуть бути легко інтегровані в різні проекти без необхідності повторного програмування. Давайте розглянемо цю тему детальніше.

Готові функціональні блоки представляють собою вузли або компоненти з певною функціональністю, які можна легко використовувати у розробці програмного забезпечення. Це може бути все, від блоків інтерфейсу та анімацій до бізнес-логіки та модулів доступу до баз даних.

Переваги готових функціональних блоків:

- Ефективність розробки: Використання готових блоків дозволяє значно прискорити розробку, оскільки розробники можуть уникати написання коду "з нуля".
- Стандартизація: Готові блоки можуть включати в себе стандартизовані рішення, що спрощує і покращує якість розробки.
- Многоразове використання: Однакові блоки можна легко використовувати в різних

Типи готових функціональних блоків:

Інтерфейсні блоки: Елементи дизайну, кнопки, форми, які можна легко включити в інтерфейс додатка чи веб-сайту.

Бізнес-логіка: Блоки, які містять готову функціональність для обробки даних, аналізу, розрахунків тощо.

Доступ до даних: Модулі для взаємодії з базами даних, зберігання та отримання інформації.

Виклики та ризики:

Обмежена гнучкість: Деякі готові блоки можуть бути обмеженими у внесенні змін або не відповідати точним потребам проекту.

Сумісність: При використанні блоків розробники повинні переконатися, що вони сумісні з іншими компонентами проекту.

Як приклад застосування розглянемо сценарій розробки електронного магазину. Замість того, щоб програмувати корзину покупок та оплату "з нуля", розробники можуть використовувати готові функціональні блоки для створення цих елементів, що дозволяє ефективно використовувати час та ресурси.

В епоху, коли технології стрімко розвиваються, використання готових функціональних блоків та no-code підходів стає критично важливим для ефективної розробки продуктів без глибоких знань програмування. Готові функціональні блоки представляють собою передбачені компоненти, які можна інтегрувати безпосередньо в різні проекти, дозволяючи розробникам уникнути великої частини рутинної роботи.

No-code, в свою чергу, спрощує процес розробки, дозволяючи користувачам створювати програмне забезпечення шляхом конфігурації та з'єднання готових елементів, замість написання коду "з нуля". Такий підхід особливо ефективний для тих, хто не є програмістами, проте має ідеї щодо створення продуктів чи застосунків.

Готові функціональні блоки та no-code інструменти разом створюють потужний союз для прискорення розробки. Невеликий бізнес, підприємці та творчі люди тепер можуть втілювати свої ідеї в життя, ефективно використовуючи існуючі компоненти та інструменти без значних витрат часу та ресурсів. Це дозволяє покращити темпи розробки, знизити вартість проектів та створює нові можливості для творчості та інновацій.

Перевикористання готових блоків також допомагає вдосконалювати їхню надійність. Коли функціональні блоки використовуються в різних проектах,

помилки та недоліки виявляються та виправляються швидше, що підвищує загальну якість продуктів.

Загалом, поєднання готових функціональних блоків та no-code може стати справжньою революцією у створенні продуктів. Це прискорює інновації та надає можливість широкому колу людей вносити свій вклад у технологічний прогрес.

В цілому, використання готових функціональних блоків та no-code інструментів сприяє не тільки ефективній розробці продуктів, але й відкриває нові горизонти для творчості та спільного вдосконалення програмного забезпечення.

2.4 Аналіз і порівняння існуючих no-code платформ

У сучасному інформаційному суспільстві, де необхідність швидкого створення та впровадження програмного забезпечення набуває величезного значення, no-code платформи стають ключовим інструментом для розробників та бізнес-аналітиків. Розглянемо докладно три з найпопулярніших no-code платформ: Bubble[26], Webflow[24] та Glide[25].

2.4.1 Bubble

Bubble[26] - це no-code платформа, яка дозволяє створювати веб-застосунки без необхідності писати код. Головною метою є полегшення розробки для тих, хто не є професійним програмістом, але має ідеї для створення програмного забезпечення.

Bubble була заснована у 2012 році Джошуа Епштейном та Емметтом Гриеном. З того часу платформа швидко здобула популярність серед розробників та бізнес-користувачів.

Bubble укладає партнерські угоди та постійно розвиває інтеграції з іншими популярними сервісами, що полегшує роботу користувачів та забезпечує додаткові можливості.

Bubble надає можливість розробникам використовувати власний код та робити власні API виклики, розширюючи можливості платформи для реалізації специфічних завдань.

Bubble часто використовується для розробки бізнес-застосунків, CRM систем, інтерактивних веб-сайтів та інших проектів, які вимагають розширених функціональних можливостей.

Основні властивості:

- Легкість використання: Bubble вирізняється своїм інтуїтивним інтерфейсом та "drag-and-drop" можливостями, що дозволяють навіть тим, хто не має глибоких знань програмування, швидко створювати веб-застосунки.

- Функціональність: Платформа пропонує широкий спектр функціональних можливостей, включаючи роботу з базами даних, автоматизацію бізнес-процесів та можливості інтеграції.

- Гнучкість: Bubble є високогнучкою платформою, придатною для створення різноманітних веб-застосунків, від простих сайтів до складних корпоративних систем.

- Інтеграція: Підтримує різноманітні інтеграції за допомогою плагінів.

2.4.2 Webflow

Webflow[24] ставить свою увагу на створенні веб-застосунків та сайтів з високоякісним та естетичним дизайном. Це інструмент для дизайнерів та розробників, які шукають зручні інструменти для роботи з веб-дизайном.

Webflow була створена у 2013 році Влодзімежем Заборовські та була випущена у 2015 році. Платформа швидко набула популярності, залучаючи веб-дизайнерів та розробників.

Однією з ключових особливостей Webflow є його візуальний редактор, що дозволяє користувачам розглядати та редагувати сторінки у режимі реального часу. Це полегшує процес роботи з дизайном та взаємодії.

Платформа має розширені можливості для розробки та налаштування електронної комерції, дозволяючи створювати стильні та функціональні онлайн-магазини.

Webflow часто використовується для створення стильних веб-сайтів, лендінгів та проєктів, які вимагають естетичного підходу до дизайну.

Основні властивості:

- Легкість використання: Webflow спрямована на дизайнерів і фронтенд-розробників, пропонуючи інтуїтивний інтерфейс для розробки веб-сайтів та веб-застосунків.

- Функціональність: Зосереджена на створенні стильних та адаптивних веб-сайтів з розширеними можливостями роботи з дизайном.

- Гнучкість: Найбільше підходить для дизайнерських проєктів, але може виявитися менш гнучкою для розробки складних бізнес-застосунків.

- Інтеграція: Має вбудовані інструменти для інтеграції з іншими сервісами.

2.4.3 Glide

Glide[25] - це no-code платформа, спеціалізована на швидкій розробці мобільних застосунків, які базуються на Google Sheets. Головний акцент робиться на легкості та швидкості розробки.

Glide була створена в 2018 році Джейсоном Шалканом, який вирішив спростити процес створення мобільних застосунків для широкого кола користувачів, особливо для тих, хто використовує Google Sheets.

Основна фокусна точка Glide - це створення мобільних застосунків без необхідності написання коду. Вона дозволяє швидко розгортати прості застосунки для збору даних.

Glide вражає швидкістю впровадження змін та оновлень. Внесення змін в Google Sheets автоматично відображається в мобільному застосунку без необхідності перекомпіляції.

Додаткову функціональність можна додавати за допомогою готових шаблонів Glide, які пропонують різноманітні можливості та стилі для застосунків.

Glide використовується для створення простих мобільних застосунків, таких як опитування, списки завдань, та інші застосунки, де основний фокус на зборі та відображенні даних.

Основні властивості:

- Легкість використання: Glide спеціалізується на швидкому створенні мобільних застосунків на базі Google Sheets, надаючи простий та доступний інтерфейс.

- Функціональність: Орієнтована на мобільні застосунки, спрощуючи створення застосунків для збору даних та роботи зі списками.

- Гнучкість: Добре підходить для швидкого створення простих мобільних застосунків, але може бути обмеженою для складніших завдань.

- Інтеграція: Орієнтована на інтеграцію з Google Sheets, але також має деякі інтеграційні можливості.

В таблиці 2.1 наведено порівняльний аналіз розглянутих платформ.

Таблиця 2.1

Порівняльний аналіз платформ

Критерій	Bubble	Webflow	Glide
Легкість Використання	Висока	Висока	Висока
Функціональність	Висока (Бізнес-застосунки)	Середня (Веб-сайти)	Низька (Мобільні застосунки)
Гнучкість	Висока	Середня	Низька
Інтеграція	Добра (Вбудовані та Сторонні)	Добра (Вбудовані)	Середня (Орієнтована на Google Sheets)

Ці платформи відкривають нові можливості для розробників і творців, але вибір між ними залежить від конкретних потреб проекту та рівня гнучкості, який вам необхідний. Bubble і Webflow підходять для різноманітних завдань,

включаючи бізнес-застосунки та дизайнерські проєкти, тоді як Glide є спеціалізованою платформою для мобільних застосунків.

2.5 Методика швидкого створення застосунків за допомогою технології no-code

Методика швидкого створення застосунків базується на використанні технології no-code та використанні готових функціональних блоків або шаблонів при створенні застосунку. Вона розширює можливості не технічних користувачів для самостійного розроблення та розгортання додатків. Це допоможе бізнесу з економити витрати на команду розробників, і направити бюджет на інші потреби бізнесу.

Основними етапами методики є:

- Вибір шаблону: Користувач обирає готовий шаблон застосунку, який відповідає його конкретним потребам. Це може бути шаблон для CRM-системи, інтернет-магазину, застосунку з геолокацією тощо.
- Конфігурація функціоналу: Вибраний шаблон надає основну структуру застосунку. Користувач може легко додавати, видаляти або налаштовувати функціональні блоки, використовуючи інтуїтивно зрозумілі інтерфейси no-code платформи.
- Інтеграція зовнішніх сервісів: Для поліпшення функціональності застосунку користувач може використовувати готові інтеграції з різними зовнішніми сервісами, такими як системи оплати, геолокації, державні служби, тощо.
- Додавання блоків функціоналу: No-code платформа надає готові блоки, які можна легко вставляти в застосунок. Це може бути блок для навігації, різноманітні готові форми для заповнення з налаштованою логікою, розрахункові модулі та інші. Користувач може складати їх у необхідному порядку.
- Персоналізація дизайну: No-code платформа дозволяє користувачеві налаштовувати дизайн застосунку, використовуючи готові елементи і стилі. Це дозволяє створювати унікальний та адаптований до бренду вигляд застосунку.

- Тестування та виправлення помилок: Після створення застосунку, користувач може провести тестування для виявлення помилок та оптимізації його роботи. No-code платформа забезпечує можливість візуального відстеження та виправлення помилок без програмування.

- Розгортання та масштабування: Завершивши створення та тестування застосунку, користувач може легко розгорнути його для використання. Крім того, застосунок можна масштабувати, додавши новий функціонал чи змінюючи існуючий, щоб відповідати розвитку бізнесу.

В результаті, методика швидкого створення застосунків за допомогою технології no-code дозволяє не технічним користувачам ефективно створювати та розвивати застосунки, прискорюючи процес розробки та забезпечуючи гнучкість у використанні технологій.

3 РОЗРОБКА ПЛАТФОРМИ ДЛЯ ШВИДКОГО СТВОРЕННЯ ПРОДУКТІВ ДЛЯ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ НА ОСНОВІ ТЕХНОЛОГІЙ NO-CODE

3.1 Опис використаних програмних засобів

MongoDB[21] - це система управління базами даних (СУБД), яка використовує концепцію NoSQL (Not Only SQL) для зберігання та обробки даних. Основна відмінність MongoDB від традиційних реляційних баз даних полягає в тому, що вона використовує JSON-подібні об'єкти для представлення даних, а не табличні структури.

MongoDB надає гнучкість у роботі з даними, не вимагаючи жорсткої визначеної схеми. Кожен документ у колекції може мати різні поля, і додавання нових полів відбувається без зайвого клопоту. Основними структурними одиницями є документи, які групуються в колекції, що відображає концепцію таблиць у реляційних базах даних.

Один із ключових елементів MongoDB - це мова запитів MQL (MongoDB Query Language), аналог JSON, яка спрощує і вдосконалює взаємодію з базою даних. Ця мова дозволяє легко конструювати запити за допомогою об'єктів та операторів.

MongoDB також знайшла широке застосування великими корпораціями, такими як Adobe, eBay, Cisco, Forbes, Bosch та іншими. Її гнучка схема, здатність до горизонтального масштабування, та підтримка індексації роблять її відмінним вибором для проектів різного роду та масштабу. MongoDB підтримує широкий спектр мов програмування, включаючи Java, Python, JavaScript (Node.js), C#, Ruby, що робить її доступною для розробників з різних технологічних екосистем.

Окрім того, для зручності користувачів існує MongoDB Atlas - облачний сервіс, який спрощує розгортання та масштабування MongoDB у хмарних середовищах, таких як AWS, Azure та Google Cloud.

Основні характеристики MongoDB наведені нижче:

- Документоорієнтована модель: Дані в MongoDB зберігаються у вигляді BSON (Binary JSON) документів. Кожен документ - це JSON-подібний об'єкт, що містить пари ключ-значення.

- Колекції: Документи групуються в колекції, що відповідає поняттю таблиці в реляційних базах даних. Колекції можуть містити різні типи документів.

- JSON-подібна запитова мова: Для взаємодії з базою даних використовується JSON-подібна мова запитів. Запити побудовані за допомогою об'єктів та операторів, які надають широкі можливості для фільтрації та обробки даних.

- Гнучка схема: MongoDB не вимагає строго визначеної схеми даних. Кожен документ в колекції може мати різні поля, і нові поля можуть додаватися безпосередньо до існуючих документів.

- Індексація: MongoDB підтримує створення індексів для поліпшення швидкодії операцій пошуку та сортування даних.

- Горизонтальне масштабування: MongoDB легко масштабується горизонтально, що означає, що можна додавати нові сервери для обробки збільшеної кількості даних.

- Реплікація та розділення даних: MongoDB підтримує реплікацію для забезпечення високої доступності та надійності. Також можливе розділення даних для більш ефективного розподілу навантаження.

- Утиліти та інструменти: MongoDB постачається з рядом утиліт та інструментів для адміністрування та моніторингу бази даних.

MongoDB часто використовується в веб-розробці, аналізі даних та інших сценаріях, де необхідна гнучкість у зберіганні та обробці невструктурованих даних.

Node.js – це інструмент, який надає можливість використовувати JavaScript для серверної розробки. Node.js дозволяє створювати ефективні та швидкодіючі застосунки з використанням однієї мови програмування як на клієнтській, так і на серверній стороні. Заснований на движку V8 від Google Chrome, Node.js дозволяє забезпечити велику продуктивність при обробці запитів. Це особливо важливо для реалізації високонавантажених застосунків і веб-сервісів. Node.js має широкі

можливості для асинхронного програмування, використовуючи концепцію подій та зворотних викликів (callbacks). Це дозволяє розробникам створювати ефективні програми, які можуть обробляти багато запитів одночасно, зменшуючи час відповіді та поліпшуючи користувацький досвід. Модульна система Node.js дозволяє використовувати різноманітні бібліотеки та фреймворки, що сприяє в швидкому розгортанні та розширенні проєктів. Однією з ключових переваг Node.js є можливість створення масштабованих застосунків за допомогою асинхронного програмування. Це дозволяє ефективно використовувати ресурси сервера, що є критично важливим у великих проєктах з великою кількістю одночасних запитів. Важливо зазначити, що Node.js не обмежується лише веб-розробкою. Він також використовується для створення інструментів командного рядка, роботи з базами даних, розробки мобільних застосунків та інших сценаріїв, де важлива висока продуктивність та гнучкість.

Redis[22] – це, високопродуктивна система управління даними, яка використовується для зберігання та обробки даних в оперативній пам'яті. Створений Сальваторе Санфіліппо у 2009 році, Redis відзначається своєю швидкістю та простотою використання, що робить його популярним в області кешування, сесійного зберігання та вирішення різних задач зберігання даних у реальному часі. Однією з ключових особливостей Redis є те, що він дозволяє зберігати дані у вигляді ключ-значення, а значення може бути рядком, списком, масивом, хешем або іншою структурою даних. Це робить Redis дуже гнучким для вирішення різних завдань. Ще однією важливою рисою Redis є його можливість працювати в режимі Master-Slave, що дозволяє створювати резервні копії даних та забезпечувати високу доступність. Redis підтримує реплікацію даних, що дозволяє вам створювати копії даних на інших серверах для поліпшення продуктивності та безпеки. Використання Redis в розробці застосунків може значно покращити продуктивність завдяки швидкому доступу до даних, який забезпечується через використання оперативної пам'яті. Крім того, Redis має багатий набір операцій для роботи з рядками, списками, множинами та іншими типами даних, що дозволяє ефективно використовувати його для різноманітних завдань. У сучасному

інтернет-середовищі Redis використовується як один із ключових компонентів для забезпечення швидкодії та масштабованості веб-застосунків.

3.2 Опис структури проекту

3.2.1 Процес розробки застосунку з використанням No-code платформи

Процес розробки застосунку за допомогою No-code Платформи є інтуїтивно зрозумілим і дозволяє користувачам швидко створити та налаштувати свої власні веб-застосунки. Нижче наведено основні кроки цього процесу:

1. Налаштування базових властивостей: Користувачі починають з визначення основних параметрів свого застосунку, таких як назва, опис, логотип та інші характеристики. Цей етап дозволяє персоналізувати застосунок відповідно до власних потреб та бренду.

2. Вибір шаблону застосунку: Користувачі обирають готовий шаблон застосунку, який відповідає їхнім конкретним потребам. Це може бути шаблон для CRM, інтернет-магазину, аналітичного інструменту чи будь-якого іншого типу застосунку.

3. Наповнення контентом та логікою: Після вибору шаблону користувачі починають наповнювати застосунок власним контентом та встановлюють логіку роботи. Це включає в себе створення різних розділів, додавання текстів, медіа-файлів та налаштування взаємодії елементів застосунку.

4. Редагування застосунку: Після реалізації базового функціоналу користувачі можуть повертатися до етапу налаштування та редагування, щоб внести будь-які зміни або доповнення. Це дозволяє легко адаптувати застосунок під нові вимоги або зміни в бізнес-процесах.

5. Маркетингова кампанія: Завершальний етап перед випуском застосунку - це налаштування маркетингової кампанії. Користувачі визначають стратегії просування, рекламу та збір зворотного зв'язку від користувачів для подальшого вдосконалення застосунку.

6. Внесення змін та оновлення: У випадку необхідності коригування чи оновлення застосунку, користувачі повертаються до етапу редагування. Це може бути зумовлено змінами в бізнес-вимогах, покращеннями функціоналу чи відгуками від користувачів.

Цей цикл дозволяє користувачам динамічно адаптувати свої застосунки, забезпечуючи швидку відповідь на зміни в бізнес-середовищі та вимоги користувачів. Розгорнута схема роботи з no-code платформою відображає динаміку та гнучкість цього процесу і зображена на рисунку 3.1



Рис 3.1. Схема роботи з платформою

3.2.2 Архітектура no-code платформи

Архітектура платформи розроблена для забезпечення високої ефективності, гнучкості та масштабованості. Архітектура складається з чотирьох основних блоків, кожен із яких відповідає за визначену функціональність:

- Блок балансування навантаження: Цей блок відповідає за розподіл трафіку між різними серверами застосунку. Використовуючи алгоритми балансування навантаження, він гарантує, що кожен запит від користувача буде направлений на доступний та оптимальний сервер. Це допомагає уникнути перевантаження окремих серверів та забезпечує рівномірний розподіл навантаження.

- Блок серверів застосунку: Цей блок містить групу серверів, які відповідають за виконання логіки та опрацювання запитів, що надходять від користувачів. Сервери застосунку горизонтально масштабуються, щоб забезпечити необхідну продуктивність при збільшенні обсягу роботи.

- Блок серверу баз даних: Цей блок відповідає за збереження та управління даними, необхідними для правильної роботи застосунку. Використовуючи потужні системи управління базами даних (СУБД), він забезпечує ефективний доступ до інформації та її надійність.

- Блок серверу кешування: Цей блок відповідає за збереження та швидке надання кешованих даних, що дозволяє платформі зменшити час відповіді на запити користувачів. Кешування використовується для оптимізації продуктивності та зниження завантаження на сервер баз даних.

Архітектурна схема зображена на рисунку 3.2.

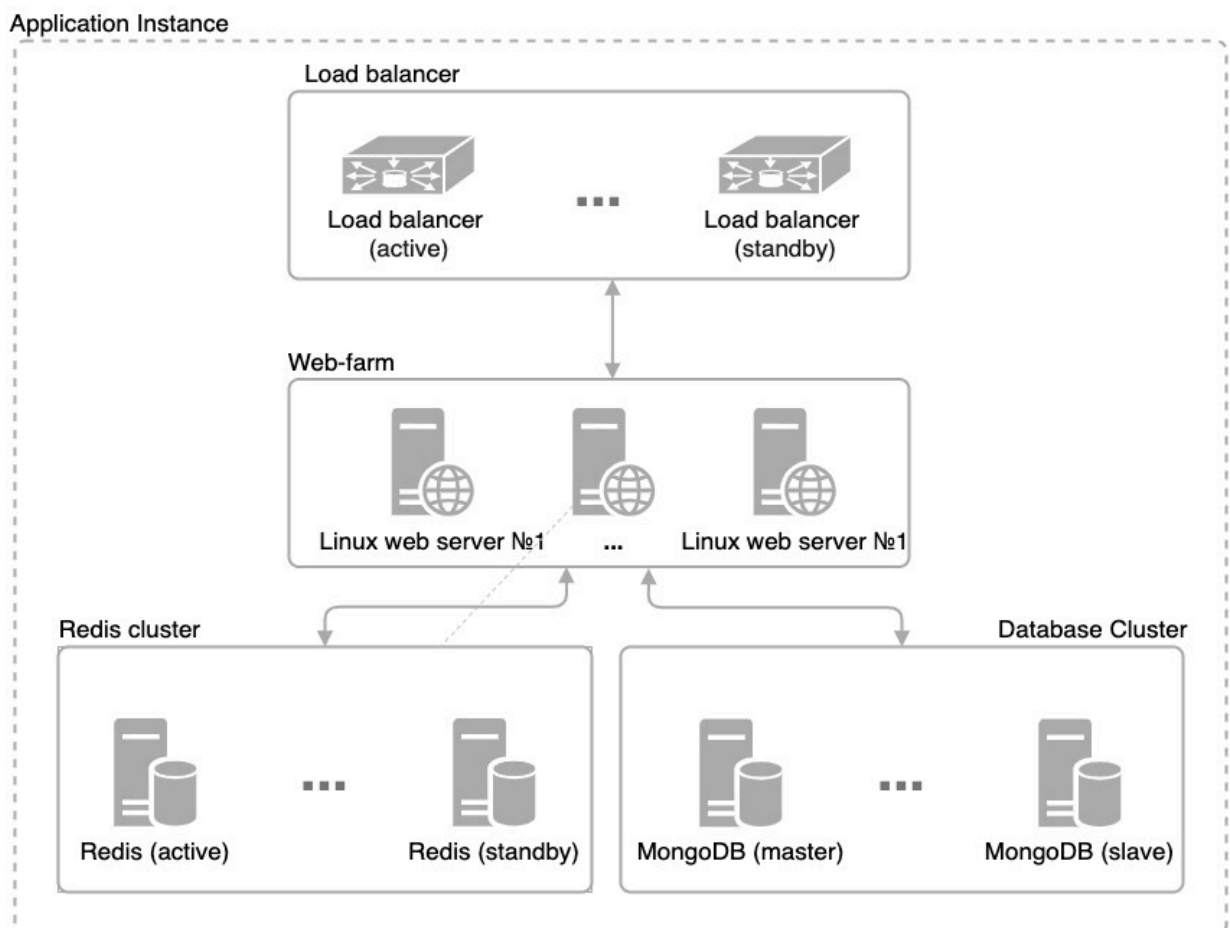


Рис 3.2. Архітектура платформи

Ця архітектура, заснована на горизонтальному масштабуванні серверів, забезпечує високу доступність та витривалість платформи, а також здатність швидко адаптуватися до зростаючого обсягу користувацького трафіку. Взаємодія

між цими блоками створює стійку та продуктивну інфраструктуру для ефективного використання no-code платформи.

3.2.3 Структура та функціональність no-code платформи

No-code платформа складається з двох ключових компонентів, кожен із яких відіграє важливу роль у процесі створення та впровадження веб-застосунків (рис.3.3).

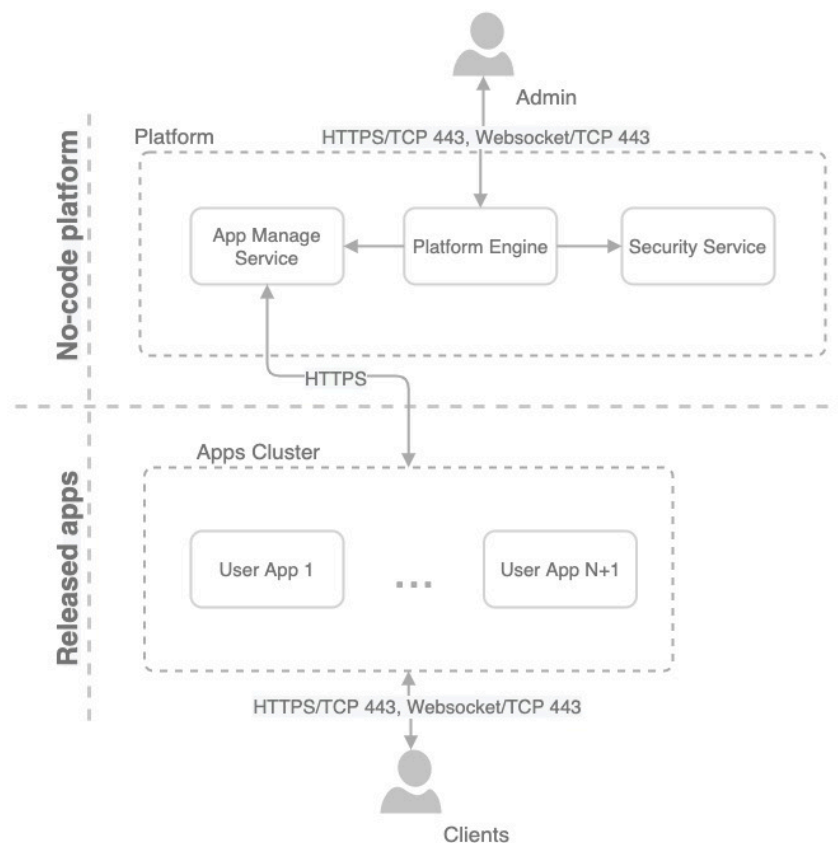


Рис 3.3. Структура платформи

No-code застосунок для налаштування: Ця частина платформи є інтерфейсом для користувачів, де вони можуть використовувати no-code інструменти для налаштування та конфігурації своїх веб-застосунків. На цьому етапі користувачі можуть обирати готові шаблони, використовувати функціональні блоки, а також взаємодіяти з інтерфейсом для визначення логіки та параметрів свого застосунку. Використовуючи інтуїтивно зрозумілі інструменти, користувачі можуть швидко створювати та налаштовувати різноманітні функції застосунків без необхідності

глибоких технічних знань.

Розгорнуті застосунки: Ця частина представляє собою вже створені та готові до використання веб-застосунки, які були розроблені за допомогою no-code інструментів у no-code Застосунку для Налаштування. Після завершення конфігураційного процесу, застосунки можуть бути розгорнуті для використання. Ця частина включає в себе весь код, інфраструктуру та необхідні ресурси для того, щоб застосунок працював на вищому рівні ефективності та безпеки. Користувачі можуть легко розгортати свої створені застосунки для широкого кола аудиторії без особливих зусиль.

3.3 Опис інтерфейсу

Інтерфейс No-code платформи розробки застосунків спрощує процес створення та налаштування застосунків, забезпечуючи користувачам доступ до готових компонентів та інструментів для налаштування зовнішнього вигляду. Важливі аспекти інтерфейсу включають:

- Готові компоненти: Користувачеві надаються готові компоненти, такі як карточки, калькулятори цін, списки, таблиці та інші, які можна використовувати для швидкої побудови функціоналу застосунку.

- Інструменти настройки: Інтерфейс включає інструменти для налаштування зовнішнього вигляду та поведінки компонентів. Користувачі можуть встановлювати параметри, такі як розмір, колір, шрифт, анімацію тощо, без використання коду.

- Створення нового блоку: на рисунку 3.4 показано створення нового блоку картки. Під час створення блоку користувачеві доступні наступні опції:

1. Заповнення загальної інформації: Визначення основної інформації для блоку, такої як назва та опис.
2. Встановлення заголовку блоку: Вказання заголовку для блоку, який буде відображатися в інтерфейсі застосунку.

3. Вибір типу блоку: Обрання типу блоку відповідно до потреб користувача.

4. Додавання фотографії: Можливість вставити фотографію для візуального відображення блоку.

- Динамічні зміни та доповнення: Інтерфейс дозволяє користувачам вносити динамічні зміни та доповнення до блоків у режимі реального часу. Це забезпечує гнучкість у процесі розробки та експериментів.

- Ілюстрація розробки блоку: на рисунку 3.4 ілюстровано процес створення нового блоку картки та його наповнення контентом. Це дає користувачам візуальний огляд та контроль над виглядом та поведінкою їхніх застосунків.

Рис 3.4. Приклад створення нового компоненту в платформі

Інтерфейс no-code платформи створення застосунків надає необхідні інструменти для ефективною та інтуїтивно зрозумілою розробки, роблячи цей процес доступним навіть для користувачів без глибоких технічних знань.

На рисунку 3.5 продемонстровано, як користувач може ефективно налаштувати зовнішній вигляд картки, використовуючи інструменти No-code платформи.

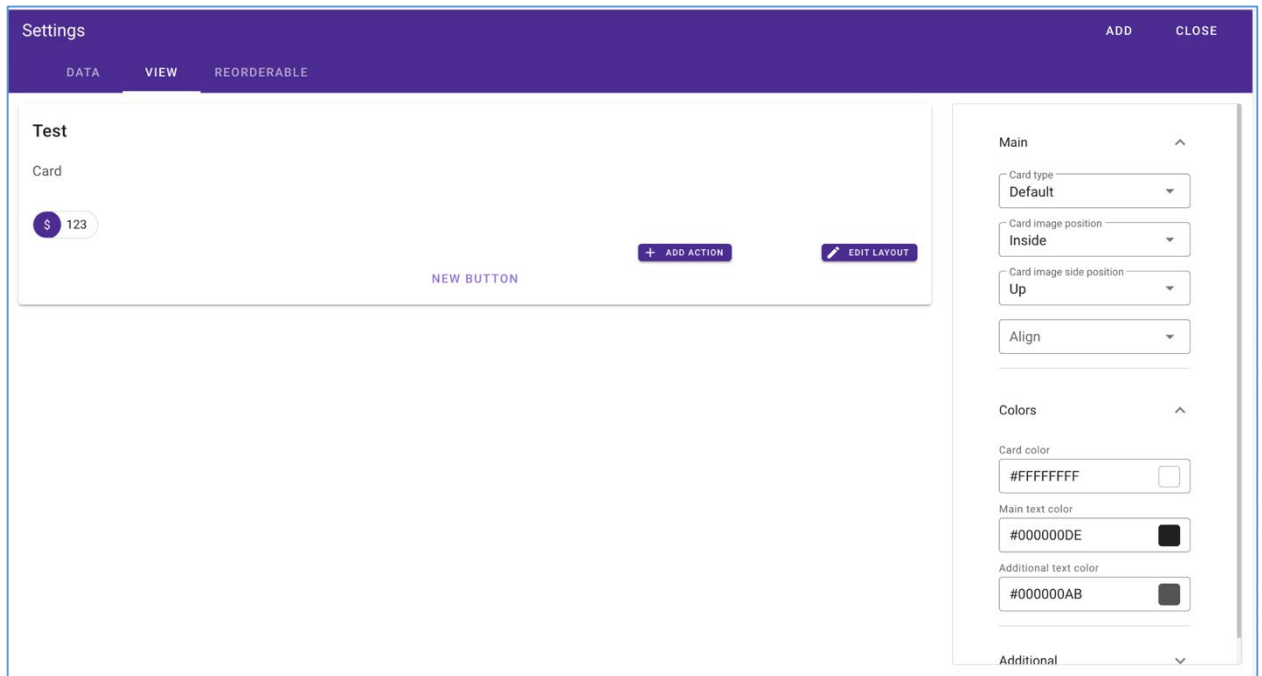


Рис 3.5. Приклад налаштування зовнішнього вигляду компоненту

Основні можливості налаштування включають:

- Додавання кнопок: За допомогою інструментів користувач може легко додавати кнопки на картку. Це відкриває можливості для інтерактивності та навігації в межах застосунку.
- Колірна гамма: Користувач може налаштувати кольори картки та тексту, забезпечуючи гнучкість у виборі палітри, яка відповідає корпоративному стилю або дизайну застосунку.
- Тінь та 3D-ефекти: Інструменти дозволяють встановлювати тінь для картки, що додає візуальний ефект та реалістичність. Також можливі 3D-ефекти для покращення вигляду.
- Тип розміщення тексту: Користувач може вибрати тип розміщення тексту на картці, такий як горизонтальне або вертикальне, забезпечуючи гнучкість у відображенні інформації.

- Інші налаштування: Додаткові параметри включають розміри шрифтів, стилі вирізування тексту, величину та вирізи для фотографій, що дозволяє користувачам докладно налаштовувати вигляд кожного елемента.

Ці інструменти дозволяють користувачам створювати не лише функціональні, але й естетично привабливі картки зовнішнього вигляду для їхніх застосунків, сприяючи кращому взаєморозумінню та привертанню уваги користувачів.

На рисунку 3.6 представлено готовий результат роботи користувача з no-code платформою. Додавання функціональних блоків, шаблонів та налаштування зовнішнього вигляду призвело до створення повноцінного застосунку.

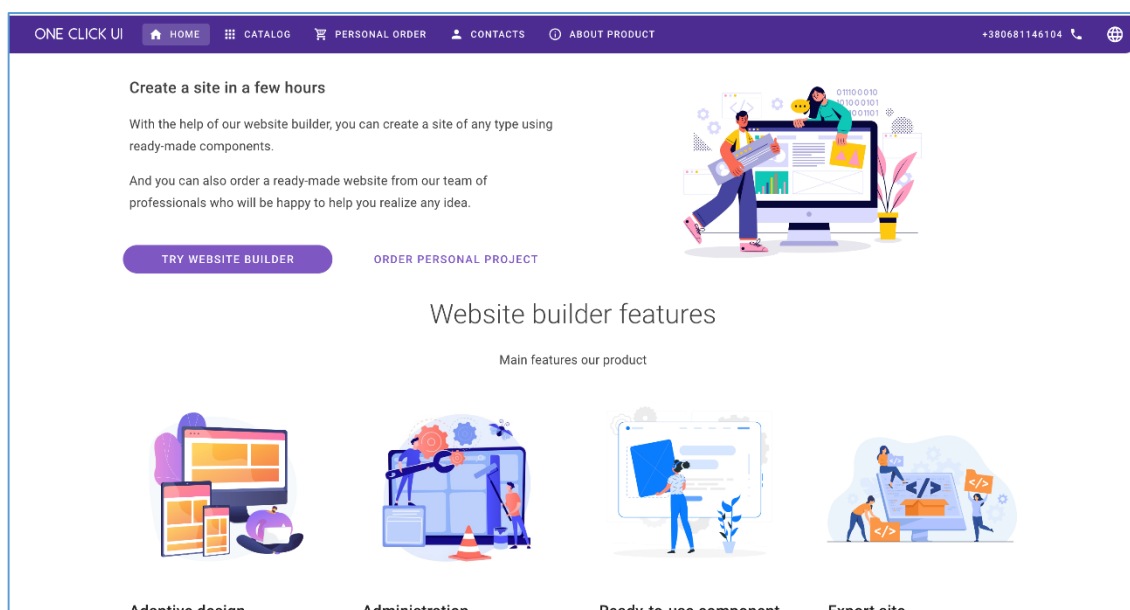


Рис 3.6. Результат створеного застосунку в платформі

Деякі ключові аспекти цього результату включають:

- Функціональність: Застосунок має вбудовані функціональні блоки, такі як картки, калькулятори цін, списки, таблиці та інші, що роблять його готовим до використання в різних сценаріях.

- Вміст та логіка: Користувач може наповнити застосунок контентом та визначити логіку його роботи, використовуючи no-code інструменти. Це дозволяє адаптувати застосунок під конкретні потреби.

- Графічний інтерфейс: Зовнішній вигляд застосунку відображається на екрані, інтерфейс створений за допомогою інструментів для налаштування зовнішнього вигляду. Користувач може докладно контролювати елементи та їхнє розташування.

- Інтеграції: Якщо потрібно, застосунок може бути легко інтегрований з зовнішніми системами оплати, геолокацією чи іншими службами.

- Ефективність витрат: Результативність no-code платформи демонструється економією часу та витрат на розробку. У порівнянні з традиційним підходом, де розробник витратив би 80 годин, застосунок був створений всього за 24 години.

Цей результат вказує на успішне використання технології no-code для швидкої та ефективної розробки застосунків без значних технічних навичок.

3.4 Специфікація розроблених шаблонів

3.4.1 Шаблон crm-системи

CRM (Customer Relationship Management) - це система управління відносинами з клієнтами, яка спрямована на ефективну роботу з клієнтами, ведення інформації про взаємодії та покращення комунікації між різними відділами підприємства. No-code платформи надають можливість використовувати готові шаблони для швидкого створення CRM-систем без програмування. Основні характеристики та елементи шаблону CRM-системи включають:

1. Контактна інформація:

- Збереження та відстеження контактної інформації клієнтів, включаючи ім'я, контактні дані, посаду, тощо.

- Можливість додавати нових клієнтів та вносити зміни у їхні дані.

2. Історія взаємодій:

- Запис усіх взаємодій з клієнтом, такі як телефонні дзвінки, листування, особисті зустрічі, тощо.

- Хронологічний перегляд історії спілкування для кращого розуміння потреб клієнта.

3. Управління продажами:

- Відстеження усіх етапів продажу, від потенційного клієнта до успішного укладення угоди.

- Аналіз обсягів продажів та прогнозування доходів.

4. Електронна пошта та комунікації:

- Інтеграція з електронною поштою для зручного листування та відправки пропозицій.

- Автоматизована реєстрація електронної комунікації з клієнтами.

5. Завдання та Календар:

- Створення та призначення завдань для учасників команди, пов'язаних з конкретним клієнтом чи угодою.

- Інтеграція календаря для планування зустрічей та подій.

6. Аналітика та звітність:

- Генерація звітів про ефективність роботи команди та успішність продажів.

- Візуалізація ключових показників продажів та залученості клієнтів.

7. Інтеграція з іншими системами:

- Можливість інтеграції з іншими платформами та сервісами для автоматизації процесів.

8. Користувальницький інтерфейс:

- Простий та інтуїтивно зрозумілий інтерфейс для швидкого доступу та роботи з функціоналом системи.

Використання шаблону CRM-системи в no-code платформі дозволяє підприємствам ефективно впорядковувати та оптимізувати взаємодію з клієнтами, що призводить до підвищення якості обслуговування та збільшення продуктивності бізнес-процесів.

3.4.2 Шаблон універсального застосунку з мапою і геолокацією

Цей шаблон надає можливість впровадження мап та геолокації в різноманітних контекстах і для різних цілей. Він дозволяє користувачам без технічних навичок створювати застосунки з інтерактивними мапами та функціоналом геолокації. Основні елементи цього шаблону включають:

1. Інтерактивна мапа:

- Легка інтеграція інтерактивної мапи у застосунок.
- Можливість додавати та відображати маркери, лінії, та полігони на мапі.

2. Пошук та маршрутизація:

- Функціонал пошуку місць та визначення маршрутів.
- Налаштування параметрів пошуку та відображення оптимальних маршрутів.

Геолокація та трекінг:

- Використання геолокації для визначення поточного положення користувача.
- Можливість включення функції трекінгу для відстеження руху користувача на мапі.

3. Інформаційні вікна:

- Додавання інформаційних вікон до маркерів на мапі для надання докладної інформації про місця.

Фільтрація маркерів:

- Можливість фільтрування маркерів за різними критеріями (категорії, рейтинг, тощо).

4. Інтеграція з зовнішніми джерелами даних:

- Підключення та використання даних з зовнішніх джерел для покращення інформації на мапі.

Мультимовний супутниковий вид:

- Відображення мультимовного супутникового виду на мапі.

5. Адаптивний дизайн:

- Підтримка адаптивного дизайну для зручного перегляду на різних пристроях.

Цей універсальний шаблон може бути використаний для створення різноманітних застосунків, таких як мобільні додатки для навігації, туристичні платформи, сервіси доставки з відстеженням, тощо. Надаючи інструменти для легкої інтеграції мап та геолокації, цей шаблон робить можливим створення застосунків, які використовують простір і місцевість для поліпшення користувацького досвіду.

3.4.3 Шаблон інтернет-магазину

Шаблон Інтернет-магазину в no-code платформі дозволяє швидко створити та налаштувати електронний магазин без програмування. Основні характеристики та елементи шаблону Інтернет-магазину включають:

1. Каталог товарів:

- Створення каталогу товарів з можливістю додавання, видалення та редагування продуктів.

- Категоризація товарів для зручності навігації клієнтів.

2. Опис товарів та зображення:

- Додавання докладної інформації та зображень до кожного товару.

- Зручний редактор для форматування та стилізації опису товару.

3. Кошик та оформлення замовлення:

- Додавання товарів до кошика та оформлення замовлення.

- Обробка платежів та інтеграція з різними системами оплати.

4. Управління замовленнями:

- Відстеження статусу та історії замовлень.

- Сповіщення клієнтів про зміни в стані їхніх замовлень.

5. Фільтри та пошук:

- Можливість швидкого пошуку та фільтрації товарів за різними параметрами.

6. Знижки та промокоди:

- Налаштування різноманітних знижок та промокодів для залучення клієнтів.

7. Відгуки та рейтинги

- Можливість залишати відгуки та оцінювати товари.

8. Інтеграція з системами доставки:

- Спрощена інтеграція з різними службами доставки для розрахунку вартості та вибору способу доставки.

9. Адаптивний дизайн:

- Дизайн, який адаптується до різних типів пристроїв, що забезпечує зручний перегляд на комп'ютерах, планшетах і смартфонах.

10. Аналітика та звітність:

- Збір та візуалізація даних щодо продажів, відвідуваності та поведінки покупців.

Шаблон Інтернет-магазину в no-code платформі дозволяє навіть непрограмістам швидко створити та управляти ефективним онлайн-магазином, прискорюючи процес розгортання та оптимізуючи роботу замовлень та клієнтського середовища.

3.5 Специфікація розроблених готових функціональних блоків

3.5.1 Каталог продуктів

Компонент "Каталог продуктів" є важливим елементом для створення електронного магазину або будь-якої платформи, де потрібно представити асортимент товарів чи послуг. Використання цього компонента в no-code платформі дозволяє користувачам легко створювати, редагувати та управляти каталогами, навіть якщо вони не мають глибоких технічних знань. Основні аспекти цього компонента включають:

1. Створення та редагування продуктів:

- Додавання продуктів: Користувач може легко додавати нові продукти в каталог, вказуючи основні характеристики, такі як назва, опис, ціна, зображення тощо.

- Редагування продуктів: Є можливість змінювати існуючі продукти, оновлюючи інформацію та зображення.

2. Категорії та Теги:

- Категорії продуктів: Продукти можуть бути організовані в різні категорії для полегшення навігації користувачів.

- Теги: Додавання тегів для продуктів, що дозволяє швидше знаходити та фільтрувати їх за різними параметрами.

3. Зображення та галереї:

- Зображення продуктів: Можливість завантажувати та відображати зображення продуктів.

- Галереї: Створення галерей для кожного продукту для подання більш детальної інформації.

4. Фільтрація та пошук:

- Фільтрація: Можливість фільтрувати продукти за різними параметрами, такими як ціна, категорія, наявність тощо.

- Пошук: Інструменти для швидкого пошуку конкретного продукту в каталозі.

5. Інтеграція з Іншими Компонентами:

- Кнопки покупки: Інтеграція з кнопками або функціями, що дозволяють користувачам додавати товари до кошика та оформляти покупки.

- Оцінки та відгуки: Взаємодія з іншими компонентами для збору відгуків і надання оцінок продуктам.

6. Аналітика:

- Статистика продажів: Збір та аналіз даних щодо продажів для прийняття управлінських рішень.

7. Адаптивність та Зовнішній Вигляд:

- Резиновий дизайн: Адаптивність до різних розмірів екрану та пристроїв для зручного відображення каталогу продуктів.

- Налаштування зовнішнього вигляду: Інструменти для налаштування дизайну та вигляду каталогу.

8. Масові Операції:

- Одночасне редагування: Здатність виконувати масові редагування та додавання продуктів для ефективного управління асортиментом.

Компонент "Каталог продуктів" дозволяє швидко створювати та управляти електронними магазинами або каталогами з продуктами без програмування, що робить його ідеальним для no-code розробки.

3.5.2 Калькулятор цін

Компонент "Калькулятор цін" є потужним інструментом для створення і налаштування розрахунків цін на продукти чи послуги без необхідності в програмуванні. Цей компонент дозволяє користувачам створювати складні логічні формули та правила для визначення ціни продукту в залежності від різних параметрів. Основні характеристики компонента "Калькулятор цін" включають:

1. Настроювані розрахунки:

- Основні розрахунки: Визначення ціни може включати в себе основні математичні операції, такі як додавання, віднімання, множення та ділення.

- Логічні операції: Можливість використовувати логічні операції для визначення умов, при яких варто застосовувати певні розрахунки.

2. Введення параметрів:

- Користувацькі параметри: Дозвіл на введення користувачем різних параметрів, які впливають на ціну (наприклад, обсяг замовлення, кількість одиниць тощо).

- Динамічні дані: Здатність отримувати дані з інших компонентів або систем для використання в розрахунках.

3. Управління знижками та акціями:

- Знижки: Додавання логіки для застосування знижок або акцій, що впливають на кінцеву ціну.

- Динамічні зміни: Зміна цін в реальному часі відповідно до зміни параметрів або наявності акцій.

4. Візуальне представлення:

- Графічний інтерфейс: Інтеграція з інтерфейсом для візуального представлення розрахунків користувачу.

- Графіки та діаграми: Візуальні елементи для розуміння, як різні параметри впливають на ціну.

5. Динамічні правила та логіка:

- Сценарії розрахунків: Можливість створення різних сценаріїв для розрахунку ціни в залежності від різних умов.

- Обмеження: Встановлення обмежень та правил для уникнення некоректних або непрактичних цін.

6. Інтеграція з іншими компонентами:

- Зв'язок з каталогом: Підключення до каталогу продуктів для автоматичного розрахунку ціни на певний товар чи послугу.

- Інтерфейс оформлення замовлення: Легка інтеграція з іншими компонентами для оформлення замовлення з правильною ціною.

7. Аналітика та звітність:

- Збір та аналіз даних: Відстеження та аналіз ефективності цінової політики за допомогою статистики.

Використання компонента "Калькулятор цін" у no-code розробці дозволяє підприємствам швидко адаптувати свої цінові стратегії без глибоких технічних знань.

3.5.3 Форма зворотнього зв'язку

Компонент "Зворотній зв'язок" є ключовою складовою для забезпечення ефективного взаємодії з користувачами та отримання конструктивного відгуку щодо продуктів чи послуг. Цей компонент дозволяє створювати і налаштовувати

форми зворотного зв'язку та опитування без програмування. Основні функції та можливості компонента "Зворотній зв'язок" включають:

1. Створення форм та опитувань:

- Готові шаблони: Наявність готових шаблонів для швидкого створення різних форм зворотного зв'язку чи опитувань.

- Настроювані поля: Можливість додавати різні типи полів (текстові, випадаючі списки, радіокнопки тощо) для отримання різноманітної інформації.

2. Логіка та умови:

- Логіка відображення: Встановлення умов для відображення певних полів або запитань в залежності від введених даних.

- Застосування умов: Змога динамічно змінювати логіку форми в реальному часі з урахуванням різних сценаріїв.

3. Зручний інтерфейс:

- Графічний редактор: Інтуїтивно зрозумілий графічний інтерфейс для створення та налаштування форм.

- Можливість перегляду: Попередній перегляд форми перед публікацією для забезпечення оптимального користувацького досвіду.

4. Інтеграція з іншими компонентами:

- Збір даних: Інтеграція з іншими компонентами для автоматичного збору і обробки отриманих відгуків.

- Взаємодія з базою даних: Збереження та управління даними в базі для подальшого аналізу.

5. Модерація та фільтрація:

- Фільтри та модерація: Можливість застосовувати фільтри та модерацію для контролю якості та відфільтрування вмісту.

- Автоматизована Обробка: Використання автоматизованих інструментів для обробки великого обсягу відгуків.

6. Аналітика та звітність:

- Збір та аналіз даних: Відстеження та аналіз відгуків для здобуття інсайтів та поліпшення продуктів чи послуг.

- Звітність: Генерація звітів та аналітичних матеріалів для розуміння рівня задоволення користувачів.

7. Інтерактивні елементи:

- Візуальні елементи: Додавання візуальних елементів для стимулювання участі та активності користувачів.

Використання компонента "Зворотній зв'язок" допомагає компаніям збирати цінні відгуки, реагувати на потреби клієнтів та покращувати якість продуктів або послуг.

3.6 Результати впровадження

Порівнюючи створення веб-застосунку за допомогою no-code платформи і традиційного підходу, можна виділити ряд очевидних характеристик, які призводять до значної економії для бізнесу.

- Швидкість розробки: Використання no-code платформи дозволяє створювати застосунки значно швидше, порівняно з традиційним програмуванням. Наприклад, при використанні платформи у конфігуратора процес розробки зайняв 24 години, оцінка розробника на створення такого ж самого застосунку при традиційному підході дорівнювала 80 годинам.

- Економія коштів: При використанні no-code платформи можна заощадити значні фінансові ресурси. При умові, що на проекті працює один розробник зі ставкою 25\$ за годину, при традиційному підході, було б витрачено $25 \cdot 80 = 1920\$$. Якщо використовувати платформу і найняти конфігуратора, то зазвичай його робота коштує 20\$ за годину, в такому разі вартість застосунку становить $20 \cdot 24 = 480\$$. В такому разі бізнес економить 1440\$ на створенні застосунку і може перенаправити кошти на інші потреби (рис.3.7).

- Гнучкість у розподілі завдань: Розробка no-code дозволяє залучити конфігураторів з меншим технічним бекграундом для створення функціональності застосунку і роботи над проектом. Бізнес-команда може ефективно розподіляти завдання і прискорювати вирішення проблем без обмеження технічними

навичками. Це спрощує і розподіляє трудові завдання, дозволяючи більш ефективно використовувати робочий час і знижувати витрати.

- Переваги резервування бюджету: Заощаджені кошти можуть бути перерозподілені на інші потреби, що може включати додатковий розвиток, маркетинг чи інші ініціативи.

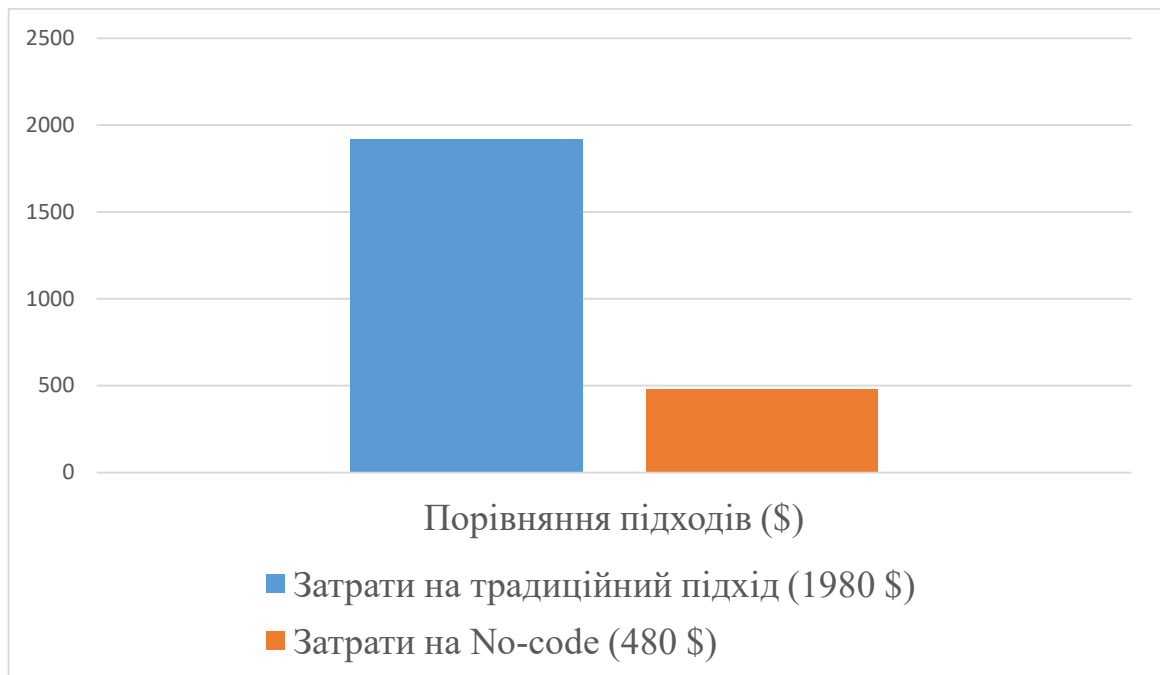


Рис. 3.7. Оцінка витрат на розробку

Таким чином, використання no-code платформи виявилось не тільки більш швидким і простим методом для створення веб-застосунків, але і призвело до значних фінансових вигод для бізнесу.

ВИСНОВКИ

1. Досліджено технологію no-code, її вплив на швидкість розробки застосунків, та можливість використання технології бізнесом задля оптимізації використання ресурсів.

2. Проаналізовано існуючі рішення що надають можливість використовувати технологію no-code для створення застосунків. Проведено аналіз особливостей та недоліків існуючих no-code it-рішень, визначено ключові потреби бізнесу при створенні веб-застосунків, визначено ключові потреби при роботі з no-code платформами.

3. Розроблено методику швидкого створення застосунків за допомогою використання технології no-code та набору готових функціональних блоків і шаблонів.

4. Проведено дослідження впливу на фінансові показники бізнесу після впровадження методики і використання платформи no-code для створення застосунків. Дослідження виявило, що впровадження методики швидкого створення застосунків за допомогою no-code технології має значний вплив на фінансові показники бізнесу. Зокрема, спостерігається значна економія часу і ресурсів у порівнянні із традиційними методами розробки. Час на розробку було пришвидшено на 70%. Вартість створення застосунку суттєво зменшилася на 75%, що сприяє оптимізації бюджету підприємства.

ПЕРЕЛІК ПОСИЛАНЬ

1. developers.google.com [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/> Дата звернення: 01.10.2023
2. Google Chrome API Reference [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://developer.chrome.com/docs/extensions/reference/> Дата звернення: 01.10.2023
3. [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://github.com/nodejs/node/README.md> Дата звернення: 02.10.2023.
4. Douglas Crockford. JSON: The Fat-Free Alternative to XML, 2009 – 55 p.
5. Mike Richardson, Ruby Leonard, Sam Amundsen. RESTful Web APIs./- “O'Reilly Media”, 2013 – с. 357 – 363.
6. Boehm, B. W. Software Engineering Economics. Prentice-Hall, 1981. - С. 112–130.
7. Sommerville, I. Software Engineering. Addison-Wesley, 2011. - С. 245–264.
8. No-Code Development: What It Is and Why You Should Care [Електронний ресурс]: – Режим доступу: <https://www.forbes.com/sites/forbestechcouncil/2022/01/12/no-code-development-what-it-is-and-why-you-should-care/?sh=516227457fd1> 18.10.2023
9. Mendes, E., Mosley, N., & Kean, T. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley, 2010. - С. 78–95.
10. The Rise of No-Code/Low-Code Development [Електронний ресурс]: – Режим доступу: <https://www.gartner.com/en/newsroom/press-releases/2021-08-18-gartner-forecasts-worldwide-no-code-low-code-market-to-reach-13-8-billion-in-2021> 20.09.2023
11. McConnell, S. Code Complete: A Practical Handbook of Software Construction. Microsoft Press, 2004. - С. 210–230.

12. No-Code and Low-Code Development: Definition, Benefits, and Challenges [Электронный ресурс]: – Режим доступа: <https://blog.hubspot.com/website/no-code-low-code> 21.10.2023

13. Brooks, F. P. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, 1995. - С. 45–62.

14. Understanding No-Code Platforms [Электронный ресурс]: – Режим доступа: <https://www.techopedia.com/2/32594/technology-trends/applications/understanding-no-code-platforms> 23.10.2023

15. McConnell, S. Rapid Development: Taming Wild Software Schedules. Microsoft Press, 1996. - С. 132–148.

16. No-Code vs. Low-Code: What's the Difference? [Электронный ресурс]: – Режим доступа: <https://www.outsystems.com/blog/low-code-vs-no-code/> 25.10.2023

17. Ambler, S. W. Introduction to UML 2 Activity Diagrams. UML 2 Activity and Action Models. IBM developerWorks, 2004. - С. 76–88.

18. No-Code Development Platforms: A Revolution in Software Development [Электронный ресурс]: – Режим доступа: <https://www.researchnester.com/reports/no-code-development-platforms-market/2849> 27.01.2023

19. Boehm, B. W. Get Ready for Agile Methods, with Care. Computer, 2002. - С. 64–69.

20. The No-Code Revolution: Democratizing Software Development [Электронный ресурс]: – Режим доступа: <https://www.cio.com/article/3325897/the-no-code-revolution-democratizing-software-development.html> 30.10.2023

21. Mongo DB [Электронный ресурс]: – Режим доступа: <https://www.mongodb.com/> 30.10.2023

22. Redis [Электронный ресурс]: – Режим доступа: <https://redis.io/> 30.10.2023

23. Node JS [Электронный ресурс]: – Режим доступа <https://nodejs.org/en/about> 30.10.2023

24. Webflow [Электронный ресурс]: – Режим доступа: <https://webflow.com/> 30.10.2023

25. Glide [Електронний ресурс]: – Режим доступу: <https://www.glideapps.com/>
30.10.2023

26. Bubble [Електронний ресурс]: – Режим доступу: <https://bubble.io/> 30.10.2023

27. Золотухіна О.А, Халецький В.С. Розробка методики швидкого створення веб-застосунків для малого та середнього бізнесу. // Науково-практична конференція «Проблеми комп'ютерної інженерії» – Київ: ДУІКТ, 2023. С. 152-153.

28. Халецький В.С., Золотухіна О.А., Довженко Т.П., Бондарчук А.П. Оптимізація створення веб-застосунків для малого та середнього бізнесу на основі технології no-code розробки. Телекомунікаційні та інформаційні технології. №1, 2024. Подано до друку.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



Кафедра інженерії програмного забезпечення

МАГІСТЕРСЬКА РОБОТА «РОЗРОБКА МЕТОДИКИ ШВИДКОГО СТВОРЕННЯ ВЕБ- ЗАСТОСУНКІВ ДЛЯ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ»

Виконав: студент групи ПДМ – 62, Халецький Віталій Сергійович

Керівник: к.т.н., доц., кафедри ПЗ, Довженко Тимур Павлович

Київ - 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

1

Мета роботи: оптимізація процесу створення веб-застосунків для малого та середнього бізнесу за рахунок використання технології по-code розробки.

Об'єкт дослідження: процес створення веб-застосунків.

Предмет дослідження: методи та технології створення веб-застосунків.

.bubble

Створення додатків на будь яку платформу
(no-code)

Webflow

Створення веб-сайтів і веб-додатків
(low-code)

glide.

Створення мобільних додатків
(no-code)

ПОРІВНЯННЯ ПІДХОДІВ ДО РОЗРОБКИ ЗАСТОСУНКІВ

Критерій	No-Code	Традиційна розробка
Швидкість та простота	Швидко створює прототипи та додатки без великого обсягу коду.	Вимагає кодування та тестування, що часто призводить до тривалих термінів розробки.
Вартість та ефективність	Знижує витрати на програмістів, оскільки не вимагає глибоких технічних навичок.	Високі витрати на програмістів та інфраструктуру для розробки.
Доступність для нетехнічних користувачів	Дозволяє нетехнічним фахівцям та бізнес-аналітикам створювати додатки без кодування.	Вимагає глибокого розуміння програмування, обмежуючи доступність для нетехнічних користувачів.
Гнучкість та розширені можливості	Годиться для швидкого розгортання простих додатків, але може обмежувати гнучкість у складних проектах.	Надає повну гнучкість та контроль, ідеально підходить для складних та масштабних проектів.
Масштабованість	Може стикатися з обмеженнями масштабування для великих та складних систем.	Забезпечує можливість масштабування та розвитку великих корпоративних систем.
Інтеграція та зовнішні системи	Забезпечує інтеграцію з деякими сторонніми сервісами, але може бути обмеженим у випадках складних інтеграцій.	Забезпечує глибоку інтеграцію з будь-якими зовнішніми системами та сервісами.

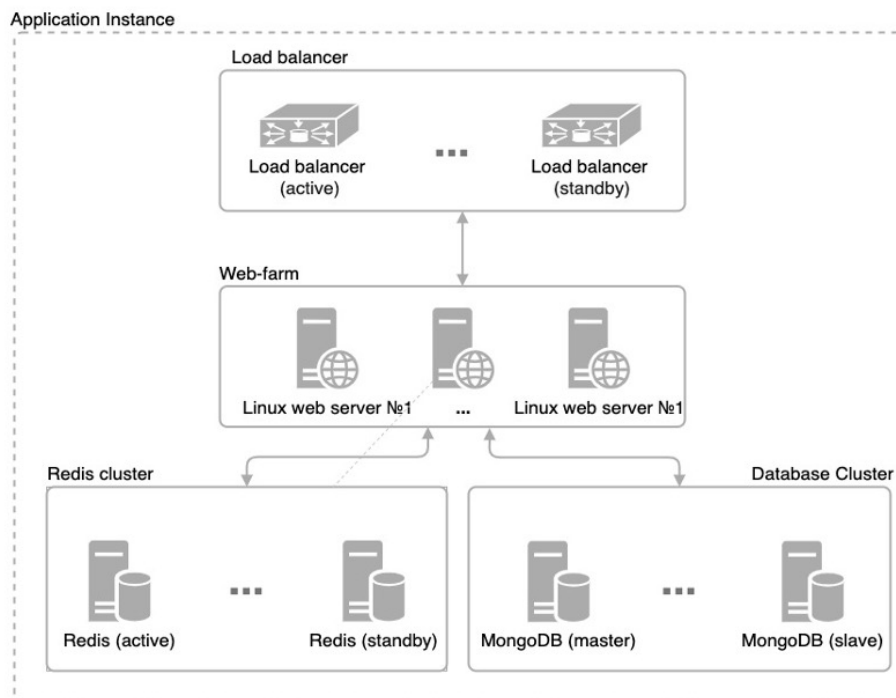
АЛГОРИТМ СТВОРЕННЯ ЗАСТОСУНКУ НА ОСНОВІ ПЛАТФОРМИ

4



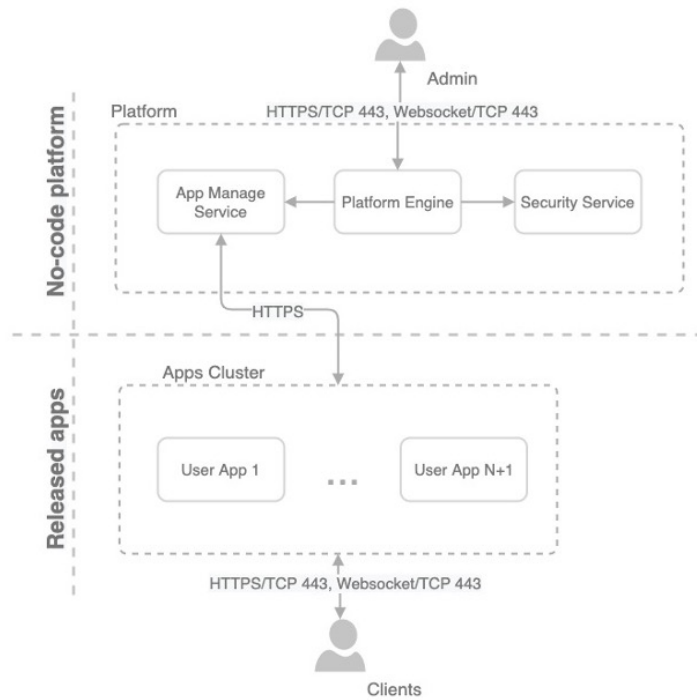
АРХІТЕКТУРА NO-CODE ПЛАТФОРМИ

5



СТРУКТУРА NO-CODE ПЛАТФОРМИ

6



ПРИКЛАД ПРОЦЕСУ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКУ

7

Settings ADD CLOSE

DATA VIEW REORDERABLE

Section: Home

Information display method: Card

Title: Test

Additional information: Card

Price: 123

Description:
[Rich text editor toolbar with icons for undo, redo, bold, italic, link, unlink, list, list, H1, H2, H3, B, <>, and a globe icon]

ПРИКЛАД ПРОЦЕСУ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКУ

8

Settings ADD CLOSE

DATA VIEW REORDERABLE

Test

Card

123

NEW BUTTON + ADD ACTION ✎ EDIT LAYOUT

Main ^

Card type
Default

Card image position
Inside

Card image side position
Up

Align

Colors ^

Card color
#FFFFFF

Main text color
#000000E

Additional text color
#000000AB

Additional v

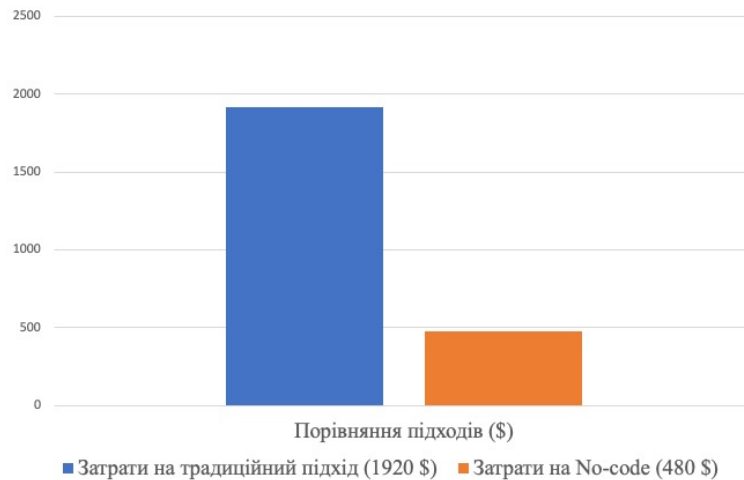
РЕЗУЛЬТАТ РОБОТИ З NO-CODE ПЛАТФОРМОЮ

9

The screenshot displays a web application interface for a website builder. The top navigation bar includes links for 'HOME', 'CATALOG', 'PERSONAL ORDER', 'CONTACTS', and 'ABOUT PRODUCT'. The main content area features a promotional message: 'Create a site in a few hours' with subtext explaining the ease of use and professional support. Below this, there are two buttons: 'TRY WEBSITE BUILDER' and 'ORDER PERSONAL PROJECT'. A section titled 'Website builder features' lists 'Adaptive design', 'Administration', 'Ready-to-use component', and 'Export site', each accompanied by an illustrative icon. The bottom section is a 'Price calculator' with a progress indicator and a question 'What type of site do you want to create?'. The calculator offers radio button options for 'Landing page', 'Online shop', 'Site catalog', 'Partial site for your business', and 'Feedback page'. The footer contains the text '2021 - ONE CLICK UI'.

РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ ПЛАТФОРМИ І ВПЛИВ НА ПОКАЗНИКИ КОМПАНІЇ

10



ВИСНОВКИ

11

1. Досліджено технологію no-code, її вплив на швидкість розробки застосунків, та можливість використання технології бізнесом задля оптимізації використання ресурсів.
2. Проаналізовано існуючі рішення що надають можливість використовувати технологію no-code для створення застосунків. Проведено аналіз особливостей та недоліків існуючих no-code it-рішень, визначено ключові потреби бізнесу при створенні веб-застосунків, визначено ключові потреби при роботі з no-code платформами.
3. Розроблено методику швидкого створення застосунків за допомогою використання технології no-code та набору готових функціональних блоків і шаблонів.
4. Проведено дослідження впливу на фінансові показники бізнесу після впровадження методики і використання платформи no-code для створення застосунків. Дослідження виявило, що впровадження методики швидкого створення застосунків за допомогою no-code технології має значний вплив на фінансові показники бізнесу. Зокрема, спостерігається значна економія часу і ресурсів у порівнянні із традиційними методами розробки. Час на розробку було пришвидшено на 70%. Вартість створення застосунку суттєво зменшилася на 75%, що сприяє оптимізації бюджету підприємства.

Стаття:

1. Халецький В.С., Золотухіна О.А., Довженко Т.П., Бондарчук А.П. Оптимізація створення веб-застосунків для малого та середнього бізнесу на основі технології no-code розробки. Телекомунікаційні та інформаційні технології. №1, 2024. Подано до друку.

Тези доповідей:

2. Халецький В.С., Золотухіна О.А. Розробка методики швидкого створення веб-застосунків для малого та середнього бізнесу. // Науково-практична конференція «Проблеми комп'ютерної інженерії» – Подано до друку. Київ: ДУТ, 2022. С. 152-153.

ДЯКУЮ ЗА УВАГУ!