

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій

(факультет)

Кафедра системного аналізу та управління

(повна назва)

ПОЯСНОВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра

Здобувача вищої освіти Казарінов Олександр Валерійович

академічної групи 124-21-2

спеціальності 124 Системний аналіз

за освітньо-професійною програмою Системний аналіз

на тему: «Аналіз та покращення інтерфейсу продуктової кошика із

використанням A/B тестування »

| Керівники | Прізвище, ініціали | Оцінка за шкалою | | Підпис |
|--------------------------|--------------------------------|------------------|---------------|--------|
| | | рейтинговою | Інституційною | |
| кваліфікаційної роботи | д-р філ., доц. Хабарлак К.С | | | |
| розділів: | | | | |
| Інформаційно-аналітичний | д-р філ., доц. Хабарлак К.С | | | |
| Спеціальний розділ | д-р філ., доц. Хабарлак К.С | | | |
| Рецензент | | | | |
| Нормоконтролер | к.ф.-м.н., доц. Хом'як Т.В | | | |

Дніпро

2025

ЗАТВЕРДЖЕНО:

завідувач кафедри

Системного аналізу та управління

(повна назва)

_____ к.т.н., доц. Желдак Т.А.

(підпис)

(прізвище, ініціали)

« _____ » _____ 20 ____ року

ЗАВДАННЯ

на кваліфікаційну роботу

ступеня бакалавра

здобувачу вищої освіти Казарінову О. В. академічної групи 124-21-2

спеціальності: 124 Системний аналіз

за освітньо-професійною програмою Системний аналіз

на тему «Аналіз та покращення інтерфейсу продуктової корзини із

використанням А/В тестування»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2025

р. №336-с

| Розділ | Зміст | Терміни виконання |
|------------------------------------|--|-----------------------|
| 1. Інформаційно-аналітичний розділ | Проаналізувати структуру об'єкта дослідження. Визначити предметну область дослідження та проблему, що розв'язується. Обґрунтувати методи виконання поставлених завдань | 09.09.2024-28.01.2025 |
| 2. Спеціальний розділ | 1) Здійснити програмну реалізацію сайту магазину товарів, що включає в себе базу даних та користувацький інтерфейс; 2) реалізувати підхід А/В тестування для сайту; 3) провести аналіз взаємодії користувачів із сайтом. На основі результатів А/В тестування проаналізувати можливість покращення користувацького інтерфейсу продуктової корзини. Провести перевірку отриманого результату на статистичну значущість. | 29.01.2025-31.05.2025 |

Завдання видано _____

(підпис)

доц. Хабарлак К.С.

(прізвище, ініціали)

Дата видачі: 09.09.2024 р.

Дата подання до екзаменаційної комісії: 09.06.2025

Прийнято до виконання _____

(підпис студента)

Казарінов О. В.

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 52с., 43 рис., 1 табл., 2 додатки, 14 джерел

Об'єктом дослідження в кваліфікаційної роботі є процес аналізу та розробка інтерфейсу продуктової корзини на веб-сайті, онлайн-магазину квітів із використанням різних методів тестування для перевірки ефективних змін.

Предметом дослідження є методи взаємодії користувача з продуктовою корзиною та на аналізі результативності інтерфейсних рішень за допомогою А/Б тестування.

Метою кваліфікаційної роботи є розробка та вдосконалення інтерфейсу перегляд обраних товарів у кошику на веб-сайті за допомогою А/Б тестування з метою підвищення зручності, ефективності та задоволення користувачів під час процесу покупок.

Методи дослідження: у цьому дослідженні застосовуються методи аналізу користувацького інтерфейсу та принципи його побудови, використання А/Б тестувань порівняння різних варіантів інтерфейсів, а також оцінка ефективності взаємодії користувача з веб-сайтом за результатами тестів і статистичний аналіз даних для вибору оптимального макету.

В *інформаційно-аналітичному розділі* здійснено аналіз підходів до створення інтерфейсів для веб-магазинів та методів тестування, що використовуються для поліпшення користувацької взаємодії. Описано різноманітні методи тестування, зокрема А/В-тестування, а також поставлені завдання дослідження і підходи до їх вирішення.

У *спеціальному розділі* представлена реалізація А/Б тестування на основі веб-сайту, розробленого з використанням мови програмування TypeScript та фреймворком Angular також наведено порівняння різних варіантів інтерфейсу для огляду кошику.

Практична цінність отриманих результатів полягає в тому, що застосування множинного тестування дозволяє отримати точні дані про

переваги різних інтерфейсів, що підвищує зручність використання веб-сайтом та сприяє покращеним продажам товарів.

Ключові слова: ІНТЕРФЕЙС, ПРОДУКТОВА КОРЗИНА, ВЕБ-САЙТ, ОНЛАЙН-МАГАЗИН, А/Б ТЕСТУВАННЯ, МЕТОДИ ТЕСТУВАННЯ, TYPESCRIPT, ANGULAR , РОЗРОБКА ІНТЕРФЕЙСУ.

124 СИСТЕМНИЙ АНАЛІЗ (НТУУ «ПІІ») (NTU «PII')

ABSTRACT

Explanatory Note: 52 pages, 43 figures, 1 tables, 2 appendices, 13 references

The object of research in the qualification work is the process of analyzing and developing the interface of the grocery cart on the website, an online flower store using various testing methods to verify effective changes.

The subject of the research is the methods of user interaction with the grocery cart and analyzing the effectiveness of interface solutions using A/B testing.

The purpose of the qualification work is to develop and improve the interface for viewing selected products in the cart on the website using A/B testing in order to increase the convenience, efficiency and satisfaction of users during the shopping process.

Research methods: this study uses methods of analyzing the user interface and the principles of its construction, using A/B testing to compare different interface options, as well as assessing the effectiveness of user interaction with the website based on test results and statistical data analysis to select the optimal layout.

The information and analytical section analyzes approaches to creating interfaces for web stores and testing methods used to improve user interaction. Various testing methods are described, in particular A/B testing, as well as research tasks and approaches to solving them.

A special section presents the implementation of A/B testing based on a website developed using the TypeScript programming language and the Angular framework, and also provides a comparison of different options for the shopping cart interface.

The practical value of the results obtained is that the use of multiple testing allows you to obtain accurate data on the advantages of different interfaces, which increases the usability of the website and contributes to improved sales of goods.

Keywords: INTERFACE, SHOPPING CART, WEBSITE, ONLINE STORE, A/B TESTING, TESTING METHODS, TYPESCRIPT, ANGULAR, INTERFACE DEVELOPMENT.

ЗМІСТ

| | |
|---|----|
| РОЗДІЛ 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ | 9 |
| 1.1 Постановка задачі | 9 |
| 1.2 Огляд сучасних онлайн-магазинів | 10 |
| 1.3 Визначення критеріїв ефективності інтерфейсу | 16 |
| 1.4 Теоретичні основи А/В тестування в контексті оптимізації інтерфейсу корзини | 18 |
| 1.4.1 Методи статистичного аналізу | 23 |
| РОЗДІЛ 2 СПЕЦІАЛЬНИЙ РОЗДІЛ | 26 |
| 2.1 Розробка бази даних товарів з використанням платформи Firebase | 26 |
| 2.2 Аналіз інтерфейсу кошика на веб-сайтах | 29 |
| 2.2.1 Візуальний та функціональний аналіз | 29 |
| 2.2.2 Порівняльна таблиця рішень | 33 |
| 2.3 Розробка веб-додатку | 34 |
| 2.3.2 Архітектура та основні функціональні можливості додатку | 34 |
| 2.3.3 Реалізація інтерфейсу кошика та інших важливих компонентів | 35 |
| 2.4 Застосування методу А/В для покращення веб-додатку | 41 |
| ВИСНОВОК | 49 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 51 |
| Додаток А | 53 |
| Додаток В | 54 |
| Додаток С | 79 |

ВСТУП

Купівля товарів через інтернет-магазини вже стала справжньою звичкою для більшості людей і, на перший погляд, не відрізняється від звичайних магазинів. Зручність у покупках онлайн є найочевиднішою перевагою сучасної культури. Однак, для того щоб досягти лояльності серед покупців, інтерфейс сайту має бути інтуїтивно зрозумілим і, що найголовніше, швидким. Тобто, практиці в юзабіліті дизайну потрібно придати не менше уваги.

Однією з найбільш дієвих методик поліпшення інтерфейсу є А/В тест. Мультитест дозволяє перевіряти по кілька варіантів одночасно і підібрати результати, які надають найбільшу зручність користувачам. В даному випадку головною задачею є розробити сайт для онлайн магазину квітів на базі Angular з викладенням аналізу рішень у вигляді тестів. Дослідження охоплює питання таких типів:

- Яка класифікація способів побудови інтерфейсу для інтернет-магазину.
- Як робляться А/В тестування на предмет дизайнерських варіантів.
- Яка архітектура сайту на Angular.
- Які етапи проходить тестування і в які способи оброблюється його результат.

Дослідження цієї теми є актуальним, адже саме зручність і простота інтерфейсів сприяють підвищенню орієнтованості інструмента на користувача. Проведення тестувань дає змогу ухвалювати рішення не інтуїтивно, а на основі зібраних і проаналізованих даних.

Було створено підготовчий сайт, який можна надалі використовувати як для роботи, так і для впровадження тестів із реальними результатами — як для авторів проекту, так і для інших фахівців у сфері веброзробки.

Отримані результати мають практичну цінність і можуть бути застосовані для подальших модифікацій інтерфейсу. Очікується, що проведені дослідження та тестування дозволять зробити сайт зручнішим для

користувачів, що, у свою чергу, сприятиме покращенню якості онлайн-торгівлі.

РОЗДІЛ 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ

1.1 Постановка задачі

Головна мета цієї роботи – спростити та полегшити процес оформлення замовлення в онлайн-магазині для покупців, аби більшій кількості людей пропонувалося завершити покупку. Як часто буває: додав товар у кошик, а потім щось не сподобалося або заплутався, на цьому етапі замовлення так і не оформлюється. Наша задача – проаналізувати, які моменти на межі оформлення замовлення відлякують і до яких змін потрібно вдаватися для їх подолання.

А як саме ми це плануємо зробити? Спочатку всесторонньо вивчимо, як працюють кошики у відомих онлайн-магазинах наявність. Також оцінимо, що в них зручне, а що могли б зробити краще. Потім розробимо критерії, за якими вестиметься оцінка успішності безперервних змін: скільки осіб дійшло до зручного огляду, скільки часу це забирає, на якому етапі найчастіше відбуваються відмови від неї.

Створимо декілька варіантів поліпшеного та зміненого інтерфейсу та, на нашу думку, спростимо форму замовлення, важливі кнопки, що збільшують корисні підказки. І далі найцікавіше: справа в тому, що треба провести справжній експеримент розгляду кошика, одній частині користувачів показувати стару версію, іншій нову.

При цьому в кінці наших рекомендацій поділимося порівнянням старого та нового, так буде ще значно простіше формулювати, а конкретніше роз'яснити, як ці зміни в оновленому інтерфейсі зроблять покупки в цьому самому магазині зручнішими. Впевнитися варто в тому, що коли приємно, то це вигідно всім.

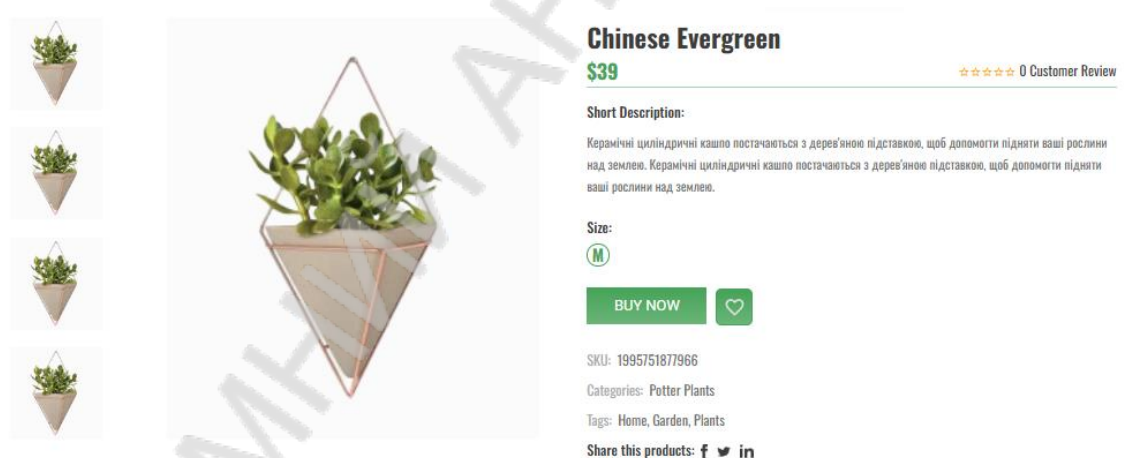
1.2 Огляд сучасних онлайн-магазинів

Створити успішний та сучасний інтернет-магазин – завдання, яке не на жарт вимагає зусиль. Можна спробувати втрутитися в процес створення самостійно, використовуючи онлайн-конструктор або розробити сайт за допомогою команди професіоналів, практично гарантуючи неконкурентоспроможність бізнесу на цифровому ринку. І те інше підходить, але незалежно від того, як підійшли ви до дизайнерського столу, успіх сайту, а отже, і існування всього бізнесу визначається неодним чинником. І ось який: плюс і майже мінус розкладається на чимало ланок. І особливість цього процесу полягає в тому, що близько 90% інтернет-магазинів зупиняють свою роботу вже протягом 1 року існування, не дізнавшись конкурентів, що їх знешкоднують. Як це можливо? Чому успішні проекти приносять власнику мільйонні прибутки, а запрошені першого року після запуску?

Ефективний і сучасний онлайн-магазин виходить за межі простих функцій торгової платформи або сховища статичних списків товарів. Це комплексна система, яка ретельно враховує кожен аспект, починаючи від простоти процесу купівлі та закінчуючи функціональністю та ресурсами, доступними для покупця. Не менш важливим є розуміння психологічної точки зору покупця. Давайте розглянемо ключові елементи додаткових ресурсів і функцій, які підносять онлайн-магазин із базової платформи до клієнтоорієнтованого облікового запису та прибуткового активу для власника. Перш ніж розробляти концепцію зовнішнього вигляду інтернет-магазину, важливо визначити особу його клієнтів. У випадках, коли магазин пропонує певні товари, доцільно враховувати переваги цієї конкретної демографічної групи. Дизайн повинен відповідати їхнім естетичним почуттям. І навпаки, якщо асортимент продукції різноманітний, наприклад, у випадку з продуктами харчування, було б доцільно прийняти нейтральний дизайн, щоб залучити як чоловіків, так і жінок різних вікових груп. Розуміння цільової аудиторії є

важливим кроком перед тим, як приступити до розробки дизайну інтернет-магазину.

Також торкнемося теми з приводу яку технологію використовувати для розташування блоків, як створювати картки товару, що входить до правильного заповнення карточки: Візуальна частина де можна показати фото або відео товару, додати близько 4-5 якісних фотографій, додати середню оцінку товару та впровадження відгуків до товару яка буде реалізована на рисунку 1.2, додати бейджі на кшталт "акція", "у топ продажах". Також є у кожного товару інформаційна зона, у ній потрібно написати основну інформацію. Туди як правило записується назва, ціна, артикул. Створюємо опис товару коротку частину і докладну [7]. Приклад картки товару зображено на рисунку 1.1.



Product Description

Reviews (0)

Сучасна декоративна рослина в мінімалістичному металевому дизайні. Чорна металева рама створює стильний геометричний акцент, а всередині знаходиться чаша з товстим штучним зеленим листям, що створює природний ефект. Цей аксесуар підійде для додавання свіжості та яскравості інтер'єру без необхідності догляду.

Living Room:

У вітальні цей декор можна розмістити на полиці, журнальному столику або шафі. Його сучасний та лаконічний дизайн чудово доповнить стиль лофт, мінімалізм або модерн, створюючи контраст між зеленою та суворими лініями металевого каркаса.

Dining Room

На обідньому столі або буфеті ця рослина додасть стильного акценту та свіжості. Компактний розмір дозволяє використовувати її як центральний елемент декору або частину композиції, створюючи затишну атмосферу.

Office:

В офісі цей аксесуар ідеально розміститься на столі, полиці чи підвіконні, додаючи простору зелені та сучасності. Його строгий дизайн поєднується з професійною обстановкою, створюючи візуальний комфорт та відчуття природної гармонії.

Рисунок 1.1 – Опис товару.

| Опис товару | Відгуки (8) | |
|--|--------------------------|-------|
| sa1nts eye | 31.05.2025, 23:05 | ★★★★☆ |
| Квіти чудові дякую за доставку | | |
| sa1nts eye | 28.05.2025, 17:37 | ★★★★★ |
| Квіти прекрасні, дякую за доставку | | |
| sa1nts eye | 01.02.2025, 12:11 | ★★★★★ |
| Квітка має насичені відтінки, виглядає дуже жваво та свіжо. Листя здорове та зелене, що свідчить про хороший догляд. Він точно стане прикрасою будь-якого інтер'єру! | | |
| sa1nts eye | 01.02.2025, 12:12 | ★★★★★ |
| Дуже красива рослина чудово вписується в будь-який куточок будинку. Квіти довго не в'януть і зберігають свою привабливість. Рекомендую всім! | | |

Рисунок 1.2 – Відгуки до товару.

Після того як впевнилися з тим яка повинна бути картка товару потрібно подбати про те, як буде виглядати семантична частина сайту, найпопулярніша семантична частина онлайн-магазинів виглядає [10] ось так на рисунку 1.3:

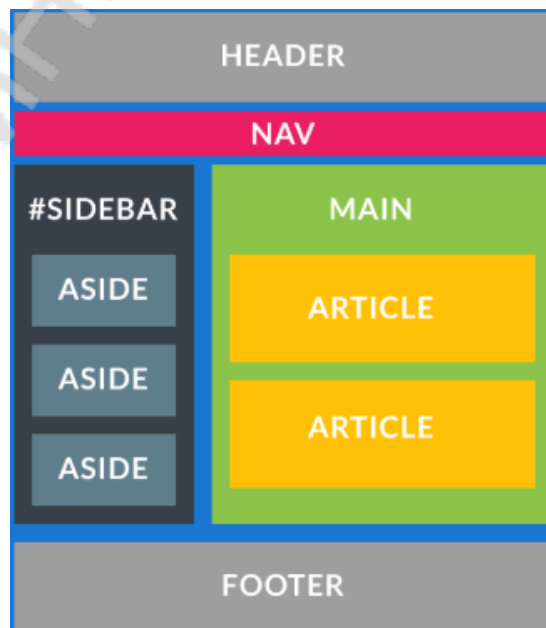


Рисунок 1.3 – семантична верстка інтернет магазину.

Таку семантику використовують великі онлайн-магазини наприклад як «ROZETKA» зображено на рисунку 1.4.

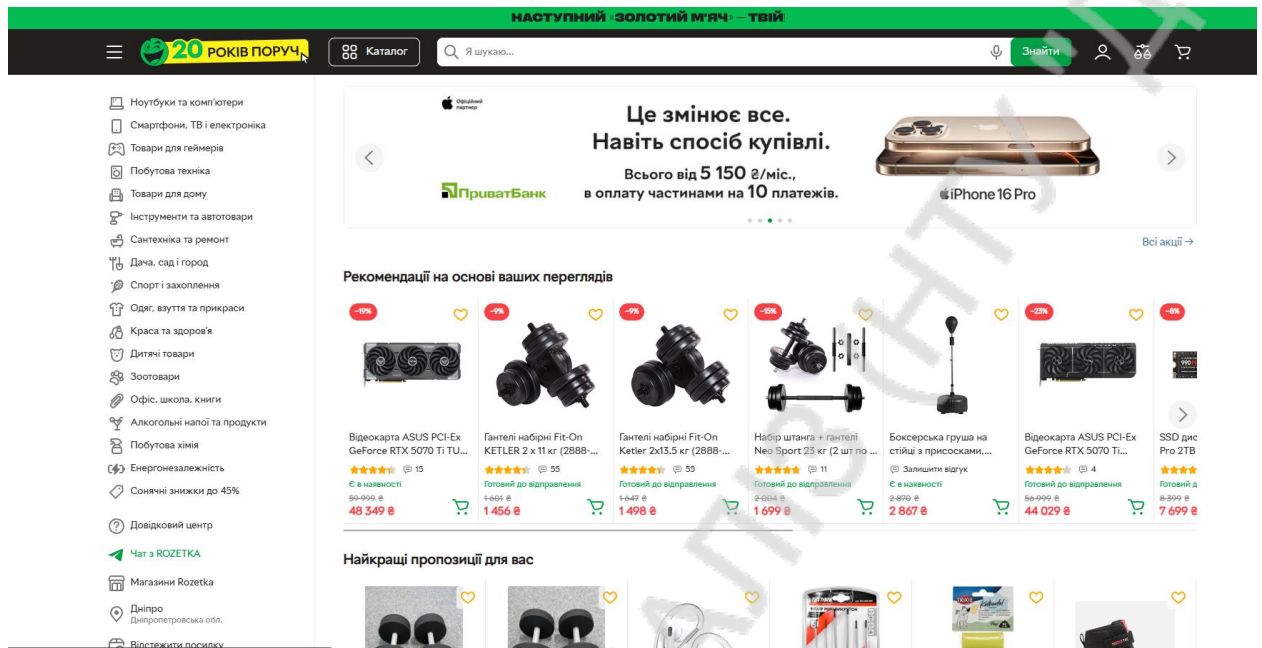


Рисунок 1.4 – семантична верстка в онлайн магазині «ROZETKA».

Для розташування блоків на сайті використовуються дві популярні технології з SCSS(Sassy Cascading Style Sheets) Flexbox та Grid

Flexbox (Flexible Box Layout) — це сучасна CSS-технологія, яка дозволяє легко керувати розташуванням, розмірами та вирівнюванням елементів у контейнері. Вона особливо корисна для створення гнучких та адаптивних макетів зображені властивості на рисунку 1.5.

Основні властивості Flexbox у SCSS:

- `display: flex` — перетворює блок на flex-контейнер.
- `flex-direction` — визначає напрямок розташування елементів (row, column, row-reverse, column-reverse).
- `justify-content` — вирівнює елементи по головній осі (center, space-between, space-around).
- `align-items` — вирівнює елементи по поперечній осі (flex-start, flex-end, center).
- `flex-wrap` — дозволяє елементам переноситись на новий рядок (wrap, nowrap). [12]

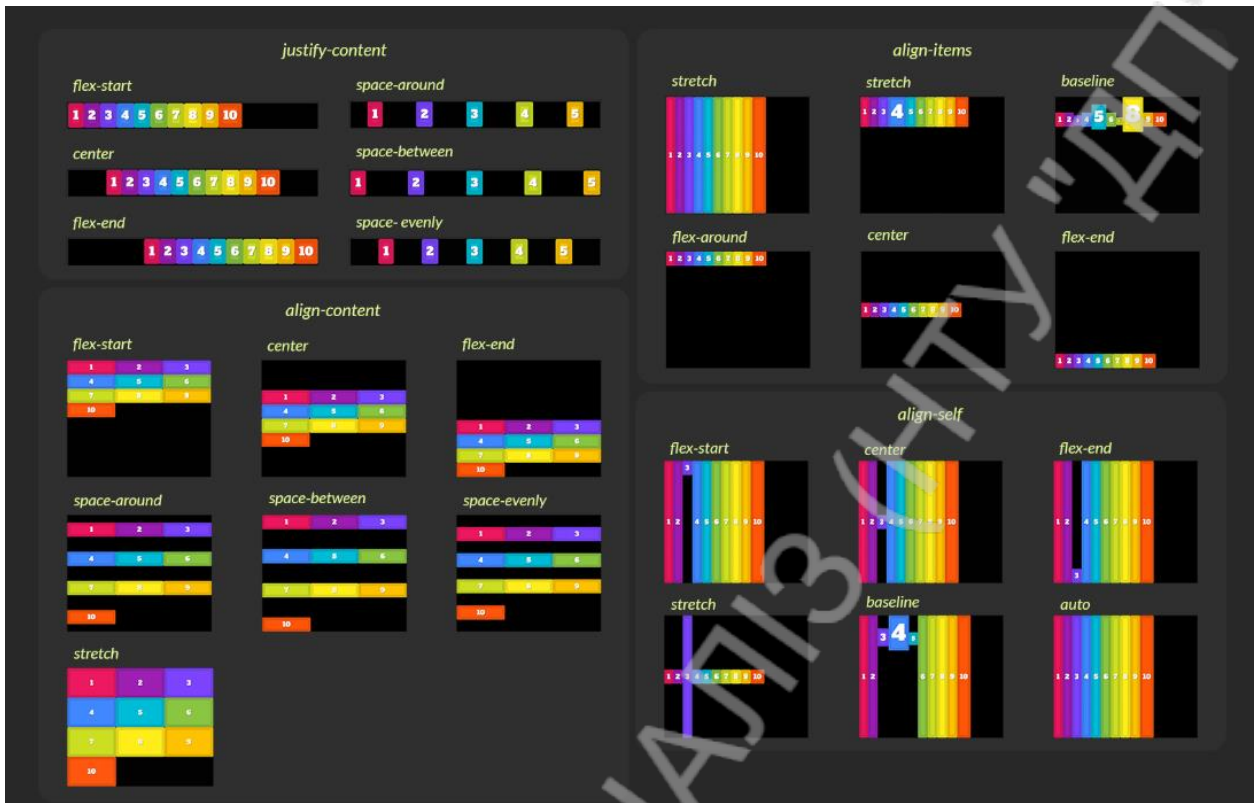


Рисунок 1.5 – Основи Flexbox.

Головні переваги flexbox це простота в використанні для одновимірних макетів та легкий для вивчення.

CSS Grid — це потужна система для створення двовимірних макетів (рядки та стовпці). Вона дозволяє точно контролювати розміщення елементів у сітці. Приклад роботи зображено на рисунку 1.6.

Основні властивості Grid у SCSS:

- `display: grid` — створює grid-контейнер.
- `grid-template-columns` / `grid-template-rows` — визначає кількість та розміри стовпців і рядків.
- `gap` — задає відступи між комірками сітки.
- `grid-area` — дозволяє елементам займати конкретні області сітки.
- `grid-column` / `grid-row` — позиціонує елементи за номерами ліній сітки.



Рисунок 1.6 – Основи CSS Grid.

Основні переваги CSS GRID – це використання його для складних макетів із багатьма елементами. Він дозволяє створювати адаптивні сітки без зайвих медіазапитів за допомогою яких можливо робити адаптацію під інші пристрої. [13]

Також можливе комбінування цих двох технологій. Наприклад, створити основну структуру сайту за допомогою Grid, а всередині окремих блоків застосувати Flexbox для вирівнювання контенту. Такий принцип роботи використовує сайт «ROZETKA».

Отже, Flexbox і Grid у SCSS значно спрощують процес верстки сучасних вебсайтів і додатків, забезпечуючи гнучкість та адаптивність макетів. Вибір між цими методами залежить від конкретної задачі: Flexbox підходить для компонентів, розташованих у одному вимірі, а Grid — для складних сіткових структур. Використання обох методів разом дозволяє досягти максимальної ефективності у веб-розробці. Однак для сайту квітів ми оберемо Flexbox, оскільки використовуватимемо одновимірний макет.

1.3 Визначення критеріїв ефективності інтерфейсу

У теперішніх умовах жорстокої боротьби в онлайн-торгівлі кожен елемент вигляду магазину потребує уважного аналізу та поліпшення. Особливу увагу потрібно звертати на вигляд кошика, оскільки саме на цьому кроці відбувається важливий перехід від перегляду товарів до покупки. Беручи до уваги особливість нашого дослідження, де для оцінки ефективності є лише дві версії вигляду (A та B) та система оцінок користувачів, потрібно детально розглянути вибраний метод оцінювання і його тлумачення.

Система 5-балів для оцінки, яка є на нашому сайті, це простий інструмент для вимірювання задоволення користувачів. Вона дає змогу оцінити як люди бачать процес замовлення. Важно знати, що кожен бал у цій шкалі має особливе значення: від "дуже погано" (1 бал) до "супер" (5 балів), що дає нам шанс не просто взяти числові дані а й розуміти емоційний фон взаємодії користувача з системою.

Процес збору інформації зроблений так, що кожен покупець обов'язково отримує запит для оцінки після завершення угоди. Це забезпечує представлення вибірки та усуває можливе упередження яке пов'язано з добровільністю відгуків. Оцінки автоматично записуються у базу даних для кожної тестової версії окремо, що дозволяє робити чіткі порівняння дослідження.

Головним аналітичним інструментом в нашому випадку є обрахунок середнього арифметичного значення оцінок. Цей показник, хоча і простий, дуже інформативний при порівнянні. Формула розрахунку проста: сума всіх отриманих балів ділиться на число відгуків. Але потрібно пам'ятати що для того щоб отримати правильні результати, треба мати достатню кількість даних - статистично важливими вважаються результати з 15 оцінок для кожної версії.

При розумінні результатів потрібно звертати увагу не тільки на абсолютні числа середніх балів, а й на зміни їх значень. Наприклад якщо

версія Б показує середній бал 4.3, а версія А - 4.1 це може свідчити про деякі переваги нової реалізації .

Важним частиною аналізу є також розподіл оцінок. Наприклад, версія із середнім балом 4.0 може мати два різні профілі:

Стабільні високі бали (більшість 4 и 5)

Поляризовані відгуки (багато п'ять, але велика кількість один-два)

Це показує про різний вид проблем в інтерфейсі. В перший раз ми маємо постійно гарний досвід, а в другий - можуть бути серйозні недоліки які викликають раптові відторгнення у частині користувачів.

Хоча метод має певні межі (немає додаткових показників, як час оформлення чи конверсія), система оцінок користувачів має декілька важливих плюсів. По-перше, він дає прямий відгук від кінцевого споживача що дуже ціно при оцінці зручності.

По-друге, дозволяє виявити неочевидні негаразди, що можуть не фіксуватися іншими інструментами аналітики. По-третє, є інтуїтивно зрозумілою для тлумачення всіма зацікавленими сторонами. Для покращення якості аналізу рекомендується встановити чіткі критерії інтерпретації результатів:

1. 4.5-5 балів – відмінний результат, інтерфейс не потребує корекції.
2. 4.0-4.4 – гарний показник, можливі незначні вдосконалення.
3. 3.5-3.9 – задовільний рівень, але потребує серйозного аналізу.
4. Нижче 3.5 - критичний стан, необхідні кардинальні зміни.

Отже, незважаючи на відносну простоту обраного методу, його правильне застосування дозволить отримати цінні інсайти щодо ефективності тестових версій інтерфейсу та прийняти обґрунтовані рішення стосовно подальшого вдосконалення системи.

1.4 Теоретичні основи A/B тестування в контексті оптимізації інтерфейсу корзини

A/B-тестування – це метод порівняння двох або більше версій компонента з головною метою визначати який із усіх вибраних компонентів найкраще працює за рахунок думки користувачів та впровадити його до проекту [5]. Щоб досягти успіху в аналізі потрібно зібрати дані користувачів, які будуть поділені на дві колонки, якщо ми тестуємо всього два компоненти, якщо три компоненти, то повна кількість людей буде поділятися на три колонки можливо побачити це на рисунку 1.7.

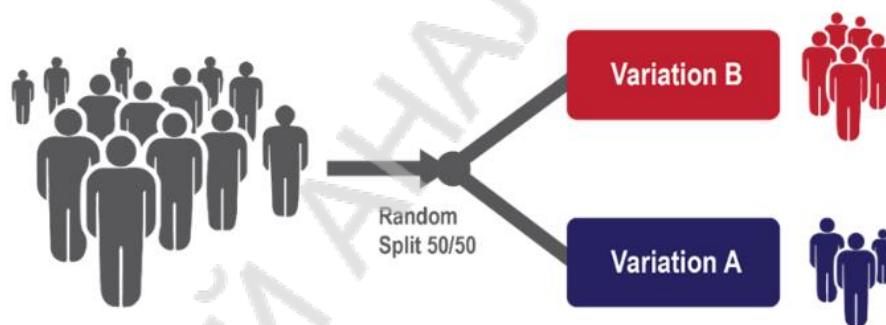


Рисунок 1.7 – розподіл користувачів на два варіанти.

Досліджувати за допомогою A/B-тестування ми можемо дизайн сторінок, кольори компонентів, шрифт тексту, розташування блоків, зображення відео та інше. Завдяки цьому ми можемо зрозуміти який компонент із тесту більше виглядатиме на око краще. Також частіше проводять тести із формами даних користувача, що заповнюються. Це робиться для того, щоб можна було зрозуміти як користувачу зручно заповнювати форму, чи є валідація у формі, де знаходяться сама форма, наприклад, на сторінці із замовленням або зроблене як окреме модальне вікно, яке дасть з концентруватися на заповненні форми більше уваги. Також не забуваємо про навігаційне меню, де користувач може оглянути каталог і

наскільки він буде зручним. Так як навігаційне меню вважається одним із найголовніших компонентів у будь-якому проєкті, тільки тому що він несе більшу частину інформації та навігації по сайту. До нього потрібно ставитись з великою увагою і тести проводити максимально величезних масштабів, перш ніж повністю переходити зі старої версії навігаційного меню на новий.[6]

Не забуваймо про основу — функціонал сайту, на скільки він може бути хорошим або поганим це ми можемо перевірити, використовуючи А/Б тестування. Наприклад створимо пошукову систему телефонів будемо використовувати слово "айфон" де компоненти будуть містити таку логіку пошуку:

1. У варіанті А: показуються лише товари, які мають у назві точне слово " айфон " .
2. У варіанті В: показуються і iPhone, і Apple iPhone 13, навіть якщо користувач помилився в написанні (айфн).

Таким методом ми можемо визначити який функціонал буде краще для фінальної версії, але якщо взяти суб'єктивну думку, то варіант В буде кращим тільки тому, що є люди які можуть друкувати в сліпу і промахуються по клавіатурі або людина не пам'ятає повну назву товару, а тільки його частину, так само цей варіант буде набагато зручніше для користувача так як він буде займати менше часу на пошук. Для будь-якого інтернет магазину ми можемо зробити тести оформлення покупки, перевірити на скільки довго користувачеві після додавання в кошик товару знадобилося часу щоб завершити покупку, скільки кліків було зроблено, при заповненні форми дізнатися з чим у нього виявилися труднощі. У будь-якому випадку чим більше кліків було зроблено і менше часу витрачено при завершенні покупки — тим краще. Приклад зроблених кліків при зміні кольору кнопки зображено на рисунку 1.8.[1]

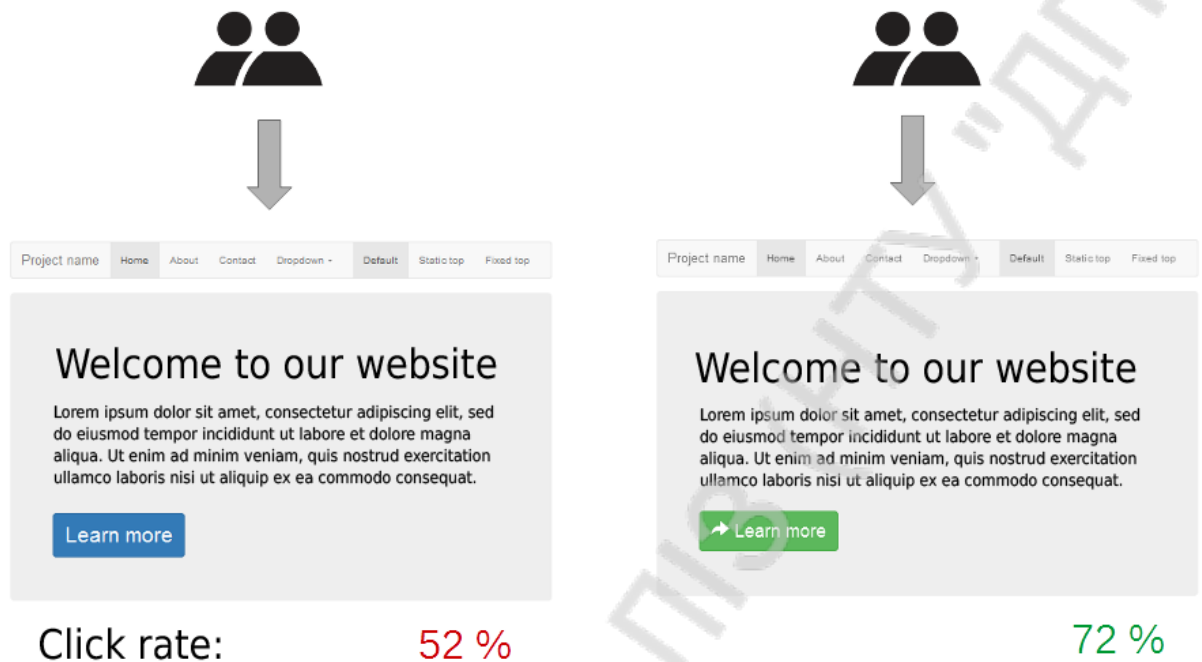


Рисунок 1.8 – Приклад А/В-тестування для вебсайту зміна кольору кнопки.

Найчастіше тестують у будь-яких магазинах маркетингову зону. Наприклад акції на товари, роблять перевірку на скільки акція зроблена стала успішною, скільки людей натиснуло на акцію і поцікавилася товаром, але перед тим як робити будь-яку акцію на товар, слід дізнатися який товар найменше купують на сайті і скільки у нього є кількість кліків від відвідувачів, так само потрібно дізнатися кількість самих товарів на складі, перевірити на скільки. Щоб розрахувати знижку товару, компанії потрібно провести аналіз скільки він втратить грошей під час продажу товару за знижкою, скільки потрібно часу з цією знижкою продати весь товар. І після ухвалення важких рішень на скільки готові продати цей товар за знижкою. Вже вибрана ціна з'являється на сайті, далі вже йде перевірка на кількість кліків, якщо кліки під час знижки, наприклад, за місяць збільшилися вдвічі або втричі, ніж за минулий місяць. Отже обране рішення було правильним, дана знижка змогла привернути увагу відвідувача.

Навіщо проводити А/В тестування сайту?

Проводити А/В тестування потрібно для оптимізації сторінки сайту або всього проекту на отриманих даних від користувачів, таке можливо зробити в

окремому компоненті, де після будь-якої події або за певну кількість часу з'являтиметься модальне вікно з побажаннями або просто залишити відгук як зручно користуватися сайтом. Також можливо зробити окрему сторінку для побажання користувачів що їм хотілося б побачити на сайті або змінити. Після збору інформації виявити, що найбільше було в списках в бажаних, створити таблицю де будуть всі торкнуті теми і збоку від них кількість людей які її торкнулися, візьмемо за приклад змінити розмір і колір кнопки під час підтвердження покупки. Після цього можна буде створити другу версію кнопки і показувати різним користувачам стару версію і нову і після цього зібрати дані де вже ми дізнаємося сподобалося їм чи ні.

Також для любої компанії А/В тест грає велику роль для того що би мінімізувати ризики і фінансові втрати від впроваджених не вдалих змін. Наведу пару прикладів які ризики можуть виникнути без А/В тестування

1) Це втрата конверсії через невдалі зміни

Маємо приклад де змінюємо колір кнопки «Купити» з зеленого на червоний без тесту користувач може замислитися, що це є помилка або попередженням, це можливо знизить клікабельність. Також якщо ця кнопка була створена для покупки товару можливе падіння продажів на великі проценти, для компанії це може вдарити по бюджету.

2) Можливе погіршення користувацького досвіду UX

Уявимо, що є користувач, який часто відвідує сайт і без використання А/В тесту розробники вирішили перенести форму відправки замовлення в інше місце, для користувача це може зайняти дуже багато часу і скласти незручності, що великий шанс ймовірності, що можна втратити клієнта.

3) Втрата довіри клієнта

Якщо робити часті невдалі зміни інтерфейсу без впровадження А/В тесту це може викликати роздратування постійних клієнтів, велика ймовірність що покинуть цей проект, або буде мінімальний трафік, так само можна очікувати негативні відгуки з боку користувачів та падіння рейтингу бренду та прибутку.

Як А/В-тестування знижує ризики? [2]

1) Перед правильним початком тестування потрібно проаналізувати кількість користувачів які відвідують сайт, перевірити який трафік відбувається за день, тиждень, місяць і тому можна взяти певний відсоток користувачів наприклад 5-10% і почати тестувати. З кожним новим днем, тижнем або місяцем можна додавати до 15-20% і тільки в тому випадку, якщо показники при 5-10% виявилися позитивними, якщо показники виявилися негативними, то краще припинити тестування, оскільки є великий шанс ймовірності втратити довіру у користувачів до проекту.

2) Потрібно приймати рішення виключно на основі статистично значущих результатів, проведених тестами, а не в особистих потребах. Наприклад, якщо команда або керівник вимагає змінити форму до мінімальних розмірів і додати червоне тло, яке може ускладнити видимість тексту, якщо він чорний, це може вплинути на втрату користувачів і на негативні відгуки.

3) Навіть якщо тест виявився позитивним, то всю зміну потрібно вносити поступово, а не відразу. Проблема полягає в тому, що є можливість негативний тренд на ранній стадії, або непередбачені ефекти наприклад нова версія працює гірше на мобільних пристроях. Що так само може вплинути на втрату користувачів та часу.

Тепер перейдемо до найголовнішого, це як правильно проводити АБ тестування.

1) Потрібно визначити мету для чого нам потрібно використовувати А/В тестування. Для цього ми для початку думаємо і вирішуємо, що ми хочемо покращити. Наприклад перемістити кошик зі сторінки в модальне вікно, що випадає.

2) Потрібно вирішити як це нам допоможе зміщення кошика зі сторінки в модальне вікно, напевно, для зручності користування та концентрації користувача тільки на кошику, щоб інші компоненти не забирали зайву увагу

3) Після вирішення з гіпотезою проаналізуємо який вид метрика будемо використовувати, це може бути : процент кліків відносно показів, Bounce Rate

– це процент користувачів які покинули сайт без дій, середній час перебування на сторінці, кількість зроблених кліків та оглянутих сторінок, наскільки довго перегортали сайт тобто глибина прокрутки, вибір оцінки після завершення покупки, або прочитання статті.

4) Як вирішили з вибором метрика пора вже створювати дві, три або більше компонентів, які будуть порівнюватися. Після створення компонентів, які будуть порівнюватися з нинішньою версією, ми починаємо експеримент, очікуємо поки почне приходити інформація, чекати доведеться до погодженого часу.

5) Після того як настав час припинити збирати дані, приступаємо до порівняння результатів за виданий термін, з'ясуємо який з компонентів краще і чи підтвердилася наша задумка при поліпшенні кошика перевівши її в модальне вікно.

б) Після підтвердження задуму та отримання позитивних даних можна вносити зміну на постійну основу. Було б чудово якщо зафіксувати отримані дані для подальших тестів, є варіант, що вони можуть стати в нагоді ще раз.

1.4.1 Методи статистичного аналізу

Байєсівські методи аналізу. Даний підхід ґрунтується на ітеративному вдосконаленні ймовірнісних оцінок шляхом інтеграції апріорних знань з новими експериментальними даними. На противагу класичним статистичним методам, він надає результати у формі ймовірнісних оцінок, що є особливо цінним при обмежених обсягах вибірки. Такі методи демонструють високу ефективність у ситуаціях, що вимагають оперативного прийняття рішень на основі обмежених даних.

Алгоритми адаптивного розподілу (МАВ). Ця інноваційна методика реалізує динамічний підхід до розподілу ресурсів, оптимально балансує між дослідженням нових варіантів і використанням найбільш ефективних рішень. Вона дозволяє суттєво знизити втрати ефективності під час

проведення тривалих експериментальних досліджень. Особливу актуальність цей підхід набуває в умовах обмеженого трафіку або при необхідності постійної оптимізації робочих процесів.

Метод поетапного аналізу. Відмінною рисою цього підходу є можливість проміжного аналізу результатів без необхідності очікування завершення повного циклу збору даних. Це дозволяє значно скоротити час отримання статистично значущих результатів. Найбільш широке застосування даний метод знайшов у сфері розробки мобільних додатків та інтерактивних сервісів, де швидкість отримання результатів має вирішальне значення.

Метод диференційованого порівняння (DiD). Ця методика передбачає ретельний аналіз динаміки змін як у контрольній, так і в експериментальній групах до та після впровадження змін. Такий підхід дозволяє ефективно ізолювати вплив сторонніх факторів на кінцеві результати. Найбільшу ефективність метод демонструє при оцінці результатів маркетингових стратегій, змін у ціновій політиці та інших довгострокових бізнес-ініціатив.[8]

Метод ітераційного ресемплінгу(Bootstrap-метод). Суть даного підходу полягає у багаторазовому аналізі даних через механізм створення віртуальних вибірок на основі наявного масиву інформації. Він не вимагає жорстких припущень щодо характеру розподілу вихідних даних і забезпечує стабільні результати навіть при обмежених обсягах інформації. Особливу цінність цей метод представляє при роботі з нестандартними даними або складними комплексними показниками. Приклад роботи зображений на рисунку 1.9. Це графічна ілюстрація методу bootstrap на невеликій вибірці, що містить $n = 3$ спостереження. Кожен набір даних bootstrap містить n спостережень, вибірково замінені з вихідного набору даних. Кожен набір даних bootstrap використовується для отримання оцінки α . [13]

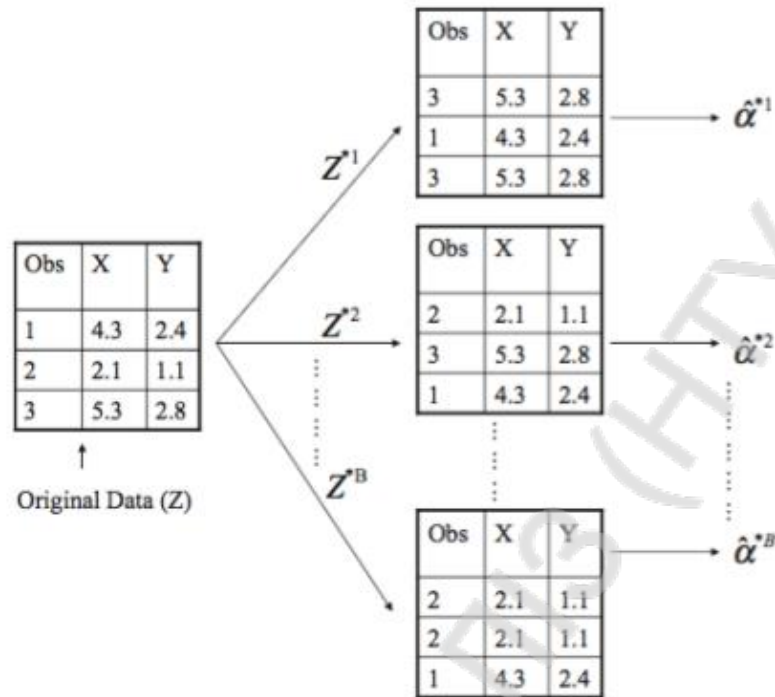


Рисунок 1.9 - Приклад роботи методу bootstrap.

Отже для кваліфікаційної роботи було використано саме Метод ітераційного ресемплінгу (Bootstrap-метод) так як він є оптимальним вибором для цієї роботи завдяки його простоті, швидкості та надійності. Angular через його масштабованість, структурованість і підтримку сучасних підходів до розробки. Як база даних використовувалася Firebase Firestore, що забезпечує надійне зберігання даних у реальному часі, зручну інтеграцію з фронтендом і гнучкі можливості для подальшого розширення функціоналу.

РОЗДІЛ 2 СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Розробка бази даних товарів з використанням платформи Firebase

На цьому етапі буде проведено дослідження інформаційної системи, що необхідна для зберігання товарів на сайті. В рамках веб-магазину FlowerShop основним об'єктом зберігання є Квіткові композиції або Декоративні рослини, кожна з них має свою характеристику, яка описує її опис, призначення, категорію і т.д.

Firebase – це платформа для розробки мобільних і веб проєктів. Її розробила компанія Firebase, а згодом купила корпорація Google. Firebase надає розробникам комплект сучасних сервісів, які роблять простішим проєктування, розробку і масштабування додатків, що надаються на ринку без створення власної архітектури. Головні можливості Firebase:

- Realtime Database та Cloud Firestore – хмарні бази даних для зберігання та синхронної передачі даних у реальному часі .
- Authentication – сервіс швидкої авторизації користувача, за допомогою email, номеру мобільного телефону, та різноманітних соціальних мереж.
- Hosting – де ми маємо змогу виставити свій веб-застосунок у мережі з підтримкою HTTPS і глобальної CDN.

Для збереження товарів будемо використовувати Cloud Firestore, оскільки він дозволяє зручно організувати дані, підтримує складні запити, краще масштабується і швидше працює з великими обсягами інформації.

Firestore використовує документи та колекції, що забезпечує більшу гнучкість в організації даних.

Тепер до головного – які риси має кожний товар на веб-застосунку:

- id – це унікальний ідентифікатор товару ,
- name - назва товару,
- categories – категорія квітки (наприклад «House Plants»),

- price – ціна товару,
- description – короткий опис товару,
- productDescription – детальний опис товару,
- img – шлях до зображення,
- grade - оцінка товару ,
- state – логічне значення наявності товару (true/false),
- livingRoom, diningRoom, office – опис можливостей розміщення у відповідних кімнатах.
- size – це об'єкт який має декілька розмірів товарів (наприклад «Small»),

Маємо таку базу даних де ми маємо колекцію «products» яка знаходиться в firestore в якому маємо 10 товарів та розроблену ER-діаграму . Товари представлені на рисунку 2.1, 2.2 та розроблений інтерфейс до додатку на рисунку 2.3 , ER-діаграма 2.4

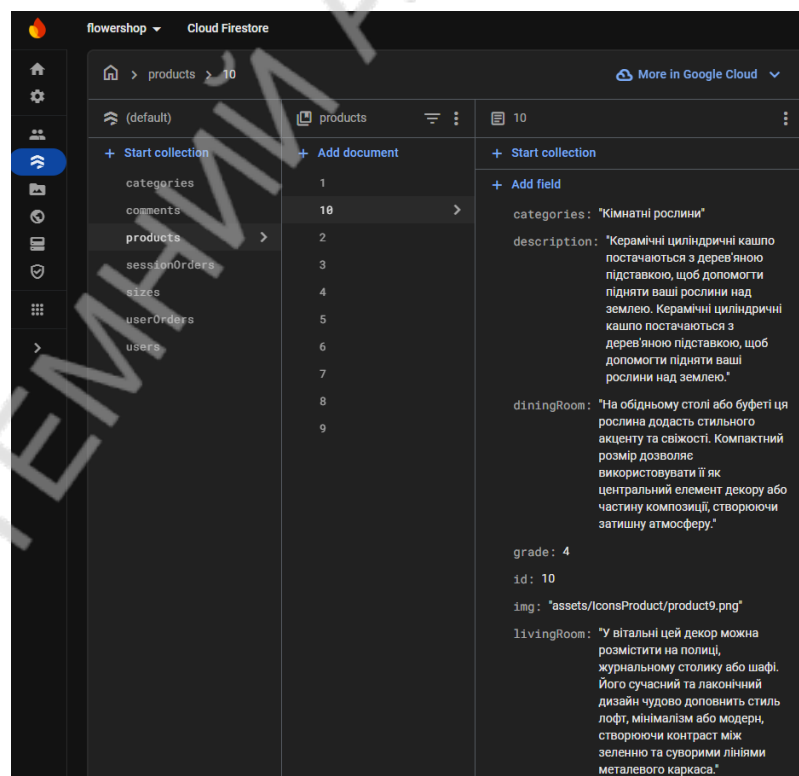


Рисунок 2.1 – Колекція «products» з першим товаром у списку та його описом.

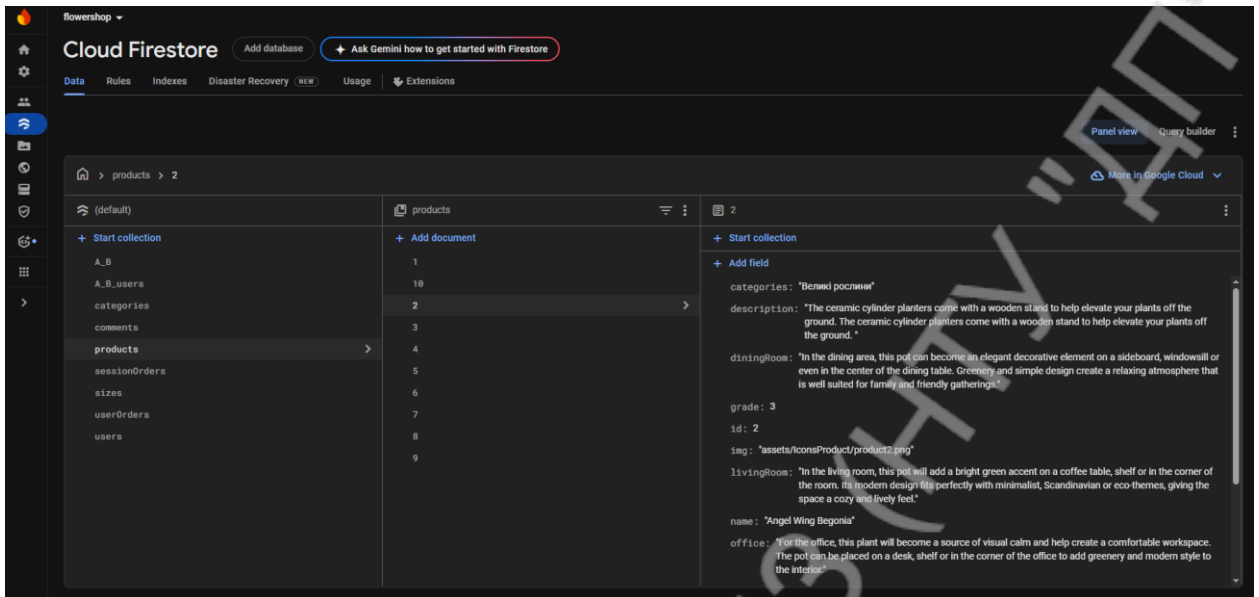


Рисунок 2.2 – Колекція «products» з другим товаром у списку та його описом.

```

export interface ProductsAll {
  img: string;
  id: number;
  name: string;
  categories: string;
  description: string;
  productDescription: string;
  quantity: number;
  livingRoom: string;
  diningRoom: string;
  office: string;
  price: number;
  grade: number;
  size: {
    name: string;
    state: boolean;
  };
};

```

Рисунок 2.3 – розроблений інтерфейс для отримання товарів з firestore.

На рисунку 2.4 нижче наведено ER-діаграму, яка ілюструє структуру даних колекції products:

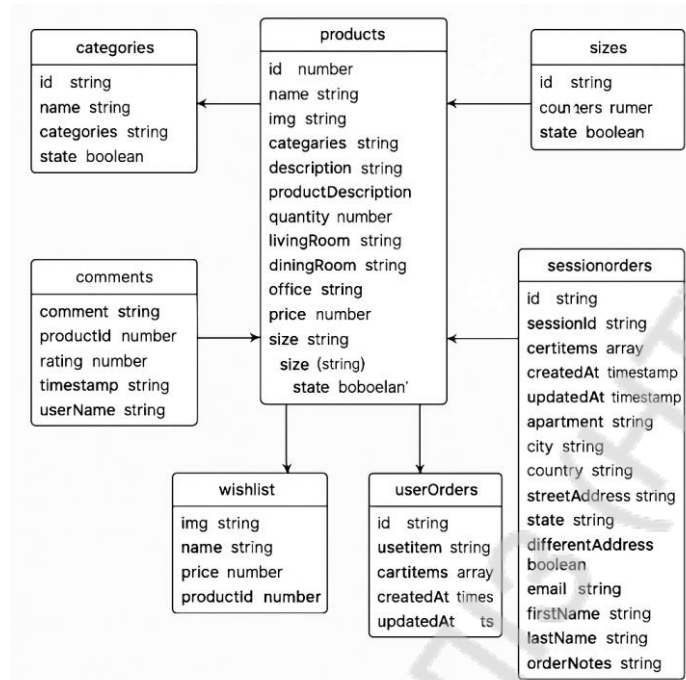


Рисунок 2.4 – ER-діаграма колекції «products».

2.2 Аналіз інтерфейсу кошика на веб-сайтах

Перед тим як починати розробляти веб-застосунок проведемо аналіз у різноманітних веб-сайтів з пошуком ефективного веб-інтерфейсу кошика. Це допоможе визначити основні принципи побудови інтерфейсу, що забезпечують зручність та ефективність для користувача.

Метою цього розділу є проведення візуального та функціонального аналізу інтерфейсів кошика на українських веб-сайтах, що займаються продажем квітів. Аналіз спрямований на виявлення сильних та слабких сторін існуючих рішень.

2.2.1 Візуальний та функціональний аналіз

Аналіз підстави вибрав три українських інтернет-сайти по продажу квітів:

1) flowers.ua – це інтернет-магазин, який популярний у нас в Україні. Вони здійснюють доставку квітів, букетів, подарунків та композицій майже у всі

1015 міст України. Сайт також містить широкий асортимент подарунків та букетів більш ніж 1015 міст по Україні.

2) buket24.dp.ua – сайт, орієнтований на доставку квітів і подарунків по Дніпру та області. Тут ви можете також знайти широкий асортимент не тільки квітів, а подарунків і інших елементів на весілля, ювілейні свята та інші.

3) royal-flowers.dp.ua – Сайт по заказу квітів та інших постачання певних товарів у м.Дніпро. Тут також можна знайти самі популярні замовлення на букети, квітки, різноманітні кошики, та декорації до шару пахучих пластикових квіточок. Лавка має гарний сервіс а також оформлення доставки квітів по місту та області.

Критерії, за якими проводився аналіз : швидкий перегляд кошика. Можливість переглядати вміст кошика не переходячи на окрему сторінку. Редагування кількості товару. Зручність зміни кількості обраних товарів .Видалення товару. Відображення загальної вартості. Видання ціна і грошові накопичення. Товарообіг. Візуальна привабливість. Естетичний вигляд інтерфейсу. Простота навігації , інтуїтивно зрозуміла та зручна.

Сайт flowers.ua має більш традиційний вигляд, який для багатьох може здаватися стандартним у порівнянні з іншими варіантами. Кошик доступний лише з окремої сторінки. У внутрішньому розділі корзини, куточку, який обмежує зліва, реалізовано форму для введення особистих даних клієнта, яка не так вже й зручна для перегляду замовлених товарів, тому що вікно, що відкривається, перекриває частину екрану зі змістом. Крім того, в правій частині екрану, вже знайомій, у нас є інформація про товари в корзині. Примітно, що дана інформація також займає досить обмежений простір, що теж ущемляє огляд. Знизу монітора або верх кошика під його рамкою, для зручності, видно вже пропозиції «по додатковому товару». Це заважає користувачу, оскільки ці елементи займають зайве місце на сторінці і відволікають увагу від основного вмісту кошика. Такі товари не повинні бути розміщені там, оскільки вони лише засмічують простір і можуть створювати

додаткові незручності під час оформлення замовлення. Рисунок 2.5 демонструє вигляд кошика на сайті.

Оформлення замовлення

1. Ваші дані

Email (необов'язково)

Ваше ім'я UA +380

Продовжити

2. Кому доставити

3. Куди доставити

Ваші товари: 6 507 грн

| | | |
|---------------------|---|-------------|
| | Букет в ЕКО упаковці "15 червоних троянд" | 1 599 грн |
| | Букет з 51 червоної троянди | 4 859 грн |
| Додатково | | |
| Листівка | | 49 грн |
| Доставка | | Безкоштовно |
| Бажаю додати | | |
| | Купа "Люблю тебе, мамо" | 299 грн |
| | Raffaello | 299 грн |
| | Бенто Торт, 300г | 699 грн |
| | 3 гелієві кульки (червоні серця) | 259 грн |

Рисунок 2.5 – кошик сайту flowers.ua.

Сайт bucket24.dp.ua має набагато приємніший і сучасніший дизайн у порівнянні з flowers.ua. Всі елементи виглядають гармонійно, інтерфейс значно приємніший. Кошик доступний через окрему сторінку, що допомагає зменшити безлад на головній сторінці.

На відміну від flowers.ua, де є розділ «Додаткові товари», який займає місце і відволікає від основного вмісту кошика, bucket24.dp.ua не має нічого подібного, але має форму інформації про користувача, яку можна було б заповнити на наступній сторінці. Це зробило навігацію по сайту більш комфортною, оскільки не було зайвої інформації, яка могла б відволікати від перегляду асортименту. Рисунок 2.6 демонструє вигляд кошика на сайті.

Рисунок 2.6 – кошик сайту buket24.dp.ua.

Сайт royal-flowers.dp.ua має більш елегантний стиль з спливаючим вікном, що дозволяє користувачам переглядати вміст кошика без потреби переходити на нову сторінку. Це значно полегшує процес покупок, оскільки не потрібно втрачати час на переходи. Інтерфейс кошика дуже простий та зручний для редагування кількості товарів, а також дозволяє швидко видаляти елементи. У кошику не має нічого зайвого, що може забирати увагу на перегляд кошику та вибору рішень. Функціонал сайту включає відображення кількості товарів та актуальної суми замовлення, можливе видалення не потрібного товару з кошику. Рисунок 2.7 демонструє вигляд кошика на сайті.

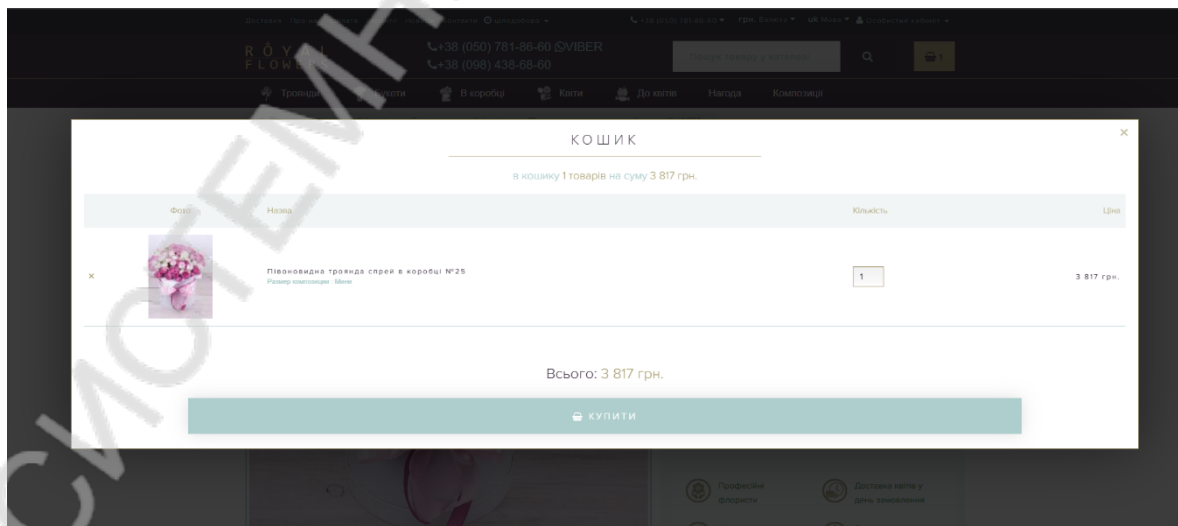


Рисунок 2.7 – кошик сайту royal-flowers.dp.ua.

2.2.2 Порівняльна таблиця рішень

За допомогою проведеного аналізу трьох сайтів ми створимо порівняльну таблицю рішень 2.1, за критеріям, які були описані раніше, що ілюструє оцінки у відсотках (від 0% до 100%) та показує, який сайт більш підходить для подальшого розгляду, було створено діаграму, яка ілюструє оцінки в графічному вигляді. Ця діаграма зображена на рисунку 2.8

Таблиця 2.1

Порівняльна таблиця рішень

| Критерій | flowers.ua | buket24.dp.ua | royal-flowers.dp.ua |
|---------------------------------|------------|---------------|---------------------|
| Швидкий перегляд кошика | 40% | 60% | 90% |
| Редагування кількості товару | 50% | 70% | 90% |
| Видалення товару | 50% | 70% | 80% |
| Відображення загальної вартості | 60% | 70% | 90% |
| Візуальна привабливість | 60% | 80% | 90% |
| Простота навігації | 50% | 80% | 90% |

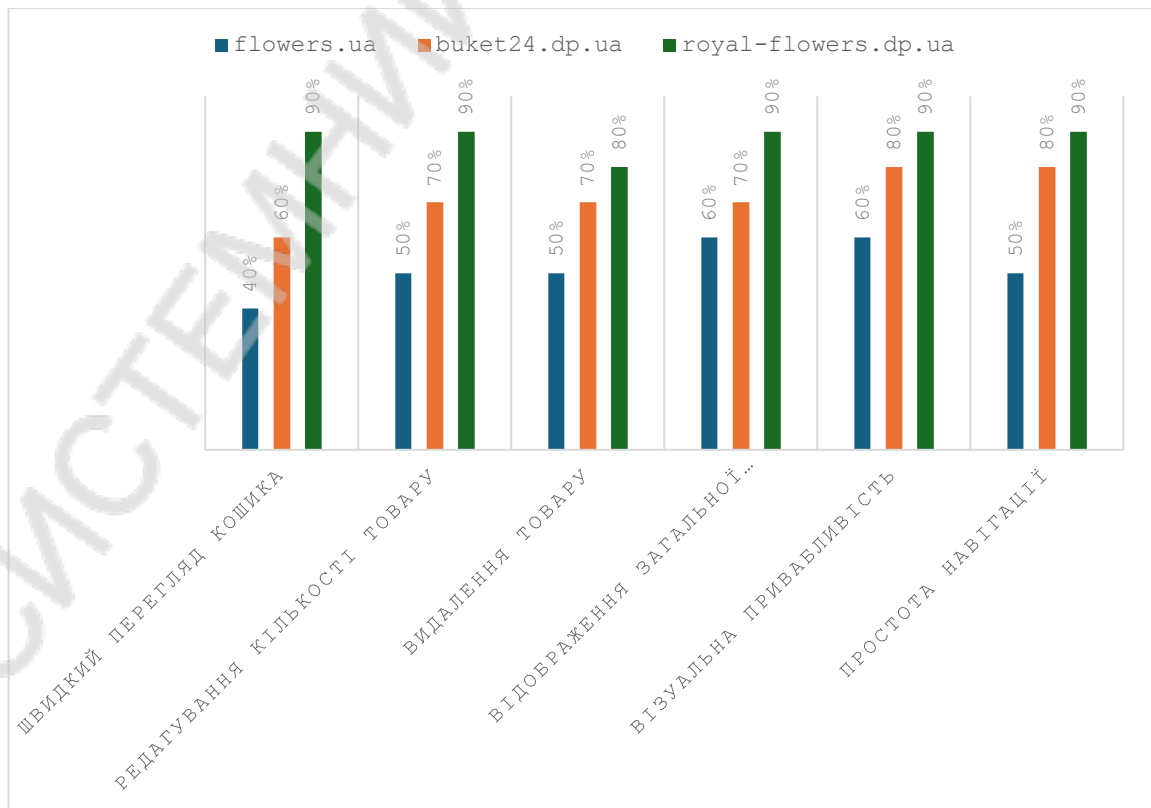


Рисунок 2.8 – графік порівняльної таблиці рішень.

Після проведеного аналізу ми можемо бачити, що найвищі оцінки має сайт `royal-flowers.dp.ua`: на рисунку 2.7 середня арифметична оцінка складає 83.33%. У сайту `bucket24.dp.ua` складає 71.67%, у `flowers.ua` найнижча оцінка 51.67%. Таким чином, більшу увагу привертають сайт `royal-flowers.dp.ua` та `bucket24.dp.ua`. Тому будемо брати приклад з цих двох сайтів який в пріоритети буде `royal-flowers.dp.ua` на рисунку 2.7.

2.3 Розробка веб-додатку

При проектуванні веб-додатку для магазинів квітів було вирішено використати сучасні, перевірені технології, щоб забезпечити швидкість, зручність та масштабованість.

Обрали фреймворк `Angular` для розробки `fronted`, оскільки він добре підходить для сфери розробки односторонникових додатків. Код програми буде реалізовано з використанням мови програмування `TypeScript`. Варто зазначити, що цьому мовному середовищу властива статична типізація, що дозволяє запобігти багатьом проблемам в процесі написання коду. Для здійснення технічної реалізації UI планується використання `HTML5` та `CSS3`. На задньому плані для роботи з даними буде обрано `Firebase` разом із `Firestore` для втілення функцій автентифікації користувачів, а також зберігання товарів у кошику і на акаунті користувача в залежності від стану замовлення на товар.

2.3.2 Архітектура та основні функціональні можливості додатку

Застосунок буде розроблений за композитною моделлю. Це означає, що кожен елемент інтерфейсу, такий як заголовок профілю та кошик для покупок, розробляється окремо як автономні одиниці з їхньою відповідною логікою та структурою. З таким підходом значно простіше вносити зміни і не боятися зламати щось.

Одним з основних компонентів цієї роботи є кошик для покупок клієнта. Він буде спроектований таким чином, щоб користувач міг переглядати обрані товари, змінювати кількість товарів, видаляти товари та бачити загальну вартість замовлення. Каталог товарів буде покращено зручним фільтром, який буде містити варіанти вибору для категорії товару, розміру товару, цінового діапазону та сортування товарів. Також передбачена реєстрація користувачів за допомогою електронної пошти та пароля або через соціальні мережі. Також передбачено особистий кабінет, де користувач може змінювати свої дані, переглядати свої замовлення та бачити нещодавно додані побажання. Головний банер, каталог продуктів, новини та актуальні акції будуть розміщені на головній сторінці, щоб зацікавити клієнтів. На другій сторінці вони зможуть знайти кошик для покупок, а додаткові новини будуть містити товари.

Взаємодія між компонентами буде здійснюватися за допомогою сервісів, які будуть відповідальні за завантаження, оновлення та обробку замовлень, а також управління кошиком для покупок. Навігація в застосунку буде здійснюватися за допомогою вбудованої системи маршрутизації Angular, яка дозволяє користувачам швидко переходити між розділами без перезавантаження сторінки.

2.3.3 Реалізація інтерфейсу кошика та інших важливих компонентів

Перед тим як створювати кошик, потрібно зробити шапку сайту, де буде логотип нашого сайту, навігація для вибору сторінки, пошуку товарів за допомогою input, особистий кабінет, кошик, та кнопку логіна. Після створення шапки ми повинні створити каталог товарів за допомогою якого користувач може вибирати зацікавлені товари та добавляти до кошика.

Для початку створюємо шапку сайту, цей компонент назвемо header на рисунку, також продемонструю шматки коду верстки 2.9



Рисунок 2.9 – компонент header.

Наступним етапом створюємо компонент productbody на рисунку 2.10, який буде вміщати у себе товари. Пишемо верстку та логіку компонента.

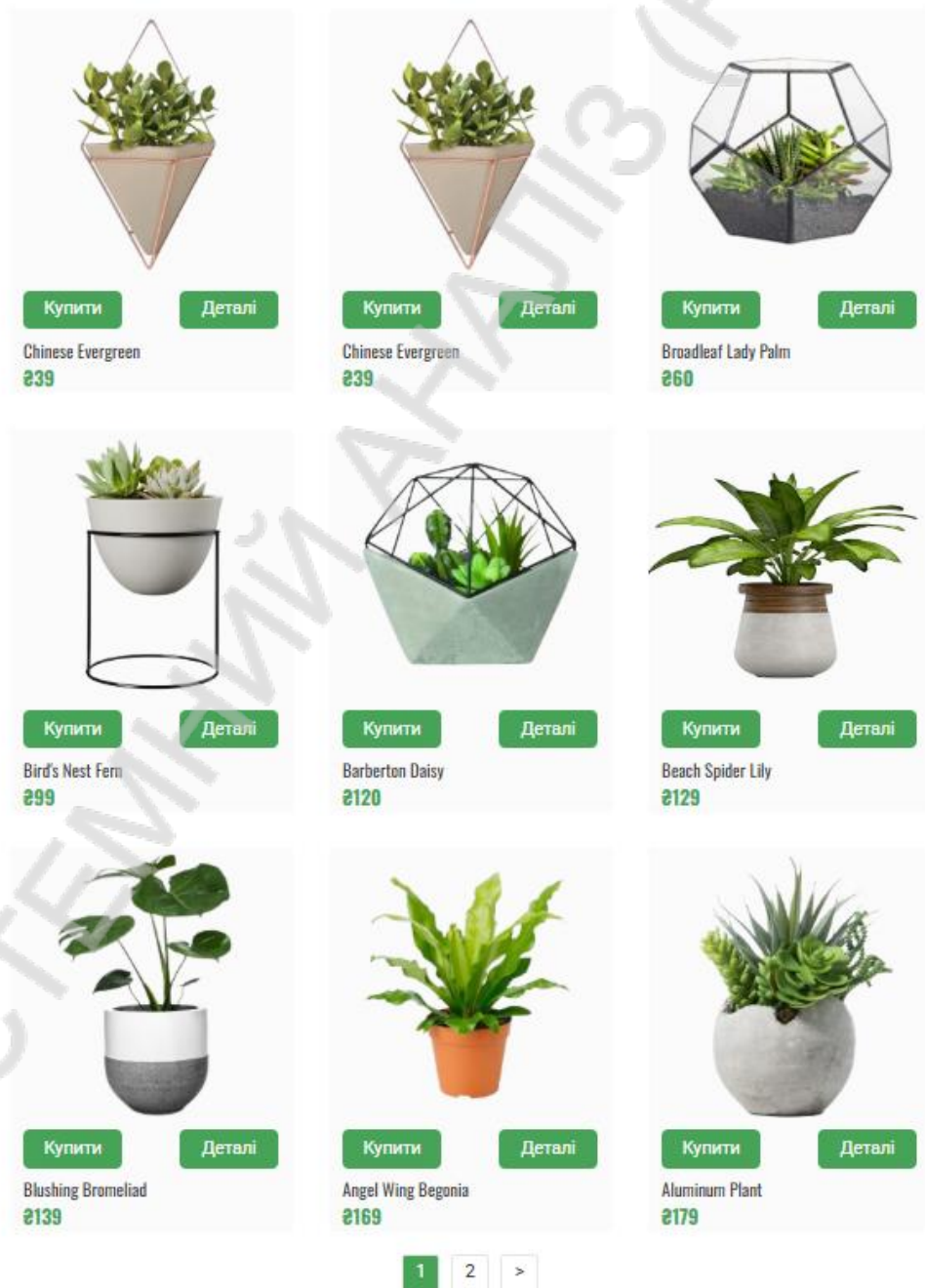


Рисунок 2.10 – компонент productbody.

Також створимо компонент sidebar, який буде відповідальний за фільтр товарів та вибір категорії та розміру товару. Створюємо логіку та верстку компоненту, та заповнюємо її каскадною таблицею стилів. Верстка зображена на рисунку 2.11.

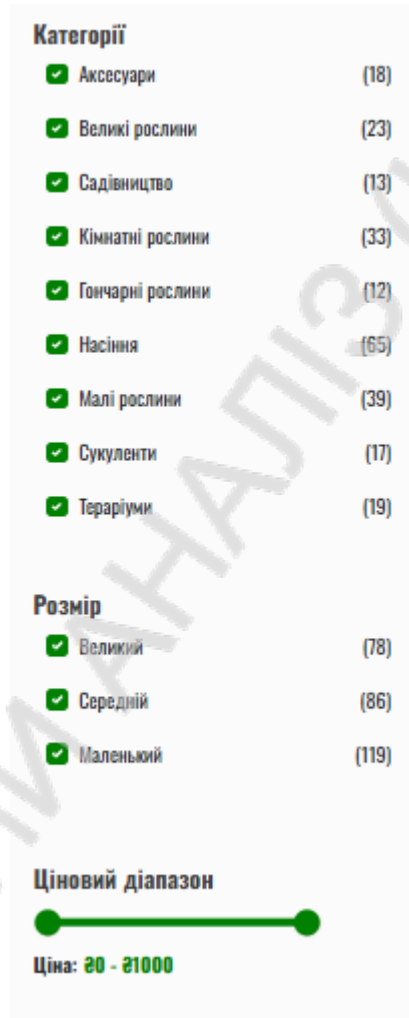


Рисунок 2.11 – компонент sidebar.

Оці два компонента productbody та sidebar виведемо на сторінці з назвою home або головна на рисунку 2.12, ці два компонента зможуть взаємодіяти між собою через сервіси.

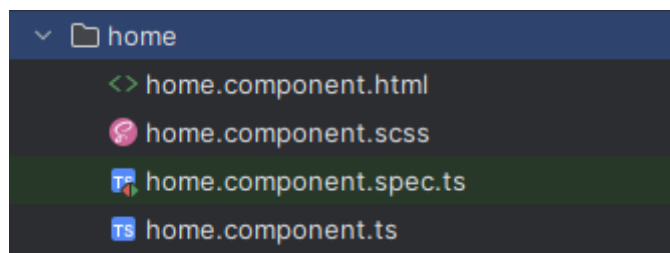


Рисунок 2.12 – компонент home.

Надалі будемо доповнювати сторінку home іншими компонентами які були описані вище . Наразі кожному товару додаємо дві кнопки це купити яка буде відповідати за наповнення нашого майбутнього кошика та деталі де ми можемо продивитись опис товару більш детальнішим , також товарам додаємо назву ,ціну та його зображення. Візуальна карточка товару зображена на рисунку 2.13



Рисунок 2.13 – товар з компонента productbody.

Після створення та з'єднання всіх цих компонентів у компонент home ми маємо такий візуальний результат який може бачити користувач коли заїде на сайт квітів який зображений на рисунку 2.14

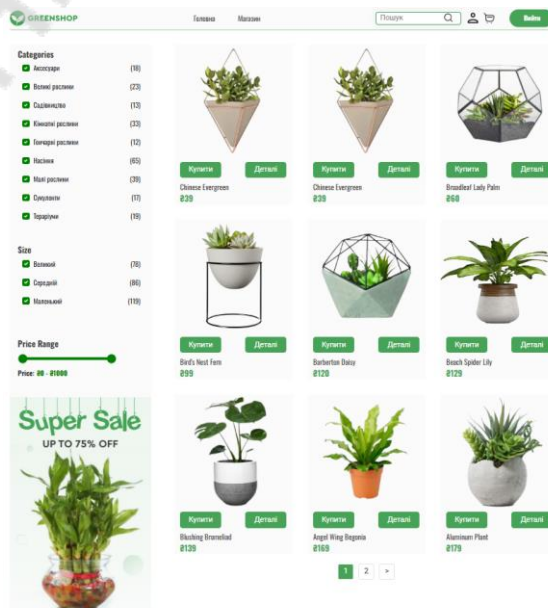


Рисунок 2.14 – компонент home.

Після створення основної сторінки настав час створювати кошик, з початку створимо компонент з назвою shopping-cart-a на рисунку 2.15.

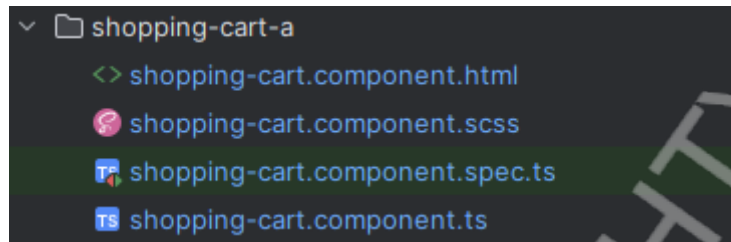


Рисунок 2.15 – компонент shopping-cart-a.

Зробимо логіку отримання товару з сторінки home у кошик, за допомогою гетера cartItems, зображено на рисунку 2.16.









```

11 export class ShoppingCartComponent implements OnInit {
12   private cartService: CartService = inject(CartService);
13   private router: Router = inject(Router);
14   private authService: AuthService = inject(AuthService);
15   couponCode: string = '';
16   discount: number = 0;
17   shipping: number = 10;
18   validCoupons: { [key: string]: number } = {
19     "DISCOUNT100": 100,
20     "SAVE100": 100
21   };
22   user$: Observable<User | null> = this.authService.user$;
23
24   constructor() {}
25
26   ngOnInit(): void {}
27
28   get cartItems(): any[] {
29     return this.cartService.getCartItems();
30   }
31
32   get cartTotal(): number {
33     const total: any = this.cartService.getTotalAmount();
34     return total - this.discount;
35   }
36
37   getItemTotal(productId: number): number {
38     return this.cartService.getTotalItem(productId);
39   }
40
41   proceedToCheckout(): void {
42     this.router.navigate(['/shop/check-out'], {
43       state: {
44         cartItems: this.cartItems,
45         discount: this.discount,
46         couponCode: this.couponCode
47       }
48     });
49   }

```

Рисунок 2.16 – логіка компонента shopping-cart.

Після цього створимо першу версію кошика, який буде функціональним і візуально привабливим і трошки схожий як на сайті bucket24.dp.ua на рисунку 2.6, але поки що без складних анімацій чи додаткових ефектів, зображення на першу версію кошика на рисунку 2.17.

| Продукти | Ціна | Кількість | Всього | |
|---|----------|-----------|----------|---|
|  Chinese Evergreen SKU: 1995751877966 | \$39.00 | - 1 + | \$39.00 |  |
|  African Violet SKU: 1995751877966 | \$219.00 | - 1 + | \$219.00 |  |
|  Beach Spider Lily SKU: 1995751877966 | \$129.00 | - 1 + | \$129.00 |  |
|  Chinese Evergreen SKU: 1995751877966 | \$39.00 | - 1 + | \$39.00 |  |

| Підсумки кошика | |
|--|--------------------|
| Застосування купону | |
| <input type="text" value="Введіть код купону..."/> | Застосувати |
| Проміжний підсумок | 2426 |
| Знижка по купону | (-) 20 |
| Доставка | 210 |
| Всього | 2436 |

Перейти до оформлення

[Продовжити покупки](#)

Рисунок 2.17 – перша версія кошика з обраними товарами.

Після створення першої версії кошика, створимо другу версію яка буде більш привабливий, простішою, у вигляді модального вікна. Зображено на рисунку 2.18.

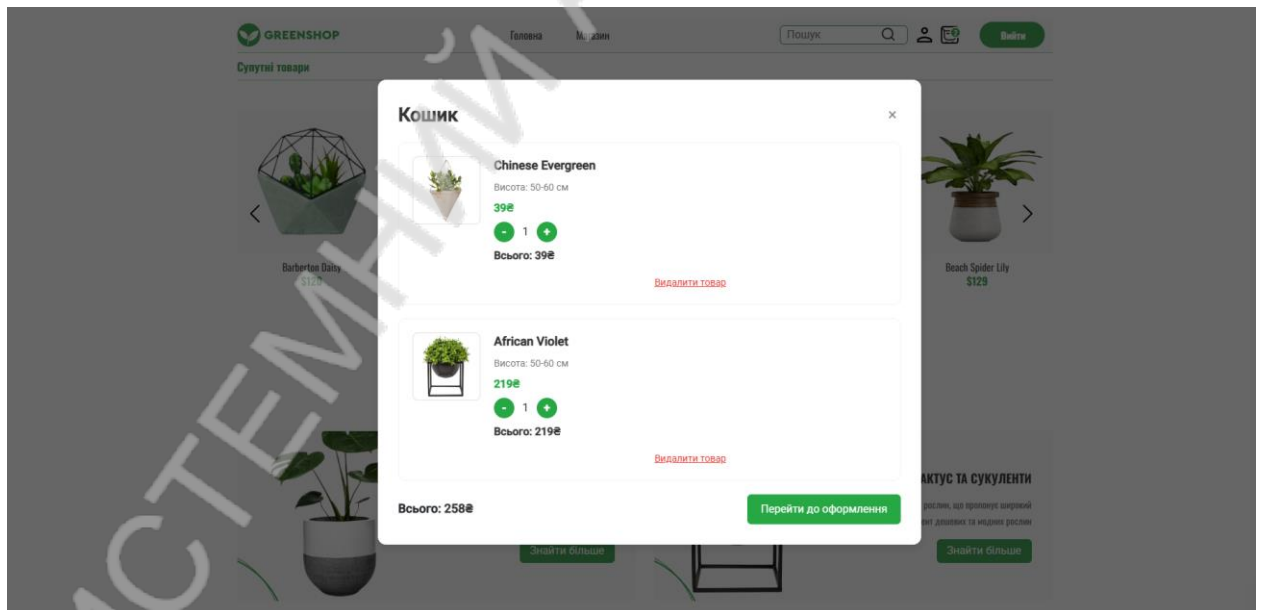


Рисунок 2.18 – друга версія кошика з обраними товарами.

Після створення двох різних компонентів кошика, розробимо компонент оформлення замовлення, який у себе буде включати: форму з інформацією про користувача, форму для введення платіжної інформації, а також підсумки

кошика з вибором способу оплати. Приклад оформлення замовлення можливо побачити на рисунку 2.19.

Адреса для виставлення рахунку

| | |
|--|--|
| Ім'я * | Прізвище * |
| <input type="text" value="Олександр"/> | <input type="text" value="Казарінов"/> |
| Країна * | Місто / Селище * |
| <input type="text" value="Україна"/> | <input type="text" value="Dnipro"/> |
| Адреса вулиці * | |
| <input type="text" value="Metrostroeviskaya 5"/> | <input type="text" value="dom 5, 776"/> |
| Область * | Електронна пошта * |
| <input type="text" value="Дніпропетровська"/> | <input type="text" value="stolbnyak2003@gmail.com"/> |
| Номер телефону * | Примітки до замовлення (необов'язково) |
| <input type="text" value="+380638902637"/> | <input type="text"/> |

Ваше замовлення






| Товари | Підсумок |
|---|-----------|
|  Chinese Evergreen Артикул: 1995751877966 (x 1) | \$39.00 |
|  African Violet Артикул: 1995751877966 (x 1) | \$219.00 |
| Підсумок | \$258.00 |
| Знижка за купоном | Купон (): |
| Доставка: | \$10.00 |
| Усього | \$268.00 |
| Спосіб оплати | |
| <input checked="" type="radio"/> PayPal    | |
| <input type="radio"/> Банківський переказ | |
| <input type="radio"/> Оплата при отриманні | |
| <input type="button" value="Оформити замовлення"/> | |

Рисунок 2.19 – оформлення замовлення.

2.4 Застосування методу А/В для покращення веб-додатку

Для застосування методу А/В розробимо модальне вікно на рисунку 2.20 після оформлення замовлення, де звернемо увагу користувача на зручність роботи веб-додатку, запропонуємо оцінити від одного до п'яти балів. Де компонентом А в нас буде перша версія кошику рисунок 2.17, а компонент В в нас буде друга версія кошику рисунок 2.18. Зачекаємо поки набереться достатньо оцінок з двох компонентів, після збору інформації буде проведено аналіз та покращення інтерфейсу продуктової корзини із використанням А/В тестування.

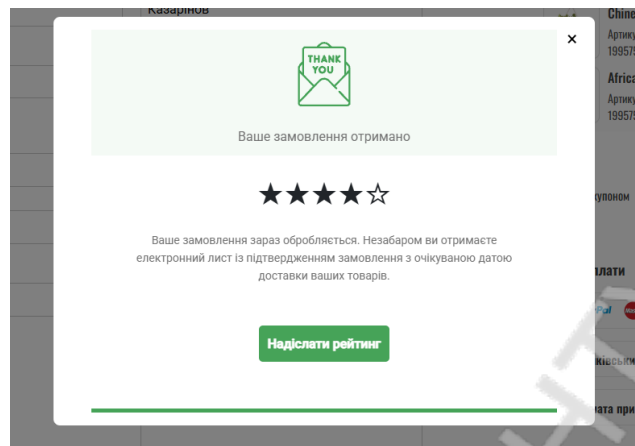


Рисунок 2.20 – підтвердження замовлення.

Замовлення буде прийняте та заблоковане після того, як користувач натисне кнопку "Надіслати оцінку". Користувач може переглянути своє замовлення у своєму особистому кабінеті рисунок 2.21.

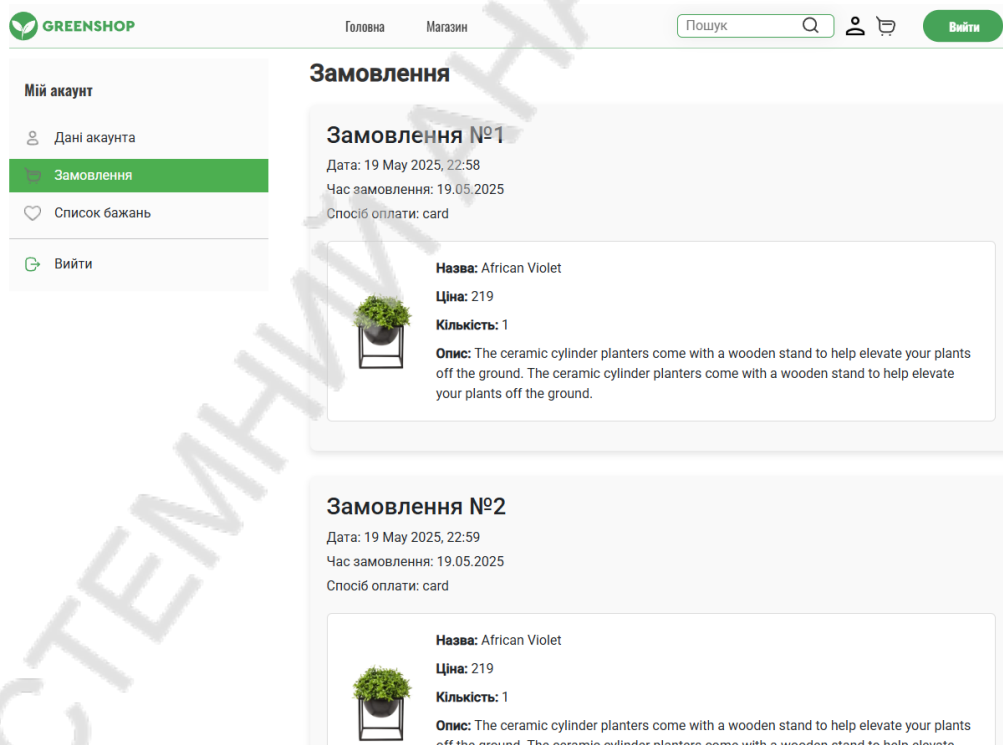


Рисунок 2.21 – Особистий кабінет користувача.

Після натискання на кнопку, оцінка веб-додатку буде зафіксована у колекції A_V у firestore рисунок 2.22. Ця колекція має два масиви:

- 1) A – оцінки користувачів, які працювали з кошиком A;
- 2) B – оцінки користувачів, які працювали з кошиком B.

Якщо користувачу дістався кошик А, його рейтинг потрапить у масив А, аналогічно для кошика В.

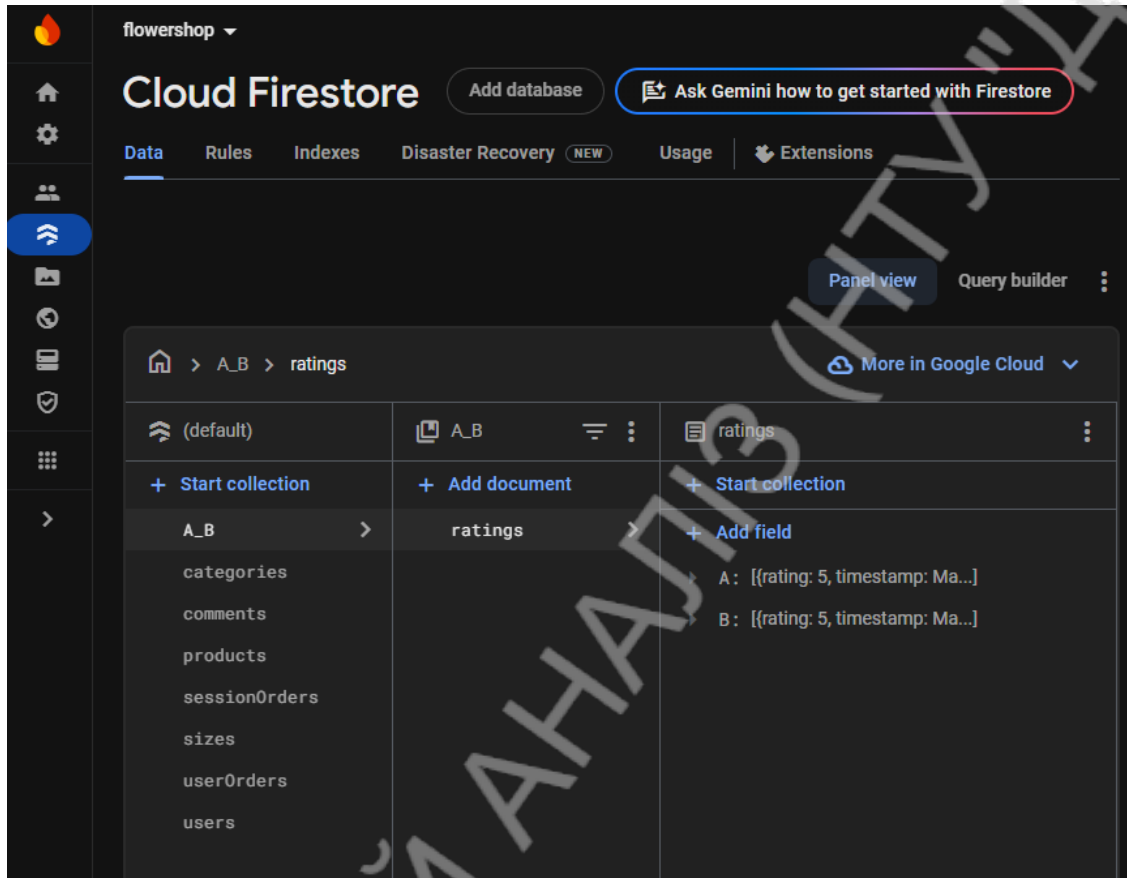


Рисунок 2.22 – Заповнена колекція А_В оцінками.

Після збору даних ми можемо починати робити А/В тестування. Для початку створимо компонент cart-ab-test в якому ми будемо робити тести. Після створення компоненту до даємо до проекту бібліотеку Chart.js, вона допоможе створювати різноманітні та інтерактивні графіки, також для Angular ми добавимо ng2-charts яка робить роботу з Chart.js більш простішою. Вона надає Angular-компоненти, які легко вставляти у шаблони та дозволяє просто налаштувати графіки через події Angular. Після встановлення та створення компоненту та бібліотек. Розробимо А/В тестування для двох варіантів кошика покупок у веб-додатку. Система буде складатися з кількох ключових частин:

- 1) Вибір варіанту для користувача – кожен новий користувач випадковим чином отримує варіант А або В, який зберігається у локальному сховищі браузера(localstorage) для послідовності досвіду.

- 2) Збір даних – ця система буде фіксувати, який варіант отримує користувач, та зберігаючи цю інформацію у базі даних firestore у колекції A_V.
- 3) Аналіз оцінок. Для обох випадків завантажуються оцінки користувачів, обчислюються середні значення та різниця між варіантами.
- 4) Статистичний аналіз. Bootstrap-метод використовується для оцінки значущості результатів..
- 5) Візуалізація результуючі bootstrap-аналізу є стовпчастою діаграмою.

Перший крок є механізм розподілу користувачів між кількома тестовими групами. Коли кожен новий користувач – реєстрація або перший візит – з вірогідністю 50/50 кидається в або В-інтерфейс кошика, комплект А використовується. Цей етап вже був обговорений.

Друга частина збирає та попередньо обчислює оцінки, зібрані від усіх користувачів, які зробили замовлення. На даний момент зафіксовано шістдесят оцінок які були розподілені на усі два варіанта кошиків рівномірно , зображено на рисунку 2.23

```

> temp1
< ▶ (30) [5, 4, 4, 3, 3, 5, 4, 5, 3, 5, 4, 2, 5, 5, 4, 5, 4, 4, 5, 4, 5, 4, 5, 4, 3, 5, 4, 5]
> temp2
< ▶ (30) [3, 3, 2, 5, 2, 4, 2, 3, 3, 4, 2, 3, 5, 4, 2, 3, 2, 3, 4, 3, 2, 2, 3, 4, 2, 3, 2, 3, 4, 2]

```

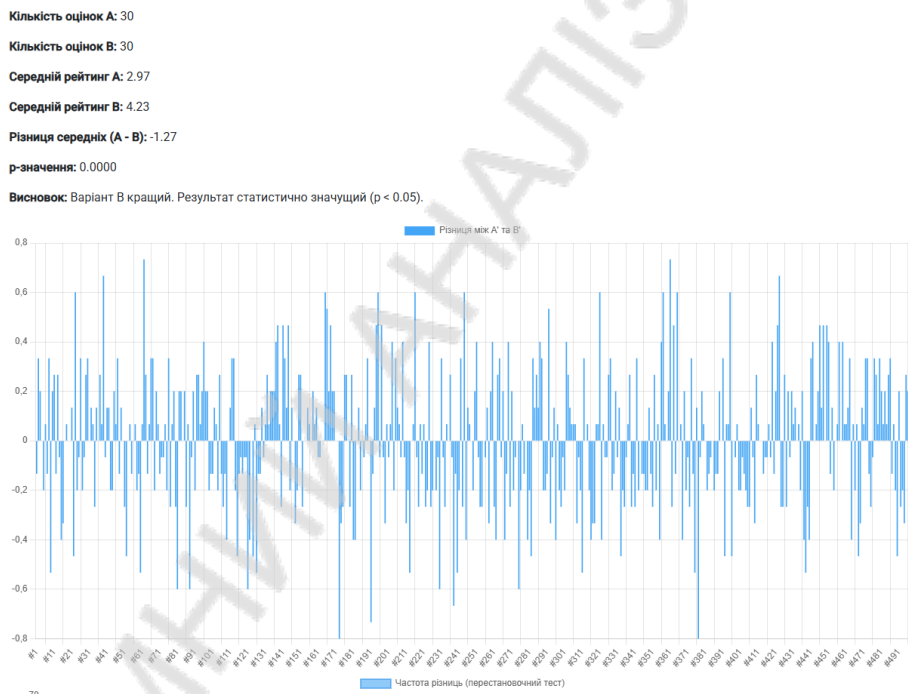
Рисунок 2.23 – Отримані оцінки з колекції A_V.

Аналізуючи дані, після того як заповнено оцінки, формуємо їх кількість, середній рейтинг кожного масиву, відставання середніх масивів та підсумкове побудує, який з компонентів є кращим. Як тільки дані стануть відомі, будемо використовувати метод Bootstrap. Метод Bootstrap - агрономічний метод із зразками, або ж його можна назвати статистичний аналіз, дуже легко працює із малими вибірками даних. Суть методу базується на такій послідовності дій:

1. Збір усіх згаданих наборів оцінок в один масив;
2. Багаторазове (500 ітерацій) виконання процедури ресемплінгу.

3. Формування емпіричного розподілу отриманих значень різниць.

Застосування методу bootstrap в даній системі є особливо доцільним, оскільки він не вимагає жорстких припущень про характер розподілу вихідних даних та залишається ефективним навіть при обмеженій кількості спостережень. Це дозволяє отримувати більш надійні результати на ранніх етапах тестування. Розробку та результати застосування bootstrap ми можемо побачити на рисунку 2.24, 2.25



```

56 assignVariant():void { Show usages
57   const stored:string|null = localStorage.getItem('cartVariant');
58   if (stored === 'A' || stored === 'B') {
59     this.selectedVariant = stored;
60   } else {
61     this.selectedVariant = Math.random() < 0.5 ? 'A' : 'B';
62     localStorage.setItem('cartVariant', this.selectedVariant);
63   }
64 }
65
66 saveVariant():void { Show usages
67   this.firestore.collection('A_B_Users').add({
68     variant: this.selectedVariant,
69     timestamp: new Date()
70   });
71 }
72
73 async loadAndAnalyzeRatings():Promise<void> { Show usages
74   const snapshot:DocumentSnapshot<unknown> = await this.firestore.collection('A_B').doc('ratings').ref.get();
75   const rawData:unknown = snapshot.data();
76
77   const data = rawData as RatingsData;
78
79   this.ratingsA = (data?.A || []).map((x:RatingEntry):number => x.rating);
80   this.ratingsB = (data?.B || []).map((x:RatingEntry):number => x.rating);
81
82   if (this.ratingsA.length === 0 || this.ratingsB.length === 0) return;
83
84   this.avgA = this.ratingsA.reduce((a, b) => a + b, 0) / this.ratingsA.length;
85   this.avgB = this.ratingsB.reduce((a, b) => a + b, 0) / this.ratingsB.length;
86   this.difference = this.avgA - this.avgB;
87
88   if (Math.abs(this.difference) < 0.05) {
89     this.conclusion = 'Різниця між варіантами незначна.';
90   } else if (this.difference > 0) {
91     this.conclusion = 'Варіант А кращий.';
92   } else {
93     this.conclusion = 'Варіант В кращий.';
94   }
95 }

```

Рисунок 2.25 – розробка A/B тестування.

Короткий опис кожної функції у компоненті cart-ab-test. Першим йде функція assignVariant – призначає користувача до варіанту А або В, зберігає вибір у localStorage.

2)saveVariant() – фіксує вибраний варіант у Firebase з міткою часу для подальшого аналізу.

3)loadAndAnalyzeRatings – завантажує оцінки користувачів з Firebase, обчислює середні значення для обох варіантів та визначає переможця тесту та ініціює bootstrap-аналіз.

4)bootstrapDiff(A, B, R) – виконує 500 ітерацій bootstrap- методу для оцінки статистичної значущості різниці між варіантами.

5)shuffle(array) – реалізує алгоритм Fisher-Yates для випадкового перемішування масиву. Використовується в bootstrap.

Аналізом результатів A/B тестування виходить варіант В переможцем, рисунок 2.17. Він набрав середній рейтинг 4.23 бали коли А варіант набрав

2.97 балів. Різниця між ними складає -1.27 балів це вказує на потенційну перевагу варіанту В.

Також було зроблено різницю середніх(А-В) за допомогою діаграми розкидання, яка реальну різницю складає -1.26667 . для цього було взято р – значення 0.05. І як ми можемо бачити на рисунку 2.26 , варіант В кращий та має результат статистично значущий ($p < 0.05$).

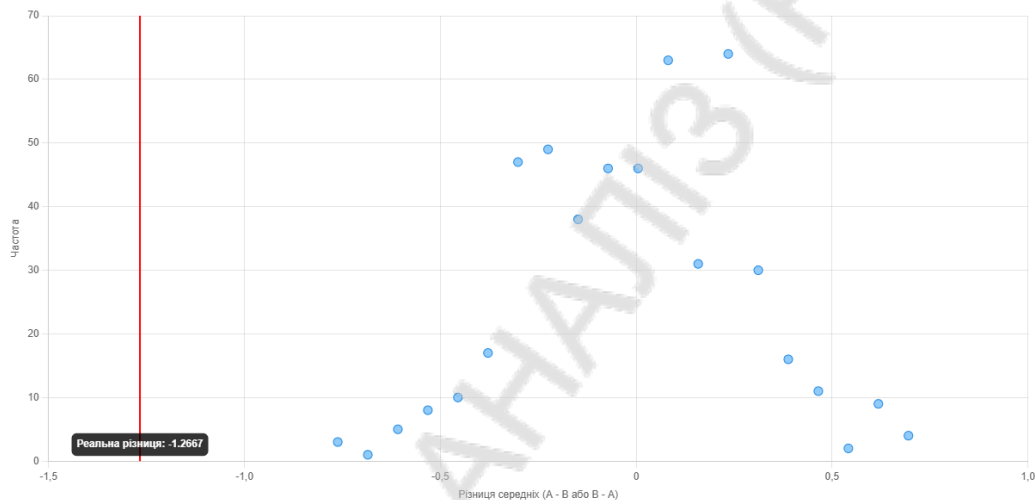


Рисунок 2.26 - Різницю середніх(А-В) діаграма розкидання.

Також додатково було розроблено діаграму стовпчасту різниці середніх(А-В) на мові python, зображено на рисунку 2.27:

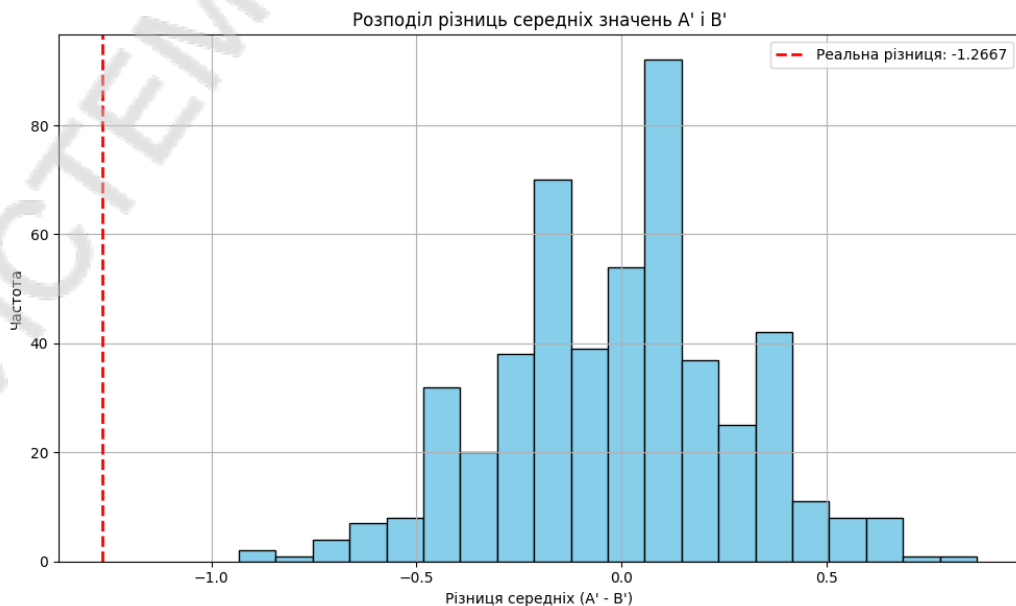


Рисунок 2.27 - Різницю середніх(А-В) діаграму стовпчаста.

Отже, результати тесту A/B говорять про статистичну значимість варіанту В у порівнянні з варіантом А. Оцінка варіанта В становить 4,23 балів, тоді як інший йде далеко позаду з 2,97 бали . Різниця між ними становить - 1.26667 бали. Допоміжний аналіз підтверджує, що різниця статистично значуща ($p < 0,05$), тому нульову гіпотезу в цілому можна відхилити й вважати обґрунтованим варіант В, для використання в основному продакшені.

ВИСНОВОК

У кваліфікаційній роботі було виконано повний цикл створення веб-додатку для квіткового онлайн-магазину, зосередившись на оптимізації інтерфейсу кошика. Головною метою дослідження було покращення зручності, інтуїтивності та ефективності взаємодії з користувачем за допомогою сучасних методів тестування інтерфейсних рішень.

Теоретична частина включає аналіз сучасних принципів UX/UI-дизайну, таких як адаптивність, компонентний підхід та використання аналітики для прийняття дизайн-рішень. Особливу увагу приділено A/B-тестуванню, зокрема bootstrap-методу — статистичному інструменту для оцінки достовірності відмінностей між варіантами інтерфейсів на основі реальних даних.

Практична частина передбачала розробку веб-додатку на Angular з використанням Firebase хмарної бази даних Firestore. Для кошика було створено два дизайн-варіанти, які випадковим чином показувалися різним користувачам. Зібрані дані (оцінки користувачів) були проаналізовані за допомогою bootstrap-методу, що дозволило визначити статистично значущу перевагу одного з варіантів.

Результати показали, що інтерфейс з компактним відображенням товарів та швидкими діями (варіант В) має вищу ефективність, особливо за показником конверсії. Bootstrap-аналіз підтвердив статистичну значимість цих результатів, що дозволяє рекомендувати цей варіант для впровадження.

Таким чином, у роботі доведено ефективність використання A/B-тестування з bootstrap-аналізом у розробці інтерфейсів. Створене рішення не лише відповідає сучасним стандартам електронної комерції, але й закладає основу для подальшого вдосконалення, включаючи персоналізацію взаємодії та глибший аналіз поведінки користувачів.

Отримані результати підтверджують, що навіть незначні зміни в інтерфейсі можуть суттєво впливати на конверсію та загальний досвід

користувача. Це свідчить про важливість постійного тестування та ітеративного вдосконалення дизайну. Система, розроблена в межах дослідження, демонструє гнучкість і готовність до масштабування, що робить її придатною для впровадження як повноцінного комерційного рішення. Крім того, використаний підхід може бути адаптований для оптимізації інших елементів інтерфейсу, що відкриває перспективи для подальших досліджень у сфері UX/UI.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А/В-тестування: що це таке та чому вам варто його використовувати <https://hostiq.ua/blog/ukr/ab-testing/> (дата звернення: 28.04.2025).
2. А/В тестування <https://esputnik.com/uk/slovnyk-email-marketologa/a-b-testuvannya> (дата звернення: 28.04.2025).
3. Хабарлак К.С. Аналіз та обробка великих даних [Електронний ресурс]: конспект лекцій для здобувачів ступеня магістра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / К.С. Хабарлак, Т.В. Хом'як ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 111 с. Доступ: <https://ir.nmu.org.ua/handle/123456789/168662> (дата звернення: 01.05.2025).
4. Хабарлак К.С. Аналіз та обробка великих даних [Електронний ресурс] : методичні рекомендації до виконання практичних робіт для здобувачів ступеня магістра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 82 с. Доступ: <https://ir.nmu.org.ua/handle/123456789/167968> (дата звернення: 02.05.2025).
5. Що таке А/В-тестування і як його провести <https://goit.global/ua/articles/shcho-take-a-b-testuvannia-i-iak-yoho-provesty/> (дата звернення: 28.05.2025).
6. Перевірено на людях, або Як і для чого починати А/В-тестування <https://dou.ua/lenta/articles/what-is-a-b-testing/> (дата звернення: 29.05.2025).
7. А/В тестування: як провести та проаналізувати? <https://speka.media/ab-testuvannya-yak-provesti-ta-proanalizuvati-9dkkev> (дата звернення: 29.05.2025).
8. Метод диференційованого порівняння (DiD) https://en.wikipedia.org/wiki/Difference_in_differences (дата звернення: 30.05.2025).

9. 15+ визначних прикладів інтернет-магазинів <https://impulse-design.com.ua/ua/15-vydayushchikhsya-primerov-internet-magazinov.html> (дата звернення: 01.06.2025).

10. Семантична структура для HTML5 сторінки <https://www.khtulhu.org.ua/veb-programmirovanie/347-semanticheskaya-struktura-dlya-html5-stranitsy.html> (дата звернення: 01.06.2025).

11. CSS Flexbox layout — технології для розташування блоків на HTML-сторінці <https://dou.ua/lenta/articles/css-flexbox-layout/> (дата звернення: 02.06.2025).

12. Основи Flexbox <https://marina-ferreira.github.io/tutorials/css/flexbox/> (дата звернення: 02.06.2025).

13. Resampling Methods: The Bootstrap https://www2.stat.duke.edu/~rcs46/lectures_2017/05-resample/05-boot.pdf (дата звернення: 01.05.2025).

14. Кваліфікаційна робота бакалавра [Електронний ресурс] : методичні рекомендації для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / уклад.: Т.А. Желдак, Т.В. Хом'як, А.В. Малієнко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 32 с.

Відомість матеріалів кваліфікаційної роботи

| № з/п | Позначення | | | | Найменування | Кількість | Примітки | | |
|-----------|-----------------|----------------|--------|------|---|---------------------------|---------------------|---------|--|
| | | | | | | аркушів | | | |
| 1 | | | | | | | | | |
| 2 | | | | | Документація | | | | |
| 3 | | | | | | | | | |
| 4 | САУ.КР.25.26.ПЗ | | | | Пояснювальна записка | 78 | Формат А4 | | |
| 5 | | | | | | | | | |
| 6 | | | | | Демонстраційний матеріал | 13 | Презентація на CD-R | | |
| 7 | | | | | | | | | |
| 8 | | | | | Копія роботи | 1 | Диск CD-R | | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |
| 11 | | | | | | | | | |
| 12 | | | | | | | | | |
| 13 | | | | | | | | | |
| 14 | | | | | | | | | |
| 15 | | | | | | | | | |
| 16 | | | | | | | | | |
| 17 | | | | | | | | | |
| 18 | | | | | | | | | |
| | | | | | САУ.КР.25.26.ДА.ПЗ. | | | | |
| Змін. | Аркуш | № докум. | Підпис | Дата | | | | | |
| Розроб. | | Казарінов О.В. | | | Матеріали кваліфікаційної роботи | Літ. | Аркуш | Аркушів | |
| К. розд. | | Хабарлак К.С. | | | | | | | |
| Керівн. | | Хабарлак К.С. | | | | НТУ «ДП», 12; 124-21-2 | | | |
| Н.контр. | | Хом'як Т.В. | | | | | | | |
| Зав. каф. | | Желдак Т.А. | | | | | | | |

Відгук

на кваліфікаційну роботу бакалавра
студента групи 124 – 21 – 2 Казарінова Олександра Валерійовича
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: «Аналіз та покращення інтерфейсу продуктової корзини із використанням A/B тестування»

Обсяг кваліфікаційної роботи 78 стор.

Мета кваліфікаційної роботи: розробка та вдосконалення інтерфейсу перегляду обраних товарів у кошику на веб-сайті за допомогою A/B тестування з метою підвищення зручності, ефективності та задоволення користувачів під час процесу покупок. Розроблений інструментарій для веб-сайтів на Angular дозволяє на ранніх етапах експериментально виявляти перспективні напрямки змін інтерфейсу, таким чином покращуючи користувацький досвід, скорочуючи час на розробку та підвищуючи прибуток.

Актуальність теми: сучасні веб-сайти постійно вдосконалюються, в той же час зміни в інтерфейсі не завжди покращують користувацький досвід цільової аудиторії, а отже виникає потреба в розробці компонентів для автоматизованого проведення A/B тестування веб-сайтів для раннього виявлення перспективних ідей для подальшої розробки на сайті.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки 1) використано сучасний підхід A/B тестування для виявлення компонента веб-сайту для покращення, за допомогою bootstrap підходу доведено статистичну значущість отриманих результатів; 2) проведено аналіз сучасної літератури, відмічено переваги та недоліки існуючих методів; 3) самостійно розроблено веб-сайт магазину квітів, інструментарій для A/B тестування, доверення статистичної значущості A/B тесту.

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра. Оригінальність наукових рішень полягає в 1) в розробці інструментарію для проведення A/B тестування, доведення статистичної значущості та аналізу користувацьких відгуків; 2) розробці нового програмного рішення, що складається з веб-сайту на Angular та хмарної БД Firebase.

Практичне значення результатів кваліфікаційної роботи полягає в експериментальному виявленню елементів користувацького інтерфейсу для покращення, що дозволяє полішити користувацький досвід, таким чином скорочуючи час на розробку та підвищуючи прибуток компанії.

Висновки підтверджують можливість використання результатів роботи в вітчизняних та закордонних Інтернет-магазинах для аналізу відгуків користувачів, оцінки покращення користувацького досвіду нових версій інтерфейсу із використанням А/В тесту.

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі.

У роботі відзначено такі недоліки:

1. Доцільно було б більш ґрунтовно описати розроблений інструментарій для А/В тестування.

Кваліфікаційна робота в цілому заслуговує оцінки: 90/відмінно.

З урахуванням висловлених зауважень авторка заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Керівник кваліфікаційної роботи бакалавра,
доктор філософії, доцент каф. САУ _____ / Хабарлак К.С.