

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

Факультет інформаційних технологій
(факультет)
Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

Здобувач вищої освіти Бублик Кирило Олегович
академічної групи 124-21-2
спеціальності 124 Системний аналіз
за освітньо-професійною програмою Системний аналіз
на тему: «Аналіз та покращення інтерфейсу вибору товарів із використанням множинного 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			

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 Бублику К. О. академічної групи 124-21-2
спеціальності: 124 Системний аналіз
за освітньо-професійною програмою Системний аналіз
на тему «Аналіз та покращення інтерфейсу вибору товарів із використанням
множинного 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) здійснити програмну реалізацію збору та аналізу даних для сайту, впровадити один з підходів множинного тестування; 3) провести аналіз взаємодії користувачів із сайтом. На основі результатів множинного тестування зробити висновок щодо можливості покращення користувацького інтерфейсу сайту.	29.01.2025- 31.05.2025

Завдання видано _____ доц. Хабарлак К.С.
(підпис) (прізвище, ініціали)

Дата видачі: 09.09.2024 р.

Дата подання до екзаменаційної комісії: 09.06.2025 р.

Прийнято до виконання _____ Бублик К. О.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 51 с., 27 рис., 6 табл., 2 додатки, 18 джерел

Об'єктом дослідження в кваліфікаційній роботі є процес аналізу та розробка веб-інтерфейсу для вибору товарів на веб-сайті, онлайн-магазину продуктів із використанням множинного тестування.

Предметом дослідження є методи та інструменти покращення користувацького досвіду під час вибору товарів на веб-сайті, застосування множинного тестування для оцінки результативності інтерфейсу.

Метою кваліфікаційної роботи – розробка та покращення інтерфейсу вибору товарів на веб-сайті за допомогою множинного тестування для покращення зручності користування та ефективності покупок.

Методи дослідження: У даній роботі використано методи аналізу інтерфейсу користувача та принципи побудови, використання множинного тестування для порівняння різних варіантів інтерфейсу, оцінка результатів тестування з точки ефективності взаємодії користувача з веб-сайтом та статистичний аналіз результатів для вибору найкращого макету.

В *інформаційно-аналітичному розділі* наведено аналіз підходів до створення інтерфейсів веб-магазину та методи тестування, що використано для покращення дій користувача. Описано різні методи тестування, зокрема A/B-тестування. Поставлені задачі дослідження та концепції їх розв'язку.

У *спеціальному розділі* розроблена реалізація множинного тестування на базі веб-сайту, створеного за допомогою язику програмування «ANGULAR» та наведено порівняння різних варіантів інтерфейсу вибору товарів.

Практична цінність отриманих результатів полягає в тому, що застосування множинного тестування дозволяє отримати дійсні дані про переваги різних інтерфейсів, що підвищує зручність використання веб-сайтом та сприяє покращеним продажам товарів.

Ключові слова: ІНТЕРФЕЙС, ВЕБ-САЙТ, МНОЖИННЕ ТЕСТУВАННЯ, A/B-ТЕСТУВАННЯ, ANGULAR, ОПТИМІЗАЦІЯ.

ABSTRACT

Explanatory note: 51 p., 27 fig., 6 tables, 2 appendices, 18 jer.

The object of research in the qualification work is the process of analysis and development of a web interface for selecting products on a website, an online grocery store using multiple testing.

The subject of the research is methods and tools for improving the user experience when selecting products on a website, the use of multiple testing to evaluate the effectiveness of the interface.

The purpose of the qualification work is to develop and improve the product selection interface on the website through multiple testing to enhance user convenience and shopping efficiency.

Research methods: This paper uses user interface analysis methods and design principles, the use of multiple testing to compare different interface options, evaluation of test results in terms of the effectiveness of user interaction with the website, and statistical analysis of the results to select the best layout.

The information and analytical section provides an analysis of approaches to creating web store interfaces and testing methods used to improve user experience. Various testing methods, including A/B testing, are described. The research objectives and concepts of their solution are presented.

In a **special section**, the implementation of multiple testing on the basis of a website created using the ANGULAR programming language is developed and a comparison of different variants of the product selection interface is presented.

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

Keywords: INTERFACE, WEBSITE, MULTIVARIATE TESTING, A/B TESTING, ANGULAR, OPTIMIZATION.

ЗМІСТ

ВСТУП	6
ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ	7
1.1 Постановка задачі	7
1.2 Багаторукий бандит і множинне тестування	8
1.2.1. Стратегії відбору варіантів для задачі багаторукого бандита..	10
1.3 Підходи до дизайну веб-сайтів.....	11
1.4 Створення зовнішнього вигляду веб-додатку.....	12
1.4.1 Базова структура сторінки	12
1.4.2 Технологія Flexbox (розташування елементів).....	13
1.4.3 Фреймворк Angular.....	14
1.4.4 Інтерфейс командного рядка Angular CLI.....	15
1.4.5 Технологія Node.js у веб-розробці	16
1.4.6 Мова програмування TypeScript	17
СПЕЦІАЛЬНИЙ РОЗДІЛ.....	19
2.1 Розробка бази даних товарів	19
2.2 Аналіз інтерфейсу веб-сайтів.....	23
2.3 Розробка веб-додатку	26
2.4 Реалізація множинного тестування інтерфейсу.....	31
2.5 Проведення множинного тестування. Аналіз результатів.....	35
2.5.1 Підключення Firebase до Angular	42
2.5.2 Результат тестування.....	42
ВИСНОВОК.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
Додаток А.....	52
Додаток В.....	53

ВСТУП

У вік цифрових технологій інтернет-магазини стають частиною кожної людини, а зручність і швидкість вибору товарів є важливим фактором користувача та підвищення продажів компаній. У зв'язку з цим зростає актуальність розробки веб-інтерфейсів, які дозволяють швидко та інтуїтивно знаходити необхідний товар. Важливим інструментом для досягнення необхідного результату є проведення множинного тестування, яке дає змогу оцінити ефективність різних варіантів інтерфейсу.

Ця кваліфікаційна робота присвячена створенню веб-сайту онлайн-магазину продуктів із використанням сучасного фреймворку «ANGULAR» та підходів множинного тестування для покращення інтерфейсу вибору і пошуку товарів. У рамках роботи буде розглянуто такі питання, як:

- Аналіз підходів при побудові інтерфейсу веб-магазину;
- Демонстрація використання множинного тестування для порівняння інтерфейсів;
- Опис архітектури веб-сайту;
- Механізм тестування та оцінка результатів.

Актуальність даної теми полягає в тому, що якісний інтерфейс підвищує ефективність взаємодії користувача з веб-сайтом. Застосування множинного тестування дозволяє знайти найефективніші блоки інтерфейсу, для впровадження їх до головної сторінки онлайн-магазину.

Практична цінність кваліфікаційної роботи полягає в тому, що реалізований веб-сайт може бути використаний як основа для інших розробок веб-додачків, а застосовані методи тестування – для покращення будь-якого веб-інтерфейсу.

Очікується, що результати дослідження підвищать ефективність взаємодії користувача із сайтом та дозволять створити більш зручні веб-рішення для онлайн-торгівель.

ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Постановка задачі

У наш час онлайн-торгівля продуктами стрімко розвивається в гору. Зі збільшенням кількості користувачів, які використовують веб-магазини для здійснення покупок, виникає потреба у покращенні веб-інтерфейсів, які дозволять ефективно знаходити необхідний товар. Зручність та швидкість взаємодії людей з сайтом впливає на рішення покупки товарів, тому якісний інтерфейс є важливим для успішного онлайн-магазину.

З огляду на це, для даної кваліфікаційної роботи ставиться задача розробки веб-додатку онлайн-магазину з використанням сучасних технологій фреймворку «ANGULAR». Основну увагу приділено аналізу та покращенню інтерфейсу вибору товарів на основі множинного тестування (алгоритм жадібних бандитів), що дозволить оцінити ефективність варіантів відображення та сортування товарів.

Спочатку потрібно провести огляд існуючих інтерфейсів інтернет магазинів. Далі – визначити ключові фактори ефективності даного сайту та реалізувати модулі множинного тестування.

Потім слід розробити архітектуру веб-додатку, а саме створити інтерфейс для взаємодії користувача, створити змогу слідкування ефективності та діями людей на даному веб-сайті.

Створити свою базу даних товарів, для подальшого використання, а саме фотографії продукту з відомим цінником.

Реалізувати динамічне використання сайтом, щоб користувач міг взаємодіяти з ним.

Метою кваліфікаційної роботи є розробка та покращення інтерфейсу вибору товарів на веб-сайті за допомогою множинного тестування для підвищення зручності користування та ефективності покупок.

Також було проаналізовано декілька схожих веб-сайтів, що працюють за схожим принципом продажу товарів у форматі онлайн-магазинів. Серед них

особливу увагу приділено таким відомим бізнес корпорацій, такі як: «АТБ», «VARUS», «СІЛЬПО», «ROZETKA», «BRAIN», «TELEMART.UA», «COMFY». Ці веб-додатки використовують адаптивні веб-інтерфейси, а також елементи аналітики поведінки користувача, щоб покращити взаємодію та продажі товарів на сайті.

На основі досвіду був розроблений власний веб-додаток на фреймворку Angular, який забезпечує покращену систему вибору товарів за допомогою множинного тестування. Реалізовано ϵ -жадібний алгоритм багаторукого бандита, що дозволить автоматично визначати найефективніші варіанти інтерфейсу.

Однією з ключових переваг розробленого сайту є можливість динамічного відображення різних варіантів інтерфейсу (наприклад, зміна палітри кольорів, кнопок, структури каталогу), які автоматично аналізуються на основі поведінки користувачів. Це дозволяє системі в режимі реального часу обирати той варіант що дає найкращі результати (найбільше кліків, переглядів чи замовлень).

Крім того, було створено адаптивну структуру веб-сайту, яка поєднує зручність та швидкість роботи на основі інтуїтивно зрозумілим інтерфейсом. Такий підхід дозволить покращити користувацький досвід, а саме кількість дій, що ведуть до покупки.

Завдяки впровадженню сучасних підходів дизайну та використанню адаптивних алгоритмів тестування, веб-додаток має потенціал бути ефективним для комерційного використання у сфері онлайн-торгівельних магазинів.

1.2 Багаторукий бандит і множинне тестування

Задача багаторукого бандита – це ситуація, при якій потрібно правильно розподілити обмежені ресурси між кількома варіантами [1,2]. Головна мета полягає в тому, що нам потрібно отримати якомога більше вигоди, навіть якщо ми не впевнені, який із варіантів є дійсно найкращим. З часом ми будемо дізнаватися все більше, пробуючи різні варіанти та вчимося приймати ефективне

рішення. У цьому полягає – що треба поєднувати експлуатацію вже відомих вигідних варіантів із системою нових, потенційно кращих, але не перевірених рішень.

Наприклад:

Людина приходить пограти в гральні автомати – він перед собою має декілька варіантів (їх називають «однорукі бандити»). Гравець не знає, який із автоматів дає кращий виграш, тому він змушений експериментувати: спробувати кілька різних а гральних автоматів, щоб зрозуміти, який із них вигідніший та вже потім грати на ньому частіше. Але є ще шанс виграти більше, на іншому – бо є ймовірність кращого результату. Це і є головна дилема цієї задачі. Грати де стабільність, чи ризикнути і пробувати нове.

Ця задача добре описує проблему багаторукого бандита, вона моделює агента, який одночасно намагається здобути нові знання (називається «розвідкою») та оптимізувати свої рішення на основі існуючих знань (називається «експлуатацією»). Агент намагається стабілізувати ці конкуруючі задачі, щоб максимізувати загальну принесену ними користь за розглянутий проміжок часу. Існує багато практичних застосувань моделі жадібного бандита, наприклад:

- створення правильного інтерфейсу сайтів;
- створення фінансового портфеля;
- зусилля створення оптимального руху по світлофорам для мінімізації затримок транспорту.

Є багато стратегій для розв'язання цієї задачі. Найпростіші з них – це напіввипромірні стратегії, які поєднують жадібний підхід (вибір найкращого варіанта на основі наявних даних) з випадковими експериментами. Не зважаючи на простоту, вони досить ефективно вирішують проблему збалансування між розвідкою і експлуатацією.

1.2.1. Стратегії відбору варіантів для задачі багаторукого бандита

Найпростішим підходом до розв'язання задачі багаторукого бандита є напіврівномірні стратегії. Вони полягають на простому принципі найкращого варіанта на основі попереднього досвіду, але інколи ми починаємо використовувати інші, щоб не втратити кращу можливість.

- Епсилон-жадібна стратегія: реалізує цей підхід до вирішення задачі через ймовірність. З великою ймовірністю обирається кращий варіант, а з малої – випадковий. Наприклад, при $\epsilon = 0.1$ – 90% рішень на досвіді, а 10% – це дослідження.
- Епсилон-перша стратегія: ділить процес на дві частини. Спочатку повна розвідка, а вже потім – лише використання кращого знайденого варіанту.
- Епсилон-зменшувана стратегія: починаємо з активного дослідження, але з часом все більше фокусується на експлуатації, поступово зменшується ϵ .

Всі методи вирішують одну проблему – як знайти збалансований метод вивчення нових можливостей та використанням уже відомих вигідних варіантів.

Задача багаторукого бандита є універсальною моделлю, що допомагає приймати аргументовані рішення в умовах невизначеності. Само тому вона дуже часто використовується в оптимізації інтерфейсів, машинному навчанні та в інших сферах, де важливо швидко адаптуватися до навколишніх змін та вдосконалювати стратегії дій.

Множинне тестування – це метод, який дозволяє одночасно перевіряти кілька варіантів змін інтерфейсу, щоб визначити які з них найбільш ефективні. На відміну від A/B тестування, яке порівнює лише два варіанти, цей підхід оцінює кілька комбінацій елементів.

Алгоритм багаторукого бандита – це стратегія оптимізації, яка дозволяє адаптивно визначати, який варіант інтерфейсу краще. Замість того, щоб

рівномірно розподіляти трафік, як у A/B тестах, він поступово зменшує показ менш ефективних варіантів і фокусується на найуспішніших. Один із простих варіантів – ϵ -жадібний алгоритм, де з ймовірністю $1-\epsilon$ – найефективніший на даний момент.

1.3 Підходи до дизайну веб-сайтів

У розробці веб-сайтів велике значення має правильна організація інтерфейсу користувача. Ефективний веб-інтерфейс забезпечуватиме зручність використання, підвищує лояльність користувачів. До основних підходів відносяться такі принципи [3]:

- мінімалізм – підхід який орієнтований на простоту та чистоту інтерфейсу. Зайві елементи прибираються, щоб не відволікати очі користувача. Акцент цього підходу робиться на функціональність, великі читабельні шрифти, логічне розміщення елементів, а також візуальну ієрархію. Мінімалізм сприяє швидкому прийняттю рішень користувачем і зменшує навантаження на увагу.
- мобільна адаптивність – підхід який повинен мати однакове працювати на комп'ютері, смартфоні і планшеті. Це забезпечує адаптивне зображення. У зв'язку з тим, що половину користувачів заходять на сайти з мобільних пристроїв, адаптивність є стандартом в цій сфері.
- UX-дослідження (User Experience) – підхід який базується на вивченні поведінки користувачів. Збираються аналітичні дані про те, як люди взаємодіють із веб-додатком: які сторінки найпопулярніші, скільки часу витрачають, де найчастіше натискають. На основі цих даних оптимізується інтерфейс: змінюється розміщення кнопок, палітра кольорів, шрифти, логіка навігації. UX-дослідження – це основа для прийняття рішень у веб-дизайні.
- Компонентний підхід – сучасні фреймворки, такі як Angular, базуються на компонентній архітектурі. Це означає, що інтерфейс розбивається на

окремі блоки – компоненти, кожен із яких має свою логіку, шаблон та стиль. Наприклад картка товару, кнопка «Додати до кошику» чи панель фільтрів – це все компоненти. Такий підхід полегшує розробку, тестування та масштабування веб-сайтів, дозволяє швидко змінювати окремі частини інтерфейсу без впливу на всю систему.

1.4 Створення зовнішнього вигляду веб-додатку

Верстка сайту – це процес створення зовнішнього вигляду веб-додатку.

Головна мета якої – зробити візуальну частину зручною, гарною та адаптивною, для подальшого покращення.

Щоб збудувати сайт, нам потрібно використовувати HTML для створення структури – це ніби дім. А також CSS, SCSS або SASS, щоб забезпечити гарне сприйняття кожного блоку веб-додатку, а також технологію Flexbox, для розташування елементів, як нам потрібно.

1.4.1 Базова структура сторінки

На кожному сайті є своя чітка структура, вона завжди складається з трьох або більше частин (рис. 1.1):

- header (верхівка сайту) – це верхня частина. Зазвичай розміщуються логотип, назва, меню для навігації або кнопки особистого кабінету (navigation),
- main (основна частина сайту) – це центральна частина. Тут розміщена вся основна інформація: товари, категорії або якісь данні (menu, main content).
- footer (нижня частина сайту) – це низ сторінки. Додається додаткова інформація: посилання, контакти та політика конфіденційності.

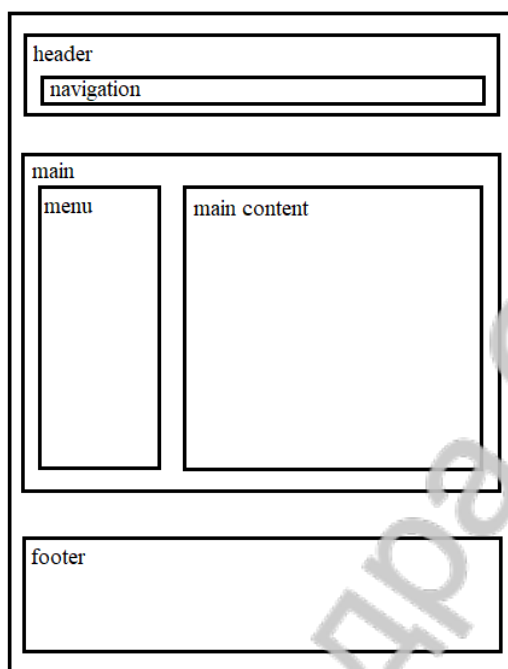


Рисунок 1.1 – Базова структура блокової системи написання сайту.

1.4.2 Технологія Flexbox (розташування елементів)

Щоб всі елементи розташовувалися зручно й були адаптивними, ми використовуємо CSS Flexbox. Це технологія, яка дозволяє вирівнювати блоки, як нам завгодно (по горизонталі або по вертикалі). Головною перевагою цього є, те що створює адаптивний вміст, який сам адаптується під різні розміри екранів (рис. 1.2).

Приклад:

- Якщо ми хочемо зробити горизонтальне меню в шапці (header), то застосовуємо код `display: flex`, а також `justify-content: space-between`. Він дозволить рівномірно розмістити вміст. Наприклад потрібно розмістити логотип зліва, меню пошуку посередині та кнопки меню праворуч. Це дозволяє нам зробити це дуже швидко та зручно;
- У головній секції (main) ми можемо відображати товари як нам захочеться. Зробити гнучку сітку. Flexbox автоматично перебудує, блок коду, залежно від ширини екрану.

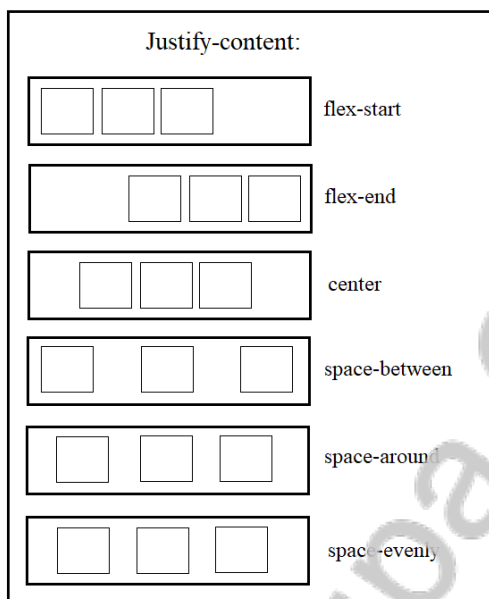


Рисунок 1.2 – Методи розміщення блоків на веб-сайті.

1.4.3 Фреймворк Angular

Angular – це платформа для розробки, яка побудована на TypeScript [4].

Платформа, Angular включає в собі:

- компонентний фреймворк для створення масштабованих веб-застосунків;
- колекція добре інтегрованих бібліотек, що охоплюють широкий спектр функцій, включаючи маршрутизацію, керування формами, зв'язок клієнт-сервер тощо;
- набір інструментів для розробників, які допоможуть вам розробляти, створювати, тестувати та оновлювати ваш код.

З Angular ви користуєтесь перевагами платформи, яка може масштабуватися від проектів з одним розробником до додатків корпоративного рівня. Найкраще те, що екосистема Angular складається з різноманітної групи з понад 1,7 мільйона розробників, авторів бібліотек та творців контенту.

Інструменти Angular

- TypeScript – постачається за замовчуванням з кожним додатком Angular, щоб забезпечити покращені інструменти та зручність обслуговування для кращого досвіду розробника.
- Command Line Interface (CLI) – Angular використовує компілятор, щоб абстрагувати складність інструментів та оптимізувати ваш код, щоб ви могли зосередитися на створенні своєї програми.

Компоненти Angular

Компоненти є будівельним блоком для створення додатків в Angular.

Використовуючи компонентну архітектуру, він прагне забезпечити структуру для організації вашого проекту в керовані, добре організовані частини з чіткими обов'язками, щоб ваш код був зручним у підтримці та масштабованим.

Кожен компонент має в собі чотири файли, а саме:

- `yourname.component.html` – html макет (шаблон компоненту)
- `yourname.component.ts` – typescript (логіка компоненту)
- `yourname.component.css` – css (стилі компонента)
- `yourname.component.spec.ts` – unit-test (тестування функціоналу компонента)

Така структура є дуже зручною, бо дозволяє чітко розділити відповідальності. HTML для структури, TS для логіки, CSS для візуалізації та Spec.ts для тестування. Це робить код більше ієрархічним.

1.4.4 Інтерфейс командного рядка Angular CLI

Angular CLI – це інструмент інтерфейсу командного рядка, який дозволяє створювати, розробляти, тестувати, розгортати та підтримувати Angular-додатки безпосередньо щ командної оболонки [5].

Команди для використання Angular CLI

Angular CLI – це рекомендований спосіб розробки Angular-застосунків, який може зробити деякі завдання безпроблемними.

Приклади поширених команд Angular CLI включають:

- `ng build` – компілює Angular-застосунок у вихідний каталог.
- `ng serve` – збирає та обслуговує вашу програму, перебудовуючи її після змін у файлах.
- `ng generate` – генерує або змінює файли на основі схеми.
- `ng test` – запускає модульні тести для заданого проекту.

У даній кваліфікаційній роботі будуть часто використовуватись такі команди як: `ng generate`, для того щоб швидко створити новий компонент, який автоматично підключиться до нашого проекту та `ng serve`, запускає локальний сервер де будується весь наш додаток.

1.4.5 Технологія Node.js у веб-розробці

Node.js – це середовище виконання JavaScript з відкритим кодом. Популярний інструмент майже для будь-якого проекту [6].

Node.js запускає рушій JavaScript V8, ядро Google Chrome, поза браузером. Це дозволяє Node.js бути дуже продуктивним.

Додаток Node.js виконується в одному процесі, не створюючи нового потоку для кожного запиту. Node.js надає набір асинхронних примітивів вводу/виводу у своїй стандартній бібліотеці, які запобігають блокуванню коду JavaScript, і загалом бібліотеки в Node.js написані з використанням неблокуючих парадигм, що робить блокувальну поведінку винятком, а не нормою.

Переваги Node.js

1. Єдиний JavaScript-стек: фронтенд і бекенд на одній мові (JavaScript) – це спрощує розробку та зменшує витрати часу на навчання;
2. Висока продуктивність: асинхронна архітектура дозволяє ефективно обробляти дуже великий потік інформації одночасно.

Node.js має унікальну перевагу, оскільки мільйони фронтенд-розробників, які пишуть JavaScript для браузера, тепер можуть писати серверний код на додаток до клієнтського коду без необхідності вивчати зовсім іншу мову.

Простими словами, Node.js – це швидкий, ефективний та зручний інструмент, для розробки вашого веб-додатку.

1.4.6 Мова програмування TypeScript

TypeScript – це варіативність JavaScript, одна із сучасних мов програмування, яка дозволяє створювати великі та масштабні веб-додатки завдяки типізації та покращеній інструментальним технологіям [7].

Щоб зрозуміти детальніше, що таке typescript треба спочатку розібратися з javascript.

Мова програмування JavaScript

JavaScript було створено для того, щоб створити функціонал веб-сайту, наприклад покупку товару, виконати вхід в аккаунт, опублікувати будь яку фотографію у соцмережі та інші схожі функції [8].

Програми цією мовою програмування називаються скриптами. Скрипти записуються та виконуються як простий текст. Для запуску їм не потрібна спеціальна підготовка чи компілятор.

Можливості JavaScript:

- Додавати новий HTML – код на сторінку;
- Реагувати на дії користувача;
- Надсилати запити мережею до серверів;

- Запам'ятовувати дані в (local storage – сховище інформації в браузері).

А отже, в роботі буде використано базу даних Firebase, для побудови інтерфейсу використано фреймворк Angular, а для проведення множинного тестування – алгоритм ϵ -жадібного багаторукого бандита з метою покращення ефективності користувача та вибору товарів.

НТУ "ДП" кафедра САУ

СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Розробка бази даних товарів

Для початку потрібно зібрати великий обсяг даних про товари, для того щоб почати робити інтерфейс веб-сайту. Включати в себе буде: фотографії, ціни та дані о продукції. Для цього збору інформації було використано фотографії з інтернету, ціни та дані продукції взяті з інтернет магазинів – «АТБ», «VARUS», «СІЛЬПО».

Було зібрано 28 іконок, 242 фотографій та 131 інформацій о товарі.

Процес знаходження інформації о товарах з інтернету представлено на (рис. 2.1, 2.2, 2.3).

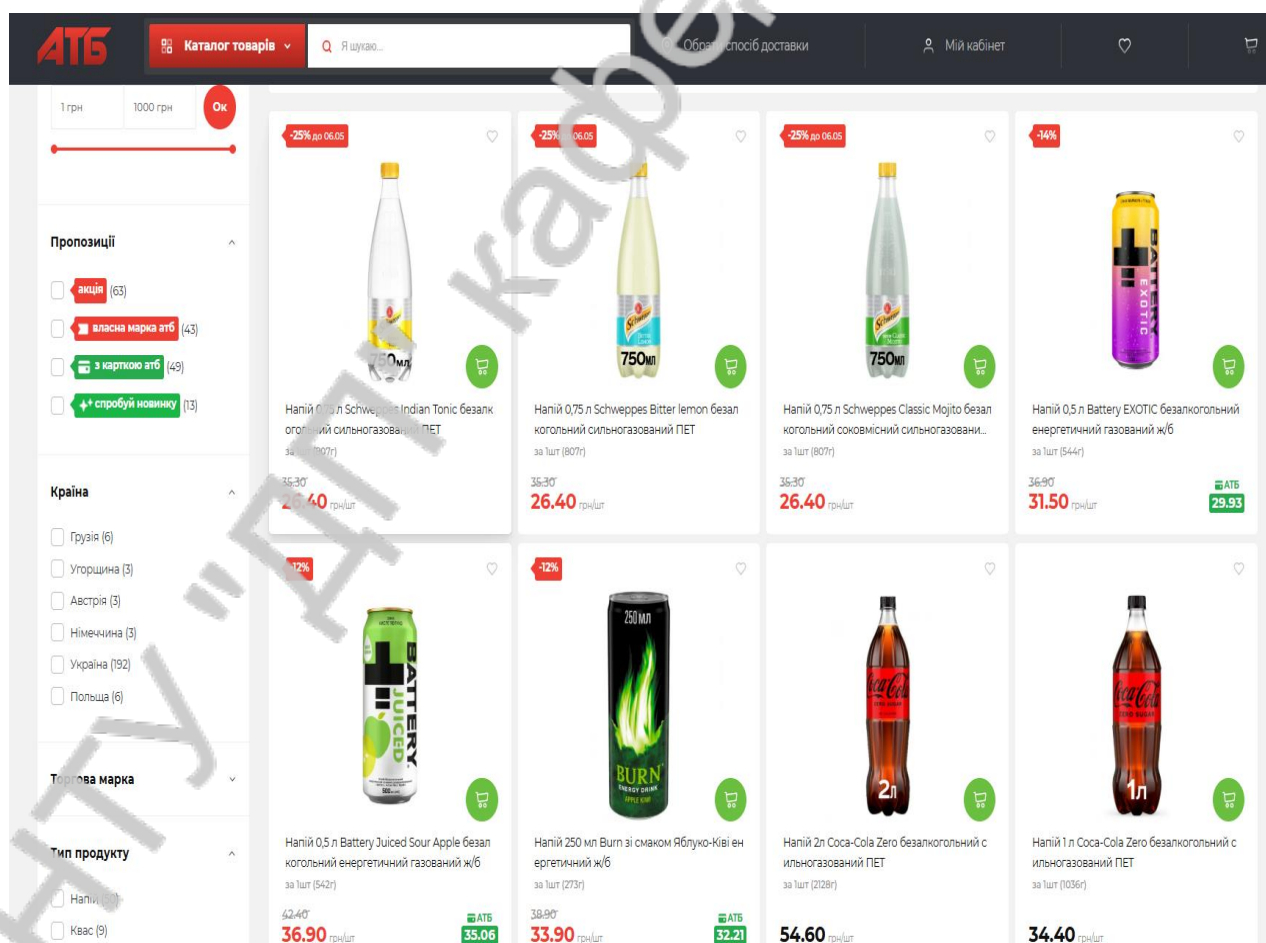


Рисунок 2.1 – Знаходження даних про товари «АТБ».

VARUS.ua Укр ▾ Клієнтам ▾ Партнерам ▾ Робота ▾ Адреса Вкажи свою адресу ▾ Вхід у кабінет

Каталог товарів Оптові закупівлі Подарункові сертифікати Акційна газета Акції Пошук товарів...

Головна – Гігієна та догляд

ГІГІЄНА ТА ДОГЛЯД

4360 товари

Отримати сьогодні Акційні товари 1107 За популярністю ▾

Всі товари

- Туалетний папір 118
- Серветки та рушники 229
- Догляд за волоссям 1000
- Догляд за тілом 1647
- Догляд за ротовою порожниною 419
- Товари для гоління 245
- Ватні засоби 42

Ціна, грн

0 — 1471

Застосувати

Бренд

До акції

ВИГІДНИЙ ШОПІНГ, ЩО ЛЕТИТЬ ДО ТЕБЕ!

Перша ціна

ЧИСТА ВІГОДА! зелені 150 шт
за 1 шт (450 г)
28.90 грн

Зубна паста Lacalut Aktiv 75 мл
за 1 шт (75 мл)
129.90 грн

Мило Safeguard туалетне тверде Класичне біле 90г
за 1 шт (90 г)
37.00 грн

Прокладки гігієнічні Kotex Ultra Soft Нормал 10 шт.
за 1 шт (54 г)
55.50 **49.00 грн** (-12% до 12.05)

Рисунок 2.2 – Знаходження даних про товари «VARUS».

Сільпо Всі товари ▾ Я шукаю... Доставка Вкажіть адресу доставки Увійти

Акції Піца, суші та бургери Тільки онлайн Знижки від ТМ Знижки від ТМ Солодощі Саїточ Знижки від ТМ Ферма Новини від Моршинська

Вино з рейтингом Vivino Знижки на Wash&Free Щойно з'явилися Знижки на перелітку Знижки від ТМ Huggies Олео Nesquik

Весняна турбота від Henkel Zewa - відчуття турботи Знижки на Бібла Акція Водний Саїт NCC LOR зі знижками Для легких моментів

Знижки до 30% на португальські вина Фестиваль українського крауду Для здоров'я хвостиків Listerine та Johnson's baby Знижки від Sensodyne Parodontax

Головна > Акції > Ціноціники

Всі 458 Спеціальні пропозиції Фрукти, овочі 20 М'ясо 6 Риба 24 Ковбаси і м'ясні делікатеси 11 Сири 9 Хліб та випічка 9

Молочні продукти та яйця 20 Власні марки 40 Лавка Традицій 3 Здорове харчування 5 Бакалія і консерви 48 Соуси і спеції 10

Солодощі 53 Спеції та чипси 15 Кава, чай 30 Напої 10 Заморожена продукція 14 Алкоголь 21 Сигарети, стики, муйки 5

Для дому 46 Гігієна та краса 10 Дитячі товари 3 Для тварин 7

Фільтри За замовчуванням

Ціноціники

89.99 грн
119.00 грн **-24%**
Авокадо Ready to eat
шт/уп * 2.6

74.99 грн
117.00 грн **-36%**
Масло солодковершкове «Ферма» «Селянське»...
180г * 4.1

169.00 грн
269.00 грн **-45%**
Хамон Proxas Serrano с/в 9-11 місяців
160г * 1.1

99.00 грн
234.00 грн **-58%**
Сосиски Фарро Дитячі в/г
500г * 3.5

21.99 грн
34.89 грн **-37%**
Пиво Kronenbourg «Бланк» 3/6
1000г * 4.6

129.00 грн
279.00 грн **-54%**
Морозиво Рудь 100%
1000г * 4.3

Рисунок 2.3 – Знаходження даних про товари «СІЛПО».

Збір графічного матеріалу для товарів та категорій

З метою створення зручного та красивого інтерфейсу було здійснено збір графічних матеріалів, зокрема іконок для категорій товарів та зображень

продуктів. Іконки категорій дозволятимуть швидко орієнтуватися у навігації, а зображення товарів підвищуватимуть інформативність і покращуватимуть сприйняття користувача.

Після збору фотографій і інформації було створено файли, які містять в собі інформацію про кожний товар. Перша база даних включала в собі назву продукту, фото, ціну, валюту та якщо є акційна ціна (за наявності). Друга частина містить фотографії категорій.

Ці дані необхідні для розробки ефективного інтерфейсу. Файли з інформацією про товари представлені на (рис. 2.4, 2.5).

```

categories-bd.json x
2      "categories": [
342      {
343      "subcategories": [
344      { "title": "Наповнявач", "link": "/dairy-eggs/yogurt", "svg": "assets/imgs/category/tittler.webp" },
345      { "title": "Дивитись всі товари категорії", "link": "/dairy-eggs/cheese", "svg": "assets/imgs/category/animals.png" }
346      ]
347      },
348      {
349      "title": "Тютюнові вироби",
350      "link": "/tobacco",
351      "svg": "assets/icons/tobacco.svg",
352      "subcategories": [
353      { "title": "Сигарети", "link": "/dairy-eggs/milk", "svg": "assets/imgs/category/cigarettes.webp" },
354      { "title": "Стіки та електронні сигарети", "link": "/dairy-eggs/yogurt", "svg": "assets/imgs/category/sticks.webp" },
355      { "title": "Дивитись всі товари категорії", "link": "/dairy-eggs/cheese", "svg": "assets/imgs/category/tobacco.png" }
356      ]
357      },
358      {
359      "title": "Сертифікати та плажні картки",
360      "link": "/certificates",
361      "svg": "assets/icons/certificates.svg",
362      "subcategories": [
363      { "title": "Благодіяча ініціатива", "link": "/dairy-eggs/milk", "svg": "assets/imgs/category/piggybank.webp" },
364      { "title": "АТБ та АТБ", "link": "/dairy-eggs/yogurt", "svg": "assets/imgs/category/atband.webp" },
365      { "title": "Дивитись всі товари категорії", "link": "/dairy-eggs/cheese", "svg": "assets/imgs/category/certificates.png" }
366      ]
367      },
368      {
369      "title": "Канцелярські товари",
370      "link": "/stationery",
371      "svg": "assets/icons/stationery.svg",
372      "subcategories": [
373      { "title": "Допоміжне приладдя", "link": "/dairy-eggs/milk", "svg": "assets/imgs/category/scissors.png" },
374      { "title": "Зошити, альбоми, блокноти", "link": "/dairy-eggs/yogurt", "svg": "assets/imgs/category/notebook.png" },
375      { "title": "Папір і картон", "link": "/dairy-eggs/cheese", "svg": "assets/imgs/category/paper.png" },
376      { "title": "Папки, файли, обкладинки", "link": "/dairy-eggs/milk", "svg": "assets/imgs/category/folder.webp" },
377      { "title": "Ручки, олівці, фломастери", "link": "/dairy-eggs/yogurt", "svg": "assets/imgs/category/pen.png" },
378      { "title": "Фарби, пластилін, клей", "link": "/dairy-eggs/cheese", "svg": "assets/imgs/category/glue.webp" },
379      { "title": "Дивитись всі товари категорії", "link": "/dairy-eggs/cheese", "svg": "assets/imgs/category/office.png" }
380      ]
381      }
382      ]
383      }
384      ]

```

Рисунок 2.4 – База даних категорій для пошуку товару (Частина коду).

```

products-bd.json x
1  {
2    "catalogItemsDiscount": [
3      {
4        "label": "-24%",
5        "photo": "assets/imgs/products/Chicken.png",
6        "title": "Гомілка курчат-бройлерів Чебатурочка охолоджена",
7        "originalPrice": 100.99,
8        "discountPrice": 75.99,
9        "unit": "грн/л"
10     },
11    {
12      "label": "-34%",
13      "photo": "assets/imgs/products/Hell.png",
14      "title": "Напій 0,5 л Hell зі смаком кавуна та тутті-фрутті безалкогольний енергетичний газований ж/б",
15      "originalPrice": 46.50,
16      "discountPrice": 30.30,
17      "unit": "грн/шт"
18     },
19    {
20      "label": "-30%",
21      "photo": "assets/imgs/products/Romantic.png",
22      "title": "Шампунь 850 мл Romantic Professional регенеруючий",
23      "originalPrice": 149.90,
24      "discountPrice": 104.90,
25      "unit": "грн/шт"
26     },
27    {
28      "label": "-25%",
29      "photo": "assets/imgs/products/FrozenVegetables.png",
30      "title": "Суміш овочева 400г Своя Родина «Суп з грибами» швидкозаморожена",
31      "originalPrice": 53.50,
32      "discountPrice": 39.90,
33      "unit": "грн/шт"
34     },
35    {
36      "label": "-37%",
37      "photo": "assets/imgs/products/Seasoning.png",
38      "title": "Приправа 100г Приправка Exclusive до курки ",
39      "originalPrice": 44.90,
40      "discountPrice": 27.70,
41      "unit": "грн/шт"
42     }
43  ]
44  }

```

Рисунок 2.5 – База даних товарів (Частина коду).

Після збору всіх даних товарів, вийшло, 28 категорій для пошуку продукції, 15 товарів. Цього достатньо щоб проаналізувати методом множинного тестування та зробити ефективний інтерфейс інтернет магазину.

Можна побачити в таблиці 2.1.

Таблиця 2.1

Кількість зібраних даних

Тип	Кількість	Кількість зображень
Категорії	28	194
Банери	5	5
Іконки	28	28
Акціонні товари	5	5
Напої	5	5
Нові товари	5	5
Загалом	76	242

Після того, як дані було зібрано та оброблено у бази даних, наступним кроком є створення інтерфейсу веб-сайту. Для того щоб зробити ефективний макет веб-додатку, потрібно, проаналізувати крупні компанії які займаються продажом товарів, наприклад такі як: «АТБ», «VARUS», «СІЛЬПО», «ROZETKA», «BRAIN», «TELEMART.UA», «COMFY».

2.2 Аналіз інтерфейсу веб-сайтів

Першим етапом у процесі пошуку ефективного веб-інтерфейсу є вибір оптимального сайту товарів. Для пошуку макету в нас є багато варіантів крупних маркет-компаній, такі як: «АТБ», «VARUS», «СІЛЬПО», «ROZETKA», «BRAIN», «TELEMART.UA», «COMFY». Кожен з цих сайтів має унікальний інтерфейс та свою ефективність, але для нашої задачі ми хочемо вибрати найкращий веб-інтерфейс.

Ключовим критеріям для вибору макету є зручність, інтуїтивна зрозумілість для користувача, що забезпечуватиме легкість к доступу до функцій веб-сайту, швидке знаходження потрібного товару та мінімізувати кількість дій для здійснення покупки. Тому, перш ніж перейти до наступного етапу, ми повинні обрати ті веб-додатки, які будуть демонструвати найліпші результати.

Таким чином, обираючи між «АТБ», «VARUS», «СІЛЬПО», «ROZETKA», «BRAIN», «TELEMART.UA», «COMFY», проведемо аналіз кожного з них з точки зору зручності інтерфейсу, функцій пошуку товарів, логіки навігації та швидкість взаємодії, що впливають на ефективність покупки користувача.

1. «АТБ» [9]

Інтерфейс АТБ є простим та зрозумілим, з мінімалістичним дизайном. Доступний пошук товару за назвою, категоріями, а також фільтрація за ціною та акційні пропозиції. З іншого боку навігація не завжди зручна: деякі категорії заховані в меню, потрібно робити багато дій. Банер розміщений на всю сторінку, що дуже незручно.

2. «VARUS» [10]

VARUS має не інтуїтивний пошук товару, каталог товарів не виділяється, банер на весь екран, категорії зроблені незручно та не ефективно. Але зроблений акцент на акційних товарах.

3. «СІЛЬПО» [11]

Пошук товарів зроблений так же само як у АТБ, каталог товарів скритий, зроблений акцент на акції, банер мінімізованого розміру, відкрив сайт можна побачити відразу товари.

4. «ROZETKA» [12]

Є одним із лідерів електронного магазину. Її інтерфейс дуже зручний: банер не закриває весь екран, акційні товари можна відразу купити, пошук по тексту, категорії на головній сторінці, а його структура дозволить швидко знайти потрібну групу продуктів без зайвих кроків.

Макет ROZTKA є прикладом ефективного поєднання функціональності та зручності, що забезпечує ефективність та наявність нових користувачів.

5. «BRAIN» [13]

Спеціалізований технічний магазин. Інтерфейс зроблений логічно, зручне сортування за характеристиками та швидкий пошук товарів. Але візуальне оформлення старе.

6. «TELEMART.UA» [14]

TELEMART має компактний вигляд та функціонал. Каталог товарів ефективний, але не дуже зручний, банер на весь екран, товарів на головній сторінці немає, кнопка каталогу показує на навігацію сайту, не зрозуміло навіщо так робити, дизайн сайту не мінімалістичний.

7. «COMFY» [15]

Сайт COMFY поєднує сучасний дизайн і простотою навігації, є акційні товари, але дуже багато зайвих анімацій, дуже яскравий колірний фон.

Таблиця 2.2

Результати аналізу веб-сайтів

№	Назва сайту	Інтерфейс	Зручність	Функціональність	Правильне розміщення банеру	Інтуїтивність
1	АТБ	85%	80%	90%	50%	95%
2	VARUS	65%	70%	75%	65%	75%
3	СІЛЬПО	70%	70%	85%	85%	80%
4	ROZETKA	90%	100%	100%	100%	95%
5	BRAIN	75%	90%	95%	95%	85%
6	TELEMART.UA	70%	65%	80%	90%	75%
7	COMFY	60%	85%	95%	80%	85%

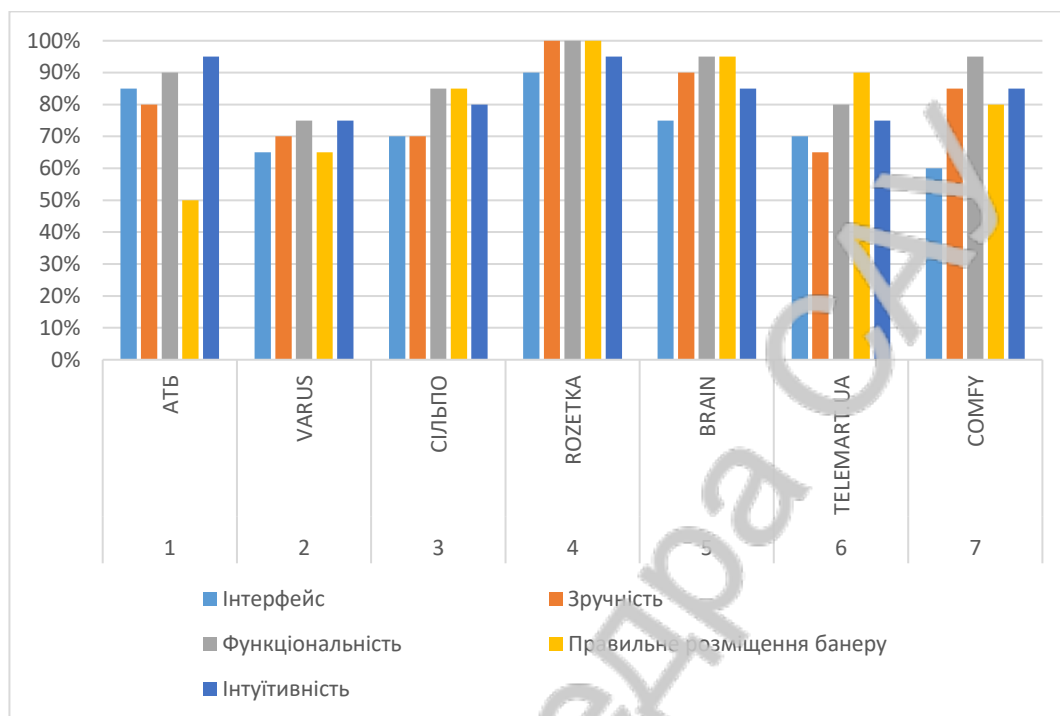


Рисунок 2.6 – Діаграма результатів аналізу веб-сайтів.

Після аналізу було вирішено обрати веб-макет АТБ, оскільки її інтерфейс має дуже добре підібраний колірний фон, зручність використання, функціонал сайту відповідає критеріям продажу товарів, інтуїтивність переміщення по веб-додатку зрозуміла, але каталог та розміщення банерів не зручний.

2.3 Розробка веб-додатку

Розробка веб-додатку розпочалася з визначення основних функцій. Ключовими завданнями були: створення зручного вибору товарів, відображення акційних пропозицій, реалізація інтуїтивного інтерфейсу, а також змогу тестування елементів інтерфейсу за допомогою множинного тестування.

Для реалізації веб-сайту було обрано фреймворк «ANGULAR», він є сучасним інструментом для побудови додатків. Angular дуже зручний своєю модульною архітектурою, що сприяє легкому розширенню нових компонентів.

Основні модулі веб-додатку

- Компонент каталогу товарів – відображення списку товарів

- Компонент картки товару – містить в собі назву товару, ціну, зображення.
- Компонент навігації – меню з переходами по сторінці.

Опис зберігання даних товарів для подальшої реалізації бази даних

На етапі розробки системи використано спрощене зберігання даних у форматі JSON-файлу, що працює через JSON Server. Це рішення не є повноцінною базою даних, але дозволить зручно зопрати та структурувати зібрану інформацію про товари, що значно спрощує подальше проектування і реалізацію повноцінної бази даних.

Кожен товар представлено у вигляді об'єкту з такими ключовими полями:

```
{
  "photo": "assets/imgs/products/Cola.png",
  "title": "Напій 2л Coca-Cola Zero безалкогольний сильногазований ПЕТ",
  "originalPrice": 51.90,
  "discountPrice": 00.00,
  "unit": "грн/л"
}
```

Ключові атрибути бази даних:

- title – назва товару;
- photo – шлях до зображення;
- originalPrice – ціна товару;
- unit – валюта.

Формат JSON дозволяє швидко відображати товари в інтерфейсі, реалізувати сортування та фільтрацію, а також підготувати дані для створення повноцінної бази даних. Такі як: MySQL, Firebase, Oracle.

У подальшому проєкті буде обрано базу даних, яка найкраще відповідатиме вимогам функціоналу та масштабу веб-сайту.

Використані технології:

- Angular версії 15+ – фреймворк для створення веб-сайту;
- TypeScript – мова програмування;
- HTML, CSS – для створення веб-додатку та стилізації;
- Angular Material – для адаптивності та візуалізації;
- Firebase та JSON-server – для зберігання даних.

Для кращої продуктивності та аналізу ефективності інтерфейсних рішень, усі компоненти реалізовані таким чином, щоб веб-сайт мав гнучку зміну елементів – такі як, різне розміщення кнопок, різні макети каталогів та навігації. Це дозволить використати методи множинного тестування без повного переписування коду.

При запуску веб-додатку за допомогою команди `npm start`, в терміналі виведеться local-server: <http://localhost:4200/> (рис. 2.7).

```
PS C:\Users\DarkLight\Desktop\Atb> npm start

> atb@0.0.0 start
> ng serve

Component HMR has been enabled.
If you encounter application reload issues, you can manually reload the page to bypass HMR and/or disable this feature with the '--no-hmr' command line option.
Please consider reporting any issues you encounter here: https://github.com/angular/angular-cli/issues

Initial chunk files | Names      | File size
main.js             | main      | 151.63 kB
polyfills.js       | polyfill  | 9.28 kB
styles.css         | styles   | 7.85 kB
                   | Initial total | 168.72 kB

Application bundle generation complete. [1.436 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
+ Local: http://localhost:4200/
+ press h + enter to show help
```

Рисунок 2.7 – Термінал веб-додатку (Запуск Local-server).

Якщо ми зайдемо по цьому посиланні, то відкриється наш веб-додаток, зображене на (рис. 2.8).

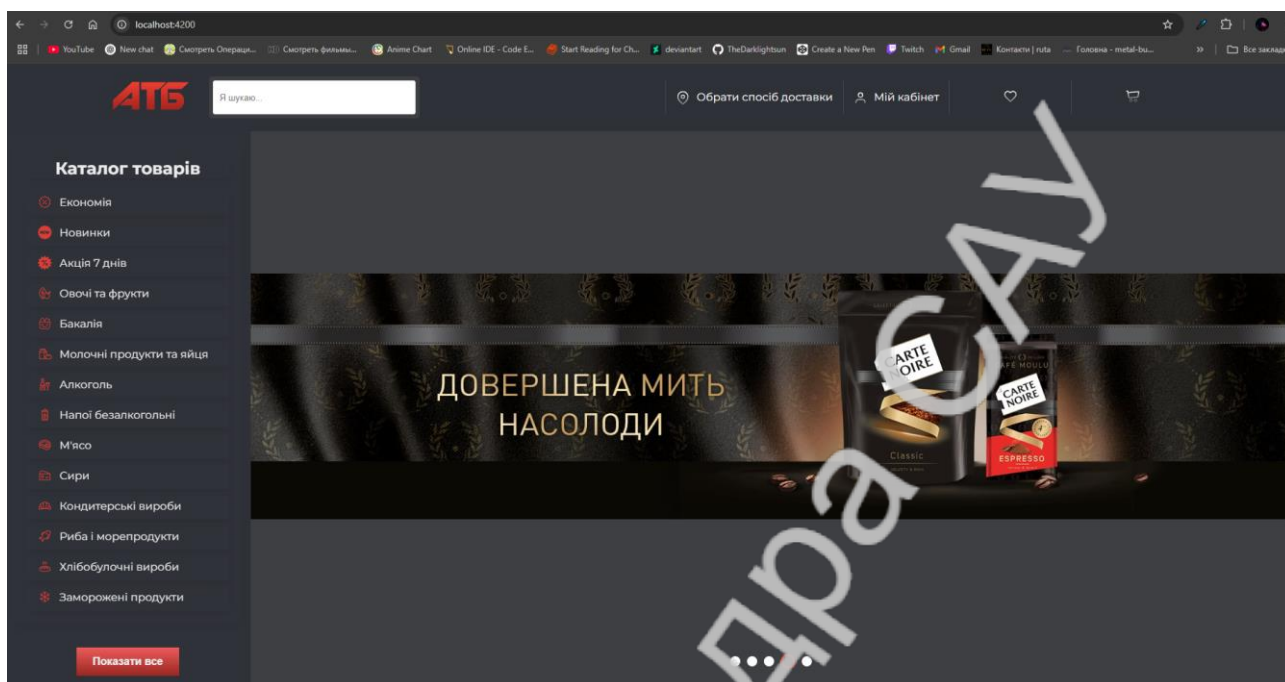


Рисунок 2.8 – Виб-податок.

При натисканні кнопки «Показати все» веб-сайт відреагує, надаючи детальний каталог товарів, як зображено на (рис. 2.9).

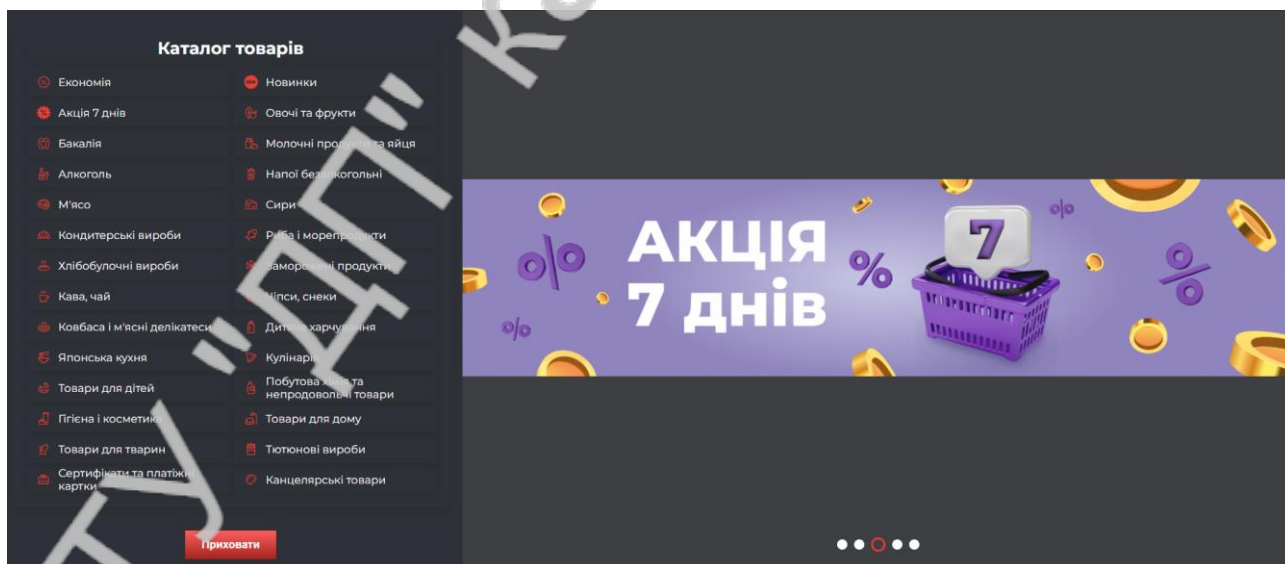


Рисунок 2.9 – Розширення каталогу товарів.

Після перегляду каталогу користувач може натиснути на будь яку категорію товару та побачити візуальний список товарів, приклад наведено на (рис. 2.10).

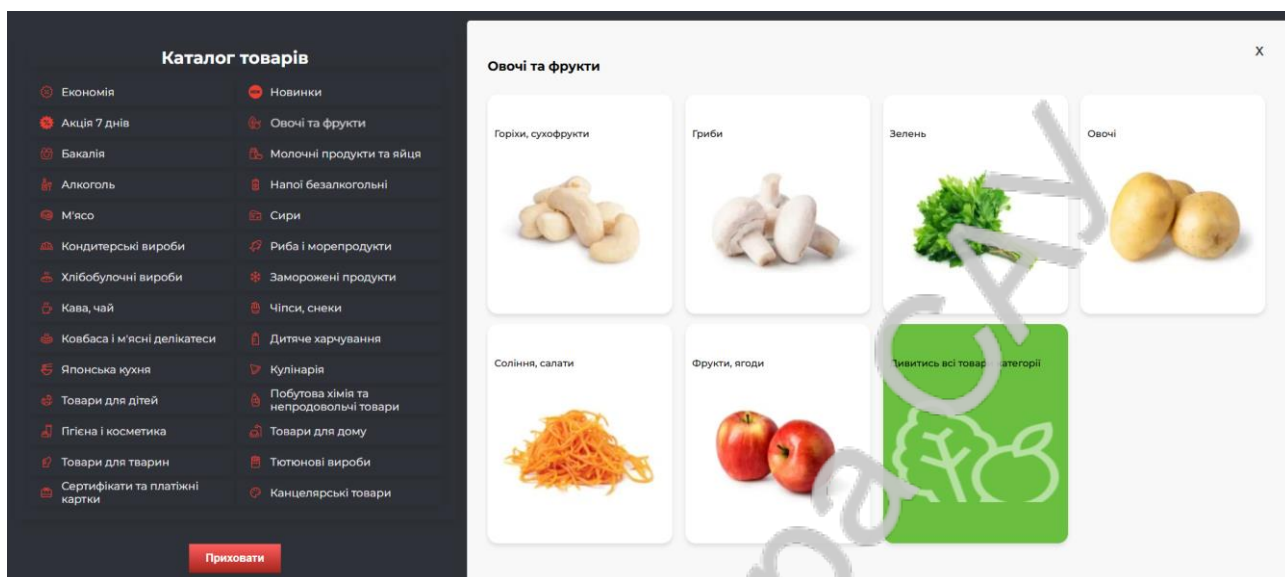


Рисунок 2.10 – Функціонал каталогу товарів.

Повний інтерфейс веб-додатку можна подивитися на (рис. 2.11).

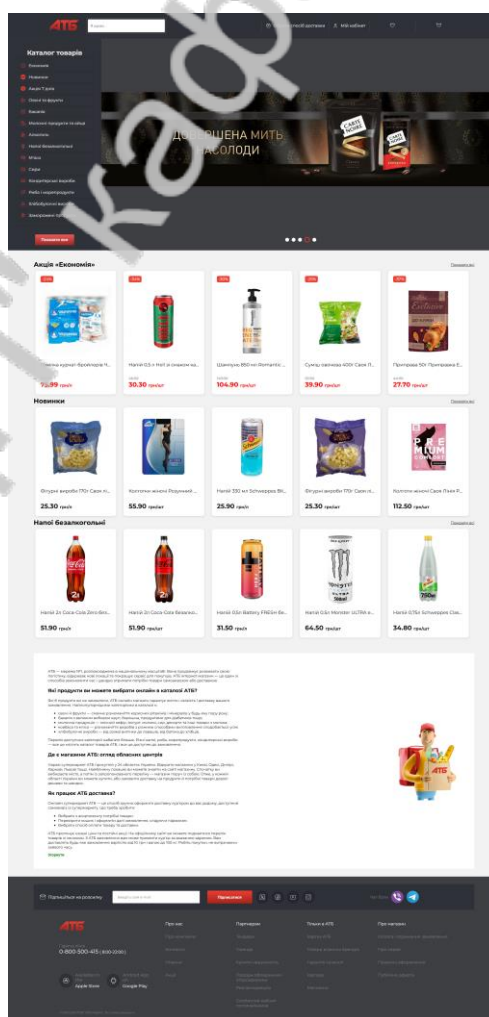


Рисунок 2.11 – Інтерфейс веб-додатку.

Розроблений веб-додаток є повноцінною платформою для демонстрації товарів, з ефективним інтерфейсом, що може адаптуватися до потреб майбутнього використання користувачів.

2.4 Реалізація множинного тестування інтерфейсу

Ключовим моментом розробленого веб-додатку є підтримка методів множинного тестування (multi-armed bandit testing) для аналізу ефективного інтерфейсного рішення. Метою даного підходу є визначення найбільш оптимального варіанту відображення компонентів інтерфейсу, наприклад, зміна кольору кнопки, каталогу товарів, пошуку продукції, який сприйматиме досягненню більшої активності на веб-сайті, а саме збільшення кількості кліків, та швидкого пошуку.

Суть множинного тестування

У множинному тестуванні застосовуються алгоритми адаптивного вибору, наприклад жадібний алгоритм, що дозволяє швидко відмовлятися від менш ефективних варіантів активності до найкращих. Теоретичний приклад можна побачити в таблиці 2.3.

Таблиця 2.3

Приклад варіантів тестування

Варіант	Змінна частина інтерфейсу
1	Каталог товарів
2	Виведення категорій
3	Візуалізація кнопок
4	Розташування банеру
5	Картки товарів
6	Навігація по веб-сайту
7	Палітра кольорів веб-сайту
8	Додати новий функціонал

Технічна реалізація

Для реалізації адаптивного множинного тестування інтерфейсу використано ϵ -жадібний алгоритм багаторукого бандита. Цей метод дозволяє динамічно визначити найбільш ефективний варіант інтерфейсу на основі кліків користувачів.

Суть алгоритму полягає в наступному:

1. З імовірністю ϵ (наприклад, 10%) випадковим чином обирається будь-який з варіантів (це називається експлорація – дослідження нових варіантів).
2. З імовірністю $1-\epsilon$ (тобто 90%) обирається той варіант, що раніше показав найкращий результат (це експлуатація – використання вже перевічених рішень).
3. Результати (перегляди, кліки) зберігаються у статистику для подальшого аналізу.

Реалізація ϵ -жадібного алгоритму на TypeScript (Angular)

У розробленому веб-сайті реалізовано ϵ -жадібний алгоритм багаторукого бандита, який дозволить автоматично визначати найефективніший варіант інтерфейсу. Реалізацію виконано за допомогою мови програмування TypeScript, у середовищі Angular.

```

type Variant = 'A' | 'B' | 'C'; // 3 варіанти інтерфейсу

interface VariantStats {
  views: number; // Кількість показів
  clicks: number; // Кількість кліків
}

// Статистика для кожного варіанта
const stats: Record<Variant, VariantStats> = {
  A: {views: 0, clicks: 0},
  B: {views: 0, clicks: 0},

```

```
C: {views: 0, clicks: 0}
};

const epsilon = 0.1; // Ймовірність випадкового вибору (10%)

// Функція для вибору інтерфейсу
function chooseVariant(): Variant {
  const random = Math.random(); // Генерація випадкового числа від 0 до 1

  if (random <= epsilon) {
    // Випадковий вибір (експлорація)
    const variant: Variant[] = ['A', 'B', 'C'];
    const randomIndex = Math.floor(Math.random() * variant.length);

    return variant[randomIndex];
  } else {
    return (Object.keys(stats) as Variant[]).reduce((best, curr) => {
      // Вибираємо найкращий варіант за ефективністю (експлуатація)
      const bestCTR = stats[best].clicks / (stats[best].views || 1);
      const currCTR = stats[curr].clicks / (stats[curr].views || 1);

      return currCTR > bestCTR ? curr : best;
    });
  }
}

// Фіксація перегляду варіантів
function recordView(variant: Variant) {
  stats[variant].views++;
}

// Фіксація кліку по варіанту
function recordClick(variant: Variant) {
  stats[variant].clicks++;
}

// Приклад використання
const selected = chooseVariant(); // Вибір варіанту

recordView(selected); // Записуємо перегляд
recordClick(selected); // Записуємо клік

// Вивід результату у консоль
console.log('Поточна статистика: ', stats);
```

Основні етапи роботи алгоритму:

1. Ініціалізація варіантів – Створюється перелік можливих варіантів інтерфейсу ('A', 'B', 'C') та об'єкт stats, в якому зберігаються статистичні дані – кількість переглядів (views) та кліків (clicks) для кожного варіанту.
2. Генерація випадкового числа – генерує числа від 0 до 1. Якщо число менше ϵ (наприклад, 0.1) – обирається випадковий варіант. Якщо більше або дорівнює ϵ , обирається той варіант, у якого співвідношення кліків до переглядів – найвища.
3. Експлорація (дослідження) – якщо спрацює 10% випадків, обирається випадковий варіант інтерфейсу із трьох можливих.
4. Експлуатація (вибір найкращого) – якщо випадкове число більше або дорівнює ϵ , проводиться порівняння всіх варіантів.
5. Фіксація переглядів і кліків – після вибору варіанту, він показується користувачу. У момент показу викликається `recordView(variant)` – фіксація переглядів. Якщо користувач натиснув на елемент, викликається `recordClick(variant)` – фіксація кліку.
6. Аналіз і збереження результатів – Отримані дані можна зберігати у файл або базу даних, а потім будувати діаграму результатів.

Як це виглядає у роботі сайту:

- кожен користувач бачить один із трьох варіантів інтерфейсу;
- веб-сайт автоматично визначає, який інтерфейс кращий за активністю користувачів;
- погані варіанти поступово відкидаються, а кращі – показуватимуть частіше;
- таким чином досягається оптимізація інтерфейсу без ручного втручання.

Варіанти інтерфейсів, які тестуються

Для множинного тестування реалізовано три варіанти інтерфейсу, які мінятимуть каталог товарів (А, В, С). Вони відрізняються такими компонентами в таблиці 2.4:

Таблиця 2.4

Приклад варіантів інтерфейсів, які тестуються

Варіант	Змінна частина інтерфейсу
А	Каталог у вигляді плитки з іконками
В	Каталог у вигляді списку
С	Каталог у вигляді іконок

2.5 Проведення множинного тестування. Аналіз результатів

З метою покращення використання користувачем веб-сайту було зроблено три варіанти інтерфейсів категорій, які відрізнялися способом подання інформації, дизайном та логікою взаємодії.

Варіант А – Каталог у вигляді плитки з іконками

Категорії відображаються у вигляді великих плиток з іконками. Цей підхід робить інтерфейс привабливим і дозволяє швидко орієнтуватися в асортименті (рис 2.12).

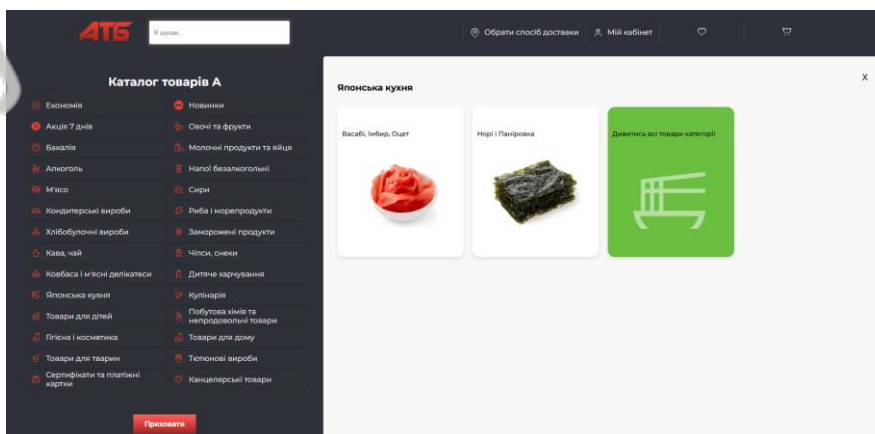


Рисунок 2.12 – Інтерфейс категорій: Варіант А (плитки з іконками).

Варіант В – Каталог у вигляді списку

Категорії представлено у вигляді звичайного текстового списку. Цей варіант компактний та простий у реалізації, однак менш виразний для користувача (рис. 2.13).

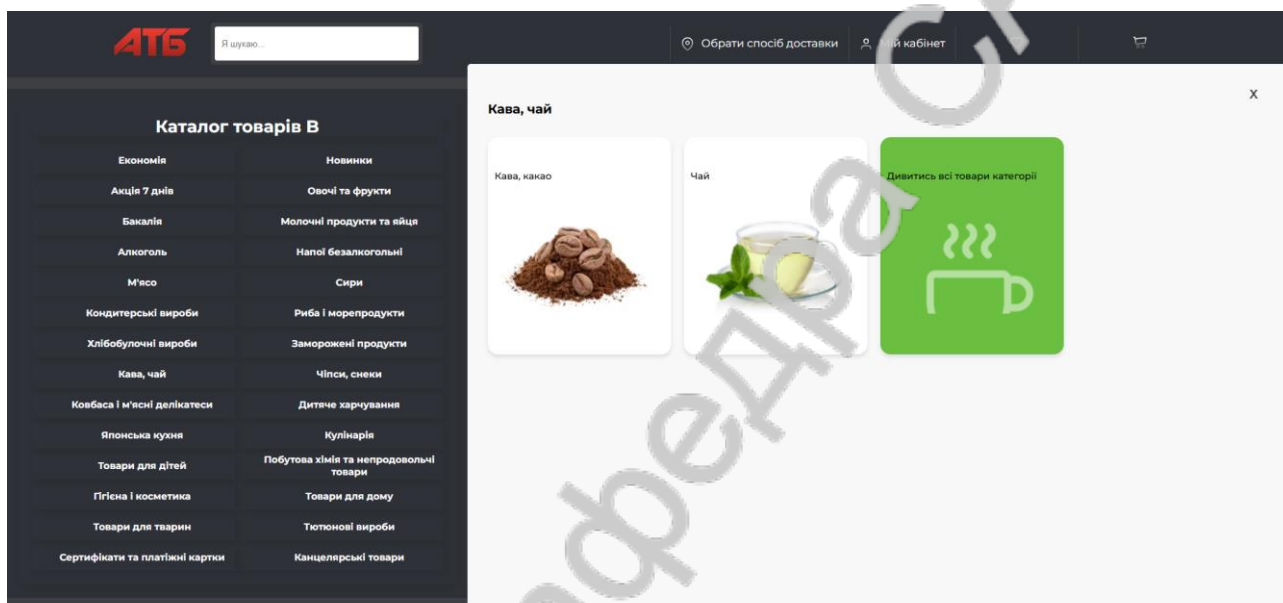


Рисунок 2.13 – Інтерфейс категорій: Варіант В (Каталог у вигляді списку).

Варіант С – Каталог у вигляді іконок

Категорії представлено у вигляді іконок. Цей варіант дуже простий, однак менш інформаційний (рис. 2.14).

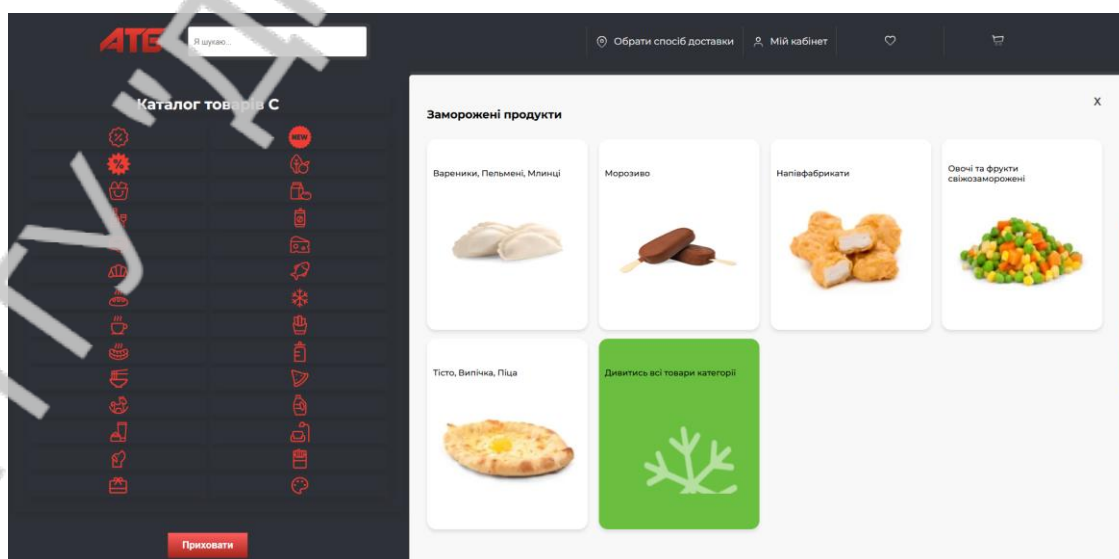


Рисунок 2.14 – Інтерфейс категорій: Варіант С (Каталог у вигляді іконок).

Створення повноцінної бази даних

Для зберігання даних категорій, товарів та результатів тестування інтерфейсів у реальному часі було обрано хмарну платформу Firebase. Це рішення дозволило забезпечити:

- зберігання динамічних даних (категорії, товари, користувацька активність);
- автоматичну синхронізацію змін у режимі реального часу;
- простоту інтеграції з Angular через Firebase SDK;
- масштабованість без необхідності ручного адміністрування серверів.

Основні переваги використання Firebase:

- автоматична синхронізація даних між клієнтами;
- легка інтеграція з Angular через @angular/firebase;
- вбудована аутентифікація та правила безпеки;
- розміщення в хмарі – не потребує власного сервера;
- Можливість відстежувати кліки / перегляди безпосередньо в базі даних.

Структура бази даних у Firebase:

Для організації зберігання даних у Firebase була розроблена структура колекцій:

1. analytics – колекція для збору аналітики кожного сеансу користувачів:

- idUser (string) – унікальний ідентифікатор користувача;
- actionLogs (objects) – об'єкт дій користувача;
 - eventType – тип події (click, view);
 - message – опис події наприклад (Клік на категорію, Напої);
 - timeOnPageSeconds – час, за який була здійснена дія;
 - timestamp – точна дата та час події;

- stats (objects) – категорія де зібрано кількість кліків та перегляду;
 - clicks – загальна кількість кліків за сеанс;
 - views – скільки переглядів за сеанс;
 - timeOnPageSeconds – загальний час перебування користувача на сторінці;
 - variant – варіант інтерфейсу (A, B або C);
2. catalogItemDiscount – колекція акційних товарів:
- id – унікальний ідентифікатор товару;
 - discountPrice – ціна зі знижкою;
 - label – розмір знижки у відсотках (-25%);
 - originalPrice – початкова ціна товару;
 - photo – фото товару;
 - title – назва товару;
 - unit – валюта товару;
3. catalogItemsDrinks – колекція товарів напоїв:
- originalPrice – початкова ціна товару;
 - photo – фото товару;
 - title – назва товару;
 - unit – валюта придбання товару;
4. catalogItemsNews – колекція нових товарів:
- originalPrice – початкова ціна товару;
 - photo – фото товару;
 - title – назва товару;
 - unit – валюта придбання товару;
5. categories – колекція основних категорій:
- link – посилання на категорію товару;
 - subcategories (objects) – об'єкт підкатегорій;
 - link – посилання на підкатегорії;

- svg – векторне зображення категорії;
- title – назва підкатегорії;

Візуальне відображення структури бази даних наведено на (рис. 2.15)

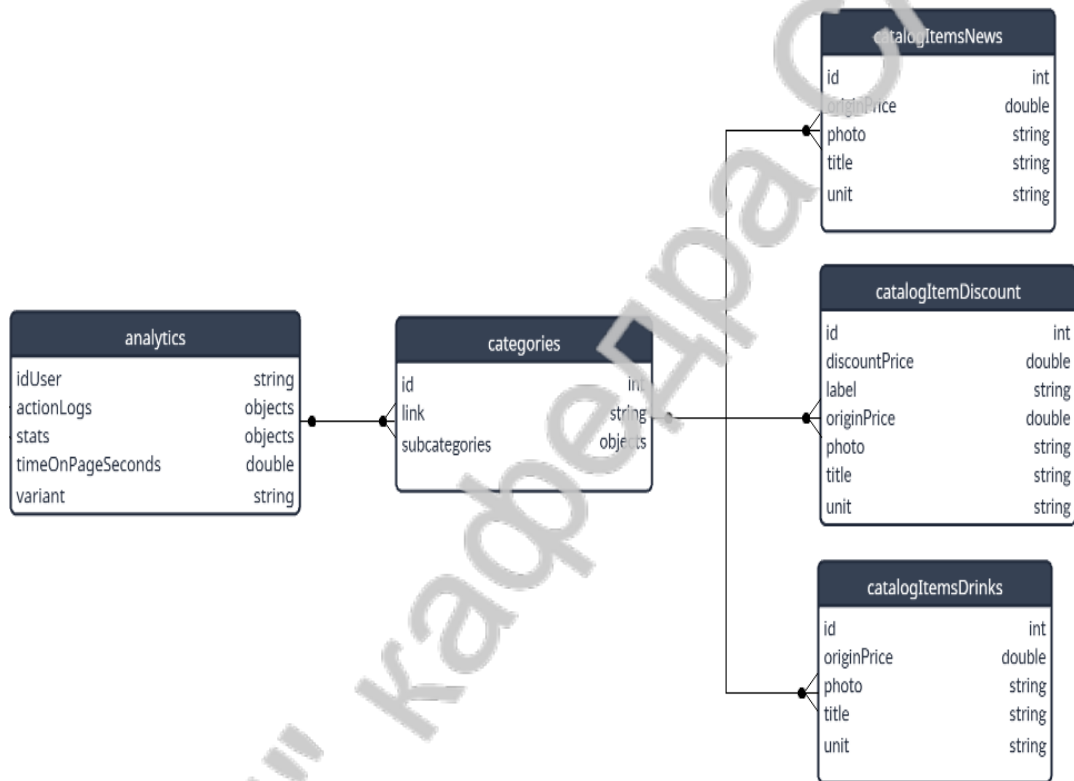


Рисунок 2.15 – Схема бази даних.

Така структура дозволяє:

- легко додавати нові категорії або підкатегорії без зміни архітектури;
- оновлювати акційні або нові товари незалежно від загального каталогу;
- ефективно збирати аналітичні дані на рівні окремих сеансів, що є ключовим для порівняння ефективності інтерфейсів;
- реалізувати персоналізацію контенту на основі кліків та переглядів.

Структурну частину бази даних Firebase наведено на (рис. 2.16, 2.17, 2.18, 2.19, 2.20).

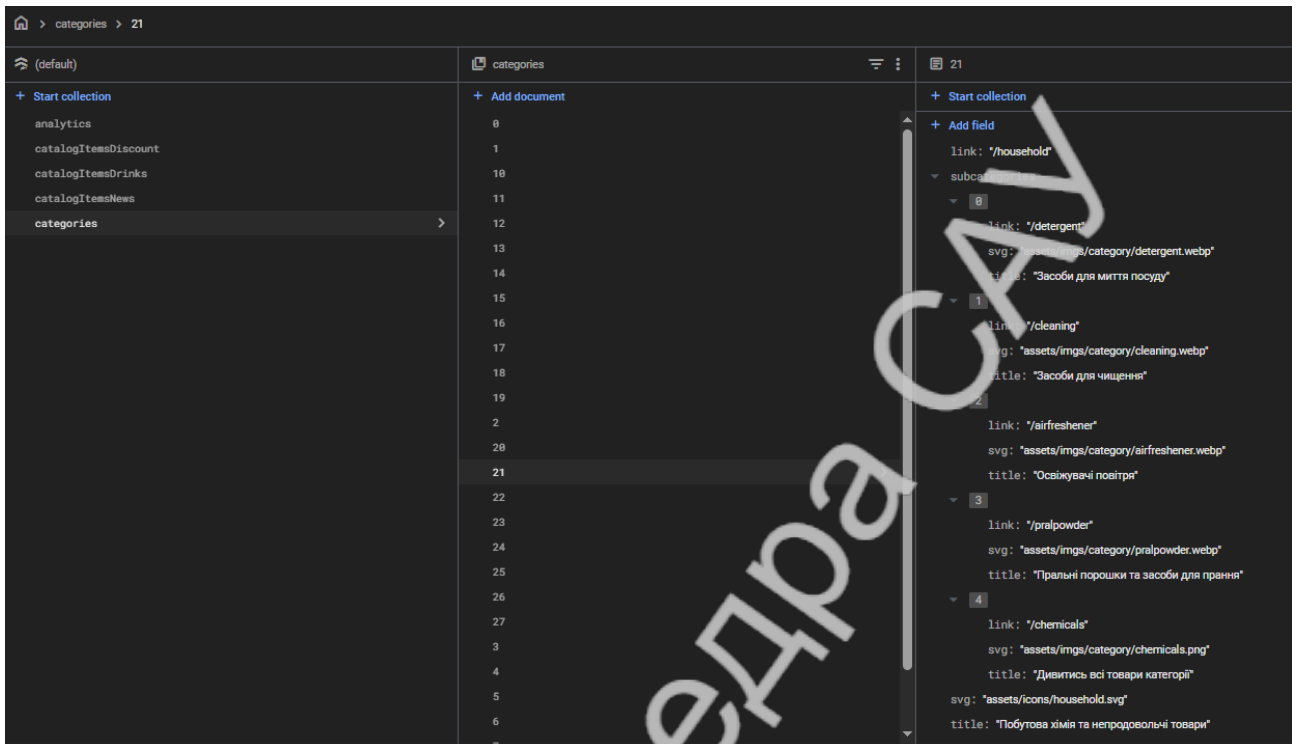


Рисунок 2.16 – Структура колекції бази даних (categories).

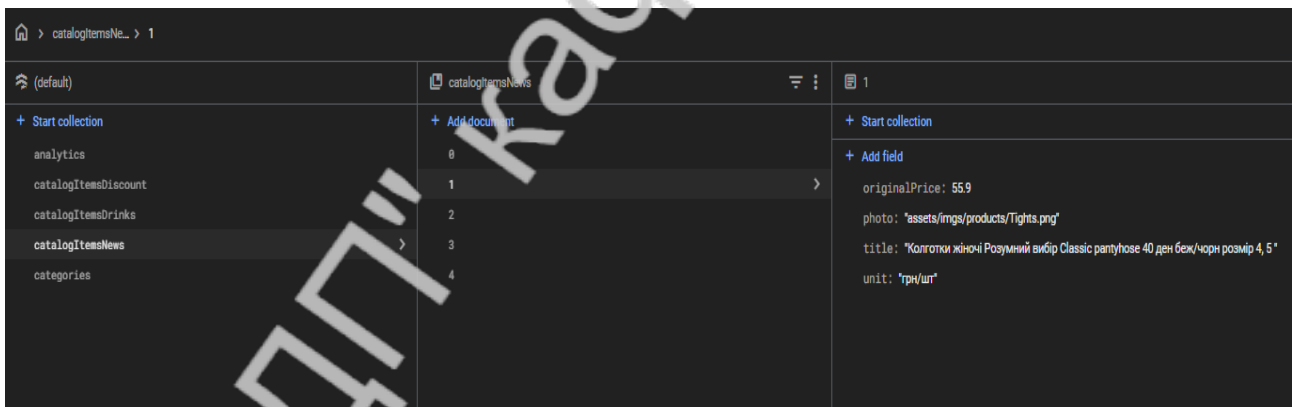


Рисунок 2.17 – Структура колекції бази даних (catalogItemNews).

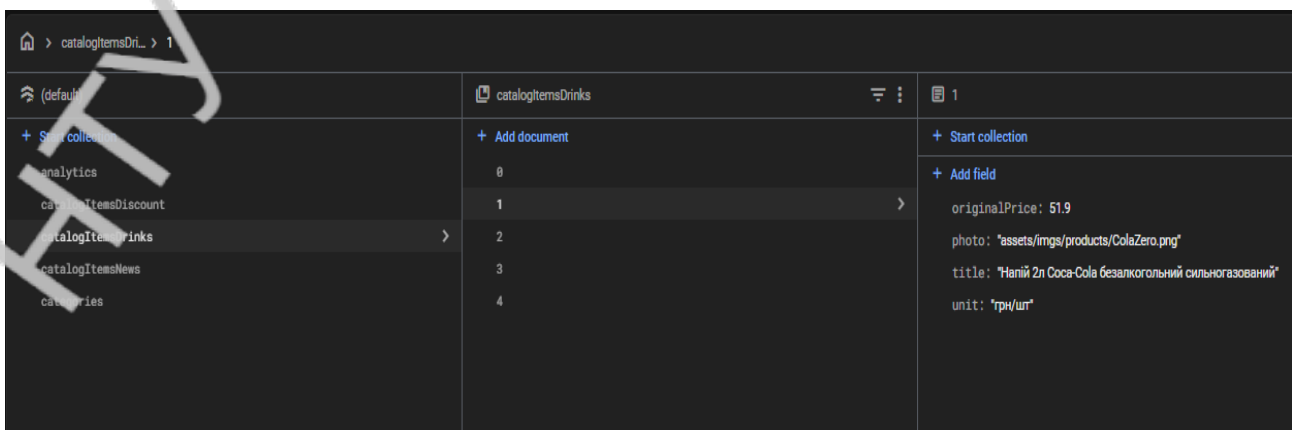


Рисунок 2.18 – Структура колекції бази даних (catalogItemDrinks).

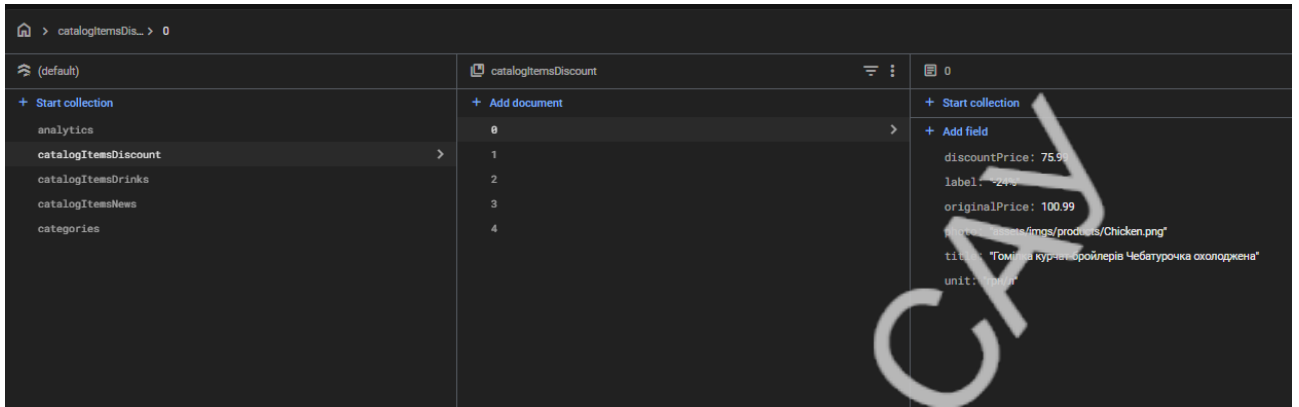


Рисунок 2.19 – Структура колекції бази даних (catalogItemDiscount).

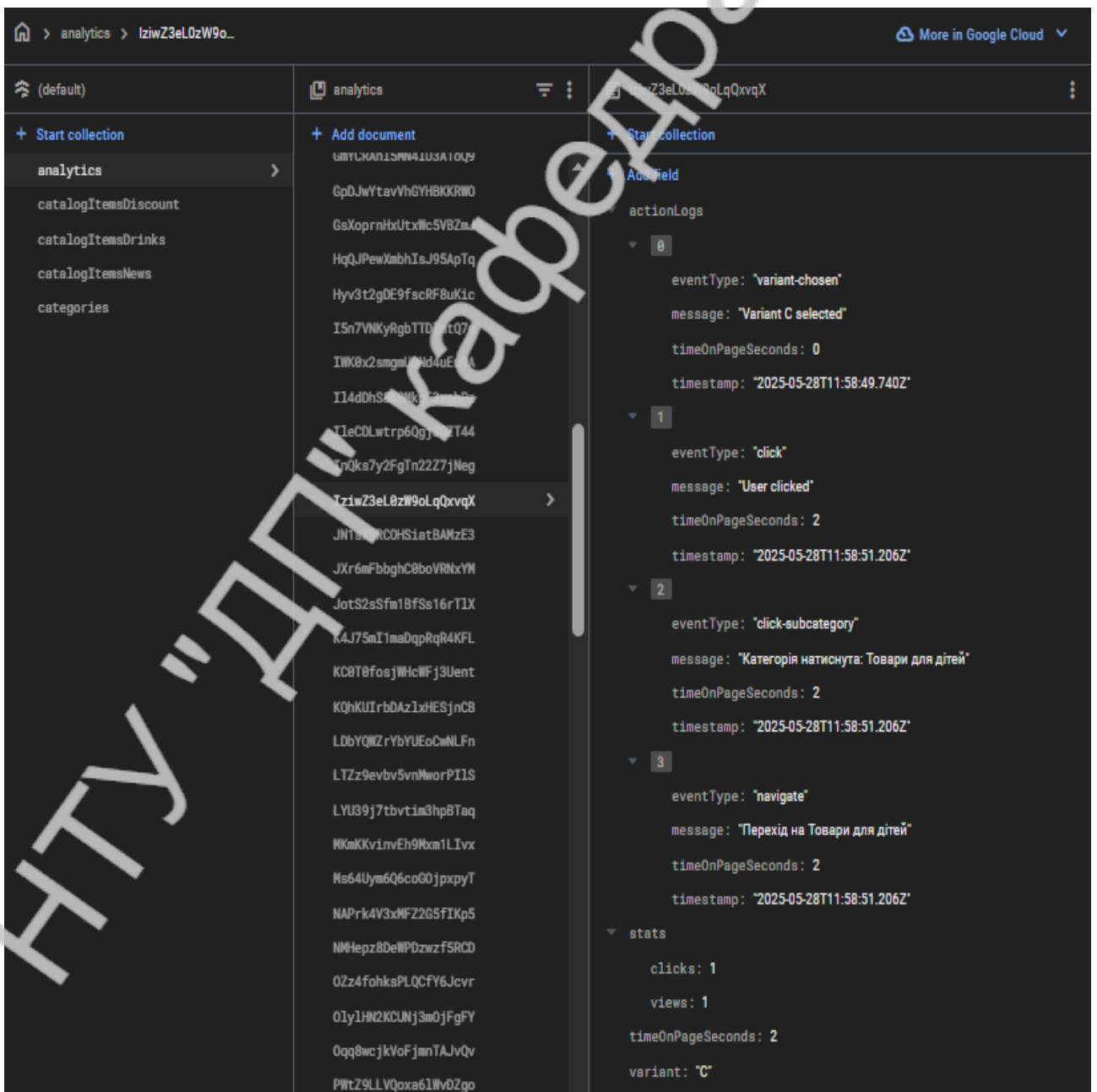


Рисунок 2.20 – Структура колекції бази даних (analytics).

2.5.1 Підключення Firebase до Angular

Для підключення Firebase до Angular було використано бібліотеку @anlular/firestore, яка дозволяє працювати з Firebase, Firestore та іншими сервісами Google.

Базові операції з Firebase

Після ініціалізації та підключення бібліотеки доступні наступні типові операції з Firestore:

1. Читання даних з колекції:
 - Отримання всіх даних з колекції (наприклад, категорії або товарів);
 - Реактивне оновлення при зміні даних у реальному часі.
2. Запис даних:
 - Додавання нових даних до колекцій (add);
 - Створення даних із заданим ID (set).
3. Оновлення:
 - Часткове або повне оновлення полів даних (update).
4. Видалення:
 - Видалення даних з колекції (delete).
5. Збір аналітики:
 - Логування дій користувача у колекцію analytics;
 - Фіксація часу перебування, кліків, фільтрів, версії інтерфейсу.

2.5.2 Результат тестування

Після запуску множинного тестування із застосуванням ε-жадібного алгоритму було зібрано статистику переглядів та кліків для кожного з інтерфейсів. Можна переглянути в таблиці 2.5, 2.6 та діаграми (рис. 2.21, 2.22, 2.23, 2.24).

Таблиця 2.5

Загальний аналіз користувацької взаємодії у варіантах А, В, С

Варіант	Переглядів	Кліки	Загальна кількість часу (хв)	CTR (%)
А	55	163	15,2	296,36
В	54	158	25	292,59
С	42	93	16	233,33
Всього:	151	419	56,2	822,29
Середнє значення (переглядів)	50,33			
Середнє значення (кліків)		139,67		
Середнє значення (часу)			18,73	

Таблиця 2.6

Аналіз взаємодії з каталогом товарів у трьох варіантів інтерфейсу

Назва	Тип	А	В	С	Середнє значення
Новинки	Категорія	2	1	4	2,33
Акція 7 днів	Категорія	1	5	3	3,00
Овочі та фрукти	Категорія	3	2	1	2,00
Молочні продукти та яйця	Категорія	0	1	1	0,67
М'ясо	Категорія	0	2	0	0,67
Кондитерські вироби	Категорія	10	6	1	5,67
Риба і морепродукти	Категорія	6	11	2	6,33
Хлібобулочні вироби	Категорія	5	2	3	3,33
Заморожені продукти	Категорія	9	10	6	8,33
Кава, чай	Категорія	2	8	1	3,67
Чипси, снеки	Категорія	5	14	3	7,33
Ковбаса і м'ясні делікатеси	Категорія	2	2	1	1,67
Дитяче харчування	Категорія	8	3	12	7,67
Японська кухня	Категорія	2	2	1	1,67
Кулінарія	Категорія	8	3	7	6,00
Товари для дітей	Категорія	2	6	8	5,33

«1»	«2»	«3»	«4»	«5»	«6»
Побутова хімія та непродовольчі товари	Категорія	3	4	0	2,33
Гігієна і косметика	Категорія	4	6	1	3,67
Товари для дому	Категорія	3	2	1	2,00
Товари для тварин	Категорія	1	2	2	1,67
Тютюнові вироби	Категорія	1	2	0	1,00
Сертифікати та платіжні картки	Категорія	3	2	1	2,00
Гриби	Підкатегорії	1	1	0	0,67
Молоко	Підкатегорії	0	0	2	0,67
Сири м'які	Підкатегорії	0	1	0	0,33
Батончики	Підкатегорії	15	8	0	7,67
Бісквіти та десерти	Підкатегорії	3	1	0	1,33
Жувальні гумки	Підкатегорії	1	0	0	0,33
Ікра та рибна паста	Підкатегорії	9	7	1	5,67
Крабові палички	Підкатегорії	1	1	1	1,00
Батон	Підкатегорії	2	0	3	1,67
Вареники, Пельмені, Млинці	Підкатегорії	3	8	2	4,33
Морозиво	Підкатегорії	2	0	0	0,67
Тісто, Випічка, Піца	Підкатегорії	0	2	0	0,67
Кава, какао	Підкатегорії	0	2	0	0,67
Снеки	Підкатегорії	2	6	3	3,67
Чипси	Підкатегорії	1	3	0	1,33
Інші м'ясні вироби	Підкатегорії	1	1	1	1,00
Вода	Підкатегорії	6	3	3	4,00
Дитяче печиво	Підкатегорії	1	0	4	1,67
Каші	Підкатегорії	1	0	6	2,33
Васабі, Імбир, Оцет	Підкатегорії	1	0	1	0,67
Бутерброди, бургери	Підкатегорії	10	2	3	5,00
Піца	Підкатегорії	2	0	1	1,00
Підгузки, підшюшки	Підкатегорії	1	3	1	1,67
Засоби для миття посуду	Підкатегорії	2	1	0	1,00
Ватні палички/диски	Підкатегорії	2	2	1	1,67
Гігієнічні прокладки/тампони	Підкатегорії	0	2	0	0,67
Батарейки	Підкатегорії	1	0	1	0,67
Наповнювач	Підкатегорії	1	1	1	1,00
Благодійна ініціатива	Підкатегорії	1	0	1	0,67

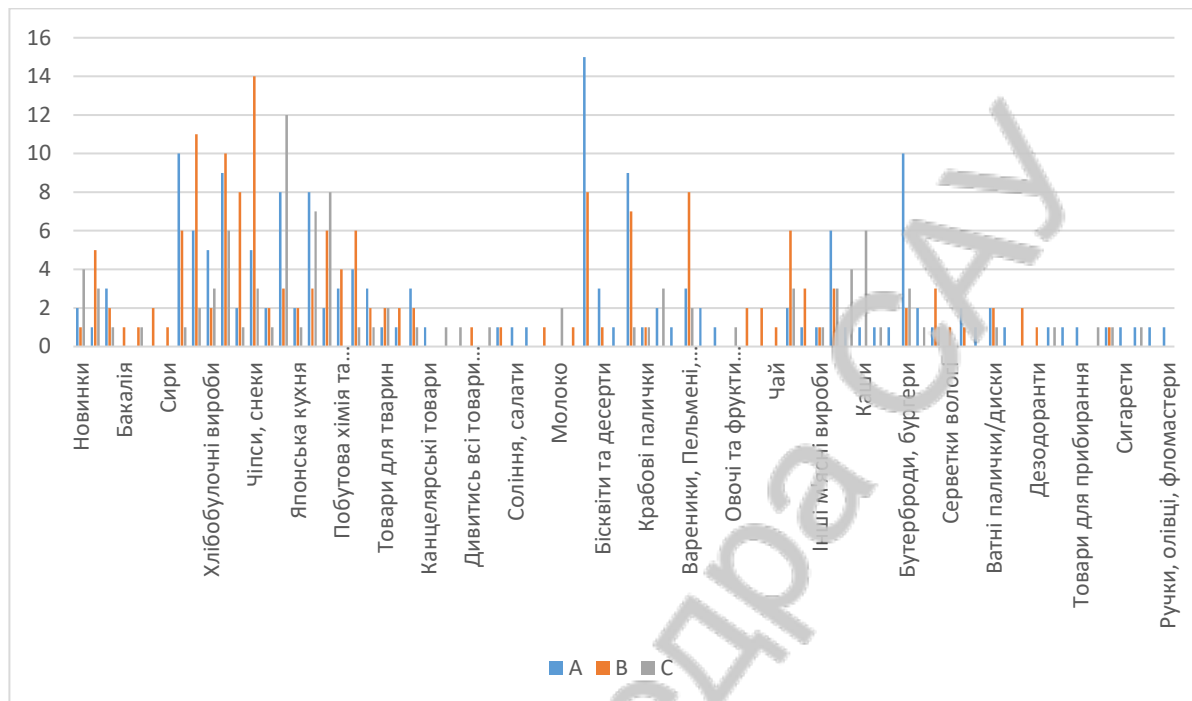


Рисунок 2.21 – Діаграма кількості переглядів категорій та підкатегорій (без непереглянутих).

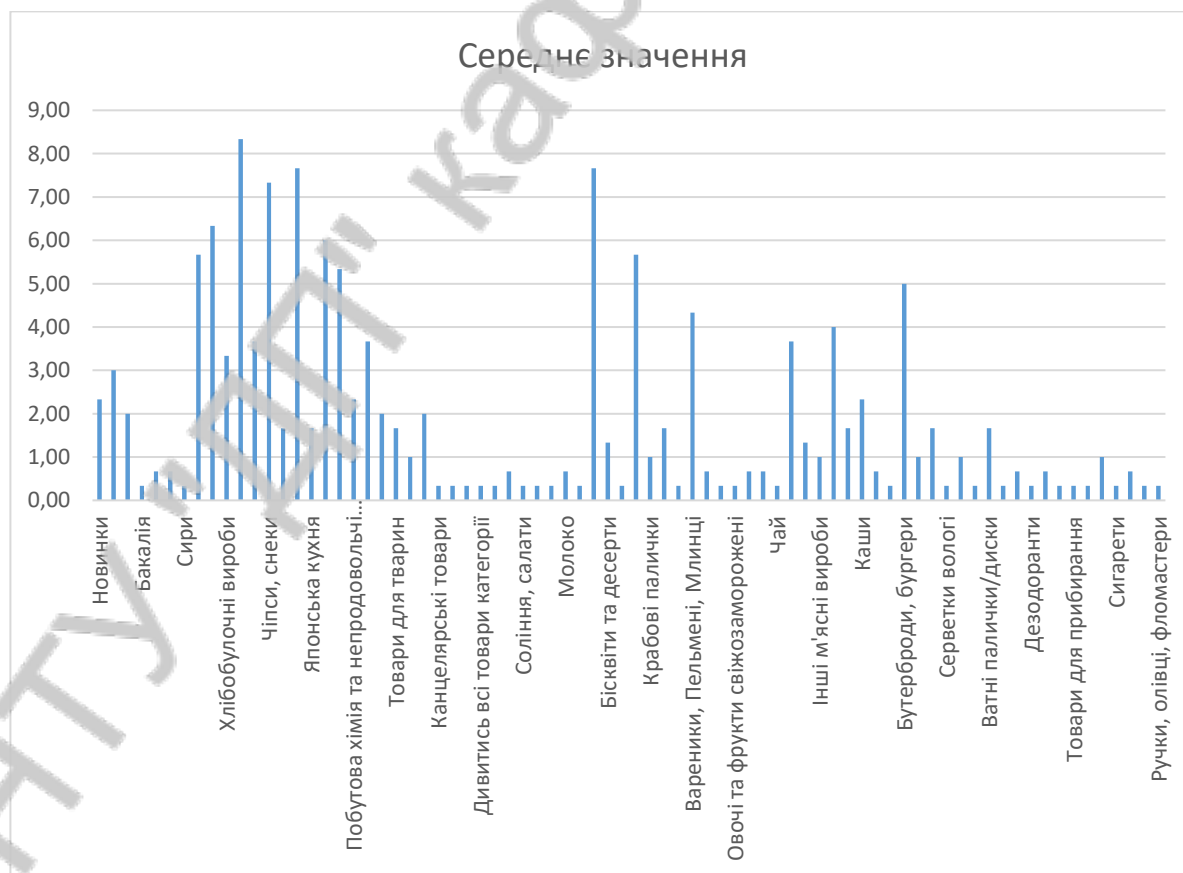


Рисунок 2.22 – Діаграма середньої кількості переглядів категорій та підкатегорій (без непереглянутих).

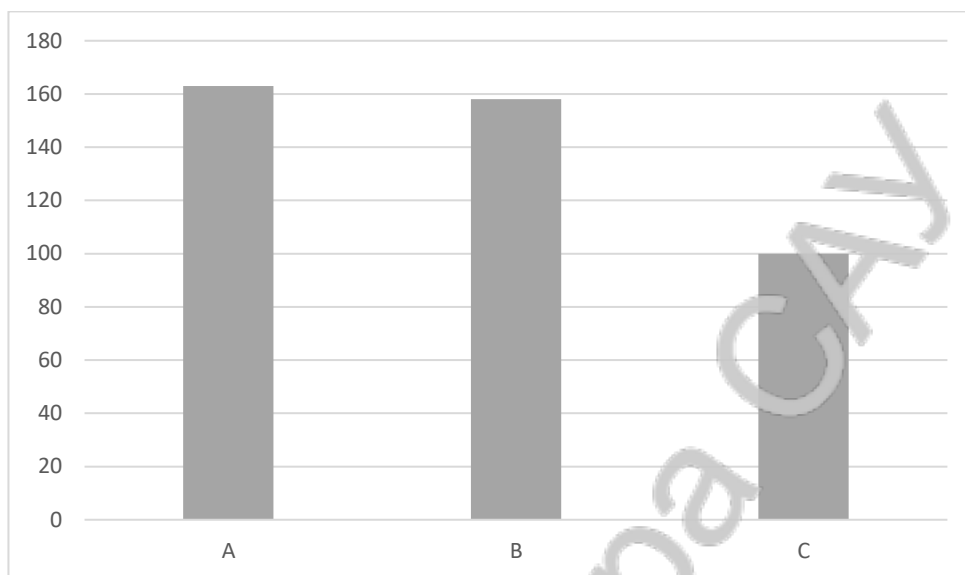


Рисунок 2.23 – Діаграма загальної кількості взаємодій з категоріями та підкатегоріями.



Рисунок 2.24 – Діаграма загальної кількості категорій та підкатегорії без взаємодій.

Висновок тестування

Результати А/В/С тестування показали, що варіант А (плитки з іконками) є найефективнішим серед трьох протестованих інтерфейсів (рис. 2.25).

Він продемонстрував:

- найвищу кількість кліків;
- найвищий (CTR – показник кліків ($\frac{\text{Кількість кліків}}{\text{Кількість переглядів}} * 100\%$));
- найбільшу взаємодію з категоріями;
- а також найкраще сприйняття з боку користувачів.

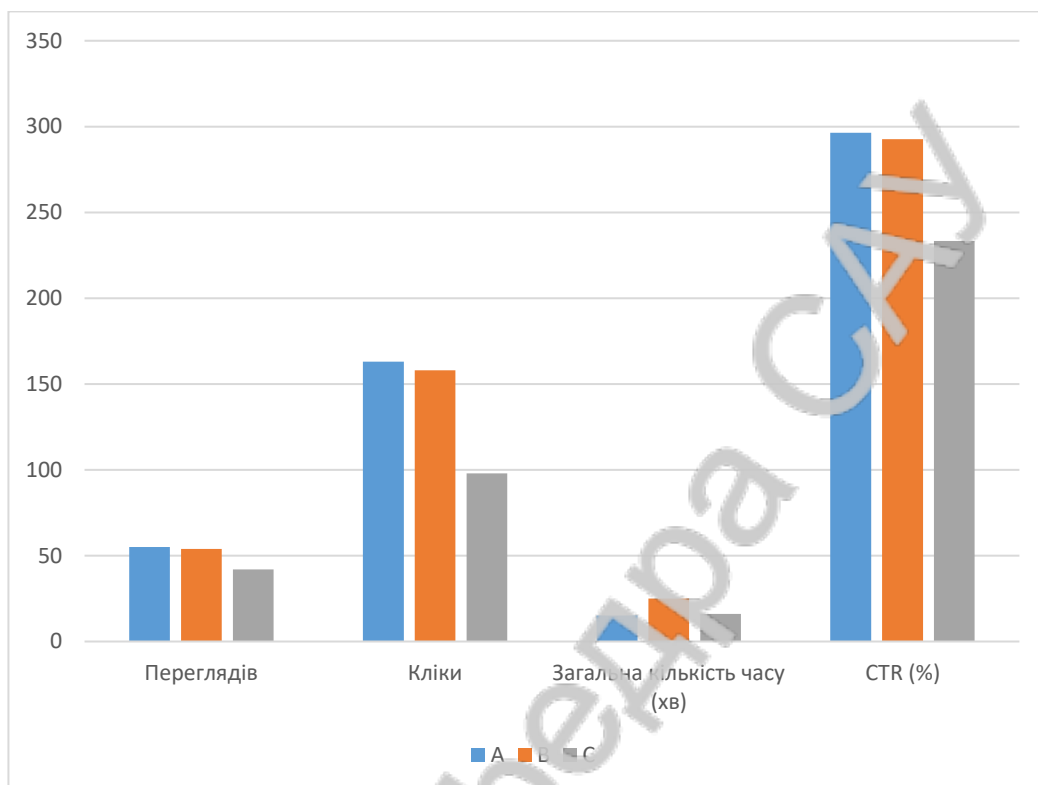


Рисунок 2.25 – Діаграма підсумкових даних.

Це свідчить про те, що візуально виразне представлення категорій значно покращує користувацький досвід і підвищує загальну ефективність веб-сайту.

На основі отриманих даних було ухвалено рішення впровадити варіант А у фінальну версію проекту.

ВИСНОВОК

У рамках дипломної роботи було реалізовано повний цикл розробки веб-додатку для онлайн-магазину з акцентом на покращення інтерфейсу вибору товарів. Основною метою дослідження було підвищення зручності та ефективності взаємодії користувача з веб-інтерфейсом за допомогою сучасних технологій і методів тестування.

У теоретичній частині роботи проведено аналіз підходів до створення інтерфейсів веб-сайтів, зокрема розглянуто концепції дизайну, адаптивності та компонентного підходу. Окрему увагу приділено методу множинного тестування, зокрема алгоритму багаторукового бандита, як ефективному способу адаптивного вибору найбільш результативного варіанту інтерфейсу.

У практичній частині реалізовано веб-додаток за допомогою фреймворку Angular. Створено базу даних товарів і категорій, що зберігається на хмарній платформі Firebase. Реалізовано три варіанти інтерфейсу категорій товарів, які тестувалися автоматично за допомогою ϵ -жадібного алгоритму. Для кожного з варіантів збиралися дані про перегляди та кліки, на основі яких визначено ефективність кожного рішення.

Проведене тестування показало, що найкращим варіантом є інтерфейс у вигляді плиток з іконками (варіант А) – він отримав найвищий показник взаємодії користувачів (CTR). Це підтверджує доцільність використання візуального привабливого подання інформації в онлайн-торгівлі.

У результаті реалізоване програмне рішення дозволяє не тільки ефективно управляти каталогом товарів, а й автоматично адаптувати інтерфейс відповідно до вподобань користувачів. Це відкриває перспективи для подальшого розширення функціоналу, впровадження персоналізованих рекомендацій, а також аналітики користувацької поведінки.

Таким чином, поставлені у роботі задачі були повністю виконані, а розроблена система може бути використана як основа для комерційного онлайн-

магазину або масштабована під потреби будь-якої веб-платформи електронної комерції.

НТУ "ДІТ" кафедра САУ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. YouScan. Маркетингові дослідження [Електронний ресурс]. Доступ: <https://youscan.io/ua/blog/market-research/>
2. 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>
3. 10 принципів гарного веб-дизайну. [Електронний ресурс]. Доступ: <https://www.smashingmagazine.com/2008/01/10-principles-of-effective-web-design/>
4. Вступ до Angular. [Електронний ресурс]. Доступ: <https://v17.angular.io/guide/what-is-angular>
5. Документація Angular CLI. [Електронний ресурс]. Доступ: <https://angular.dev/tools/cli>
6. Вступ до Node.js. [Електронний ресурс]. Доступ: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
7. TypeScript для програмістів-початківців. [Електронний ресурс]. Доступ: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>
8. Вступ до JavaScript. [Електронний ресурс]: Доступ: <https://uk.javascript.info/intro>
9. «АТБ» Онлайн магазин продуктів [Електронний ресурс]. Доступ: <https://www.atbmarket.com/>
10. «VARUS» Онлайн магазин продуктів [Електронний ресурс]. Доступ: <https://varus.ua/>
11. «СІЛЬПО» Онлайн магазин продуктів [Електронний ресурс]. Доступ: <https://silpo.ua/>
12. «ROZETKA» Онлайн магазин техніки [Електронний ресурс]. Доступ: <https://rozetka.com.ua/>

13. «BRAIN» Онлайн магазин техніки [Електронний ресурс]. Доступ: <https://brain.com.ua/ukr/>
14. «TELEMART.UA» Онлайн магазин техніки [Електронний ресурс]. Доступ: <https://telemart.ua/ua/>
15. «COMFY» Онлайн магазин приладів [Електронний ресурс]. Доступ: <https://comfy.ua/>
16. 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>
17. GitHub [Електронний ресурс]: завантажений повний проект. Доступ: <https://github.com/Kyrylo-Bublyk/atb-market-thesis>
18. 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. Маленко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 32 с. Доступ: <https://do.nmu.org.ua>

Додаток А

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки			
1										
2					Документація					
3										
4	САУ.КР.25.04.ПЗ				Пояснювальна записка	72	Формат А4			
5										
6					Демонстраційний матеріал	12	Презентація на CD-R			
7										
8					Копія роботи	1	Диск CD-R			
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
					САУ.КР.25.04.ДА.ПЗ.					
Змін.	Аркуш	№ докум.	Підпис	Дата						
Розроб.		Бублик К.О.			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.								

Додаток В

Аналіз взаємодії з каталогом товарів у трьох варіантів інтерфейсу

Назва	Тип	А	В	С	Середнє значення
Економія	Категорія	0	0	0	0,00
Новинки	Категорія	2	1	4	2,33
Акція 7 днів	Категорія	1	5	3	3,00
Овочі та фрукти	Категорія	3	2	1	2,00
Бакалія	Категорія	0	1	0	0,33
Молочні продукти та яйця	Категорія	0	1	1	0,67
Алкоголь	Категорія	0	0	0	0,00
Напої безалкогольні	Категорія	0	0	0	0,00
М'ясо	Категорія	0	2	0	0,67
Сири	Категорія	0	1	0	0,33
Кондитерські вироби	Категорія	10	6	1	5,67
Риба і морепродукти	Категорія	6	11	2	6,33
Хлібобулочні вироби	Категорія	5	2	3	3,33
Заморожені продукти	Категорія	9	10	6	8,33
Кава, чай	Категорія	2	8	1	3,67
Чіпси, снеки	Категорія	5	14	3	7,33
Ковбаса і м'ясні делікатеси	Категорія	2	2	1	1,67
Дитяче харчування	Категорія	8	3	12	7,67
Японська кухня	Категорія	2	2	1	1,67
Кулінарія	Категорія	8	3	7	6,00
Товари для дітей	Категорія	2	6	8	5,33
Побутова хімія та непродуктові товари	Категорія	3	4	0	2,33
Гігієна і косметика	Категорія	4	6	1	3,67
Товари для дому	Категорія	3	2	1	2,00
Товари для тварин	Категорія	1	2	2	1,67
Текстильні вироби	Категорія	1	2	0	1,00
Сертифікати та платіжні картки	Категорія	3	2	1	2,00
Капцелярські товари	Категорія	1	0	0	0,33
Автоаксесуари	Підкатегорії	0	0	1	0,33
Будівництво, ремонт, інструменти	Підкатегорії	0	0	1	0,33
Галантерея	Підкатегорії	0	0	0	0,00
Господарські товари	Підкатегорії	0	0	0	0,00
Декор и інтер'єр	Підкатегорії	0	0	0	0,00

«1»	«2»	«3»	«4»	«5»	«6»
Дитячі іграшки	Підкатегорії	0	0	0	0,00
Краса и здоров'я	Підкатегорії	0	0	0	0,00
Мультимедія и аксесуари	Підкатегорії	0	0	0	0,00
Одяг, взуття та аксесуари	Підкатегорії	0	0	0	0,00
Освітлення	Підкатегорії	0	0	0	0,00
Посуд и аксесуари для кухні	Підкатегорії	0	0	0	0,00
Сад та город	Підкатегорії	0	0	0	0,00
Текстиль для дому	Підкатегорії	0	0	0	0,00
Дивитись всі товари категорії	Підкатегорії	0	1	0	0,33
Горіхи, сухофрукти	Підкатегорії	0	0	1	0,33
Гриби	Підкатегорії	1	1	0	0,67
Зелень	Підкатегорії	0	0	0	0,00
Овочі	Підкатегорії	0	0	0	0,00
Соління, салати	Підкатегорії	1	0	0	0,33
Фрукти, ягоди	Підкатегорії	1	0	0	0,33
Борошно	Підкатегорії	0	1	0	0,33
Бутибродні масла, закуски, пасти	Підкатегорії	0	0	0	0,00
Джем, паства, мед	Підкатегорії	0	0	0	0,00
Діабетична та дієтична продукція	Підкатегорії	0	0	0	0,00
Їжа швидкого приготування	Підкатегорії	0	0	0	0,00
Крупи	Підкатегорії	0	0	0	0,00
М'ясні та рибні консерви	Підкатегорії	0	0	0	0,00
Майонез	Підкатегорії	0	0	0	0,00
Макаронні вироби	Підкатегорії	0	0	0	0,00
Овочева та фруктовая консервація	Підкатегорії	0	0	0	0,00
Олія та Оцет	Підкатегорії	0	0	0	0,00
Приправи та маринади	Підкатегорії	0	0	0	0,00
Соуси, кетчупи	Підкатегорії	0	0	0	0,00
Сухі сніданки	Підкатегорії	0	0	0	0,00
Товари для випічки	Підкатегорії	0	0	0	0,00
Цукор	Підкатегорії	0	0	0	0,00
Вершки	Підкатегорії	0	0	0	0,00
Дитяча молочна продукція	Підкатегорії	0	0	0	0,00
Згущене молоко	Підкатегорії	0	0	0	0,00
Йогурти	Підкатегорії	0	0	0	0,00
Кисломолочні напої	Підкатегорії	0	0	0	0,00
Масло і маргарин	Підкатегорії	0	0	0	0,00
Молоко	Підкатегорії	0	0	2	0,67

«1»	«2»	«3»	«4»	«5»	«6»
Молочні десерти	Підкатегорії	0	0	0	0,00
Сир кисломолочний	Підкатегорії	0	0	0	0,00
Сирки глазуровані	Підкатегорії	0	0	0	0,00
Сметана	Підкатегорії	0	0	0	0,00
Яйця курячі/перепелині	Підкатегорії	0	0	0	0,00
Вермут	Підкатегорії	0	0	0	0,00
Вино	Підкатегорії	0	0	0	0,00
Горілка	Підкатегорії	0	0	0	0,00
Ігристе вино	Підкатегорії	0	0	0	0,00
Інший алкоголь	Підкатегорії	0	0	0	0,00
Коньяк	Підкатегорії	0	0	0	0,00
Лікери, настоянки, бальзами	Підкатегорії	0	0	0	0,00
Пиво	Підкатегорії	0	0	0	0,00
Слабоалкогольні напої	Підкатегорії	0	0	0	0,00
Дитяче шампанське	Підкатегорії	0	0	0	0,00
Енергетичні	Підкатегорії	0	0	0	0,00
Квас	Підкатегорії	0	0	0	0,00
Мінеральна і питна вода	Підкатегорії	0	0	0	0,00
Напої	Підкатегорії	0	0	0	0,00
Соки, нектари	Підкатегорії	0	0	0	0,00
М'ясні напівфабрикати	Підкатегорії	0	0	0	0,00
М'ясо охолоджене	Підкатегорії	0	0	0	0,00
Сири м'які	Підкатегорії	0	1	0	0,33
Сири плавлені	Підкатегорії	0	0	0	0,00
Сири тверді	Підкатегорії	0	0	0	0,00
Батончики	Підкатегорії	15	8	0	7,67
Бісквіти та десерти	Підкатегорії	3	1	0	1,33
Вафлі	Підкатегорії	0	0	0	0,00
Драже	Підкатегорії	0	0	0	0,00
Жувальні гумки	Підкатегорії	1	0	0	0,33
Крекери	Підкатегорії	0	0	0	0,00
Круасани та зефір	Підкатегорії	0	0	0	0,00
Новорічні подарунки	Підкатегорії	0	0	0	0,00
Пасти шоколадно-горіхові	Підкатегорії	0	0	0	0,00
Печиво, пряники	Підкатегорії	0	0	0	0,00
Східні солодощі	Підкатегорії	0	0	0	0,00
Торти і тістечка	Підкатегорії	0	0	0	0,00
Цукерки	Підкатегорії	0	0	0	0,00

«1»	«2»	«3»	«4»	«5»	«6»
Шоколад	Підкатегорії	0	0	0	0,00
Ікра та рибна паста	Підкатегорії	9	7	1	5,67
Крабові палички	Підкатегорії	1	1	1	1,00
Морепродукти	Підкатегорії	0	0	0	0,00
Пресерви	Підкатегорії	0	0	0	0,00
Риба	Підкатегорії	0	0	0	0,00
Рибна нарізка	Підкатегорії	0	0	0	0,00
Батон	Підкатегорії	2	0	3	1,67
Булочні, листкові вироби	Підкатегорії	1	0	0	0,33
Гаряча випічка	Підкатегорії	0	0	0	0,00
Лаваш	Підкатегорії	0	0	0	0,00
Сушка та сухарі	Підкатегорії	0	0	0	0,00
Хліб	Підкатегорії	0	0	0	0,00
Хлібці	Підкатегорії	0	0	0	0,00
Вареники, Пельмені, Млинці	Підкатегорії	3	8	2	4,33
Морозиво	Підкатегорії	2	0	0	0,67
Напівфабрикати	Підкатегорії	1	0	0	0,33
Овочі та фрукти свіжозаморожені	Підкатегорії	0	0	1	0,33
Тісто, Випічка, Піца	Підкатегорії	0	2	0	0,67
Кава, какао	Підкатегорії	0	2	0	0,67
Чай	Підкатегорії	0	1	0	0,33
Снеки	Підкатегорії	2	6	3	3,67
Чипси	Підкатегорії	1	3	0	1,33
Інші м'ясні вироби	Підкатегорії	1	1	1	1,00
Ковбаса	Підкатегорії	0	0	0	0,00
Нарізки та делікатеси	Підкатегорії	0	0	0	0,00
Сосиски, сардельки	Підкатегорії	0	0	0	0,00
Вода	Підкатегорії	6	3	3	4,00
Дитяче печиво	Підкатегорії	1	0	4	1,67
Каші	Підкатегорії	1	0	6	2,33
Молочні суміші	Підкатегорії	0	0	0	0,00
Пюре	Підкатегорії	0	0	0	0,00
Соки, чай	Підкатегорії	0	0	0	0,00
Васабі, Імбир, Оцет	Підкатегорії	1	0	1	0,67
Норі і Паніровка	Підкатегорії	1	0	0	0,33
Бутерброди, бургери	Підкатегорії	10	2	3	5,00
Піца	Підкатегорії	2	0	1	1,00
Підгузки, пелюшки	Підкатегорії	1	3	1	1,67

«1»	«2»	«3»	«4»	«5»	«6»
Серветки вологі	Підкатегорії	0	1	0	0,33
Засоби для миття посуду	Підкатегорії	2	1	0	1,00
Засоби для чищення	Підкатегорії	1	0	0	0,33
Освіжувачі повітря	Підкатегорії	0	0	0	0,00
Пральні порошки та засоби для прання	Підкатегорії	0	0	0	0,00
Ватні палички/диски	Підкатегорії	2	2	1	1,67
Гелі для душу	Підкатегорії	0	0	0	0,00
Гігієна ротової порожнини	Підкатегорії	1	0	0	0,33
Гігієнічні прокладки/тампони	Підкатегорії	0	2	0	0,67
Дезодоранти	Підкатегорії	0	1	0	0,33
Догляд за волоссям	Підкатегорії	0	0	0	0,00
Догляд за тілом	Підкатегорії	0	0	0	0,00
Засоби для гоління	Підкатегорії	0	0	0	0,00
Мило	Підкатегорії	0	0	0	0,00
Підгузки для дорослих	Підкатегорії	0	0	0	0,00
Презервативи	Підкатегорії	0	0	0	0,00
Серветки, рушники	Підкатегорії	0	0	0	0,00
Туалетний папір	Підкатегорії	0	0	0	0,00
Універсальні засоби	Підкатегорії	0	0	0	0,00
Шампуні	Підкатегорії	0	0	0	0,00
Батарейки	Підкатегорії	1	0	1	0,67
Електролампи	Підкатегорії	0	0	0	0,00
Засоби по догляду за взуттям	Підкатегорії	0	0	0	0,00
Канцтовари	Підкатегорії	0	0	0	0,00
Колготи, гольфи, шкарпетки	Підкатегорії	0	0	0	0,00
Медикаменти	Підкатегорії	1	0	0	0,33
Мобільний зв'язок	Підкатегорії	0	0	0	0,00
Одноразовий посуд	Підкатегорії	0	0	0	0,00
Пакети для покупок	Підкатегорії	0	0	0	0,00
Пакети для сміття	Підкатегорії	0	0	0	0,00
Подарункові набори	Підкатегорії	0	0	0	0,00
Сірники, запальнички	Підкатегорії	0	0	0	0,00
Товари для випічки і упаковки їжі	Підкатегорії	0	0	0	0,00
Товари для кухні	Підкатегорії	0	0	0	0,00
Товари для прибирання	Підкатегорії	1	0	0	0,33
Корм	Підкатегорії	0	0	1	0,33
Наповнювач	Підкатегорії	1	1	1	1,00

«1»	«2»	«3»	«4»	«5»	«6»
Сигарети	Підкатегорії	1	0	0	0,33
Стіки та електронні сигарети	Підкатегорії	0	0	0	0,00
Благодійна ініціатива	Підкатегорії	1	0	1	0,67
Карта АТБ	Підкатегорії	1	0	0	0,33
Допоміжне приладдя	Підкатегорії	0	0	0	0,00
Зошити, альбоми, блокноти	Підкатегорії	0	0	0	0,00
Папір і картон	Підкатегорії	0	0	0	0,00
Папки, файли, обкладинки	Підкатегорії	0	0	0	0,00
Ручки, олівці, фломастери	Підкатегорії	1	0	0	0,33
Фарби, пластилін, клей	Підкатегорії	0	0	0	0,00
Всього переглядів:		163	158	100	140,33
Всього не переглянутих:		137	146	152	118

Головний компонент коду веб-додачу з методом багаторугого бандита

store.component.html

```

<section class="store-component">
  <div class="store-container">
    <div class="store-category-list" [ngClass]="{ 'collapsed': isCollapsed }">
      <div class="store-block">
        <div class="store-title">
          <h1 class="store-category-title">{{ variantTitle }}</h1>
        </div>
        <div class="store-list-wrapper">
          <ul class="store-list-block" [ngClass]="variant">
            <li class="store-list" *ngFor="let category of visibleCategories" (click)="toggleSubcategory(category)">
              <a class="store-list-link">
                <ng-container [ngSwitch]="variant">
                  <ng-container *ngSwitchCase="A">
                    <img [src]="category.svg" alt="{{ category.title }}" width="24" height="24" class="store-list-img" />
                    <span class="store-list-text">{{ category.title }}</span>
                  </ng-container>
                  <ng-container *ngSwitchCase="B">
                    <span class="store-list-text">{{ category.title }}</span>
                  </ng-container>
                  <ng-container *ngSwitchCase="C">
                    <img [src]="category.svg" alt="{{ category.title }}" width="24" height="24" class="store-list-img" />
                  </ng-container>
                </ng-container>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
</div>

<div class="store-btn" *ngIf="variant !== 'B'">
  <button class="toggle-btn" (click)="toggleCategories()">
    {{ isCollapsed ? '' : '' }}
  </button>
</div>
</div>

<div class="subcategory-menu" [ngClass]="{'show': isSubcategoryVisible}">
  <div class="subcategory-btn-close">
    <button class="close-btn" (click)="hideSubcategory()">x</button>
  </div>
  <div class="subcategory-container">
    <h3>{{ currentCategory?.title }}</h3>
    <ul class="submenu">
      <li class="submenu-item" *ngFor="let subcategory of currentCategory?.subcategories">
        <a class="submenu-link"
          (click)="onSubcategoryClick(subcategory)"
          style="cursor: pointer;">
          <span>{{ subcategory.title }}</span>
          <img class="submenu-img" [src]="subcategory.svg" alt=" {{ subcategory.title }}">
        </a>
      </li>
    </ul>
  </div>
</div>

<div class="store-advertising-box">
  <div class="store-advertising-contain">
    <div class="store-advertising-img-block">
      <img *ngFor="let img of images; let i = index"
        [src]="img"
        [alt]="Image ' + i'"
        class="store-advertising-img"
        [ngStyle]="{'display': currentIndex === i ? 'block' : 'none'}">
    </div>
  </div>
  <div class="store-advertising-dots">
    <span *ngFor="let dot of images; let i = index"
      class="dot"
      [class.active]="currentIndex === i"
      (click)="goToSlide(i)">
    </span>
  </div>
</div>

<div *ngIf="currentSubcategory" class="subcategory-modal">
  <button (click)="currentSubcategory = null" class="close-subcategory-btn"></button>
  <div class="subcategory-modal-content">

```

```

<img [src]="currentSubcategory.svg" alt="{{ currentSubcategory.title }}" style="max-width: 200px; height: 200px; margin-bottom: 60px;" />
<p *ngIf="currentSubcategory.description">{{ currentSubcategory.description }}</p>
<h2>{{ currentSubcategory.title }}</h2>
<div>
  <div style="font-size: 25px; text-decoration: none; color: #121212; font-weight: bold;">
    {{ randomPrice }}
    <span style="font-size: 16px;"></span>
  </div>
</div>
</div>
</div>
</div>
</div>
</section>

```

store.component.ts

```

import { Component, OnDestroy, OnInit, inject } from '@angular/core';
import { Firestore, collection, doc, setDoc, updateDoc, getDocs } from '@angular/fire/firestore';
import { DecimalPipe, NgClass, NgForOf, NgIf, NgStyle, NgSwitch, NgSwitchCase } from '@angular/common';

```

```

type Variant = 'A' | 'B' | 'C';

```

```

interface VariantStats {
  views: number;
  clicks: number;
}

```

```

@Component({
  selector: 'app-store',
  templateUrl: './store.component.html',
  styleUrls: ['./store.component.css'],
  imports: [
    NgForOf,
    NgStyle,
    NgClass,
    NgIf,
    NgSwitchCase,
    NgSwitch,
  ],
  standalone: true
})

```

```

export class StoreComponent implements OnInit, OnDestroy {
  private firestore: Firestore = inject(Firestore);
  private userDocRef: any = null;

  public categories: any[] = [];
  public currentCategory: any;
  public isSubcategoryVisible = false;
  public isCollapsed = true;

```

```

public currentIndex = 0;
public currentSubcategory: any = null;
randomPrice: number = 0;

images: string[] = [
  'assets/imgs/advertising/1.png',
  'assets/imgs/advertising/2.png',
  'assets/imgs/advertising/3.png',
  'assets/imgs/advertising/5.png',
  'assets/imgs/advertising/6.png'
];

variant: Variant = 'A';
defaultProductImage = 'assets/imgs/products/default.jpg';

stats: VariantStats = { views: 0, clicks: 0 };
actionLogs: Array<{ timestamp: string; eventType: string; message?: string; timeOnPageSeconds: number }> = [];

private startTime!: number;

ngOnInit(): void {
  this.variant = this.chooseVariant();

  this.stats.views = 1;

  this.startTime = Date.now();

  this.loadCategories();

  this.createUserDocument()
    .then(() => {
      this.logAction('variant-chosen', `Variant ${this.variant} selected`);
      this.syncAnalytics();
    });

  setInterval(() => this.nextSlide(), 5000);

  if (this.variant === 'B') {
    this.isCollapsed = false;
  }
  console.log('!', this.variant);
}

async loadCategories(): Promise<void> {
  try {
    const categoryCollection = collection(this.firestore, 'categories');
    const snapshot = await getDocs(categoryCollection);

    this.categories = snapshot.docs.map(doc => doc.data());
  } catch (error) {
    console.error('! Firestore:', error);
  }
}

```

```

    }
  }

  ngOnDestroy(): void {
    const timeSpent = Math.round((Date.now() - this.startTime) / 1000);

    this.logAction('time-on-page', `Time spent: ${timeSpent} seconds`);
    this.syncAnalytics(true);
  }

  chooseVariant(): Variant {
    const epsilon = 0.1;
    const random = Math.random();

    if (random < epsilon) {
      const variants: Variant[] = ['A', 'B', 'C'];

      return variants[Math.floor(Math.random() * variants.length)];
    } else {
      //
      return (['A', 'B', 'C'] as Variant[])[Math.floor(Math.random() * 3)];
    }
  }

  recordClick(): void {
    this.stats.clicks++;
    this.logAction('click', 'User clicked');
    this.syncAnalytics();
  }

  trackClick(message: string): void {
    this.recordClick();
    this.logAction('click-detail', message);
    this.syncAnalytics();
  }

  async toggleSubcategory(category: any): Promise<void> {
    this.recordClick();
    this.logAction('click-subcategory', ` ${category.title}`);

    if (category.title === "" || category.title === " ") {
      this.logAction('navigate', ` ${category.title}`);
    }

    try {
      await this.syncAnalytics();
    } catch (err) {
      console.error(' ', err);
    }

    setTimeout(() => {
      window.location.href = category.link;
    });
  }
}

```

```

    }, 200);
  } else {
    if (this.currentCategory === category && this.isSubcategoryVisible) {
      this.hideSubcategory();
    } else {
      this.currentCategory = category;
      this.isSubcategoryVisible = true;
    }
    await this.syncAnalytics();
  }
}

```

```

async onSubcategoryClick(subcategory: any): Promise<void> {
  this.recordClick();
  this.logAction('click-subcategory-item', `: ${subcategory.title}`);

  await this.syncAnalytics();

  this.currentSubcategory = subcategory;
  this.isSubcategoryVisible = false;
  this.randomPrice = Math.floor(Math.random() * 101);
}

```

```

private async sendAdditionalLog(data: any) {
  if (!this.userDocRef) return;
  try {
    await updateDoc(this.userDocRef, {
      actionLogs: [...this.actionLogs, data]
    });
    console.log(' ');
  } catch (error) {
    console.error(' ', error);
  }
}

```

```

hideSubcategory(): void {
  this.isSubcategoryVisible = false;
  this.currentCategory = null;
}

```

```

nextSlide(): void {
  this.currentIndex = (this.currentIndex + 1) % this.images.length;
}

```

```

goToSlide(index: number): void {
  this.currentIndex = index;
}

```

```

toggleCategories(): void {
  this.isCollapsed = !this.isCollapsed;
}

```

```

}

get visibleCategories() {
  return this.isCollapsed
    ? this.categories.slice(0, Math.ceil(this.categories.length / 2))
    : this.categories;
}

get variantTitle(): string {
  switch (this.variant) {
    case 'A': return ' A';
    case 'B': return ' B';
    case 'C': return ' C';
    default: return '';
  }
}

private async createUserDocument(): Promise<void> {
  const analyticsCollection = collection(this.firestore, 'analytics');
  const newDocRef = doc(analyticsCollection);

  this.userDocRef = newDocRef;

  const initialData = {
    variant: this.variant,
    stats: this.stats,
    timeOnPageSeconds: 0,
    actionLogs: []
  };

  await setDoc(newDocRef, initialData);
}

private logAction(eventType: string, message?: string) {
  const timeOnPage = Math.round((Date.now() - this.startTime) / 1000);
  this.actionLogs.push({
    timestamp: new Date().toISOString(),
    eventType,
    message,
    timeOnPageSeconds: timeOnPage
  });
}

private async syncAnalytics(isFinal: boolean = false): Promise<void> {
  if (!this.userDocRef) return;

  const timeOnPage = Math.round((Date.now() - this.startTime) / 1000);

  try {
    await updateDoc(this.userDocRef, {
      variant: this.variant,

```

```

    stats: this.stats,
    timeOnPageSeconds: timeOnPage,
    actionLogs: this.actionLogs
  });
  if (isFinal) console.log(' ');
} catch (error) {
  console.error(' ', error);
}
}
}

```

store.component.css

```

store-component {
  width: 100%;
  height: 100%;
}

```

```

store-container {
  display: flex;
  justify-content: space-around;
  align-items: center;
  background: #3e3f43;
}

```

```

store-category-list {
  display: flex;
  flex-direction: column;
  align-items: center;
  height: auto;
  padding: 20px;
  width: 500px;
  background-color: #2f3138,
  transition: max-height 0.4s cubic-bezier(0.4, 0, 0.2, 1), width 0.6s cubic-bezier(0.4, 0, 0.2, 1), opacity 0.4s ease-in-out;
  overflow: hidden;
  opacity: 1;
  max-height: 250px;
}

```

```

store-category-list.collapsed {
  max-height: 100%;
  width: 600px;
  opacity: 0.9;
}

```

```

.store-category-list:not (collapsed) {
  max-height: 1100px;
  opacity: 1;
  width: 1000px;
}

```

```
}
store-block {
  display: flex;
  flex-direction: column;
  justify-content: center;
  height: 100%;
  padding: 20px;
  box-shadow: rgba(0, 0, 0, 0.1) 2px 4px 12px;
}

store-title {
  display: flex;
  justify-content: center;
  align-items: center;
  margin-bottom: 15px;
  box-shadow: rgba(0, 0, 0, 0.1) 2px 4px 12px;
}

store-category-title {
  font-size: 25px;
  color: white;
}

store-btn {
  display: flex;
  justify-content: center;
  align-items: center;
  margin-top: 15px;
}

store-list-wrapper {
  overflow: hidden;
  box-shadow: rgba(0, 0, 0, 0.1) 2px 4px 12px;
}

store-list-block {
  display: grid;
  grid-template-columns: 1fr;
  gap: 5px;
}

.store-category-list:not(collapsed) store-list-block {
  grid-template-columns: 1fr 1fr;
  opacity: 1;
  transition: all 1s ease, transform 1s cubic-bezier(0.4, 0, 0.2, 1), grid-template-columns 1s ease-in-out;
}

store-list {
  display: flex;
  align-items: center;
  width: 300px;
}
```

```
height: 40px;
box-shadow: rgba(0, 0, 0, 0.1) 2px 4px 12px;
transition: transform 0.5s cubic-bezier(0.4, 0, 0.2, 1), opacity 0.5s ease-in-out;
cursor: pointer;
}
```

```
store-list:hover {
  transform: translateX(5px) scale(1.05);
  opacity: 0.9;
}
```

```
store-list-link {
  display: flex;
  align-items: center;
  width: 100%;
  margin-left: 15px;
}
```

```
store-list-img {
  margin-right: 10px;
}
```

```
store-list-text {
  color: #FFFFFF;
}
```

```
toggle-btn {
  margin-top: 20px;
  padding: 12px 24px;
  font-size: 16px;
  font-weight: bold;
  background: linear-gradient(to bottom, #ff5f5f, #a5251d);
  color: white;
  border: none;
  cursor: pointer;
  transition: background 0.3s ease;
}
```

```
toggle-btn:hover {
  background: linear-gradient(to bottom, #ff7f7f, #c53028);
}
```

```
store-category-list {
  width: 250px;
  transition: width 0.3s ease-in-out;
}
```

```
store-category-list.collapsed {
  width: 400px;
}
```

```
subcategory-menu {  
  display: none;  
  background-color: #f8f8f8;  
  padding: 30px;  
  border-radius: 5px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  width: calc(100% - 40px);  
  max-width: 1150px;  
  max-height: 1100px;  
  height: 725px;  
  overflow-y: auto;  
  opacity: 0;  
  transition: opacity 0.3s ease-in-out;  
  position: absolute;  
  left: 680px;  
  z-index: 2;  
}
```

```
subcategory-menu.show {  
  display: block;  
  opacity: 1;  
  scrollbar-color: #ff0000 white;  
  scrollbar-gutter: stable;  
  overflow-y: auto;  
  scrollbar-width: thin;  
}
```

```
subcategory-btn-close {  
  display: flex;  
  justify-content: end;  
  align-items: end;  
  width: 100%;  
}
```

```
close-btn {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  background: none;  
  border: none;  
  color: #333;  
  cursor: pointer;  
  font-size: 25px;  
  width: 25px;  
  height: 25px;  
}
```

```
.close-btn:hover {  
  color: #ff0000;  
}
```

```
subcategory-container h3 {
  margin-bottom: 30px;
}

submenu {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  align-items: center;
  gap: 15px;
}

submenu-item {
  display: flex;
  flex-direction: column;
  align-items: center;
  height: 300px;
  width: 250px;
  padding: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  border-radius: 15px;
  background: white;
}

submenu-item span {
  display: flex;
  justify-content: start;
  flex-wrap: wrap;
  align-items: center;
  width: 100%;
  height: 75px;
  margin-bottom: 15px;
  font-size: 14px;
  font-weight: 500;
  line-height: 1.15;
}

submenu-item:last-child {
  background: #6ab040;
}

submenu-link {
  display: flex;
  justify-content: center;
  flex-direction: column;
  align-items: center;
  height: auto;
  padding: 10px;
  border-radius: 5px;
  width: 100%;
  cursor: pointer;
}
```

```
submenu-img {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  width: 100%;  
  height: 150px;  
  overflow: hidden;  
}
```

```
submenu-img img {  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
}
```

```
store-advertising-box {  
  display: flex;  
  flex-direction: column;  
  width: 100%;  
  height: 784px;  
  z-index: 1;  
}
```

```
store-advertising-contain {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  width: 100%;  
  height: 100%;  
}
```

```
store-advertising-img-block {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  width: 100%;  
  position: relative;  
  overflow: hidden;  
}
```

```
store-advertising-img {  
  width: 100%;  
  height: auto;  
  display: none;  
}
```

```
store-advertising-dots {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  padding: 10px;  
}
```

```
.dot {  
  height: 15px;  
  width: 15px;  
  margin: 0 5px;  
  background: white;  
  border-radius: 50%;  
  display: inline-block;  
  transition: background-color 0.3s ease;  
  cursor: pointer;  
}
```

```
.dot.active {  
  border: 3px solid #e53b30;  
  background: 0 0;  
}
```

```
store-list-block.B .store-list-link {  
  justify-content: center;  
  align-items: center;  
  text-align: center;  
  padding-left: 10px;  
  font-size: 14px;  
  font-weight: bold;  
  color: white;  
}
```

```
store-list-block.C {  
  padding: 5px;  
}
```

```
store-list-block.C .store-list-link {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  padding: 0 10px;  
}
```

```
store-list-block.C .store-list-link img {  
  height: 45px;  
  width: 50px;  
}
```

```
subcategory-modal {  
  background: #fff;  
  border: 1px solid #ccc;  
  padding: 20px;  
  margin-top: 20px;  
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
  border-radius: 8px;  
  width: 1000px;  
  height: 700px;
```

```
position: absolute;
left: 682px;
z-index: 2;
}

close-subcategory-btn {
background-color: #e74c3c;
color: white;
border: none;
padding: 8px 16px;
font-size: 14px;
border-radius: 4px;
cursor: pointer;
float: right;
}

subcategory-modal-content {
display: flex;
flex-direction: column;
align-items: center;
height: 300px;
width: 250px;
padding: 10px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
border-radius: 15px;
background: white;
}

.close-subcategory-btn:hover {
background-color: #c0392b;
}

@media (max-width: 768px) {
store-category-list {
width: 100%;
max-height: 500px;
}

store-list-block {
grid-template-columns: 1fr;
}

store-advertising-box {
height: auto;
}
}
```

Відгук
на кваліфікаційну роботу бакалавра
студента групи 124 – 21 – 2 Бублика Кирила Олеговича
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: «Аналіз та покращення інтерфейсу вибору товарів із використанням множинного тестування»

Обсяг кваліфікаційної роботи 72 стор.

Мета кваліфікаційної роботи: розробка та покращення інтерфейсу вибору товарів на веб-сайті за допомогою множинного тестування для покращення зручності користування та ефективності покупок.

Актуальність теми: при створенні веб-сайтів часто дизайнери генерують відразу декілька ідей, класичний A/B тест не дозволяє їх втілити одночасно. Розроблений застосунок для проведення множинного тестування дозволяє вирішити цю проблему.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки 1) використано сучасний підхід множинного тестування – епсилон-жадібний бандит, що дозволяє оцінювати одночасно декілька ідей щодо покращення інтерфейсу; 2) проведено аналіз сучасної літератури, відмічено переваги та недоліки існуючих методів; 3) самостійно розроблено веб-сайт продуктового магазину квітів, проведено множинний тест та покращено інтерфейс вибору товарів за результатами аналізу.

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра. Оригінальність наукових рішень полягає в 1) в розробці застосунку для проведення множинного тестування інтерфейсу веб-сайту за алгоритмом епсилон-жадібного бандита; 2) розробці нового веб-сайту продуктового магазину на базі Angular та хмарної БД Firebase, на прикладі якого проведено множинний тест.

Практичне значення результатів кваліфікаційної роботи полягає в отриманні дійсних даних про переваги різних інтерфейсів за допомогою множинного тестування для підвищення зручності використання веб-сайту та сприянню покращення продажу товарів.

Висновки підтверджують можливість використання результатів роботи в вітчизняних та закордонних Інтернет-магазинах для аналізу ідей дизайнерів для поліпшення веб-сайтів та виявленню найкращої з них з використанням відгуків користувачів.

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 показати як алгоритм епслон-жадібного бандита дозволяє аналізувати та враховувати зміни в перевагах користувачів протягом часу експерименту.

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.