

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Інформаційна технологія розширення
функціональності платформи salesforce для
замовлення продукту у пробний період»**

Виконав: студент 2-го курсу,
групи 1КН-20м спеціальності 122 –
Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

Зарицький О.С.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Барабан С.В.

(прізвище та ініціали)

« ____ » _____ 2021 р.

Опонент: д.т.н., професор каф. ПЗ

Ліщинська Л.Б.

(прізвище та ініціали)

« ____ » _____ 2021 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« ____ » _____ 2021 р.

ЗАТВЕРДЖУЮ
Завідувач кафедри _____ КН _____
д.т.н., проф.. Яровий А.А.

(підпис)
“ _____ ” _____ 2021 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.015.20.537

Магістранта групи 1КН-20м Зарицького Олександра Сергійовича
(назва групи) (прізвище, ім'я і по батькові)

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія розширення функціональностей платформи Salesforce для замовлення продукту у пробний період»

Вхідні дані: мова програмування – JavaScript, базовий функціонал – Sales Force Commerce Cloud; для демонстрації можливостей розширень – SFRA Storefront; серверна частина – SFRA.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: схема алгоритму викупу або повернення продукту, схема алгоритму оформлення замовлення, схема алгоритму вибору продукту, етапи отримання продукту у пробний період

2. Текстова (пояснювальна записка): вступ, обґрунтування доцільності розробки інформаційної технології розширення функціональностей платформи Salesforce для замовлення продукту у пробний період; моделювання інформаційної технології розширення функціональностей платформи Salesforce для замовлення продукту у пробний період, Програмна реалізація інформаційної технології розширень функціональностей платформи Salesforce для замовлення продукту у пробний період, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Обґрунтування доцільності розробки			Розділ 1
2	Моделювання інформаційної технології			Розділ 2
3	Програмна реалізація інформаційної технології			Розділ 3
4	Підготовка економічної частини			Розділ 4
5	Апробація результатів дослідження			Тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ к.т.н, доц., кафедри КН
 (підпис) _____
 наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____
С. В. Барабан
 ініціали та прізвище

2. Економічна частина _____
 (підпис) _____
 наук. ступінь, вчене звання (посада)

 ініціали та прізвище
 “ ____ ” _____ 20__ р.
 Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____
 (підпис) _____
д.т.н., проф.кафедри ПЗ
 наук. ступінь, вчене звання (посада)
Л.Б. Ліщинська
 ініціали та прізвище

Завдання видав
 науковий керівник _____
 (підпис) _____
к.т.н, доц., кафедри КН
 наук. ступінь, вчене звання (посада)

С. В. Барабан
 ініціали та прізвище
 “ ____ ” _____ 20__ р.

Завдання отримав магістрант _____
 (підпис) _____
О. С. Зарицький
 ініціали та прізвище
 “ ____ ” _____ 20__ р.

АНОТАЦІЯ

У магістерській кваліфікаційній роботі проведено аналіз існуючого функціоналу Salesforce та запропонованих розширень. Проведено дослідження актуальних технологій які дозволяють працювати з eCommerce проектами, здійснено порівняння найпопулярніших рішень.

Дана робота допомагає проаналізувати існуючі можливості платформи Salesforce, та допомагає розширити платформу для додавання гнучкого та налаштовуваного функціоналу отримання продукту у пробний період.

Метою дослідження є розробка програмного модуля розширення платформи Salesforce.

В роботі приведені зразки з використанням функціоналу розроблених розширень та аналіз їх роботи та тестування.

ABSTRACT

In this Master's thesis the analysis of the existing functionality of the Salesforce and the offered extensions are carried out. A study of current technologies that allow working with eCommerce projects, a comparison of the most popular solutions.

This paper helps to analyze the existing capabilities of Salesforce platform, as well as to help expand the platform for adding configurable functionality of getting products in trial period.

The purpose of the study is to develop a software module to expand Salesforce platform.

The paper presents samples using the functionality of the developed extensions and analysis of their work and testi

ЗМІСТ

ВСТУП	8
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТЕЙ ПЛАТФОРМИ SALESFORCE ДЛЯ ОТРИМАННЯ ПРОДУКТУ У ПРОБНИЙ ПЕРІОД....	100
1.1 Аналіз предметної області отримання продукту у пробний період	100
1.2 Аналіз існуючих технологій для отримання продукту у пробний період.....	144
1.3 Постановка задачі дослідження.....	26
1.4 Висновки до розділу	26
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТЕЙ ПЛАТФОРМИ SALESFORCE ДЛЯ ОТРИМАННЯ ПРОДУКТУ У ПРОБНИЙ ПЕРІОД	27
2.1 Обґрунтування вибору мови програмування.....	27
2.2 Обґрунтування вибору клієнтських технологій	37
2.3 Обґрунтування вибору серверної технології	43
2.4 Розробка архітектури та алгоритмів інформаційної технології розширення функціональностей платформи Salesforce для отримання продукту у пробний період	45
2.5 Висновки	48
3 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТЕЙ ПЛАТФОРМИ SALESFORCE ДЛЯ ОТРИМАННЯ ПРОДУКТУ У ПРОБНИЙ ПЕРІОД	49
3.1 Програмна реалізація.....	49
3.2 Тестування інформаційної технології.....	56
3.3 Висновки	59
4 ЕКОНОМІЧНА ЧАСТИНА	60

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	60
4.2 Розрахунок витрат на здійснення науково-дослідної роботи роботи	62
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	67
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ.....	75
ДОДАТОК А Інструкція користувача	76
ДОДАТОК Б Лістинг вихідного коду	78
ДОДАТОК В Графічна частина.....	100

ВСТУП

Актуальність теми. Електронна комерція (від англ. Electronic commerce) — це сфера цифрової економіки, що включає всі фінансові та торгові транзакції, які проводяться за допомогою комп'ютерних мереж, та бізнес-процеси, пов'язані з проведенням цих транзакцій.

Основними складовими електронної комерції є мобільна комерція, електронний переказ коштів, управління ланцюгами поставок, Інтернет-маркетинг, обробка онлайн-транзакцій, електронний обмін даними (EDI), системи управління запасами та автоматизовані системи збору даних.

Активний розвиток цього напрямку зумовлений технологічними досягненнями напівпровідникової промисловості (електронної промисловості).

Електронна комерція бізнес-до-користувача, або торгівля між компаніями й споживачами, включає збирання інформації клієнтами, купівлю фізичних речей чи інформаційних або електронних товарів, і, для інформаційних товарів, одержування товару (програми, електронної книги) по електронній мережі. Це друга по величині й найперша форма електронної комерції. Приклади B2C моделей — мережні компанії продажу в роздріб типу Amazon.com, Drugstore.com, Beyond.com.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка спеціалізованих засобів штучного інтелекту на основі інтелектуального аналізу даних та машинного навчання» та плану наукової та навчально-методичної роботи кафедри.

Мета дослідження – розробка програмного модуля розширення функціональностей платформи Salesforce.

Об'єкт дослідження – процес розширення платформи Salesforce.

Предмет дослідження – методи та програмні засоби що надають змогу розробляти розширення.

Наукова новизна одержаних результатів полягає в розширенні функціональності платформи Salesforce для отримання продукту у пробний період.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням

математичних методів під час доведення наукових положень, чітким та лаконічним виведенням аналітичних співвідношень, порівнянням результатів з

відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано тези доповіді на конференції [1].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТЕЙ ПЛАТФОРМИ SALESFORCE ДЛЯ ОТРИМАННЯ ПРОДУКТУ У ПРОБНИЙ ПЕРІОД

1.1 Аналіз предметної області отримання продукту у пробний період

Робити покупки в інтернет-магазинах або відвідувати звичайний магазин - це вибір, з яким стикається кожен споживач після ухвалення рішення про покупку.

Чинники, що впливають на це рішення, можуть залежати від таких речей, скільки є часу і де користувач отримує максимальну віддачу, але зрештою рішення зводиться до зручності.

Для багатьох ця тенденція змістила увагу на покупки в Інтернеті. Цей варіант має багато переваг:

- 1) Не треба вставати з ліжка,
- 2) Можна вибрати доставку у найближчий час,
- 3) Є доступ до додаткової інформації про продукт,
- 4) Можна мати більш індивідуальний досвід покупок.
- 5) Індустрія електронної комерції та онлайн-шопінгу зростає швидкими темпами.

Очікується, що лише у Сполучених Штатах до 2023 року продажі електронної комерції перевищать 740 мільярдів доларів.

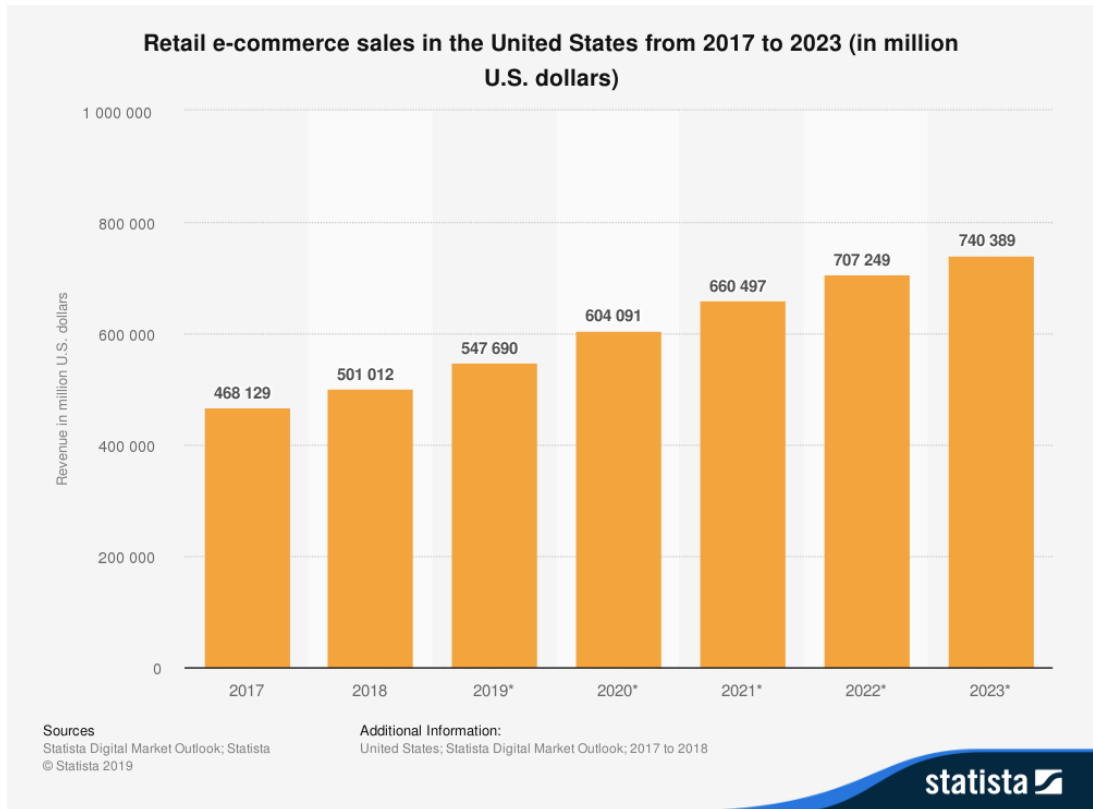


Рисунок 1.1 – Прогнози продажі у електронній комерції США на 2023 рік

Зростання покупок в Інтернеті є наслідком багатьох факторів, таких як:

- 1) Збільшення використання смартфонів та мобільних покупок,
- 2) Соціальні мережі та соціальна комерція,
- 3) Перетворююча технологія,
- 4) Інтернет-майданчики,
- 5) Те, як купівельна поведінка змінювалося від покоління до покоління – від бeбi-бумерів до мiленiалів та покоління Z.

З цієї причини відстеження тенденцій в електронній торгівлі може допомогти вам розробити передові стратегії, які допоможуть вам досягти успіху в електронній торгівлі як у найближчому, так і далекому майбутньому.

59% покупців у США купували одяг через Інтернет у період з другого кварталу 2018 року до другого кварталу 2019 року. З великим відривом

відстають книги, фільми, музика та ігри, взуття та побутова електроніка.

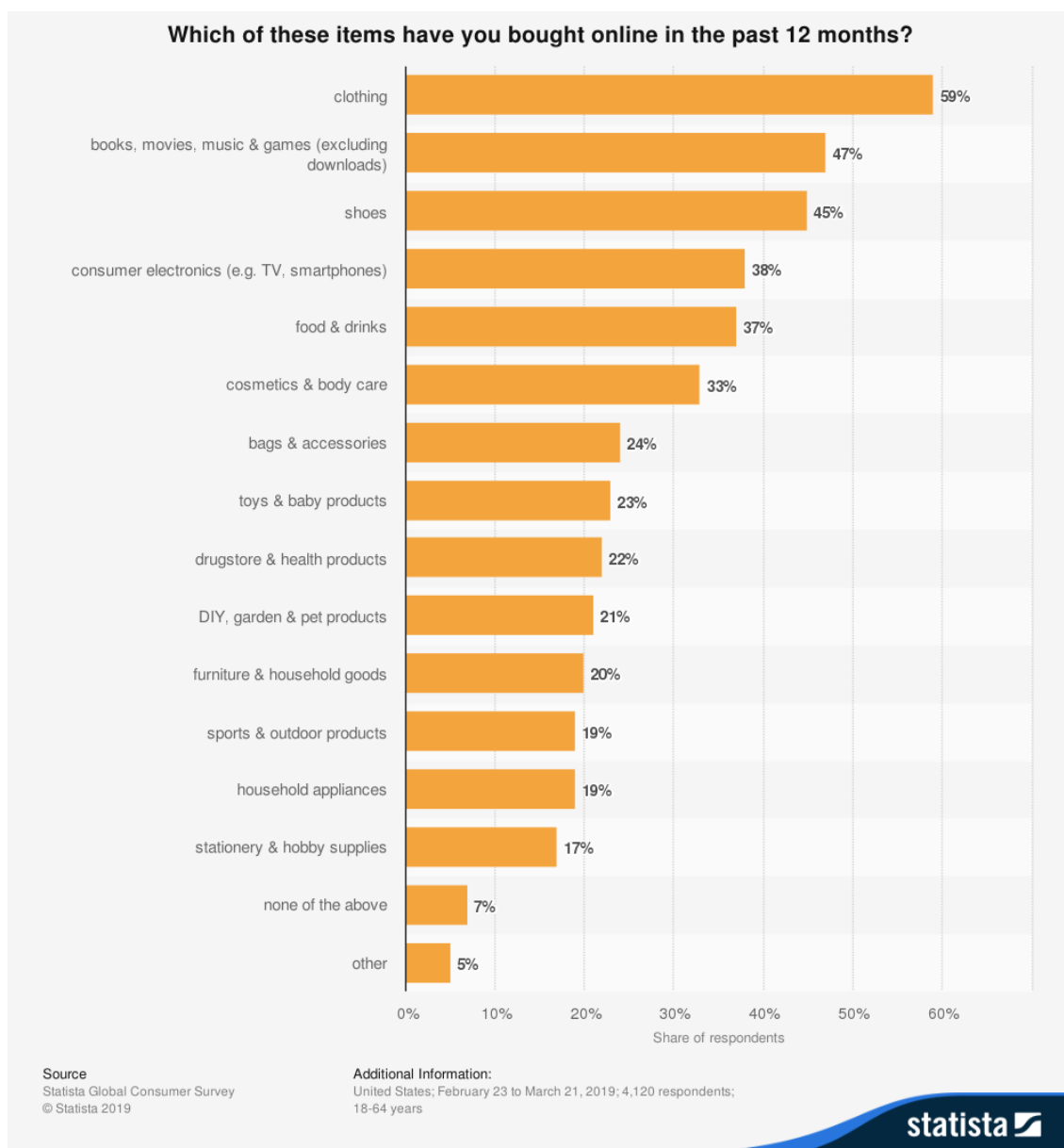


Рисунок 1.2 – Статистика онлайн купівель у США

Успіх електронної комерції – це більше, ніж просто красивий веб-сайт – потрібен хороший обсяг трафіку для інтернет-магазину. У недавньому опитуванні Statista прямі посилання на сайт, електронну пошту та пошукову

оптимізацію були визнані провідними драйверами трафіку.

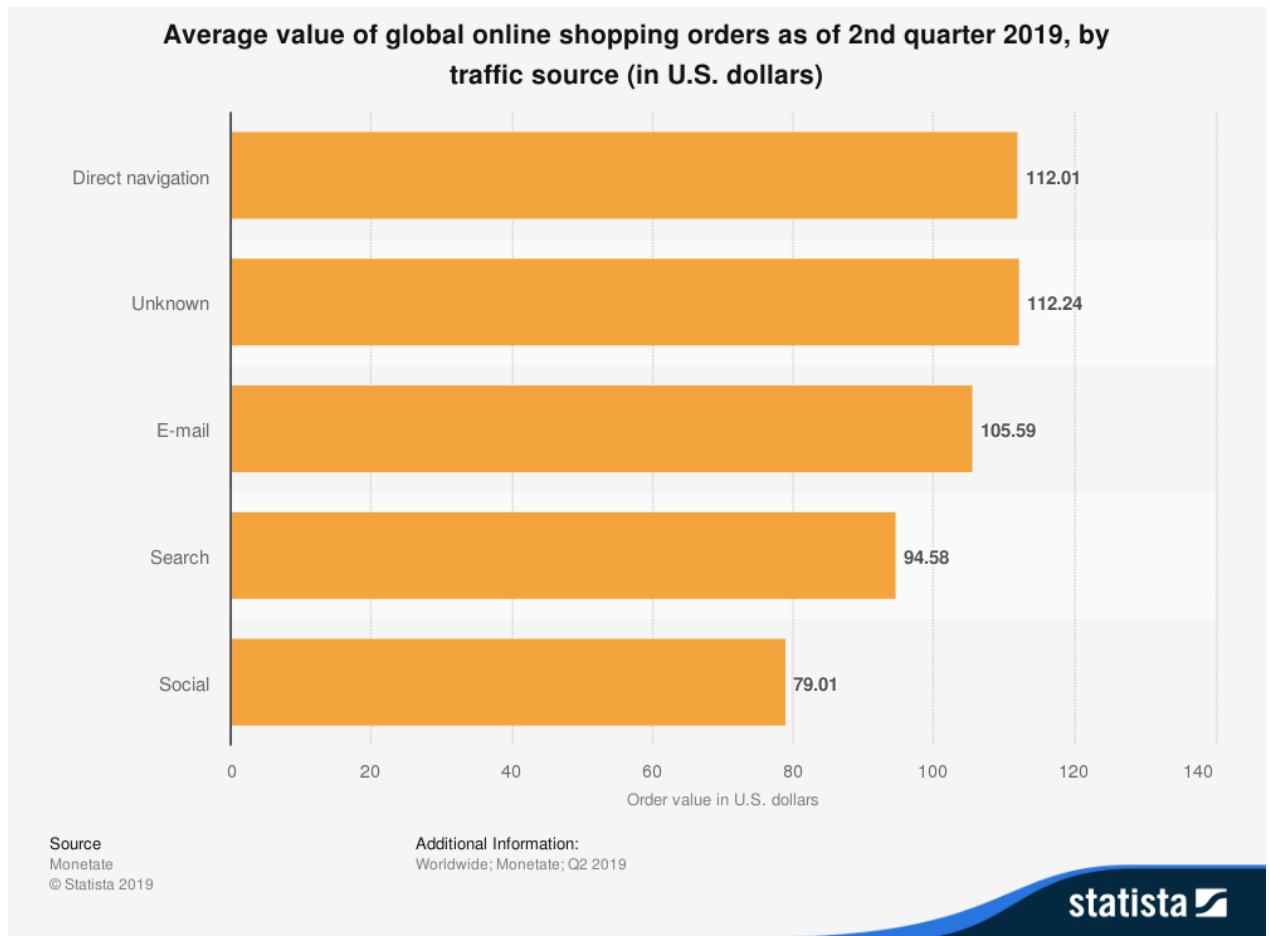


Рисунок 1.3 – Статистика трафіку інтернет-магазину

Епідемія коронавірусу відкрила несподівані можливості для інтернет-магазинів. Наприклад, онлайн-продаж Sun Art Retail Group, найбільшого китайського оператора гіпермаркетів, майже подвоївся під час Китайського Нового року порівняно з аналогічним періодом минулого року. За словами виконавчого директора Пітера Хуанга, близько 80% із 486 гіпермаркетів, що знаходяться в управлінні Sun Art, були закриті в розпал епідемії 4 лютого, проте їх доходи залишалися стабільними, оскільки мешканці перейшли до замовлення свіжих продуктів у режимі онлайн.

Такі продукти, як свіжі продукти, які раніше продавалися в офлайн, отримали величезне зростання онлайн-замовлень. За словами французького ритейлера Carrefour, постачання овочів збільшилося на 600% порівняно з минулим роком у період Китайського Нового року. Китайський онлайн-

ритейлер JD.com повідомив, що його онлайн-продаж зріс на 215% порівняно з аналогічним періодом минулого року - до 15 000 тонн протягом 10-денного періоду з кінця січня до початку лютого. За даними аналітичного сайту Sensor Tower, програми для доставки продуктів харчування отримали величезний сплеск. Fresh Nema, додаток для доставки продуктів, що належить китайському технологічному гіганту Alibaba, 8 лютого завантажили 100 000 разів, що більш ніж утричі перевищує середній щоденний показник установки – 29 000.

У зв'язку з різким підвищенням онлайн купівель потрібен механізм, який буде розвивати лояльність покупців до інтернет-магазину. Таким механізмом є отримання продукту у пробний період, він дозволяє не тільки утримати певних покупців, а ще охопити додаткові об'єми клієнтів, які не були готові до покупок, або були не впевнені в якості товару.

1.2 Аналіз існуючих технологій для отримання продукту у пробний період

Salesforce Commerce Cloud

Demandware – компанія-розробник програмного забезпечення зі штаб-квартирою в місті Берлінгтон, штат Массачусетс, яка забезпечує хмарну уніфіковану платформу електронної комерції з мобільними пристроями, персоналізацією штучного інтелекту, можливостями управління замовленнями та супутніми послугами для роздрібних торговців B2C та B2B та виробників брендів з усього світу.

Компанія Demandware була заснована в лютому 2004 року Стефаном Шамбахом для створення, що дозволить компаніям розробляти та керувати простими у користуванні, настроюваними веб-сайтами електронної комерції, а не створювати їх з нуля. Послуга була запущена в першому кварталі 2005 року. Раніше Шамбах заснував ранній інтернет-магазин Intershop у 1992 році.

Початкові гроші для Demandware надали фірми венчурного капіталу General Catalyst Partners та North Bridge Venture Partners.

15 березня 2012 року компанія Demandware розпочала торгівлю на Нью-Йоркській фондовій біржі, залучивши 88 мільйонів доларів у своєму первинному розміщенні в розмірі 16 доларів за акцію. Після IPO акції зросли більш ніж на 50% від ціни IPO наступного ранку. У листопаді 2013 року компанія Demandware оголосила про публічне розміщення публічних акцій.

Salesforce – це платформа, яка повністю хоститься на серверах компанії Salesforce у клауді. Компанія Salesforce була заснована у 1999 році колишнім виконавчим директором Oracle Марком Беніоффом. Головна ідея створення — це побудова доступного програмного забезпечення та впровадження його повністю онлайн як сервіс.

Salesforce вже давно вийшла за межі просто CRM. Salesforce – це клауд-платформа, на базі якої, крім CRM частини, є багато цікавого. Salesforce дозволяє створювати та розгортати індивідуальні рішення, автоматизувати бізнес-процеси, інтегруватися із зовнішніми додатками.

Більшість світових ентерпрайз-компаній є клієнтами компанії Salesforce і використовують цю платформу як рішення для своїх бізнес-потреб. Серед них: Adidas, AWS, Canon, Philips, Toyota, American Express, Western Union, Cisco, KLM та багато інших. Візовий центр США використовує Salesforce для збирання запитів на візи, їхньої обробки та видачі.

Salesforce, як компанія, є партнером різноманітних представників IT-світу і, відповідно, вони пропонують готові інтеграційні рішення, удосконалені сервіси для задоволення вимог кінцевих клієнтів та ще багато різних можливостей. Серед них – Apple, Microsoft, Google, Amazon.

На 2019 рік Salesforce пропонує цілий набір продуктів: Sales Cloud, Service Cloud, Marketing Cloud, e-Commerce Cloud, Heroku, Integration, Community Cloud, Einstein Analytics та інші.

Кілька фактів:

- 1) Salesforce – перший номер у списку «100 найкращих компаній для роботи за 2018 рік» за версією Fortune.
- 2) Salesforce лідирує у списку найбільш інноваційних компаній за версією Forbes у 2017 році та №3 у 2018 році.
- 3) Salesforce – найкраща CRM за версією G2 Crowd Grid.
- 4) Salesforce займає найвищу позицію у списку CRM систем за версією Software Advice.

Більш ніж 150 000 компаній використовують Salesforce CRM для розвитку свого бізнесу.

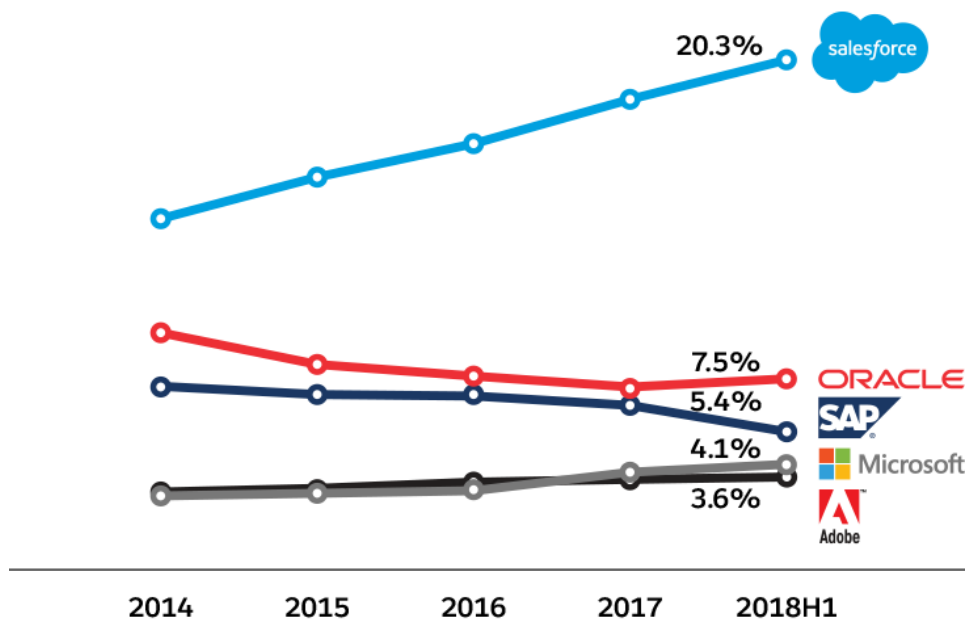


Рисунок 1.4 – Всесвітні CRM-додатки 2018H1 частка ринку IDC

Простий початок роботи одна з найбільших переваг Salesforce. Не потрібно встановлювати жодних програм, немає ніяких вимог до обладнання. Розміщення в хмарі надає доступ до інформації в режимі реального часу, тобто ви можете користуватися платформою будь-де і будь-коли.

Для роботи з Salesforce потрібно мати лише доступ до інтернету. Все, що потрібно, — це перейти в браузері на сайт, ввести свій логін і пароль. Можна створювати власні функціонали або користуватися стандартними, уже створеними Salesforce.

Усі користувачі Salesforce мають спільну інфраструктуру та екземпляр програмного забезпечення. Це дає можливість автоматичного та одночасного оновлення для всіх користувачів на платформі. Найсвіжіші та найновіші функції з автоматичним оновленням користувачі отримують тричі на рік.

Кожен має можливість створити безкоштовну організацію для навчання, практикування чи тестування. Це фактично повнофункціональне середовище для розробки, яке можна також використовувати для створення власних продуктів — пакетів, які можуть випускатися як додатки в AppExchange.

AppExchange у Salesforce дозволяє розробляти та продавати власні продукти або отримувати доступ до тисяч корисних, захищених та перевірених продуктів чи інтеграцій, створених іншими користувачами. Наприклад, `apex-lang` — це відкрита бібліотека допоміжних класів, написаних на Apex, метою якої є усунення недоліків у основних класах платформи. Пакет `Rollup Helper` дозволяє агрегувати дані за певним набором критеріїв у Salesforce без використання Apex коду.

Salesforce дає можливість для будь-якої інтеграції. Крім того, пропонується багато вбудованих інтеграцій — наприклад, з Heroku, Outlook, Gmail.

Двостороння інтеграція між Salesforce і .NET аплікацією для менеджменту і синхронізації звернень клієнтів. Для вхідних викликів використовується `apex REST` веб-сервіс. Для вихідних викликів використовується `Apex callout` — HTTP request з Apex коду.

Інтуїтивна і зрозуміла панель навігації. Всі вкладки розділені залежно від категорії. Для доступу до налаштувань чи кастомних розробок використовується Setup меню. Меню розділене на різні секції в залежності від мети використання. Наприклад, налаштування об'єктів і полів відбувається через Object Manager. Створити, переглянути чи редагувати Apex клас можна через Platform Tools -> Custom Code.

Salesforce-спеціаліст має можливості зростання і розвитку як в технічному напрямку, так і в менеджменті проектів і консалтингу. Хороших Salesforce фахівців сьогодні небагато. Статистика показує, що рівень заробітної плати Salesforce розробника вищий, ніж, до прикладу, у Java або .NET спеціаліста, якщо говорити про однаковий досвід.

Однією з найбільших переваг Salesforce є те, що велика кількість функціоналу, яку вимагають для більшості продуктів, уже імплементована і протестована. Наприклад, вбудований функціонал Email-to-Case дозволяє автоматично створювати Case записи в системі з електронної пошти. Або Web-to-Lead, який створює HTML форму, яку можна вставити на свій веб-сайт і створювати записи лідів безпосередньо з сайту (Lead — потенційний клієнт в розумінні SF).

Значну частину функціоналу можна просто конфігурувати без єдиної лінійки коду за короткий період часу. А це піднімає Salesforce на рівень вище порівняно з традиційною розробкою з нуля. Розробка з нуля несе набагато більші ризики для клієнта і займає набагато більше часу.

Salesforce — це платформа, яка постійно змінюється і розвивається, даючи вам нові можливості для вивчення і вдосконалення. Компанія займає активну позицію стосовно стартапів. У 2010 році була придбана одна з перших хмарних платформ Heroku, яка дозволяє розгортати, запускати та управляти додатками, написаними на різних відкритих мовах і фреймворках.

Протягом останніх років Salesforce активно інтегрує елементи штучного інтелекту (AI) в свою платформу. Технологія використовує всі дані в межах системи: дані клієнта, інформацію з листувань, електронної пошти, календаря та електронної комерції, потоки даних із соціальних мереж, такі як твіти та фотографії, для створення моделей машинного навчання. Один з прикладів AI в Salesforce — функціонал Einstein Bots for Service Cloud, який використовує машинне навчання для кращого та швидшого обслуговування клієнтів. Це дозволяє передавати запити клієнтів ботам, які застосовують

поєднання машинного навчання та обробки історії клієнтів для прийняття рішення.

Salesforce Trailhead — це онлайн-платформа для вивчення Salesforce, яка дає можливість дотримуватися вже готових способів навчання або обрати свій шлях — і все у своєму власному темпі.

Magento



Рисунок 1.5 – логотип Magento

Magento — eCommerce платформа з відкритим кодом, призначена для створення інтернет-магазинів. Поширюється за ліцензією OSL 3.0.

Magento — одна з найпопулярніших відкритих систем для організації електронної комерції в мережі: на базі цієї платформи створено понад 100 тисяч інтернет-магазинів, сторонніми розробниками створено більше 2 тисяч розширень, спільнота проєкту налічує близько 375 тисяч учасників, код платформи був завантажений більше 2,5 млн разів, через системи на базі Magento в 2010 році було продано товарів на 25 мільярдів доларів. За кілька років свого існування платформа кілька разів була лауреатом премій «Best of Open Source Software Awards» і «SourceForge Community Choice Awards». Платформа Magento Commerce складає близько 30% загальної частки ринку.

Це програмне забезпечення створене з використанням Zend Framework. Сирцевий код Magento поширюються в рамках схваленої організацією Open Source Initiative (OSI) ліцензії Open Software License (OSL) v3.0, близької за своєю суттю до ліцензії AGPL, але не сумісної з GPL. Особливість ліцензії OSL полягає в тому, що якщо хтось модифікує програму, але не буде поширювати результат своєї роботи, а просто задіює його на своєму сайті, то він все одно зобов'язаний відкрити код внесених змін і поширювати його під тією ж ліцензією. При цьому ліцензія OSL не забороняє комерційне використання продуктів.

Freshworks



Рисунок 1.5 – логотип Freshworks

Freshworks - це програмне забезпечення CRM, що здобуло безліч нагород та рекомендацій від таких платформ, як G2, GetApp, Gartner і Capterra, в основному через його ефективну функціональність.

У вас є одна програма, щоб правильно спрямовувати всі ваші відносини з клієнтами, узгоджуючи ваші команди продажів та маркетингу.

Freshworks використовує розширені можливості штучного інтелекту, щоб зробити CRM максимально простий. Він стає більш адаптивним у міру того, як його використовують, та надає аналітичні дані та прогнози на основі ваших даних CRM. Завдяки цьому можна розуміти клієнтів та легко приймати розумніші рішення.

Крім отримання кращих потенційних клієнтів і спілкування на платформі , вдосконалені функції штучного інтелекту Freshwork також допомагають краще взаємодіяти з клієнтами за допомогою персоналізованих і заснованих на поведінці розмов .

Загалом Freshworks CRM - відмінне програмне забезпечення, яке дає вам все, що потрібно знати про всіх клієнтів, а також можливість створювати звіти для підвищення загальної ефективності бізнесу.

NetSuite CRM



Рисунок 1.6 – логотип NetSuite CRM

NetSuite CRM - це хмарне програмне забезпечення CRM, яке пропонує швидке перетворення потенційних клієнтів за допомогою вбудованої веб-форми у форми потенційних клієнтів. Веб-форми для лідів автоматично надають торговому персоналу детальну інформацію про ліди та іншу важливу інформацію про потенційного клієнта, як тільки вони її заповнюють. Це

робиться в режимі реального часу, а це означає, що навіть якщо ваші потенційні клієнти не надішлють форму, інформація про них все одно буде отримана.

Ці форми дозволяють швидко та ефективно спрямовувати потрібних маркетингових пропозицій потрібним торговим представникам за допомогою потужного механізму маршрутизації. Торговий персонал отримує можливість оперативно реагувати, що, своєю чергою, збільшує коефіцієнт конверсії потенційних клієнтів.

Продуктивність продажів також може бути покращена за допомогою детального аналізу взаємодій за допомогою телефонних дзвінків, завдань, зустрічі та нотатки.

Netsuite CRM також допомагає визначити рентабельність інвестицій у поточні проекти. Подібна функція дає правильні варіанти управління ризиками та допомагає визначити вмайбутню політику щодо інвестицій. Завдяки цьому можна знайти цільові області для маркетингу залежно від розташування.

В цілому, Netsuite надає гнучке хмарне програмне забезпечення CRM, яке пропонує розширені можливості, просте в навігації та допомагає покращити всю вашу маркетингову стратегію за допомогою аналітичних графіків та діаграм.

ActiveCampaign

The logo for ActiveCampaign is displayed on a solid blue rectangular background. The text "ActiveCampaign" is written in a white, sans-serif font, followed by a white chevron symbol pointing to the right.

Рисунок 1.7 – логотип ActiveCampaign

ActiveCampaign надає можливість додати ще більше 850 рішень щодо взаємодії з клієнтами, безкоштовні функції автоматизації. Його численні функції автоматизації дозволяють легко взаємодіяти з аудиторією.

Вбудована CRM, цільові сторінки, карта автоматизації, дизайнер листів з перетягуванням, безкоштовні попередньо відформатовані шаблони листів та можливості роздільної автоматизації для досягнення маркетингових цілей.

Є доступ до даних, які дозволяють оцінювати контакти та потенційних клієнтів, функцій прогнозування повідомлень, автовідповідачам, докладним звітам та можливостям відстеження GEO.

ActiveCampaign відповідає вимогам HIPAA та пропонує широкі можливості інтеграції з різними платформами в Інтернеті. Можна автоматично імпортувати файли з інших сервісів, відстежувати події на зовнішньому веб-сайті, а також користуватися інтеграцією з індивідуальною аудиторією Facebook та багатьма іншими.

ActiveCampaign надає настроювану URL-адресу програми, можливість ребрендингу згенерованих форм, домени поштового сервера, що настроюються, а також налаштований номер SMS для відправки повідомлень і негайного розпізнавання контактами.

В цілому ActiveCampaign дає можливість автоматизувати бізнес та отримати доступ до важливих рішень з продажу, обслуговування та маркетингу для зміцнення відносин із клієнтами.

Zoho CRM



Рисунок 1.8 – логотип Zoho CRM

Zoho CRM патронують понад 150 000 компаній у 180 країнах світу та високо оцінені PCMag за ефективні мобільні програми. Його сервісам довіряють такі популярні бренди, як Amazon, Netflix, Suzuki та багато інших.

Це CRM-сервіс для підприємств, що покладаються на віддалене робоче середовище. Як хмарна платформа для спільної роботи служба CRM надає просте у використанні програмне забезпечення корпоративного рівня, яке можна налаштувати та використовувати з будь-якого місця та в будь-який час.

Дозволяє керувати глобальною командою продажів за допомогою функції віддаленого офісу. Це надбудова, яка пропонує інтеграцію для зустрічей онлайн. Можна здійснювати групові аудіо та відео дзвінки, забезпечуючи оптимальне спілкування.

Персонал може вести візуальні бесіди з клієнтами в режимі реального часу, а також може відтворити досвід проведення фізичної індивідуальної презентації пропозиції.

Zoho CRM дає оповіщення в режимі реального часу, коли хтось відвідує веб-сайт, відкриває електронну пошту або згадує бренд у будь-якій соціальній мережі.

Insightly

The logo for Insightly, featuring the word "insightly" in a lowercase, rounded, orange-red script font.

Рисунок 1.9 – логотип Insightly

Insightly спрощує автоматизацію бізнес-процесів, оскільки дозволяє зробити бізнес унікальним за допомогою CRM, що настроюється, що відповідає конкретним потребам.

Організовує процеси продажу за допомогою дій з управління контактами, потенційними клієнтами та можливостями, використовуючи поля, що настроюються, і динамічні макети сторінок .

Є єдина платформа, яка дозволяє об'єднати маркетингові та CRM-рішення. Потоки даних про взаємини з клієнтами збираються на єдиній платформі без шансів бути втраченими чи пошкодженими.

Функція "Залучення клієнтів" представляє розумні пропозиції "наступної найкращої дії" відповідно до даних, вивчених на основі попередніх взаємин з клієнтами.

Є доступ до показників залученості, таких як відкритість, показник відмов, рейтинг кліків та багато іншого завдяки функції інтелектуальних кампаній Insightly. Надаються докладні звіти, а також можливості A/B-тестування для тестування та визначення ефективних кампаній.

Можете створювати електронні листи за допомогою попередньо відформатованих та легко настроюваних шаблонів, щоб покращити маркетинг. Можна покращити маркетингові кампанії з клієнтами за допомогою двосторонніх каналів зв'язку, відкритих та вбудованих на веб-сайт та цільові сторінки.

Найактуальніша інформація про потенційних клієнтів відстежується шляхом вивчення джерела попередньої маркетингової кампанії, електронних листів, телефонних дзвінків, зустрічей та завдань. Ліди також автоматично надсилаються відповідному торговому представнику, доки вони все ще можуть бути використані.

Insightly дозволяє створювати робочий простір для бізнесу, що настроюється, за рахунок синхронізованої інтеграції із зовнішніми платформами, такими як SAP, Oracle та інші. Інші інтегровані платформи

включають 8×8, ActiveCampaign, Adobe Analytics, Adobe Experience Manager та багато інших.

1.3 Постановка задачі дослідження

Задача даної роботи полягає у розробці інформаційної технології розширення функціональностей платформи salesforce для замовлення продукту у пробний період, а саме:

- Аналіз актуальності теми
- Аналіз предметної області
- Аналіз систем аналогів
- Розробка архітектури та алгоритмів інформаційної технології
- Програмна реалізація
- Економічна частина дослідження.

1.4 Висновки до розділу

У даному розділі було проаналізовано розробку інформаційну технологію отримання продукту у пробний період, детально розглянуто системи аналогів, проведено постановку задачі дослідження та підведено підсумки.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТЕЙ ПЛАТФОРМИ SALESFORCE ДЛЯ ОТРИМАННЯ ПРОДУКТУ У ПРОБНИЙ ПЕРІОД

2.1 Обґрунтування вибору мови програмування

Будь-яка Java

Мова програмування Java була розроблена компанією Sun Microsystems на початку 1990-х років. Хоча вона в основному використовується для Інтернет-додатків, Java є простою, ефективною мовою загального призначення. Спочатку Java була розроблена для вбудованих мережеских програм, що працюють на кількох платформах. Це портативна, об'єктно-орієнтована, інтерпретована мова.

Java надзвичайно портативна. Одна й та сама програма Java працюватиме однаково на будь-якому комп'ютері, незалежно від апаратних можливостей чи операційної системи, якщо в ній є інтерпретатор Java. Крім портативності, ще однією з ключових переваг Java є набір функцій безпеки, які захищають ПК, на якому запущена програма Java, не тільки від проблем, викликаних помилковим кодом, але й від шкідливих програм (наприклад, вірусів). Можна безпечно запускати аплет Java, завантажений з Інтернету, оскільки функції безпеки Java перешкоджають цим типам аплетів отримати доступ до жорсткого диска ПК або мережеских з'єднань. Аплет, як правило, є невеликою програмою на Java, яка вбудована в сторінку HTML.

Java можна вважати як компільованою, так і інтерпретованою мовою, оскільки її вихідний код спочатку компілюється у двійковий байт-код. Цей байт-код працює на віртуальній машині Java (JVM), яка зазвичай є програмним інтерпретатором. Використання скомпільованого байт-коду дозволяє інтерпретатору (віртуальній машині) бути невеликим і ефективним (і майже

таким же швидким, як і процесор, який виконує власний скомпільований код). Крім того, цей байт-код надає Java портативності: він працюватиме на будь-якій JVM, яка правильно реалізована, незалежно від комп'ютерної апаратної чи програмної конфігурації. Більшість веб-браузерів (наприклад, Microsoft Internet Explorer або Netscape Communicator) містять JVM для запуску Java-аплетів.

У порівнянні з C++ (іншою об'єктно-орієнтованою мовою), код Java працює трохи повільніше (через JVM), але він більш портативний і має набагато кращі функції безпеки. Віртуальна машина забезпечує ізоляцію між ненадійною програмою Java та ПК, на якому запущено програмне забезпечення. Синтаксис Java схожий на C++, але мови зовсім інші. Наприклад, Java не дозволяє програмістам реалізувати перевантаження операторів, тоді як C++ це робить. Крім того, Java є динамічною мовою, де ви можете безпечно змінювати програму під час її виконання, тоді як C++ цього не дозволяє. Це особливо важливо для мережевих програм, які не можуть дозволити собі простої. Крім того, всі основні типи даних Java є попередньо визначеними і не залежать від платформи, тоді як деякі типи даних можуть змінюватися в залежності від платформи, що використовується в C або C++ (наприклад, тип `int`).

Програми на Java мають більш високу структуру, ніж еквіваленти C++. Усі функції (або методи Java) і виконувані оператори в Java повинні знаходитися в межах класу, тоді як C++ дозволяє визначенням функцій і рядкам коду існувати за межами класів (як у програмах у стилі C). Глобальні дані та методи не можуть перебувати за межами класу в Java, тоді як C++ дозволяє це. Ці обмеження, хоча часом і громіздкі, допомагають підтримувати цілісність і безпеку програм Java і змушують їх бути повністю об'єктно-орієнтованими.

Ще однією ключовою особливістю Java є те, що це відкритий стандарт із загальнодоступним вихідним кодом. Sun Microsystems контролює мову Java

та пов'язані з нею продукти, але ліберальна політика ліцензування Sun сприяла тому, що Інтернет-спільнота прийняла Java як стандарт.

```
public class AverageProgram    // start of class definition
{
    public static void main(String[] args)
        // start of method definition
    {
        int npoints, counter, acc, average;    // declare variables

        System.out.println("Enter the number of points to average: ");
        npoints = ConsoleIn.readInt();    // read npoints
        counter = 0;    // initialize variables
        acc = 0;
        while (counter < npoints)
        {
            // start of while loop
            System.out.println("Enter value: ");
            acc = acc + ConsoleIn.readInt();    // add in current value
            counter = counter + 1;    // increment counter
        }    // end of while loop
        average = acc / npoints;    // calculate average
        System.out.println("Average value = " + average);
        // display result
    }    // end of method definition
}    // end of class definition
```

Рисунок 2.1 – Приклад коду JAVA

C#

C# — сучасна об'єктно-орієнтована мова програмування загального призначення, розроблена компанією Microsoft і схвалена Європейською асоціацією виробників комп'ютерів (ЕСМА) та Міжнародною організацією зі стандартів (ISO).

Був розроблений Андерсом Хейлсбергом та його командою під час розробки .Net Framework.

C# розроблено для загальномовної інфраструктури (CLI), яка складається з виконуваного коду та середовища виконання, що дозволяє використовувати різні мови високого рівня на різних комп'ютерних платформах та архітектурах.

Наступні причини роблять C# широко вживаною професійною мовою:

- 1) Це сучасна мова програмування загального призначення
- 2) Він об'єктно-орієнтований.
- 3) Він компонентно-орієнтований.
- 4) Навчитися легко.

- 5) Це структурована мова.
- 6) Він створює ефективні програми.
- 7) Його можна компілювати на різних комп'ютерних платформах.
- 8) Він є частиною .Net Framework.

Хоча конструкції C# тісно відповідають традиційним мовам високого рівня, C і C++ є об'єктно-орієнтованою мовою програмування. Він дуже схожий на Java, має численні потужні функції програмування, які роблять його привабливим для багатьох програмістів у всьому світі.

Нижче наведено список кількох важливих функцій C#:

- 1) Булеві умови
- 2) Автоматичний збір сміття
- 3) Стандартна бібліотека
- 4) Версійна збірка
- 5) Властивості та події
- 6) Делегати та управління подіями
- 7) Прості у використанні генерики
- 8) Індексатори
- 9) Умовна компіляція
- 10) Проста багатопоточність
- 11) LINQ і лямбда-вирази
- 12) Інтеграція з Windows

```

1514 public function getColorProductListByProductIds($product_rs,$product_sales,$product_relation){
1515     $colorids = array();
1516     $rs = array();
1517     if (!empty($product_rs)) {
1518         $noTotalProducts = array ();
1519         foreach ( $product_rs as $vs ) {
1520             if (!empty($product_sales)) {
1521                 foreach ( $product_sales as $ps ) {
1522                     if ($ps ["product_id"] == $vs ["product_id"]) {
1523                         $rs [$vs ["product_id"]] [$ps ["color_code_id"]] = $ps;
1524                         $rs [$vs ["product_id"]] [$ps ["color_code_id"]] ["stock"] = 0;
1525                         if ($ps ["is_active"] == 1 || $ps ["has_stock"] == 1) {
1526                             $rs [$vs ["product_id"]] [$ps ["color_code_id"]] ["stock"] = 1;
1527                         }
1528                     }
1529                 }
1530             }
1531         }
1532         if (empty ( $rs [$vs ["product_id"]] [$product_relation [$vs ["product_id"]] ["color_code_id"]] )) {
1533             $rs [$vs ["product_id"]] [$product_relation [$vs ["product_id"]] ["color_code_id"]] = $product_relation [$vs ["product_id"]];
1534         }
1535     }
1536
1537     // this is order list by totle sale
1538     $rlist_order_sale = array ();
1539     foreach ( $rs as $kr => $vr ) {
1540         $rlist_order_sale [$kr] = $this->sortSaleRand ( $vr );
1541     }
1542
1543     // this is order list by stock,totle sale,color_code_id
1544     $rlist_order_stock_sale = array ();
1545     foreach ( $rs as $kr => $vr ) {
1546         $rlist_order_stock_sale [$kr] = $this->sortSaleStockRand ( $vr );
1547     }

```

Рисунок 2.2 – Приклад коду C#

C++

C++ — це статично типізована, скомпільована, універсальна, чутлива до регістру, мова програмування вільної форми, яка підтримує процедурне, об'єктно-орієнтоване та загальне програмування.

Вважається мовою середнього рівня, оскільки містить комбінацію мов високого та низького рівня.

C++ був розроблений Б'ярном Страуструпом, починаючи з 1979 року в Bell Labs в Мюррей-Хілл, штат Нью-Джерсі, як доповнення до мови C і спочатку називався C with Classes, але пізніше він був перейменований в C++ в 1983 році.

Є надмножиною C, і практично будь-яка легальна програма C є легальною програмою C++.

Повністю підтримує об'єктно-орієнтоване програмування, включаючи чотири стовпи об'єктно-орієнтованої розробки:

- 1) Інкапсуляція
- 2) Приховування даних
- 3) Спадкування
- 4) Поліморфізм
- 5) Стандартні бібліотеки

Стандартний C++ складається з трьох важливих частин.

Основна мова, що містить усі будівельні блоки, включаючи змінні, типи даних, літерали тощо.

Стандартна бібліотека C++, що надає багатий набір функцій для керування файлами, рядками тощо.

Стандартна бібліотека шаблонів (STL), що надає багатий набір методів для маніпулювання структурами даних тощо..

```
#include <cstdio>

int main()
{
    // 1. prompt user to enter coefficients
    std::puts("please enter a and b for `ax + b = 0`");

    // 2. read coefficients for equation `ax + b = 0`
    float a = 0;
    float b = 0;
    std::scanf("%f %f", &a, &b);

    // 3. solve equation `ax + b = 0`
    // solution: `x = -b / a`
    float x = -b / a;
    std::printf("solution: %f\n", x);
}
```

Рисунок 2.3 – Приклад коду C++

Python

Python — це інтерпретована, інтерактивна, об'єктно-орієнтована мова програмування загального призначення та високорівневої мови. Він був створений Гвідо ван Россумом у 1985-1990 роках. Як і Perl, вихідний код Python також доступний під загальною суспільною ліцензією GNU (GPL). Цей підручник дає достатньо розуміння мови програмування Python.

Високорівнева, інтерпретована, інтерактивна й об'єктно-орієнтована мова сценаріїв. Python розроблений так, щоб він був добре читабельним. Він часто використовує ключові слова англійською мовою, де, як і інші мови, використовуються розділові знаки, і він має менше синтаксичних конструкцій, ніж інші мови.

Деякі з ключових переваг вивчення Python.

Python інтерпретується — Python обробляється під час виконання інтерпретатором. Вам не потрібно компілювати вашу програму перед її виконанням. Це схоже на PERL і PHP.

Є інтерактивним — ви насправді можете сидіти за підказкою Python і безпосередньо взаємодіяти з інтерпретатором, щоб писати свої програми.

Є об'єктно-орієнтованим – Python підтримує об'єктно-орієнтований стиль або техніку програмування, що інкапсулює код в об'єкти.

Характеристики Python.

Він підтримує функціональні та структуровані методи програмування, а також ООП.

Його можна використовувати як мову сценаріїв або компілювати в байт-код для створення великих програм.

Він забезпечує дуже високорівневі динамічні типи даних і підтримує динамічну перевірку типів.

Він підтримує автоматичний збір сміття.

Його можна легко інтегрувати з C, C++, COM, ActiveX, CORBA та Java.

```
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.ensemble import GradientBoostingClassifier

#Sorting hat classification

StudentsData = pd.read_sql("SELECT * FROM StudentsData", engine)

X = pd.drop('faculty')
Y = pd['faculty']

X_train, X_test, y_train, y_test =
train_test_split(X, Y, test_size=0.33, random_state=42)

Classifier = GradientBoostingClassifier(n_estimators=500)
Classifier.fit(X_train, y_train)

y_prediction = Classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_prediction)
print("Model accuracy = " + accuracy_score
```

Рисунок 2.4 – Приклад коду Python

PHP

PHP Hypertext Preprocessor (PHP) — це мова програмування, яка дозволяє веб-розробникам створювати динамічний вміст, який взаємодіє з базами даних. PHP в основному використовується для розробки веб-додатків.

PHP починався як невеликий проект з відкритим кодом, який розвивався, оскільки все більше і більше людей дізнавалися, наскільки він корисний. Расмус Лердорф випустив першу версію PHP ще в 1994 році.

PHP є рекурсивним акронімом від «PHP: Hypertext Preprocessor».

Це мова сценаріїв на стороні сервера, яка вбудована в HTML. Він використовується для керування динамічним вмістом, базами даних, відстеження сеансів, навіть для створення цілих сайтів електронної комерції.

Він інтегрований з низкою популярних баз даних, включаючи MySQL, PostgreSQL, Oracle, Sybase, Informix і Microsoft SQL Server.

PHP підтримує велику кількість основних протоколів, таких як POP3, IMAP і LDAP. У PHP4 додано підтримку Java та архітектур розподілених об'єктів (COM і CORBA), що вперше зробило n-рівневу розробку можливою.

Синтаксис PHP схожий на C.

PHP є однією з найбільш широко використовуваних мов в Інтернеті.

Виконує системні функції, тобто з файлів у системі він може створювати, відкривати, читати, записувати та закривати їх.

Може обробляти форми, тобто збирати дані з файлів, зберігати дані у файл, через електронну пошту ви можете надсилати дані, повертати дані користувачеві.

Надає доступ до змінних файлів cookie та встановлення файлів cookie.

Він може шифрувати дані.

```

1 <?hh
2 class MyClass {
3     public function alpha(): int {
4         return 1;
5     }
6
7     public function beta(): string {
8         return 'hi test';
9     }
10 }
11
12 function f(MyClass $my_inst): string {
13     // Fix me!
14     return $my_inst->alpha();
15 }

```

Рисунок 2.5 – Приклад коду Python

JavaScript

JavaScript — це легка інтерпретована мова програмування. Він призначений для створення мережових додатків. JavaScript дуже простий у реалізації, оскільки він інтегрований з HTML. Він відкритий і кросплатформний.

Javascript є найпопулярнішою мовою програмування у світі, і це робить її чудовим вибором для програміста.

Javascript допомагає створювати дійсно красиві та швидкі веб-сайти.

Використання JavaScript поширилося на розробку мобільних додатків, розробку додатків для настільних комп'ютерів та розробку ігор. Це відкриває багато можливостей для програміста Javascript.

У зв'язку з високим попитом для тих, хто знає JavaScript, існує величезна кількість робочих місць і висока оплата.

Існує багато корисних фреймворків і бібліотек Javascript:

- 1) Angular.js
- 2) React
- 3) jQuery
- 4) Vue.js
- 5) Ext.js
- 6) Ember.js
- 7) Node.js
- 8) Backbone.js

Javascript є однією з найбільш широко використовуваних мов програмування (Front-end, а також Back-end). Він присутній майже в кожній області розробки програмного забезпечення.

Перевірка на стороні клієнта. Це дійсно важливо для перевірки будь-якого введення користувача перед відправкою його на сервер, і Javascript відіграє важливу роль у перевірці цих введених даних на самому інтерфейсі.

Маніпулювання сторінками HTML - Javascript допомагає маніпулювати сторінками HTML на льоту. Це допомагає дуже легко додавати та видаляти будь-який HTML-тег за допомогою javascript і змінювати ваш HTML, щоб змінити його зовнішній вигляд на основі різних пристроїв і вимог.

Внутрішнє завантаження даних - Javascript надає бібліотеку Ajax, яка допомагає завантажувати внутрішні дані.

Презентації – JavaScript також надає можливість створювати презентації, які надають веб-сайту зовнішній вигляд. JavaScript надає бібліотеки RevealJS і BespokeJS для створення веб-презентацій слайдів.

Серверні програми – Node JS побудований на середовищі виконання Chrome Javascript для створення швидких і масштабованих мережеских програм. Це бібліотека на основі подій, яка допомагає розробляти дуже складні серверні програми, включаючи веб-сервери.

Salesforce Commerce Cloud використовує JavaScript та JavaScript-подібні мови програмування.

```

1
2 var image = new Image();
3 image.addEventListener("click", bitWolf, false);
4
5 function createPicture(){
6     var timer= window.setInterval("tick()", 150);
7 }
8
9 function bitWolf(event){
10     getGoal();
11     tick();
12 }
13
14 // window.clearTimeout
15 function tick(){
16     image.src = "wolf.png";
17     var imageX= getNumber(4);
18     var imageY=getNumber(3);
19     var idPic ="pic".concat(imageX,imageY);
20     document.getElementById(idPic).appendChild(image);
21 }
22
23 function getNumber(max_number){
24     return Math.floor(Math.random()*max_number);
25 }
26
27 function getGoal(){
28     var element = document.getElementById("number").innerHTML;
29     var num = Number(element) + 1;
30     document.getElementById("number").innerHTML= num;
31 }

```

Рисунок 2.6 – Приклад коду JavaScript

Проаналізувавши та порівнявши мови програмування, у роботі буде використано мову програмування JavaScript.

2.2 Обґрунтування вибору клієнтських технологій

AngularJS

AngularJS — це дуже потужна платформа JavaScript. Він використовується в проектах Single Page Application (SPA). Він розширює HTML DOM додатковими атрибутами і робить його більш чуйним до дій користувача. AngularJS є відкритим вихідним кодом, повністю безкоштовним і використовується тисячами розробників по всьому світу. Він ліцензований під ліцензією Apache версії 2.0.

AngularJS — це фреймворк веб-додатків з відкритим кодом. Спочатку він був розроблений у 2009 році Міско Хевері та Адамом Абронсом. Зараз його підтримує Google. Його остання версія – 1.2.21.

AngularJS надає розробникам можливість писати програми на стороні клієнта за допомогою JavaScript чистим контролером представлення моделі (MVC).

Програми, написані на AngularJS, сумісні з різними браузерами. AngularJS автоматично обробляє код JavaScript, який підходить для кожного браузера.

AngularJS є відкритим вихідним кодом, повністю безкоштовним і використовується тисячами розробників по всьому світу. Він ліцензований під ліцензією Apache версії 2.0.

Загалом, це платформа для створення великомасштабних, високопродуктивних і простих у обслуговуванні веб-додатків.ції – JavaScript також надає можливість створювати презентації, які надають веб-сайту

зовнішній вигляд. JavaScript надає бібліотеки RevealJS і BespokeJS для створення веб-презентацій слайдів.

```

1 var app = angular.module('app', []);
2 app.directive('parentDir1', function () {
3     return {
4         restrict: 'EA',
5         template: '<div>{{name1}}<child1></child1></div>',
6         compile: function(tElem, tAttrs){
7             console.log(': compile called');
8         },
9         link: {
10            post: function(scope, elem, attr){
11                console.log(': Pre called');
12                scope.name1 = 'Parent directive Without pre link';
13                scope.parentName1 = 'Parent 1';
14            }
15        }
16    };
17 });

```

Рисунок 2.7 – Приклад коду AngularJS

ReactJs

ReactJS — це проста, багатофункціональна бібліотека інтерфейсу JavaScript на основі компонентів. Її можна використовувати для розробки невеликих програм, а також великих, складних програм. ReactJS надає мінімальний і надійний набір функцій для запуску веб-програми. Спільнота React надає великий набір готових компонентів для розробки веб-додатків у рекордно короткий термін. Спільнота React також надає розширені концепції, такі як керування станом, маршрутизація тощо, поверх бібліотеки React.

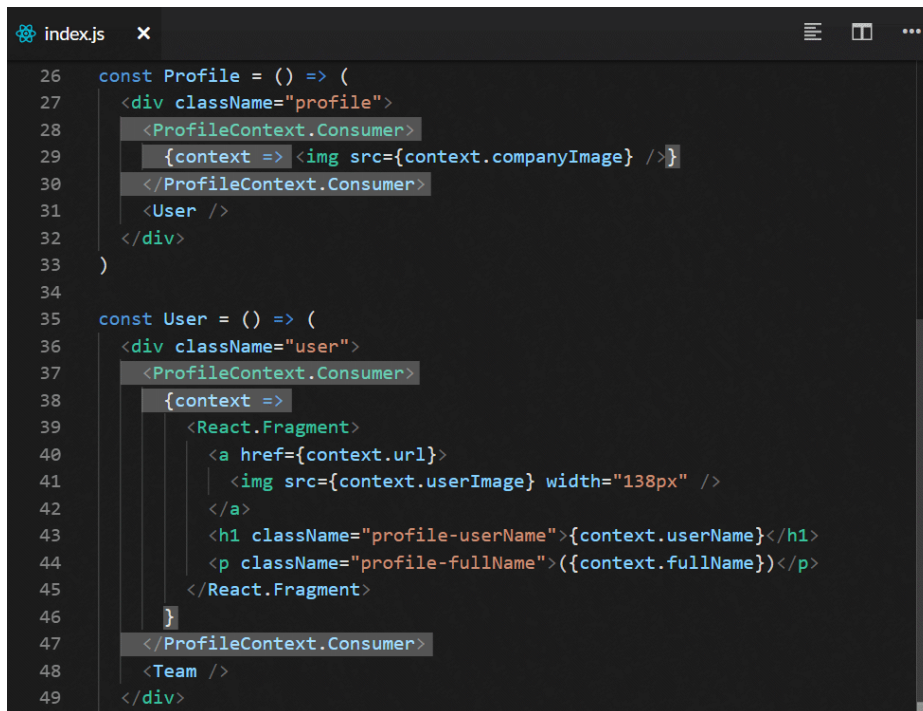
Початкова версія, 0.3.0 React, випущена в травні 2013 року, а остання версія, 17.0.1, випущена в жовтні 2020 року. Основна версія вносить критичні зміни, а другорядна версія вводить нову функцію, не порушуючи існуючу функціональність.

Основні особливості бібліотеки React полягають у наступному:

- 1) Суцільна базова архітектура
- 2) Розширювана архітектура
- 3) Бібліотека на основі компонентів
- 4) Архітектура дизайну на основі JSX
- 5) Декларативна бібліотека інтерфейсу користувача

Кілька переваг використання бібліотеки React полягають у наступному:

- 1) Легко навчитися
- 2) Легко освоїти як сучасні, так і застарілі програми
- 3) Швидший спосіб кодування функціоналу
- 4) Наявність великої кількості готових компонентів



```

26 const Profile = () => (
27   <div className="profile">
28     <ProfileContext.Consumer>
29       {context => <img src={context.companyImage} />}
30     </ProfileContext.Consumer>
31     <User />
32   </div>
33 )
34
35 const User = () => (
36   <div className="user">
37     <ProfileContext.Consumer>
38       {context =>
39         <React.Fragment>
40           <a href={context.url}>
41             <img src={context.userImage} width="138px" />
42           </a>
43           <h1 className="profile-userName">{context.userName}</h1>
44           <p className="profile-fullName">({context.fullName})</p>
45         </React.Fragment>
46       }
47     </ProfileContext.Consumer>
48     <Team />
49   </div>

```

Рисунок 2.8 – Приклад коду ReactJS

VueJS

VueJS — це прогресивний JavaScript-фреймворк з відкритим кодом, який використовується для розробки інтерактивних веб-інтерфейсів. Це один із відомих фреймворків, що використовуються для спрощення веб-розробки. VueJS зосереджується на шарі перегляду. Його можна легко інтегрувати у великі проекти для розробки інтерфейсу без будь-яких проблем.

Розпочати встановлення VueJS дуже легко. Будь-який розробник може легко зрозуміти та створити інтерактивні веб-інтерфейси за лічені хвилини. VueJS створено Еваном Ю, колишнім співробітником Google. Перша версія VueJS була випущена в лютому 2014 року. Нещодавно вона отримала 64 828 зірок на GitHub, що робить її дуже популярною.

VueJS використовує віртуальний DOM, який також використовується іншими фреймворками, такими як React, Ember тощо. Зміни в DOM не вносяться, замість цього створюється копія DOM, яка представлена у вигляді структур даних JavaScript. Щоразу, коли потрібно внести будь-які зміни, вони вносяться в структури даних JavaScript, і остання порівнюється з вихідною структурою даних. Остаточні зміни потім оновлюються до реального DOM, який користувач побачить, як змінюється. Це добре з точки зору оптимізації, це дешевше, а зміни можна вносити швидше.

Функція зв'язування даних допомагає маніпулювати або призначати значення атрибутам HTML, змінювати стиль, призначати класи за допомогою директиви зв'язування, яка називається `v-bind`, доступної у VueJS.

Компоненти є однією з важливих функцій VueJS, яка допомагає створювати власні елементи, які можна повторно використовувати в HTML.

`v-on` — це атрибут, доданий до елементів DOM для прослуховування подій у VueJS.

VueJS надає різні способи застосування переходу до елементів HTML, коли вони додаються/оновлюються або видаляються з DOM. VueJS має вбудований компонент переходу, який потрібно обернути навколо елемента для ефекту переходу. Ми можемо легко додати сторонні бібліотеки анімації, а також додати більше інтерактивності в інтерфейс.

Це одна з важливих особливостей VueJS. Він допомагає прослухати зміни, внесені в елементи інтерфейсу користувача, і виконує необхідні обчислення. Для цього не потрібно додаткового кодування.

VueJS надає шаблони на основі HTML, які пов'язують DOM з даними екземпляра `Vue`. `Vue` компілює шаблони у віртуальні функції візуалізації DOM. Ми можемо використовувати шаблон функцій візуалізації, і для цього нам потрібно замінити шаблон на функцію візуалізації.

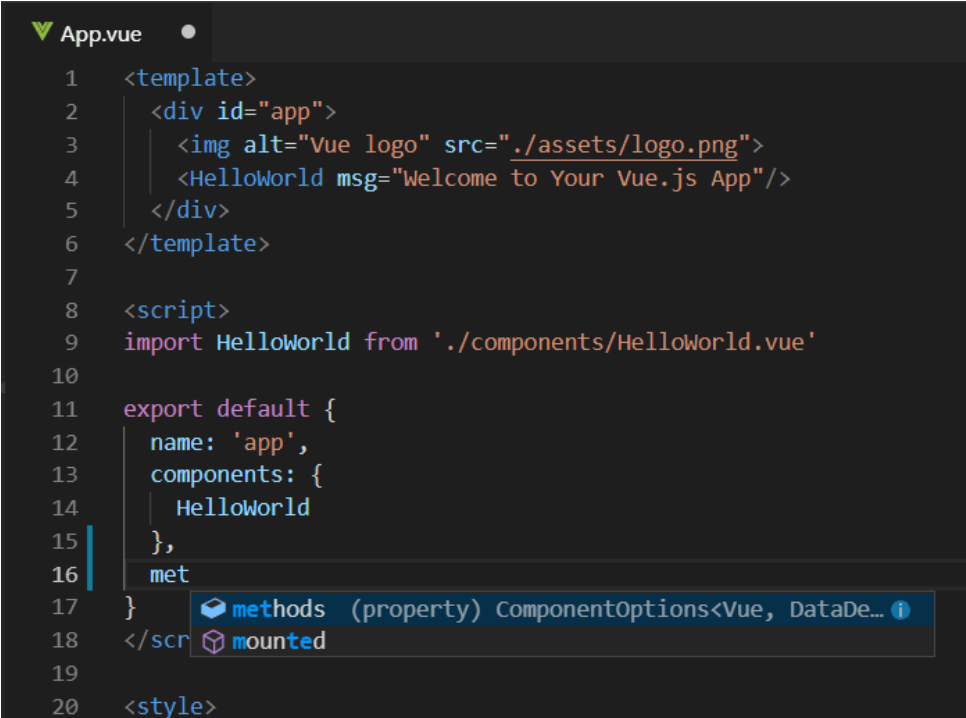
VueJS має вбудовані директиви, такі як `v-if`, `v-else`, `v-show`, `v-on`, `v-bind` і `v-model`, які використовуються для виконання різних дій у інтерфейсі.

Спостерігачі застосовуються до даних, які змінюються. Наприклад, елементи введення форми. Не потрібно додавати додаткові події. Watcher піклується про обробку будь-яких змін даних, що робить код простим і швидким.

Навігація між сторінками здійснюється за допомогою vue-router.

Сценарій VueJS дуже легкий, а продуктивність також дуже швидка.

VueJS можна встановити в командному рядку за допомогою інтерфейсу командного рядка vue-cli. Це допомагає легко побудувати та зібрати проект за допомогою vue-cli.



```

1 <template>
2   <div id="app">
3     
4     <HelloWorld msg="Welcome to Your Vue.js App"/>
5   </div>
6 </template>
7
8 <script>
9 import HelloWorld from './components/HelloWorld.vue'
10
11 export default {
12   name: 'app',
13   components: {
14     HelloWorld
15   },
16   met
17 }
18 </scr
19
20 <style>

```

Рисунок 2.9 – Приклад коду ReactJS

jQuery

jQuery — це швидка та лаконічна бібліотека JavaScript, створена Джоном Резігом у 2006 році з гарним девізом: пишіть менше, робіть більше. jQuery спрощує перехід по документу HTML, обробку подій, анімацію та взаємодію Ajax для швидкої веб-розробки. jQuery — це набір інструментів JavaScript, призначений для спрощення різних завдань шляхом написання менше коду. Ось список важливих основних функцій, які підтримує jQuery.

Маніпулювання DOM — jQuery спрощує вибір елементів DOM, узгодження їх та зміну їх вмісту за допомогою міжбраузерного механізму вибору з відкритим кодом під назвою Sizzle.

Обробка подій – jQuery пропонує елегантний спосіб фіксувати широкий спектр подій, наприклад, коли користувач натискає посилання, без необхідності захищувати сам HTML-код обробниками подій.

Підтримка AJAX — jQuery дуже допомагає вам розробити адаптивний і багатофункціональний сайт за допомогою технології AJAX.

Анімація — jQuery поставляється з великою кількістю вбудованих ефектів анімації, які ви можете використовувати на своїх веб-сайтах.

Дуже легка бібліотека — розміром близько 19 КБ (мініфікована та заархівована).

Підтримка між браузерами — jQuery має міжбраузерну підтримку і добре працює в IE 6.0+, FF 2.0+, Safari 3.0+, Chrome і Opera 9.0+

Новітні технології — jQuery підтримує селектори CSS3 та основний синтаксис XPath.

```
$("#selectCulture").change(function () {  
    // Set culture from select list  
    $.preferCulture(this.value);  
  
    // Format dates  
    $("span[data-date]").html(function () {  
        var dateString = $(this).attr("data-date");  
        var dateValue = $.parseDate(dateString, "d", "en-US");  
        return $.format(dateValue, "D");  
    });  
});
```

Рисунок 2.10 – Приклад коду jQuery

Після аналізу та порівняння фронт-енд технологій було обрано jQuery, так як це дуже проста та швидка технологія, шаблонний сайт на Salesforce Commerce Cloud платформі використовує jQuery за замовчуванням і розроблювана інформаційна технологія не потребує великих обсягів фронт-енд роботи.

2.3 Обґрунтування вибору серверної технології

SiteGenesis

SiteGenesis JavaScript Controllers (SGJC) — це демонстраційний довідковий додаток для електронної комерції, який дає змогу вивчити Salesforce B2C Commerce та його можливості. Ви можете використовувати його як основу свого власного сайту, хоча SFRA рекомендується для нових проектів.

Функції, характерні для SGJC, можуть бути незнайомими, якщо система базується на попередній версії SiteGenesis. Більшість функцій зосереджені на повторному використанні коду і можуть бути корисними під час міграції коду.

Модулі CommonJS дозволяють створювати сценарії з багаторазовою функціональністю. Модуль — це файл .ds або .js. Модуль можна розмістити в папці сценарію картриджа або в папці модулів, яка знаходиться на тому ж рівні, що й інші картриджі. Ви можете отримати доступ до модулів, які знаходяться у вашому картриджі, в інших картриджах і в папці модулів.

Salesforce B2C Commerce підтримує шляхи CommonJS до модулів доступу та відносні шляхи.

Використовуйте наступний синтаксис:

- 1) ~ - назва поточного картриджа. Приклад: `require('~~/cartridge/scripts/guard')`
- 2) . - та сама папка (як у CommonJS). Приклад: `require('./shipping')`;
- 3) .. - батьківська папка (як у CommonJS). Приклад: `require('../..util')`

Хуки налаштовують частину функціональності для виклику в певній точці потоку програми або під час певної події.

З SiteGenesis можна використовувати три типи хуків:

Хуки OCAPI – B2C Commerce надає точки розширення, які дозволяють автоматично викликати сценарії до або після певних викликів OCAPI.

Хуки B2C Commerce – B2C Commerce надає хуки `onSession` і `onRequest` для заміни Pipeline `onSession` і `onRequest`. Він також надає гачки для Apple Pay в Інтернеті.

Спеціальні хуки – ви можете визначити користувацькі точки розширення та викликати їх у коді свого магазину за допомогою методів класу `HookMgr` системного пакету B2C Commerce.

Визначення хука.

Файл `package.json` вказує на файл `hook` для картриджа, використовуючи ключове слово `hooks`.

Файл-хук визначає точку розширення з унікальною назвою та сценарій для запуску. Хук-скрипти повинні бути реалізовані як модулі `CommonJS`. Таким чином, ідентифікатор сценарію є ідентифікатором модуля і може бути відносним шляхом або будь-яким іншим дійсним ідентифікатором модуля.

SFRA

Storefront Reference Architecture – найновіша версія платформ, яка має весь функціонал SiteGenesis та дозволяє дуже гнучко кастомізувати сайт та перевикористовувати різні картриджі

Storefront Reference Architecture надає клієнтам еталонну архітектуру, яка служить відправною точкою для створення веб-магазину. Еталонна архітектура знаходиться за межами рівня API платформи і повністю налаштовується клієнтом.

Використовується ExpressJs-подібна модель розробки контролерів JavaScript та моделлю налаштування, призначеною для полегшення командам розробників використовувати підхід MVC.

Також використовується Bootstrap, щоб оптимізувати покупки на мобільних пристроях, водночас забезпечуючи чудовий досвід роботи на настільному сайті.

Після аналізу та порівняння северних технологій було обрано SFRA, так як ця технологія у всьому краща за аналоги.

2.4 Розробка архітектури та алгоритмів інформаційної технології розширення функціональностей платформи Salesforce для отримання продукту у пробний період

Процес отримання продукту у пробний період можна розділити на декілька етапів

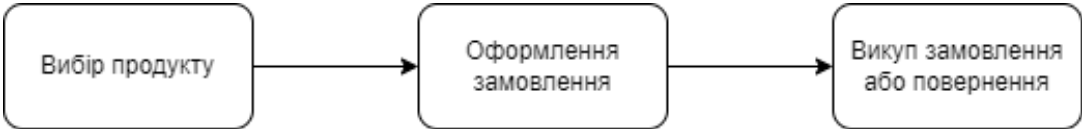


Рисунок 2.11 – Етапи отримання продукту у пробний період

Кожен з цих етапів можна описати за допомогою схем алгоритмів.

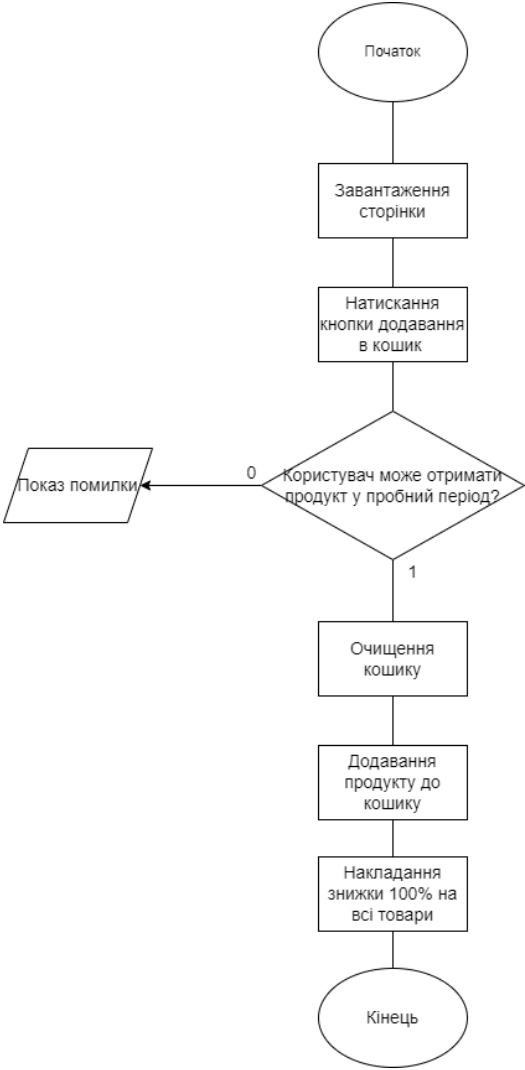


Рисунок 2.13 – Схема алгоритму вибору продукту

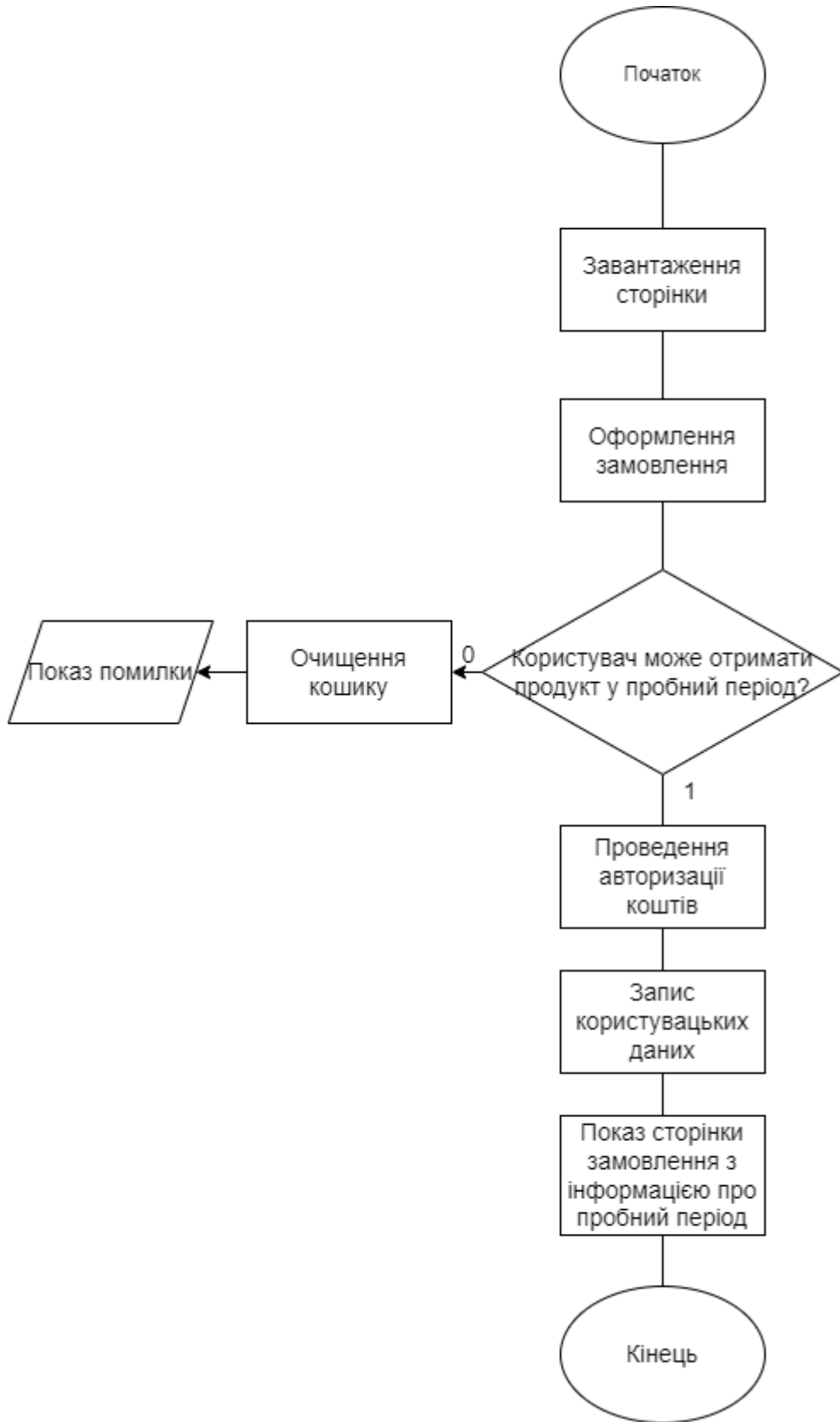


Рисунок 2.14 – Схема алгоритму оформлення замовлення



Рисунок 2.15 – Схема алгоритму викупу або повернення продукту

Потрібно написати програмний код який буде перевіряти замовлення у пробному періоді з певною періодичністю. Кошти мають бути списані у разі, коли у користувача закінчився пробний період і він ще не повернув продукт. Потрібно написати ендпоінт який буде приймати дані зі стороннього сервісу про оффлайн замовлення.

2.5 Висновки

У даному розділі було обгрунтовано вибір мови програмування, фронтенд та серверної технологій, розроблено та описано архітектуру, алгоритми та модулі інформаційної технології та підведено підсумки.

Основна мова програмування JavaScript. JQuery було обрано як клієнтську технологію. SFRA було обрано як серверну технологію.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТЕЙ ПЛАТФОРМИ SALESFORCE ДЛЯ ОТРИМАННЯ ПРОДУКТУ У ПРОБНИЙ ПЕРІОД

3.1 Програмна реалізація

Сторінку вибору продукту буде реалізовано за допомогою PageDesigner.

PageDesigner — це функція B2C Commerce, яка підтримує стандартні процеси розробки та інструменти для створення багаторазових типів сторінок і типів компонентів. Можна використовувати візуальний редактор у Business Manager для розробки, планування й публікації спеціальних сторінок сайту, перетягуючи типи сторінок і компонентів. Можна використовувати одні й ті самі типи сторінок і компонентів для створення різних сторінок незалежно, не звертаючись до розробника за оновленнями.

Типовий процес впровадження PageDesigner включає ці кроки.

Розробник створює багаторазові типи сторінок і типи компонентів, які підтримують вимоги.

Розробник завантажує на сайт користувацький картридж, який містить типи сторінок і компонентів, а також інший користувальницький картридж, що містить підтримуючі артефакти інтерфейсу користувача.

Продавець використовує візуальний редактор Page Designer у Business Manager, щоб створювати сторінки, використовуючи типи сторінок і компонентів, які реалізував розробник.

Для кожного типу сторінки та компонента потрібен файл мета-визначення JSON. Файл мета-визначення для типу сторінки описує регіони сторінки, де можна розміщувати компоненти. Файл мета-визначення для типу компонента описує атрибути, які визначаються під час використання типу компонента, а також може визначити регіони всередині типу компонента.

Потрібно помістити файл мета-визначення в один з наступних каталогів спеціального картриджа або будь-який довільний підкаталог у цих каталогах:

Типи сторінок: `<your_cartridge>/cartridge/experience/pages`

Типи компонентів: `<your_cartridge>/cartridge/experience/components`

Ім'я файлу мета-визначення може включати лише буквено-цифрові символи або символи підкреслення. Якщо ви помістили файл мета-визначення в підкаталог у каталогах `/experience/pages` або `/experience/components`, назви рівнів у підкаталозі також мають використовувати лише буквено-цифрові символи або символи підкреслення.

Щоб переглянути файли схеми, які описують форматування JSON для файлу мета-визначення.

Цей приклад файлу мета-визначення типу сторінки реклами під назвою `promorage.json` визначає три регіони для сторінки рекламних акцій: верхній, основний та нижній. Регіони верхнього та нижнього колонтитулів можуть одночасно відображати лише один компонент, як зазначено `max_components = 1`. Визначення `region` для основного регіону не містить значення `max_components`, що вказує на те, що основна область може відображати будь-яку кількість компонентів. Основний регіон використовує `component_type_exclusion`, щоб вказати, що компоненти типу банер не дозволені в головному регіоні.

Значення `max_components` обмежує кількість компонентів, які відображаються для регіону за раз, але регіон може містити будь-яку кількість компонентів. Тому ви можете відображати різні компоненти для різних користувачів або за різними графіками. Наприклад, ви можете мати область банера, яка відображає один компонент взимку і інший компонент влітку.

```

promopage.json
{
  "name": "Promopage",
  "description": "A promotional page.",
  "region_definitions": [
    {
      "id": "header",
      "name": "Name",
      "description": "Header of the promopage",
      "max_components": 1
    },
    {
      "id": "main",
      "name": "Main",
      "description": "Main area of the promopage",
      "component_type_exclusions": [
        { "type_id": "banner" }
      ]
    },
    {
      "id": "footer",
      "name": "Footer",
      "description": "Footer of the promopage",
      "max_components": 1
    }
  ]
}

```

Рисунок 3.1 – Приклад мета-файлу сторінки

Цей приклад файлу мета-визначення типу компонента банера під назвою `banner.json` визначає чотири атрибути: файл зображення, його альтернативний текст, його розмір та атрибут під назвою `Magic Unicorn`. Атрибут `Magic Unicorns` налаштовується за допомогою спеціального елемента керування інтерфейсом користувача `com.sfcc.magical`. У візуальному редакторі атрибути відображаються в групах. Наприклад, чотири атрибути, визначені в цьому файлі, відображаються у візуальному редакторі в групі з написом «Конфігурація зображення банера».

Параметр `default_value` для атрибута використовується лише для відтворення сторінки у вітрині магазину або попереднього перегляду у візуальному редакторі. Параметр `default_value` не відображається як попередньо вибраний у візуальному редакторі, який продавець використовує

для налаштування компонента. Поведінка візуалізації на основі налаштувань для обов'язкових і `default_value` виглядає так:

Обов'язкове значення `true` і налаштовано значення `default_value`: атрибут `default_value` призначається для відтворення сторінки, але не відображається як попередньо вибраний у візуальному редакторі. Якщо продавець не налаштує значення для атрибута і не збереже компонент, повідомлення про помилку не відображається.

Обов'язкове значення `false` і налаштовано значення `default_value`: атрибут `default_value` призначається для відтворення сторінки, але не відображається як попередньо вибраний у візуальному редакторі. Якщо продавець не налаштує значення для атрибута і не збереже компонент, повідомлення про помилку не відображається.

Обов'язкове значення `true`, і значення `default_value` не налаштовано: якщо продавець не налаштує значення для атрибута та збереже компонент, з'явиться повідомлення про помилку.

Обов'язкове значення `false` і значення `default_value` не налаштовано: якщо продавець не налаштує значення для атрибута і не збереже компонент, повідомлення про помилку не відображається.

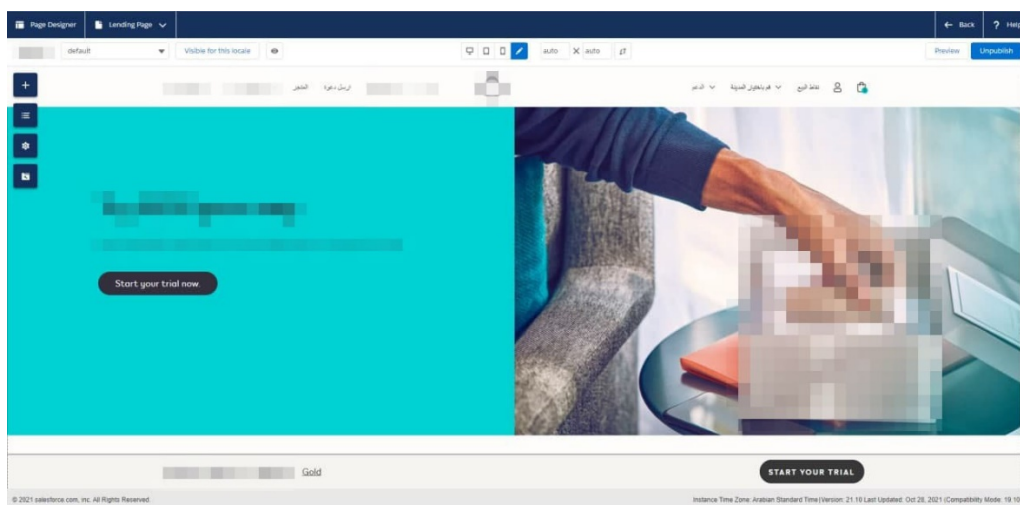


Рисунок 3.2 – Інтерфейс PageDesigner

Для реалізації технології потрібно зорширити декілька системних об'єктів.

За допомогою модуля System Object Type можна:

- 1) Переглянути всі типи системних об'єктів.
- 2) Переглянути всі атрибути, визначені для системного об'єкта.
- 3) Відредагуйте атрибути, включаючи створення нових значень атрибутів.
- 4) Переглянути груп атрибутів.
- 5) Створити нові групи атрибутів.

Окрім додавання спеціального атрибута, ви також можна вибрати, як згрупувати ці атрибути за допомогою цього самого модуля.

Системні об'єкти можуть зберігатися глобально, що робить екземпляри доступними для всієї організації та всіх сайтів, або локально (на сайт). Сам тип системного об'єкта завжди доступний для всієї організації.

Виберіть Administration > Site Development > System Object Types.

На сторінці System Object Type оберіть назву об'єкта, який потрібно редагувати.

Натисніть вкладку Attribute Definitions.

На вкладці Attribute Definitions натисніть New.

На сторінці Attribute Definition введіть або виберіть таку інформацію:

ID: введіть унікальний ідентифікатор для атрибута. Ми рекомендуємо не використовувати лише буквено-цифрові символи для ідентифікатора.

Можна використовувати такі спеціальні символи в ідентифікаторі атрибута: «+», «-», «\$», «.», «%», «§», «&», «*», «#», «/».

Відображуване ім'я: введіть відображуване ім'я. Це ім'я, яке відображається в Business Manager.

Тип значення: виберіть тип даних. Тип даних змінює інші параметри конфігурації для атрибута після його застосування.

Додаткові поля з'являються залежно від вибраного типу даних.

У таблиці нижче наведено зміни у системних об'єктах.

Таблиця 3.1 – Нові атрибути у системних об'єктах

Об'єкт	Атрибути
ProductLineItem	isLendingProductLineItem lendingProductLineItemRegion
Order	lendingFullyPaid lendingExtended isLendingOrder lendingReturnReason lendingReturned deliveryDate keptFailed retryAttempts lastRetryDate keepSscSyncStatus keepPaymentStatus lendingVouchers isSfscKeep keepAmount keepPaymentMethod
SitePreferences	lendingDevicePromotionId enableLendingCitySelector trialDays trialExtensionDays trialAutochargeDays trialReturnReasons trialNotificationDays lendingFeePromotionId enableKeepPromotion keepPromotionObject

Для обміну даними з сторонніми сервісами будуть використовуватись кастомні SFRA ендпоінти та service framework.

Salesforce B2C Commerce надає фреймворк веб-сервісів, який допомагає керувати викликами до веб-сервісів та аналізувати ефективність. Фреймворк встановлює обмеження, які налаштовуються — обмеження на загальну кількість викликів, дозволених за часовий інтервал, і обмеження на кількість невдалих викликів, дозволених за інтервал часу. Він також збирає аналітику про виклики ваших веб-служб.

Для кожної веб-служби фреймворку потрібна конфігурація служби, конфігурація профілю служби та конфігурація облікових даних служби. Ви створюєте ці конфігурації в Business Manager.

Конфігурація служби генерує об'єкт ServiceConfig. Облікові дані служби використовуються для базової аутентифікації. Профіль служби визначає, як B2C Commerce керує тайм-аутами та обмеженнями швидкості для веб-сервісу.

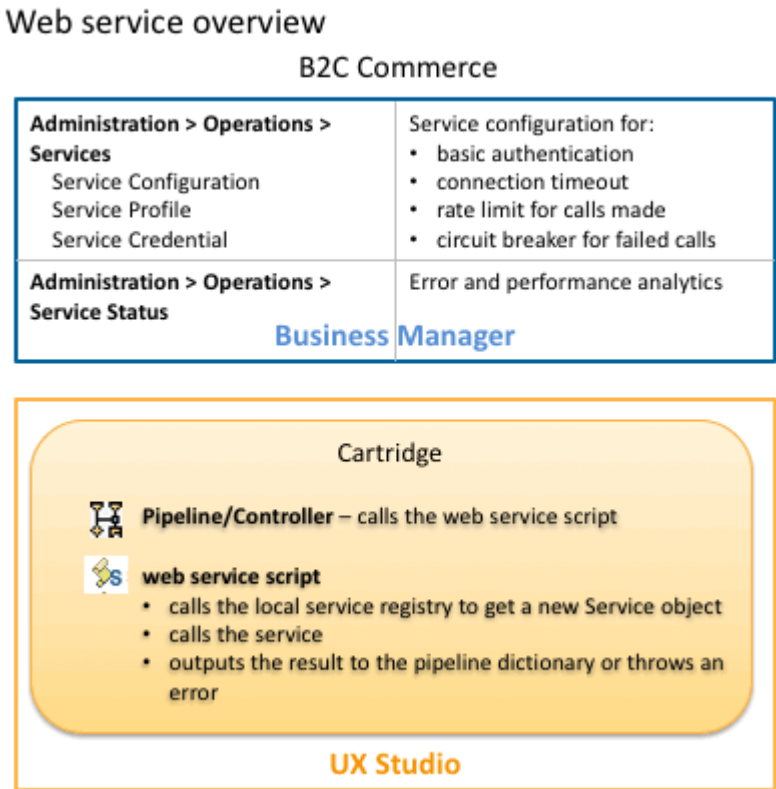


Рисунок 3.2 – Схема роботи service framework

Сервіс скрипт отримує об'єкт `Service`, викликаючи метод `LocalServiceRegistry.createService`. Метод приймає два аргументи: ім'я конфігурації служби та об'єкт сценарію, який містить код зворотного виклику.

Потім сценарій викликає службу, викликаючи `Service.call`. Коли викликається цей метод, фреймворк перевіряє, чи не перевищено налаштований ліміт швидкості або ліміт вимикача. Якщо ні, фреймворк виконує зворотні виклики, які ви реалізували в об'єкті `Service`.

Після завершення виклику скрипт або передає результат успішного виклику до `pipeline dictionary`, або видає помилку.

3.2 Тестування інформаційної технології

У даному розділі буде протестовано інформаційну технологію.

Нижче наведено декілька тестових сценаріїв.

Перевірити те, що товар додається до кошика.

Очікуваний результат: товар додався до кошика.

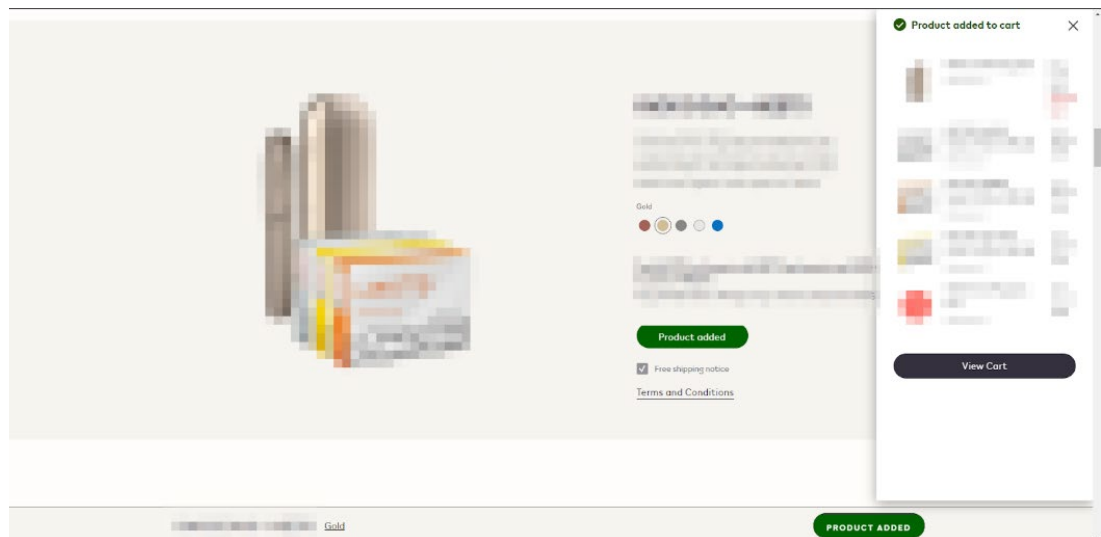


Рисунок 3.3 – Перевірка додавання у кошик

Перевірити те, що користувач не може додати більше одного продукту в кошик.

Очікуваний результат: користувач не може додати більше одного продукту до кошика.

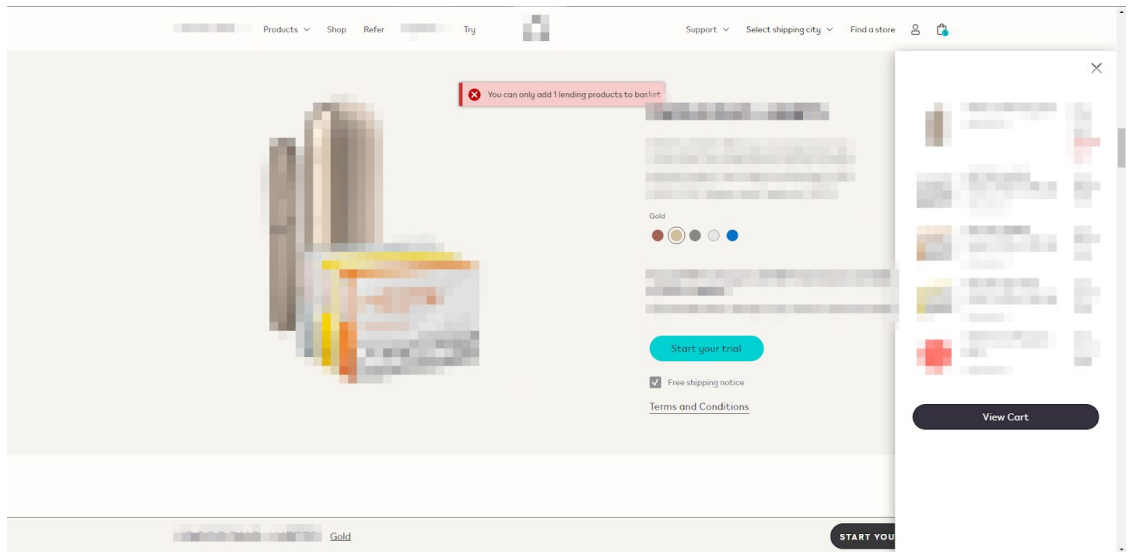


Рисунок 3.4 – Перевірка додавання декількох продуктів у кошик

Перевірити збереження даних на сервері.

Очікуваний результат: дані збережені на сервері.

Details for Order '00540466'

Information: Contains 5 line items to 1 shipping location. The total price is 100,000.00.

Qty	Product ID	Name	Manufacturer	Tax Rate	Unit Sales Price	Tax Basis	Item Total
1	7622190766124			5,000 %	350,000.00	350,000.00	350,000.00
1	7622190766826			5,000 %	25,000.00	25,000.00	25,000.00
1	7622190766738			5,000 %	25,000.00	25,000.00	25,000.00
1	7622190766707			5,000 %	25,000.00	25,000.00	25,000.00
1	lending fee			5,000 %	10,000.00	10,000.00	10,000.00
							-10,000.00
							25,000.00
							25,000.00
							25,000.00
1	Coupon LENDINGFEE						
Shipping Total:							25,000.00
Total:							100,000.00
Tax Total Included:							4,760.00

Рисунок 3.5 – Перевірка збереження даних на сервер

Перевірити у користувача доступ до сторінки пробного замовлення.

Очікуваний результат: користувач має доступ до сторінки пробного замовлення.

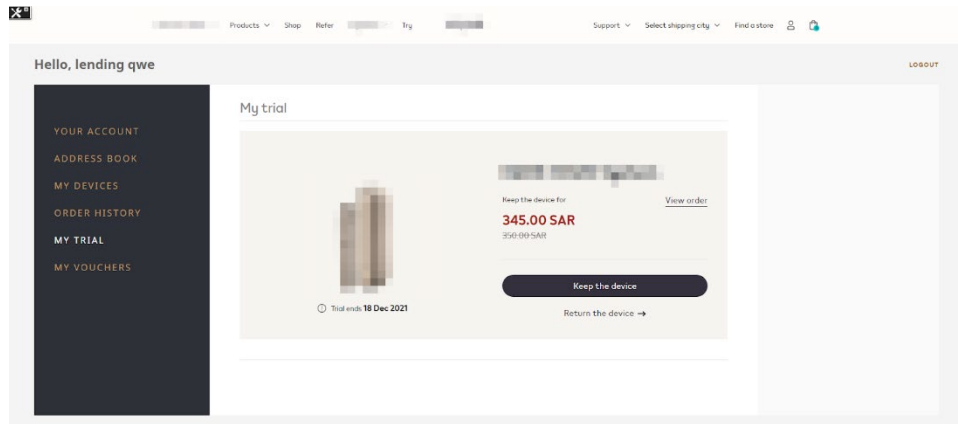


Рисунок 3.6 – Перевірка сторінки пробного замовлення

Перевірити доступ користувача до сторінки викупу пробного продукту.
Очікуваний результат: користувач має доступ до сторінки викупу пробного продукту.

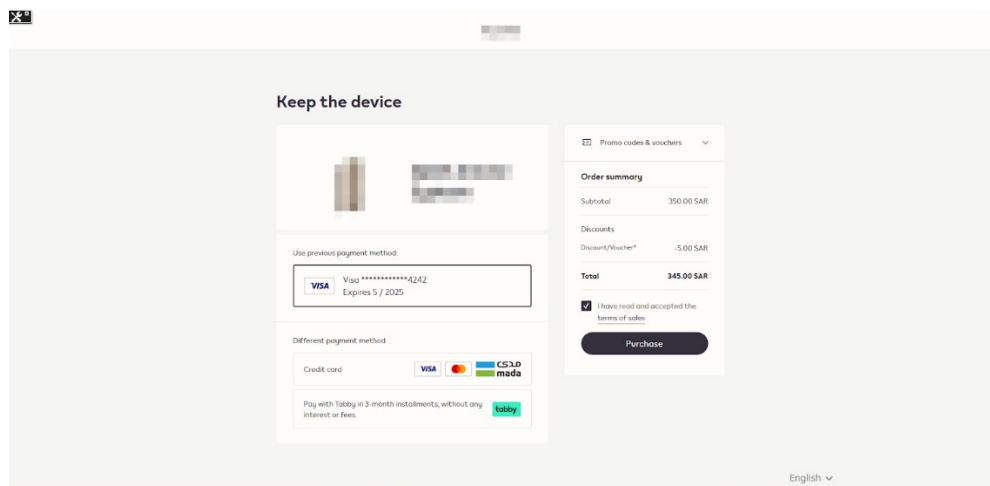


Рисунок 3.7 – Перевірка сторінки викупу пробного продукту

Перевірити доступ користувача до сторінки повернення пробного продукту.

Очікуваний результат: у користувача є доступ до сторінки повернення пробного продукту.

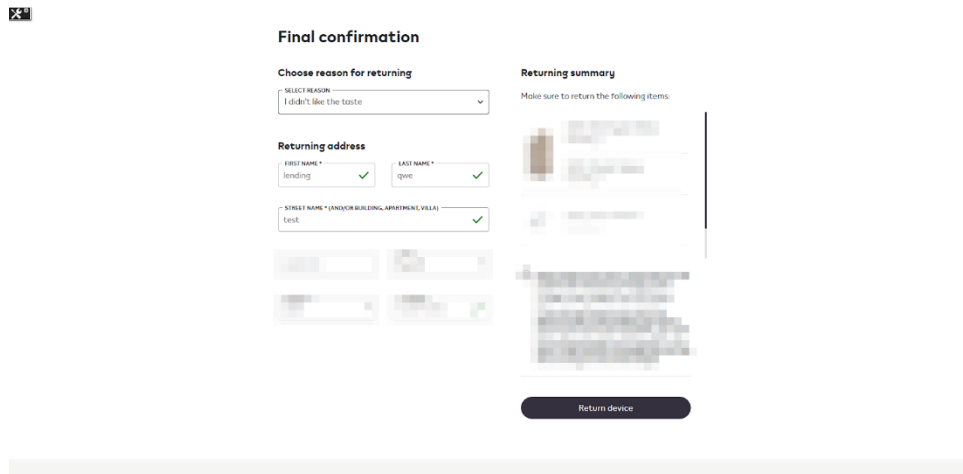


Рисунок 3.8 – Перевірка сторінки повернення продукту

Перевірити доступ користувача до сторінки продовження пробного періоду.

Очікуваний результат: користувач має доступ до сторінки продовження пробного періоду.

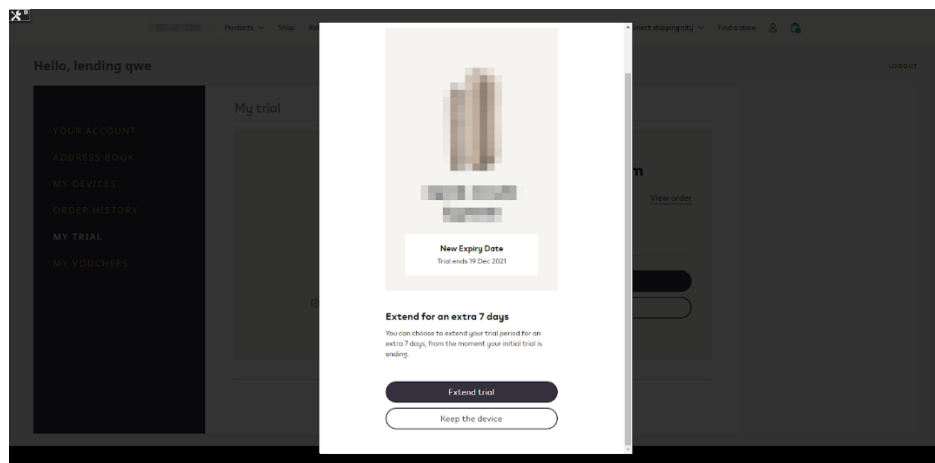


Рисунок 3.9 – Перевірка сторінки продовження пробного періоду

3.3 Висновки

У даному розділі було програмно реалізовано, протестовано інформаційну технологію та підведено підсумки.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного і технологічного аудиту залучимо 3-х незалежних експертів. У нашому випадку такими експертами будуть провідні викладачі випускової та споріднених кафедр.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу будемо здійснювати за 12-а критеріями згідно рекомендацій.

Результати оцінювання комерційного потенціалу розробки заносимо до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	4	3	4
Ринкові переваги (наявність аналогів)	4	3	3
Ринкові переваги (ціна продукту)	4	4	3
Ринкові переваги (технічні властивості)	4	3	4
Ринкові переваги (експлуатаційні витрати)	3	3	3

Ринкові перспективи (розмір ринку)	3	4	4
Ринкові перспективи (конкуренція)	3	4	3
Практична здійсненність (наявність фахівців)	4	3	3
Практична здійсненність (наявність фінансів)	3	4	4
Практична здійсненність (необхідність нових матеріалів)	3	3	4
Практична здійсненність (термін реалізації)	3	3	3
Практична здійсненність (розробка документів)	4	4	4
Сума балів	42	40	41
Середньоарифметична сума балів \overline{CB}	41		

За даними таблиці 4.1 робимо висновок щодо рівня комерційного потенціалу розробки. При цьому користуємося рекомендаціями, наведеними в таблиці 4.2.

Таблиця 4.2 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів, розрахована на основі висновків	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 50	Високий

Оскільки середньоарифметична сума балів складає 41, то рівень комерційного потенціалу розробки високий, тому дана розробка є реальною для подальшої її реалізації та впровадження.

Можливі декілька шляхів реалізації розробки.

У магістерській кваліфікаційній роботі проведено аналіз існуючого функціоналу Salesforce та запропонованих розширень. Проведено дослідження актуальних технологій які дозволяють працювати з eCommerce проектами, здійснено порівняння найпопулярніших рішень.

Дана робота допомагає проаналізувати існуючі можливості платформи Salesforce, та допомагає розширити платформу для додавання гнучкого та налаштовуваного функціоналу отримання продукту у пробний період.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи роботи

4.2.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p – середня кількість робочих днів в місяці, $T_p \approx 21 \dots 23$ дні;

t_i – кількість днів роботи конкретного дослідника.

Дану розробку буде проводити інженер, величина окладу буде становити 11000 грн. на місяць. Кількість робочих днів у місяці складає 21, а кількість робочих днів дослідника складає 50.

Зведемо сумарні розрахунки до таблиця 4.3.

Таблиця 4.3 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12000	571,42	6	3428,52
Інженер	11000	476,19	50	23809,5
Всього				27238,02

Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата Z_d розраховується як 10-12% від суми основної заробітної плати дослідників та робітників за формулою

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{N_{\text{дод}}}{100\%}, \quad (4.2)$$

де $N_{\text{дод}}$ – норма нарахування додаткової заробітної плати.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 10% від основної заробітної плати.

$$Z_d = 0,10 \cdot 27238,02 = 2723,80(\text{грн.})$$

4.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату $N_{\text{зп}}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою :

$$Z_{\text{дод}} = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{N_{\text{зп}}}{100\%}, \quad (4.3)$$

де $H_{зп}$ – норма нарахування на заробітну плату.

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді:

$$H_{зп} = (27238,02 + 2723,80) \cdot \frac{22}{100} = 6591,60 \text{ (грн.)}$$

Отже, нарахування на заробітну плату складають 6591,60 грн.

4.2.3 Розрахунок витрат на комплектуючі.

Витрати на комплектуючі, які використовують при дослідженні нового технічного рішення, розраховуються, згідно з їхньою номенклатурою за формулою:

$$K_v = \sum_{j=1}^n H_j \cdot C_j \cdot K_j, \quad (4.4)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j - коефіцієнт транспортних витрат, (1,1...1,15);

Проведені розрахунки зводимо до таблиці 4.4.

Таблиця 4.4 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, кг	Ціна за штуку, грн.	Разом
Папір	150	1	150
Ручка	10	1	10
Флешка	300	1	300
Всього			504

4.2.4 Амортизація обладнання, програмних засобів та приміщення.

У спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.5)$$

де $Ц_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн.;

$t_{\text{вик}}$ – термін корисного використання обладнання, програмних засобів, приміщень під час досліджень, місяців.

$T_{\text{в}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер	15000	5	6	1800
Всього				1800

4.2.5 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію розраховують за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (4.6)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн;

$K_{впi}$ – коефіцієнт, що враховує використання потужності;

η_i – коефіцієнт корисної дії обладнання.

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Комп'ютер	0,5	70	126
Освітлення приміщення	0,6	50	108
Всього			234

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B = Z_o + Z_d + Z_n + K + A_{обл} + B_e, \quad (4.7)$$

$$B = 27238,02 + 2723,80 + 6591,60 + 504 + 1800 + 234 = 39091,42 \text{ (грн)}$$

Загальні витрати на завершення науково-дослідної роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{\text{заг}}}{\eta}, \quad (4.8)$$

Загальні витрати складають

$$ЗВ = \frac{39091,42}{0,9} = 43434,91 \text{ (грн.)}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

Розрахуємо можливе збільшення чистого прибутку у потенційного інвестора для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки за формулою:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\nu}{100}\right), \quad (4.9)$$

де $\pm\Delta\Pi_0$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної діяльності;

Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році;

ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги), рекомендується приймати 0,2...0,5;

ϑ – ставка податку на прибуток.

В середньому в рік продається 500 розробок. Середня вартість такої розробки становить 800 грн.

Впровадження зразка розробки дозволяє збільшити ціну кожного зразка на 100 грн, враховуючи ціни конкурентів. Також прогнозується, що попит на даний продукт зростає, оскільки даний продукт відрізняється якістю від конкурентних.

Попит збільшиться за перший рік на 300 примірників, за наступний на 250 та протягом третього року – ще на 200 примірників.

Ставка податку на додану вартість в 2021 році залишилась на рівні 20% , а коефіцієнт $\lambda=0,8333$. Ставка податку на прибуток складає 18%.

Коефіцієнт, який враховує рентабельність продукту, дорівнює 0,3.

Отже, розрахуємо збільшення чистого прибутку підприємства на 2022 - 2024 рр.:

$$\Delta\P_{2022} = (800 \cdot 500 + (800 + 100) \cdot 300) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 137344,50 \text{ (грн.)}$$

$$\Delta\P_{2023} = (800 \cdot 500 + (800 + 100) \cdot (300 + 250)) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 183467,66 \text{ (грн.)}$$

$$\Delta\P_{2024} = (800 \cdot 500 + (800 + 100) \cdot (300 + 250 + 200)) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 220366,18 \text{ (грн.)}$$

Далі розрахуємо приведену вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\P_i}{(1+\tau)^{t^i}} \quad (4.10)$$

де $\Delta\P_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{137344,50}{(1 + 0,1)^1} + \frac{183467,66}{(1 + 0,1)^2} + \frac{220366,18}{(1 + 0,1)^3} = 442049,16 \text{ (грн.)}$$

Далі розрахуємо величину початкових інвестицій, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ, \quad (4.11)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науко-во-технічної розробки та її комерціалізацію.

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 4 \cdot 43434,91 = 173739,64 \text{ (грн)}$$

Тоді абсолютний економічний ефект або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = ПП - PV, \quad (4.12)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн.

$$E_{\text{абс}} = (442049,16 - 173739,64) = 268309,52 \text{ (грн.)}$$

Внутрішня економічна дохідність інвестицій, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (4.13)$$

де $E_{\text{абс}}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$E_B = \sqrt[3]{1 + \frac{268309,52}{173739,64}} - 1 = 0,36 = 36\%$$

Далі визначимо бар'єрну ставку дисконтування, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій визначається за формулою:

$$\tau_{min} = d + f, \quad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках;

f – показник, що характеризує ризикованість вкладення інвестицій.

$$\tau = 0,12 + 0,05 = 0,17$$

Далі розрахуємо період окупності інвестицій, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_B}, \quad (4.15)$$

де E_B – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = \frac{1}{0,36} = 2,77 \text{ роки}$$

Термін окупності складає 2,77 роки, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

ВИСНОВКИ

У даній роботі було проведено аналіз актуальності теми, аналіз предметної області, аналіз систем аналогів, обрано та обгрунтовано мову програмування, клієнтську та серверну технології, спроектовано, розроблено та описано архітектуру та алгоритми інформаційної технології, програмно реалізовано та протестовано інформаційну технологію, проведено економічне дослідження та підведено підсумки. Мету дослідження було досягнуто, а саме розроблено програмний модуль розширення функціональностей платформи Salesforce.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Зарицький О.С., Барабан С.В. «Аналіз розробки е-commerce проектів на платформі salesforce». Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2021)», Вінниця, 2021. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/view/14157>
Дата звернення: грудень, 2021р.
2. What is SALESFORCE. URL: <https://www.salesfix.com.au/blog/reasons-why-salesforce-is-the-best-crm/>
3. Salesforce benefits. URL: <https://www.webdesign-inspiration.com/article/5-benefits-of-salesforce-commerce-cloud-for-your-business/>
4. What is CRM. URL: <https://www.salesforce.com/crm/what-is-crm/>
5. Appexchange. URL: <https://appexchange.salesforce.com/>
6. Sales Cloud. URL: <https://www.salesforce.com/products/sales-cloud/overview/>
7. Salesforce in life. URL: <https://www.gspann.com/resources/blogs/business-benefits-of-salesforce-commerce-cloud/>
8. Альтернативи Salesforce. URL: <https://www.affde.com/ru/salesforce-alternatives.html>
9. Pipelines. URL: <https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2FLegacyDevDoc%2FWorkingwithPipelines.html&resultof=%22%70%69%70%65%6c%69%6e%65%22%20%22%70%69%70%65%6c%69%6e%22%20>
10. Angular. URL: <https://www.tutorialspoint.com/angularjs/index.htm>
11. ReactJS. URL: https://www.tutorialspoint.com/reactjs/reactjs_introduction.htm
12. Site Genesis. URL: <https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2FLegacyDevDoc%2FSiteGenesisDe>

[velopmentOverview.html&resultof=%22%73%69%74%65%22%20%22%67%65%6e%65%73%69%22%20](#)

13. VueJS URL: https://www.tutorialspoint.com/vuejs/vuejs_overview.htm

14. What is PageDesigner. URL:

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fpage_designer%2Fb2c_what_is_page_designer.html

15. jQuery URL: <https://www.tutorialspoint.com/jquery/jquery-overview.htm>

16. Web Services. URL:

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fpage_designer%2Fb2c_pd_script_file.html

17. Configure web services. URL:

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fweb_services%2Fb2c_configuring_web_services_in_business_manager.html

18. Create web service profile. URL:

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fweb_services%2Fb2c_create_web_service_profile.html

19. Create web service credentials. URL:

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fweb_services%2Fb2c_create_web_service_credential.html

20. Code the web service call. URL:

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fweb_services%2Fb2c_coding_your_web_service.html

21. Class ProductLineItem [Электронный ресурс] – Режим доступа :

https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2Fcontent%2Fb2c_commerce%2Ftopics%2Fcustom_objects%2Fb2c_system_objects.html&resultof=%22%73%79%73%74%65%6d%22%20%22%6f%62%6a%65%63%74%22%20%22%74%79%70%65%73%22%20%22%74%79%70%65%22%20