

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

БУРАГА ІВАН ЮРІЙОВИЧ

Допускається до захисту:

в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент

_____ О. В. Зелінська

«_____» _____ 2025 р.

**ВЕБДОДАТОК ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧА
СУПЕРМАРКЕТУ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Потапова Н. А., доцент кафедри
інформаційних технологій,
к. е. н., доцент

Оцінка: _____ / _____ / _____
(бали за шкалою ЄКТС / за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2025

АНОТАЦІЯ

Бурага І. Ю. Вебдодаток для аналізу вподобань споживача супермаркету. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2025.

У кваліфікаційній (бакалаврській) роботі досліджено теоретичні основи та практична розробка вебдодатку для аналізу вподобань споживача супермаркету. Ключовим аспектом розробки являється реалізація вебзастосунку, де функціонал дозволяє споживачам супермаркету здійснювати аналіз своїх вподобань завдяки опитуванню та переглядати його результат.

Ключові слова: вебдодаток, споживач, вподобання, алгоритм, Flask, HTML, SQLite.

53 с., 25 рис., 40 посилань.

ABSTRACT

Buraha I. Y. Web application for analysing supermarket consumer preferences. Specialty 122 «Computer Science», educational program «Computer Science». Vasyl Stus Donetsk National University, Vinnytsia 2025.

The qualification (bachelor's) work considers the theoretical foundations and practical development of a web application for analysing supermarket consumer preferences. The key aspect of the development is the implementation of a web application, where the functionality allows supermarket consumers to analyse their preferences through a survey and view its results.

Keywords: web application, consumer, preferences, algorithm, Flask, HTML, SQLite.

53 p., 25 figures., 40 sources.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБДОДАТКУ ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧІВ СУПЕРМАРКЕТУ	6
1.1 Концептуальні підходи та сутність використання технологій створення вебдодатків	6
1.2 Теоретичні основи технологій аналізу вподобань	11
1.3. Сучасні тенденції розробок вебдодатків аналізу вподобань споживачів.	17
РОЗДІЛ 2. ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБДОДАТКУ ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧІВ СУПЕРМАРКЕТУ	21
2.1 Аналіз інструментарію для створення вебдодатку аналізу вподобань	21
2.2 Функціональні вимоги та технології аналізу вподобань	23
2.3 Модель вебдодатку та база даних для забезпечення основних операцій вебдодатку	26
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ ВЕБДОДАТКУ ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧІВ СУПЕРМАРКЕТУ	29
3.1 Проєкт дизайну вебдодатку для аналізу вподобань	29
3.2 Програмна реалізація вебдодатку для аналізу вподобань	34
3.3 Оцінка функціональності роботи вебдодатку для аналізу вподобань споживачів супермаркету	42
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	51
ДОДАТКИ.....	54

ВСТУП

В сучасну епоху розвитку вебтехнологій, супермаркети намагаються оптимізувати взаємодію між споживачами. Однією з головних цілей магазинів є персоналізація досвіду користувача та аналізу їх вподобань. На жаль не існує такого вебдодатку, наявні тільки загальні інструменти, які не дуже зручні середньому споживачу. У зв'язку з цим створення вебдодатку для аналізу вподобань споживача супермаркету стане гарним інструментом для того, щоб дізнатися про товари, які користувачі ще для себе не відкрили, на основі вибраних характеристик.

Сучасний споживач прагне не тільки переглядати доступний асортимент товарів та вибирати самому, а й отримати зручний сервіс, який проаналізує його вподобання та запропонує перелік товарів, базуючись на відправлених відповідях. Тому вебдодаток для аналізу вподобань споживача супермаркету повинен вирішити цю проблему.

Метою бакалаврської роботи є розробка вебдодатку для аналізу вподобань споживача супермаркету, функціонал якого надасть користувачу змогу зробити вибір товарів, який орієнтується на власні вподобання, та отримувати додаткові пропозиції.

Об'єктом дослідження визначено процес розробки вебдодатку, з урахуванням його можливостей та функціоналу, які спрямовані на врахування вподобань споживача супермаркету.

Предметом дослідження є моделі, методи та інструменти, які застосовуються для розробки вебдодатку, збору вподобань, обробки та створення персональних пропозицій для споживача.

Для досягнення поставленої мети поставлено наступні *завдання*:

1. Визначити основні теоретичні положення та концептуальні підходи в технологічних рішеннях розробки сучасного вебдодатку.
2. Обґрунтувати вибір інструментів розробки вебдодатку для аналізу вподобань споживача супермаркету.

3. Окреслити вимоги та механізм аналізу вподобань, а також інших функціональних можливостей вебдодатку.

4. Розробити структуру бази даних для забезпечення операцій зберігання, обробки та аналізу даних по визначенню вподобань споживача супермаркету.

5. Розробити дизайн та програмно реалізувати вебдодаток для аналізу вподобань споживача супермаркету.

6. Провести перевірку працездатності вебдодатку та оцінити функціональність його роботи.

Основний зміст бакалаврської роботи викладено в 3 розділах. У першому розділі проведений аналіз теоретичних основ та сучасних тенденцій розробки вебдодатку. У другому розділі розглянуті інструменти, які використовуються для розробки додатку, його модель та база даних для операцій з даним. У третьому розділі розглянуто складові програмної реалізації клієнтської та серверної частини вебдодатку, а також способи взаємодії з ним користувача.

Практична цінність даної роботи полягає у створенні інтерактивного вебдодатку, який допоможе споживачу краще дізнатися про свої вподобання та здійснити обґрунтований вибір товару. Цей вебдодаток буде слугувати користувачу, як додатковий інструмент для визначення майбутніх покупок під час походу до супермаркетів.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБДОДАТКУ ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧІВ СУПЕРМАРКЕТУ

1.1 Концептуальні підходи та сутність використання технологій створення вебдодатків

У сучасному світі вебдодатки стали однією з незамінних складовими нашого життя, адже користувачі кожного дня відвідують їх, а сам вебдодаток – це комп'ютерна програма, доступ до якої здійснюється через веб-браузер через Інтернет або інтрамережу. Він кодується такими мовами, як HTML, JavaScript або Java, і для його виконання потрібен веб-браузер [1]. У кожної людини своя мета перегляду сайту: перегляд новин, придбання товарів, перегляду відео, спілкування з друзями, тощо.

Вебдодатки працюють не так як звичайні додатки, а саме сайти роблять по моделі “клієнт-сервер”, де клієнтом є браузер, який користувач використовує для перегляду, а логіка та зберігання даних реалізовані на серверній частині. Така модель дозволяє розробникам легко та ефективно реалізовувати або змінювати функціонал без залучення кінцевого користувача. [2]

Упродовж розвитку Інтернету, вебдодатки також зазнали суттєвих змін. Умовно їх можна поділити на декілька типів: MPA (односторінкові додатки), SPA (багаторічкові додатки), PWA (Прогресивні вебдодатки).

MPA, або Multi Page Application, вважають, як класичний підхід, який був популярний у розробників починаючи з кінця 90-х років. У цьому типі кожен клік користувача на інтерактивне посилання на вебсторінці, що веде на нову сторінку призводить до завантаження нової сторінки у браузері. Сам HTML-код генерується на стороні сервера та передається користувачу у вигляді повної нової сторінки [3]. Найпопулярнішим прикладом є Wikipedia, тому що, коли людина клікає на нову статтю, то сторінка повністю перезавантажується. Ще у цього вебдодатку повністю відсутні Javascript-фреймворки. Сайт до сих пір використовує просту архітектуру PHP та базу даних MySQL.

SPA, або Single Page Application, є вже більш сучаснішою версією, ніж MPA. Через те, що розвинувся Javascript та появи його фреймворків спочатку Angular, який вийшов у 2010 році, а потім React, що дозволило людям перейти на SPA. У такій моделі вся вебсторінка завантажувалась один раз, а далі працювала без перезавантаження. Такий підхід дозволив користувачам взаємодіяти з вебдодатками швидше, зменшив різні затримки та покращив загальне враження користувачів [4]. Популярними прикладами вважаються: Gmail, Google Maps, Facebook, тощо.

PWA, або Progressive Web Application, є новіший тип вебдодатків, що поєднує у собі зручність доступу через браузер з функціональністю нативних мобільних додатків, який виник приблизно у 2015 році. Сама концепція була представлена компанією Google метою покращити досвід користувачів у мобільному вебсередовищі. Основними характеристиками PWA є: можливість роботи вебсторінки в офлайн-режимі, можливість додати значок сайту на головний екран телефону та push-повідомлення. Таким чином, PWA можна вважати, як гібрид мобільного додатку з моделлю SPA. [5]

Також не слід забувати, що завдяки інструментів, бібліотек та фреймворків створюються вебдодатки, а весь цей набір називають технологічний стек. Сам стек включає різні рівні, кожен з яких виконує свою відповідну роль у створенні сайту. Так, наприклад, якщо стек включає в собі набір інструментів для фронтенду (інтерфейс), бекенду (логіка сайту) та бази даних для зберігання різних даних, то його називають повний стек. [6]

Розглянемо основні рівні повного стеку: frontend, backend та базу даних, із прикладами відповідних технологій:

Frontend – це частина вебдодатку, яка відповідає за інтерфейс користувача і як він буде з нею взаємодіяти [6]. Для розробки використовують такі інструменти:

- HTML – це мова розмітки, яка використовується для створення та відображення вебдодатків і їхнього контенту у браузерах. Сам код пишеться за допомогою хештегів, таких як: `<html>`, `<head>`, `<body>`. Вони, в свою чергу,

визначають, де буде заголовок, абзац, різні посилання, зображення, тощо. [7]

- CSS - це каскадні таблиці стилів, які визначають зовнішній вигляд вебдодатку, тобто колір, шрифти та їх розмір, де і як будуть відображатися елементи, тощо. [8]

- JavaScript - мова програмування, що використовується в браузері, задля додавання інтерактивності до вебсторінки, а саме додавання різних анімацій, тощо. [9]

Backend – це серверна сторона вебдодатку, яка відповідає за логіку, обробку запитів, зберігання інформації, тощо, які захищені від очей користувача [6]. Для розробки використовують такі інструменти:

- Node.js – це середовище виконання запуску JavaScript-коду поза браузером. Його застосовують для виконання JavaScript на сервері, що дає змогу розробляти серверні та мережеві додатки. [10]

- Django – це фреймворк для створення вебдодатків, який написаний на мові Python. Django пропонує багато різних інструментів, які роблять розробку вебдодатків швидкою та легкою. [11]

- Ruby – це мова програмування, яка поєднує в собі простоту у використанні та потужність для створення складних вебдодатків. Для backend зачасто використовують фреймворк Rails, який є найпопулярнішим серед фреймворків Ruby. [12]

База даних – це сховище вебдодатку, яке відповідає за зберігання, доступ та організацію інформації, що надходять від користувача на сервер. Саме вона зберігає дані користувачів, такі як: логін, пароль, пошта і т.д; інформацію про замовлення, самі замовлення та товар, історію перегляду, тощо [13]. Для розробки такої структури зазвичай використовують такі популярні інструменти, як:

- MySQL – це система для управління реляційними базами даних, яка зберігає дані у вигляді пов'язаних таблиць. Оскільки вона використовує мову SQL для створення та модифікування даних, новому користувачу легше адаптуватися. [14]

- PostgreSQL - це система для управління реляційними базами даних, яка також зберігає дані у вигляді таблиць. Вона підтримує мову SQL, але на відміну від MySQL має розширену функціональність, що дозволяє працювати зі складними структурами даних, масивами та JSON, тощо. [15]

- MongoDB - це система для управління нереляційними базами даних, яка зберігає інформацію у вигляді документів формату BSON. Якщо у розробника проєкт з великим обсягом динамічної інформації, то MongoDB чудово підійде, через свої гнучкі схеми зберігання та можливості швидко змінювати структуру даних. [16]

Також не слід забувати, що сьогодні вебтехнології стрімко розвиваються через потребу людей у швидкості і стабільності під час користування продуктом й тому свою популярність набирають хмарні технології та методологія DevOps. Вони не лише полегшили спосіб сповиження ресурсів, а й оптимізували процес розробки вебдодатку, дозволяють запускати додатки на віддалених серверах і т.д. Всі ці складові дозволяють розробникам розроблювати їхні вебдодатки швидше і ефективніше.

Хмарні технології – це технології, які надають доступ до ресурсів таких, як: сховища, сервери, мережі, тощо через Інтернет. Головні переваги хмарних технологій можна вважати самообслуговування, тобто розробник сам може обрати потрібні йому ресурси; обсяг ресурсів, які можна налаштовувати по власному бажанню. Також існують різні типи хмар – це публічні (спільні хмарні технології, які можуть використовувати різні компанії), приватні (хмарні технології, які створюються для однієї компанії) та гібридні (поєднання двох попередніх). [17]

Найпоширенішими платформами хмарних сервісів вважають Amazon Web Service, Google Cloud Platform, Microsoft Azure. Саме вони надають найкращі інструменти для обробки запитів та даних, моніторингу стану, зберігання даних, тощо.

Важливими являються різні типи хмарних послуг через те, що вибір залежить від рівня керування ресурсами. Виділяють кілька моделей:

- IaaS (Infrastructure as a Service) – це модель, коли користувач отримує в оренду доступ до серверів, сховищ, мереж, тощо. [17]
- PaaS (Platform as a Service) – це модель, коли розробник отримує все необхідне для розробки, заходячи і оплачуючи послуги на платформі. [17]
- SaaS (Software as a Service) – це модель, коли користувач отримує готову програму через браузер, наприклад, щоб не скачувати Microsoft Word, можна знайти в Інтернеті Google Docs. [17]

DevOps – це методика розробки якогось додатку, яка поєднує процес розробки, тобто Development, разом із експлуатацією, тобто Operations. Головна мета DevOps полягає у суттєвому скороченні розробки додатку та кращої комунікації між людьми, хто пише код та людьми, які відповідають за операційну частину. [18]

Основними перевагами DevOps є прискорення випуску повноцінної версії додатку, покращення якості продукту та підвищення стабільності роботи додатку. Все це досягається завдяки автоматизації процесів та тісної взаємодії між командою. Через ці переваги багато команд, які працюють над великими та динамічними проектами, імплементують систему DevOps, що дозволяють їм виконувати поставлені задачі швидко й ефективно. [19]

Ключовими інструментами DevOps вважають:

- Git – це система контролю версій, яка має в своєму наборі інструментів такі функції: відстежування зміни в коді, повернення до минулих версій, можливість працювати декільком розробникам одночасно над одним проектом. [20]
- Grafana – інструмент для візуалізації даних, яка може приєднуватись до джерел з різними метриками. Grafana дозволяє на основі метрик створювати дашборди, графіки, звіти, тощо і команда буде бачити зміни в реальному часі. [21]
- Docker – це платформа, що збирає програму разом з її бібліотеками, фреймворками, тощо у спеціальний контейнер. Цей контейнер можна запускати неважливо на якому сервері і вона буде працювати стабільно та видавати такий

самий результат. [22]

- Jenkins – це інструмент, що дозволяє автоматизувати рутинний процес розробки. Його задача – це автоматичне тестування програми після кожної її зміни та встановлення нових версій програми на сервері. [23]

Також не слід забувати, що одним з важливих компонентів сучасного вебдодатку також є API, тобто інтерфейс, через який взаємодіють різні частини додатку.

API – це інтерфейс програмування застосунків, що дозволяє одним компонентам програми взаємодіяти з іншими [24]. Якщо брати контекст вебдодатків, то API використовується, щоб інтерфейс користувача, або frontend, взаємодівав з серверною частиною, або backend. Тобто запити відправляються до серверу та отримуються у структурованому вигляді у форматі JSON та XML.

Головним, або домінуючим архітектурним стилем для створення API у веб-розробці можна вважати REST, або Representational State Transfer. Свою популярність REST здобув завдяки своїй простоті в реалізації та можливостях у масштабуванні, що базуються на стандартних HTTP-запитах таких, як: GET, POST, PUT, DELETE. Також обмін даними між клієнтом-сервером відбуваються у форматі JSON. [25]

Забезпечення безпеки API є також важливим аспектом у створенні додатку, тому щоб захистити його використовують різні механізми автентифікації та авторизації, токени доступу, типу JWT, стандарт OAuth2.0 та інше. JWT — це компактний зашифрований текстовий токен, який передається разом із запитом до API [26]. Стандарт OAuth 2.0 — це протокол авторизації, який дозволяє третім сторонам отримувати обмежений доступ до ресурсів без розкриття пароля користувача [27].

1.2 Теоретичні основи технологій аналізу вподобань

Технології аналізу вподобань споживача є важливим елементом для розуміння, що саме цікавить споживача та його поведінки, коли він бачить різницю у цінах на один продукт у різних супермаркетах. Коли людина

використовує Інтернет, щоб щось зробити, вони залишають цифровий слід, а саме: історію переглядів товарів, саму взаємодію з відвідуваним сайтом, можливо оцінку товару, тощо, що в свою чергу дозволяє різним супермаркетам виявити якийсь спільний патерн і завдяки ньому зробити покращення проаналізувати і виявити якому товару, якого бренду і коли вони надають свою перевагу. [28]

Тому основою метою аналізу вподобань споживача можна вважати надання людині зручного та персоналізованого досвіду з вебдодатком, підвищення рівня лояльності та задоволення сервісом у користувача та оптимізація системи знижок і різних акційних пропозицій у влучні дати.

Джерела даних для аналізу вподобань споживача, умовно можна розділити на: first-party data(власні), third-party data(сторонні). Власні джерела, або first-party data – це термін, який відсилається на дані, які збираються на пряму з користувачів, які використовують додаток. Ці взаємодії відбуваються через різні точки доступу такі, як: вебдодатки, мобільні додатки, соціальні мережі компанії, різні опитування, тощо [29]. До first-party data зачасти входять такі типи даних:

- Демографічна інформація – це дані, які включають персональну інформацію про споживача, а саме: вік, стать, місце проживання, рівень освіти, тощо. [30]
- Поведінкова інформація – це дані, які відносяться до поведінки споживача під час його користування продуктом, а саме: час перебування на сторінках, його взаємодія з додатком і послухами, його привички під час перегляду, тощо. [30]
- Психографічна інформація – це дані, які описують особисті уподобання, інтереси, цінності, переконання, тощо. Ці дані допомагають компаніям зрозуміти споживача на більш новому рівні, що допомагає пропонувати цікаві йому товари. [30]

Сторонні джерела, або third-party data – це термін, який відсилається на дані, які спеціальні компанії збирають з різних джерел, щоб згрупувати їх задля подальшої продажі іншим. Зазвичай такі дані збираються з різни джерел усюди,

щоб була вибірка була більшою. До third-party data входять такі ж типи даних, як і до first-party data, але власні джерела орієнтовані більш на один продукт однієї компанії, коли в свою чергу сторонні джерела орієнтовані на широку аудиторію без прямої взаємодії з кінцевим користувачем. [29]

Для того, щоб аналіз вподобань споживача був ефективний, він повинен базуватися на якісному та продуктивному зборі інформації, детальній структуризації та в кінцевій обробці даних. Компанії та розробники використовують як і технічні, так і аналітичні методи, які охоплюють різні етапи аналізу, такі як: збір інформації, очищення інформації від шумів, перетворення та кінцевого збереження у відповідний формат. [31]

Спочатку йде збір, який є етапом отримання чи придбання даних про користувачів. Ця інформація отримується завдяки first-party data і third-party data. Власні джерела збирають інформацію про споживача з додатків, які компанії створюють, що дозволяють покращити додаток, рекомендувати продукт для кожного користувача індивідуально, тощо. В свою чергу third-party data збирає загальну інформацію зі всього Інтернету та групує її, щоб потім продати її. Часто компанії використовують технічні інструменти для збору інформації, такі як: різні форми опитування, API-інтеграції, Web scraping, тощо. [30]

Далі йде очищення інформації від шумів, яка допомагає очистити дані після збору. Тобто інформація, якою компанії оволодівають містять якійсь непотрібні, повністю неправильно або повторювані записи, що паплюжить чистоту аналізу. Зазвичай основними завданнями є видалення однакових записів, тобто дублікатів, запитів, що виходять за рамки дозволеного, застарілих даних, виправлення помилок, тощо. Після того, як дані успішно пройшли очистку, вони відправляються знову до набору даних, а інформація, яка не пройшла її – видаляється. [32]

Наступний етап – це перетворення. Цей пункт дозволяє зробити більш глибоку аналітичну обробку даних, щоб перетворити необроблені значення в ті, які будуть мати велике значення в аналітиці. Ці дії можуть включати щось просте, типу форматування усіх дат так, щоб виділялися дні тижня, яке свято

було в це день, і який час; або складне, типу створення нових полів даних, які об'єднують інформацію, що міститься в декількох раніше існуючих полях. В загалом, цей етап дозволяє збагати існуючі дані, надаючи їм додаткову інформацію, що дозволяє краще проводити аналітичні дії. [32]

Останнім етапом вважається збереження перетворених даних у відповідний формат. Інформацію, яка пройшла перетворення, зберігають у файлових форматах (CSV, JSON), в базах даних (MySQL, PostgreSQL, MongoDB і тому подібне) та в різних хмарних сховищах. Дані зберігають саме так, щоб було зручно їх використовувати в подальшому аналізі або легше передавати їх іншим компаніям під час продажу. [32]

Отже, важливо притримуватися цих етапів через те, що пропуск хоча б одного з них може призвести до спотворених даних, недостовірних результатів під час аналізу або збоїв у системі аналізу вподобань.

Аналіз вподобань споживача існує для того, щоб допомагати супермаркетам відстежувати поведінку покупця, що дозволить сказати які товари їм подобаються та які переглядають, коли вони купляють продукти та за скільки, тощо. Щоб спростити цей процес існують різні моделі, такі як: традиційні та сучасні моделі.

Традиційні моделі – це модель, яка базується на статистичних методах та правилах, закладений аналітиками вручну. Традиційні моделі були розроблені економістами та аналітиками, щоб зрозуміти, що саме цікавить споживача, що спонукає його придбати товар, виходячи з їхніх потреб, переглядів та бажань. До таких моделей відносять: асоціативні правила, аналіз часових рядів, RFM-аналіз, тощо.

Асоціативні правила – це метод аналізу даних, що дозволяє виявити зв'язки між різними товарами. У контексті аналізу вподобань споживача цей метод є цінним через те, що існує відомий алгоритм – Apriori, який використовується, наприклад, щоб виявити закономірності типу, “Якщо споживача зацікавив товар А, то ймовірніше всього його зацікавить товар Б”. Метриками цього методу є підтримка (support), достовірність (confidence) та

підйом (lift). [33]

Аналіз часових рядів – це метод статистичного аналізу, який застосовується до даних, що змінюються в часі та були зібрані у різні часові періоди. В загалом аналіз часових рядів допомагає прогнозувати продаж, планування акцій під час свят, тощо. [34]

RFM-аналіз – це метод, який базується на трьох метриках споживацької поведінки: Давність (Recency), Частота (Frequency) та Грошова цінність (Monetary Value). Давність відповідає за вимір часу, коли споживач востаннє здійснював покупку і чим менше цей показник, тим краще. Частота відображає, як часто людина взаємодіє з супермаркетом за певний період часу та загалом демонструє рівень лояльності клієнта. Грошова цінність визначає скільки споживач витрачає грошей, скільки він загалом витратив та розраховує фінансову цінність для компанії. [35]

Час не стоїть на місці, тому розробники постійно придумують підходи, які дозволяють автоматизувати процес аналізу, що дозволить пришвидшити та підняти точність прогнозів за допомогою машинного навчання та штучному інтелекту. Тому такі моделі кличуть сучасними.

Сучасні моделі – це методи, які базуються на алгоритмах машинного навчання, штучному інтелекті. Вони дозволяють прогнозувати майбутню поведінку користувача, адаптуватися під його потреби і все це у режимі реального часу. Їхнє застосування дає змогу системам швидко адаптуватися до змін у поведінці користувачів. Все це є важливим інструментарієм для супермаркетів, адже вони зможуть отримати розширені знання про поведінку споживачів, що допомагає підбирати персональний підхід для кожного з них. До популярних сучасних методів відносяться: кластеризація, аналіз послідовностей дій та колаборативна фільтрація.

Кластеризація – це метод, який дозволяє об'єднувати споживачів, які мають схожу поведінку, вподобання. За допомогою алгоритму K-means, розробники вебдодатків групують користувачів за їхніми переглядами, покупками, частотою відвідування різних сторінок, середнім часом перебування

на них, тощо. Це допомагає налаштовувати систему так, щоб цим групам під час перебування на сайтах рекомендувало різні пропозиції, акції, знижки. [36]

Аналіз послідовностей дій – це метод, який аналізує закономірність у перегляді, покупках та різних дій споживача. Наприклад, система спостерігає, що користувач переглядав, чи є по знижкам м'ясо та після цього переглянув спеції до нього, в наступний раз система порекомендує йому різні прянощі після перегляду м'ясних продуктів. Тому аналіз послідовностей дій є важливим, адже він враховує не лише факт взаємодії споживача зі сторінками, а й порядок, який він дотримувався.

Останнім в черзі є колаборативна фільтрація. Цей метод базується на теорії, що споживач з вподобаннями, які в минулому йому подобались, скоріш за все матиме подібні смаки в майбутньому. Колаборативній фільтрації не потрібні знання про товари, адже вона проводить аналіз на взаємодії користувача із продуктом. Цей метод більш походить на рекомендацію через те, що вона шукає споживачів зі схожими покупками та переглядами і може запропонувати якийсь товар з кошика іншого. [37]

У сучасних вебдодатках всі ці методи працюють як і поодиночці, так і разом для аналізу вподобань споживача, що дозволяє розробникам та самим супермаркетам формувати майбутні плани, персональні пропозиції та покращувати перебування споживачів.

Захист персональних даних споживача також є невід'ємною складовою процесу, адже у сучасних вебдодатках, у сховищах в яких знаходиться ця інформація є важливою, щоб зберегти довіру користувача та репутацію супермаркету. Щоб забезпечити надійне зберігання застосовують шифрування, систему авторизації, ідентифікатори замість особистої інформації, контроль за доступом до даних та аудит безпеки.

Шифрування існує для того, щоб захищати дані у разі несанкціонованого доступу до них. Інформація кодується так, що без відповідного ключа її неможливо було б прочитати. Окрім цього, застосовуються комплексні системи авторизації, серед яких двофакторна аутентифікація, що суттєво підвищує рівень

безпеки доступу від сторонніх осіб.

В свою чергу, ідентифікатори дозволяють проводити аналіз, не використовуючи особисту інформацію споживача напряду. Тобто ім'я та номер телефона користувача система замінює на унікальний числовий або символічний код. Навіть якщо витік коду стався, то особисту інформацію користувача неможливо встановити без знань, що цей числовий чи символічний код означають.

Контроль за доступом до даних – це коли кожен співробітник має рівень доступу, що дозволяє фільтрувати людей, які можуть бачити особисті дані споживачів. Ця система існує для того, щоб максимально нейтралізувати витік даних від співробітників, які нещодавно прийшли у компанію працювати.

Важливою складовою також є аудит безпеки, який включає в собі різні регулярні перевірки системи на вразливості. Мова йдеться про тестування на проникнення через різні “дірки”, оцінка самої системи та перевірка логів активності на несанкціоновані входи. Отже, аудит безпеки дозволяє своєчасно оновлювати безпеку та ремонтувати слабкі місця системи, якщо такі є.

Не слід забувати, якщо вебдодатки збирають персональну інформацію, то розробники повинні вказати чітку політику конфіденційності. Тобто має бути чітко зазначено, які дані збирає та зберігає, з якою метою, як довго вона буде зберігатися, чи буде вона передана комусь, тощо.

1.3 Сучасні тенденції розробок вебдодатків аналізу вподобань споживачів

Сьогодні тенденції розробки вебдодатків не стоять на місці і тому розробникам, які працюють на різні супермаркети потрібно встигати інтегрувати сучасні технології, щоб не відставати від інших. Вебдодатки, які орієнтовані на вподобання споживача повинні швидко адаптуватися, підлаштовуючись під новітні технології.

Одна з цих тенденцій є впровадження нейромережевого аналізу вподобань. Нейромережі дозволяють не тільки працювати за заданими правилами, а й

виявляти неочевидні закономірності в поведінці споживача. Її навчають на великих обсягах даних, що дозволяє бачити ці патерни, які вказують на потенційний інтерес до якоїсь категорії товарів. Щоб виявляти такі закономірності, нейромережа обробляє непрямі ознаки, до яких входять: перебування на товарах певної категорії більше часу, ніж на інших, повторне відвідування цих сторінок, запити в пошуковому полі, тощо.

Ще однією з тенденцій є впровадження персоналізованої взаємодії. Завдяки машинному навчанню, розробники можуть впровадити, щоб система реагувала на поведінку кожного споживача окремо унікально. Тобто, одна й та сама сторінка вебдодатку для користувачів може виглядати по різному, враховуючи їхні звички, інтереси, тощо.

Через велику популярність соціальних мереж та загалом люди проводять багато часу за телефонами супермаркети почали інтегруватися з ними. Це дозволяє збирати дані, які не витягнеш з вебдодатку, адже зовнішнім платформам можна переглядати поведінку споживачів, на що вони підписані і куди ставлять лайки, які у них переконання, що в майбутньому можна застосувати під час складання персональних пропозицій. Це дозволяє аналізу вподобань вийти за межі основного сервісу та працювати більш глибоко.

Не менш важливою також є наскільки вебдодаток має адаптивний інтерфейс. Завдяки фреймворкам, які продовж років розвинулись, розробники можуть динамічно змінювати кольорову палітру сайту, наприклад під час свят, розташування елементів та пріоритетність блоків з товарами залежно від переваг користувачів, що дозволить затримувати їх тим самим збільшувати шанс на придбання продуктів.

Також популярності набуло використання інтегрованих у вебдодаток ігрових механік. Наприклад, споживачам можуть пропонуватися різні щоденні завдання: пройти коротке опитування щодо нової функції, перегляд недавно появленої категорії, тощо. За виконання цих дій вони отримують бонусні бали, які в свою чергу можна витратити, щоб зменшити ціну продукту. Таке

впровадження дозволить затримати споживача та зібрати інформацію, щоб в подальшому її проаналізувати.

Також до адаптивного інтерфейсі можна додати пункт про адаптацію вебдодатків до цифрової грамотності споживача. Все частіше розробники враховують різні можливості користувачів, тобто рівень технічної обізнаності, вік чи, можливо, якісь людські вади та обмеження. Тому краще мати адаптивний інтерфейс, де споживач зможе міняти розміри шрифтів, колір сторінок, бажано впроваджувати базові інструкції щодо користування сайтом, тощо.

Також впровадження системи CRM (Customer Relationship Management) можна вважати сучасною тенденцією у розробці вебдодатку для аналізу вподобань споживача. Все це через те, що CRM-системи накопичують велику базу даних про користувачів, такі як: його історію покупок, обрахунок середньої суми витрат на продукти, реакція на акції, чи піднімається ця середня сума чи ні, тощо. Тобто така інтеграція дозволить реалізувати додаткові системи, наприклад, система буде підлаштовувати персональні рекомендації, базуючись на вище написаному, автоматично.

Ще однією сучасною тенденцією являється адаптація геолокації споживача до вебдодатку. Все через те, що завдяки цій технології, супермаркети можуть вказати користувачу на найближчий магазин і покаже доступний асортимент там, що зекономить час людині, яка буде шукати продукти не в тому супермаркеті. Також завдяки геолокації може формуватися персональні рекомендації споживача, враховуючи його вподобання та минулі покупки. Це все дозволяє фільтрувати товари, відштовхуючись від локального асортименту супермаркету.

І на останок, слід розібрати найновішу тенденцію, яка існує – це впровадження систем емоційного аналізу. Цей метод дозволяє сфокусуватися не тільки на діях, які споживач виконує, а й що він при цьому відчуває. Для цього застосовують метод обробки природної мови (NPL) для розпізнавання емоційного тону, який вкладений у запити та (або) відгуки. Існують також алгоритм комп'ютерного зору, який може використовувати камеру, щоб

проаналізувати вираз обличчя користувача та технологія аналізу голосу, але їх застосовують тільки тоді, коли людина дала свою згоду. Система аналізу тонального тону дозволяють визначати, які емоції були використані для написання відгуку та відокремлювати конкретні з них, типу вдячності, розчарування, задоволення, тощо. Це дозволяє визначати проблемні місця вебдодатку, які розробники не побачили під час тестування.



РОЗДІЛ 2

ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБДОДАТКУ ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧІВ СУПЕРМАРКЕТУ

2.1 Аналіз інструментарію для створення вебдодатку аналізу вподобань

Створення вебдодатку аналізу вподобань споживача супермаркету потрібно обрати інструменти, які допоможуть обробляти введені користувачем дані, візуально представити інформацію, створення зручного інтерфейсу для людей та зберігання введених даних. Тому від вибору залежить як і технічна сторона та ефективність, так і зручність у використанні клієнтської системи.

Для цього проекту було обрано такий стек технологій: Flask, як серверний фреймворк, HTML та CSS для реалізації клієнтського інтерфейсу, також не слід забувати про систему керування базами даних – тому вибір впав на SQLite. Дані технології допоможуть у реалізації гнучкої, практичної вебсистеми з легкістю.

Спочатку, слід розповісти про фреймворк Python – Flask. Цей фреймворк був розроблений на мові Python і він надає корисні та потужні інструменти для швидкої та простої побудови вебдодатків. Також Flask є розширюваним і не нав'язує якусь певну структуру каталогів або не вимагає надзвичайно складного шаблону для початку роботи з ним. Всі ці факти надають розробникам певну свободу у вигляді та виборі вебдодатку. [38]

Для динамічного вигляду сайту Flask використовує систему шаблонів Jinja2 для створення динамічних HTML-сторінок. Крім цього, фреймворк надає зручну маршрутизацію, роботу з різними формами, файлами cookie, тощо. Всі ці функції надають розробнику реалізовувати функції реєстрації та входу в профіль, проходити різні опитування, додавати вибрані товари в улюблене, тощо. [38]

Також Flask чудово інтегрується з базами даних через свою бібліотеку Flask-SQLAlchemy. SQLAlchemy дозволяє працювати з таблицями бази даних, використовуючи класи та об'єкти Python, тобто замість SQL-запиту, можна

використовувати конструкцію фреймворку, що значно полегшує розробку вебдодатку. Також можна створювати таблиці автоматично через його конструкцію, що спрощує процес налаштування бази даних. [39]

Наступним в черзі йде мова розмітки, яка допомагає визначити, як буде представлена інформація – HTML. Вона відповідає за елементи, які будуть на сторінках вебдодатку, а саме: заголовки, кнопки, різні форми, текст, тощо. Слід уточнити, HTML описує, що повинно бути відображено на сторінках, а саме вигляд і поведінку – це вже CSS і JavaScript. Сучасні версії HTML, а саме версії 5, підтримують теги типу `<nav>`, які покращують гнучкість, доступність та читабельність коду. Також HTML підтримує інтерактивні елементи, гарним прикладом слугує кнопки `<button>`, які можуть перенести користувача на сторінку чи щось інше, дивлячись на що розробник залишив посилання.

Завдяки системі шаблонів Jinja2, HTML структуру можна писати з динамічними вставками, наприклад, підставляти дані у HTML-сторінку, які прийшли з бази даних. Це дає змогу швидко та ефективно підставляти нові оновлення для користувача.

CSS – це мова стилів, яка співпрацює з HTML тим, що саме вона визначає зовнішній вигляд елементів мови розмітки. CSS дозволяє розробнику керувати кольорами на сторінках, шрифтами, положеннями різних елементів типу фотографії, а також адаптувати інтерфейс для різних пристроїв. Додатково можлива реалізація плавних переходів, коли людина наводиться на елементи дизайну, підсвітки, тощо, що покращує вигляд UX.

Мова стилів також чудово підключається до Flask через функцію `url_for`, яка в свою чергу допомагає вказувати шлях до статичних файлів, які ніяк не змінюються під час виконання програми. Це допомагає зберігати проект в чистій структурі, де логіка розділена від представлення та стилів.

Останнім в черзі залишився SQLite – це вбудована реляційна база даних, яка відрізняється від інших систем, наприклад MySQL, PostgreSQL, тим що вона не потребує окремого серверного середовища. Вона працює по принципу, де всі дані та схеми зберігаються у файлі з розширенням `.db`, що робить її ідеальним

кандидатом для різних середніх по обсягу вебдодатків, мобільних додатків, тестування, тощо. [40]

Також SQLite чудово підтримується бібліотекою SQLAlchemy, що робить розробку вебдодатку зрозумілішою та простішою. Для перегляду та редагування можна встановити зручний інструмент DB Browser for SQLite. Він примітивно зрозумілий, що дуже влучно для людей, які майже не знайомі з базами даних, легко редагуються та відкриваються дані з розширенням .db.

Отже, даний стек технологій був обраний через свою практичність, функціональність, простоту та ефективність. Зі серверної сторони, Flask дасть змогу швидко розгорнути весь проект, покладаючись на свої бібліотеки та своїй розширюваності. В свою чергу HTML та CSS допоможуть створити зручний та гарний для споживача інтерфейс, що дозволить йому повністю переглянути можливості вебдодатку. Також не слід забувати про зберігання введених даних, де SQLite виконає чудову та лаконічну роботу разом з бібліотекою SQLAlchemy.

2.2 Функціональні вимоги та технології аналізу вподобань

Розробка вебдодатку для аналізу вподобань споживача супермаркету слід чітко визначити функціональні вимоги, що дозволить забезпечити привабливість для користувача та створити продуктивну архітектуру системи. Однією з ключових вимог є персоналізація взаємодії з користувачем через збір та обробки його вподобань.

Слід розділити функціональні частини вебдодатку на дві групи, а саме: серверний та користувацький функціонали. Через те, що основний акцент стоїть саме на споживацьких вподобаннях, основна увага буде приділятися йому. Тому першою ціллю має бути створення зручної можливості реєстрації та входу в акаунт. Потрібно зібрати мінімальну кількість інформації про користувача, щоб йому було затишно користуватися додатком та забезпечити захист персональних даних, таких як: пошта, пароль ім'я, тощо.

Далі, після реєстрації споживач має пройти невеличке опитування на 10 питань, яке допоможе системі визначити його вподобання. Інформацію, яку

системі надасть користувач, включає в собі питання щодо типів товарів, які цікавлять його, харчові звички або можливо стиль життя. Ці відповіді будуть збережені у базі даних і застосовані для створення персональних пропозицій.

Після опитування, споживачу дадуть можливість переглянути рекомендовані товари, прочитати їхній короткий опис, подивитися як вони виглядають та додати в “Улюблені”. В самій вкладці користувач може переглянути все що він додав та або залишити їх, або видалити їх. Також це є складовою подальшого аналізу вподобань.

Система пропозицій є важливою частиною додатку. Вона спочатку враховує результати опитування і після того, як користувач додав товари до своїх “Улюблених”, система дивиться на них і пропонує щось подібне. Це дає змогу споживачу більш детально дізнатися про продукти, які можливо йому сподобаються та він їх буде вживати в майбутньому.

Варто згадати, що вебдодаток може мати не лише інтерфейс для користувачів, а й для адміністраторів – адміністративна панель. Вона дозволяє перевіряти загальну статистику на сайті, що дозволить в режимі реального часу керувати ним. Також у адміністраторів є доступ до управління товарами у вебдодатку, змінювати їм опис, тощо.

Слід також розглянути алгоритмічні частини вебдодатку, які забезпечують роботу описаного функціоналу. Першим йде обробка результатів опитування. Відповіді користувача заносяться до бази даних, потім система присвоює цим відповідям певну набір ознак і на основі цього формується певний профіль вподобань споживача, який визначає його інтерес до певних категорій товарів.

Наступним йде сам аналіз списку рекомендованих вподобань споживача. Також база даних збирає інформацію з сторінки “Пропозиції” користувача, система дивиться, які категорії співпадають, після цього знову повертається до бази даних та витягує з неї ті товари, які ще не були показані користувачу.

Отже, підсумовуючи вище написане, можна чітко конкретизувати функціональні вимоги до вебдодатку. Вони допоможуть визначити необхідний функціонал для споживача та побудувати зручну архітектуру додатку для нього.

Перелік ключових функціональних вимог вебдодатку виглядає так:

- механізми реєстрації нового користувача та авторизації існуючого;
- після реєстрації, користувач повинен пройти опитування на 10

запитань, що дозволить системі визначити його вподобання;

- після реєстрації відбувається автоматичне перенаправлення споживача до сторінки з рекомендованими товарами, базуючись на відповідях з опитування, де користувач зможе побачити рекомендовані товари;

- можливість додати продукти до сторінки “Улюблене” для подальшого перегляду.

Додаткові можливості:

- можливість повторного проходження опитування, для зміни рекомендованих товарів;
- можливість редагувати свій профіль, а саме: зміна імені, пошти та паролю;
- блок з додатковими пропозиціями на сторінці “Пропозиції”.

Слід також розглянути одну з ключових частин у розробці вебдодатку для аналізу вподобань – алгоритм обробки результатів опитування та формування пропозицій, на основі цих відповідей. Сам алгоритм побудований по принципу: користувач проходить опитування і вписує свої відповіді, далі система підставляє ці відповіді до параметрів продуктів, які вказані у базі даних, якщо система знайшла такі товари, то вони будуть додані до сторінки “Пропозиції” у користувача, якщо ні, то вебдодаток запропонує пройти заново тест.

Даний алгоритм можна розглянути детальніше, розбивши його на декілька пунктів:

1. Після реєстрації, користувач потрапляє на сторінку з опитуванням, яка містить таких 10 питань: група товару, галузь застосування, вікова група, специфіка використання, обмеження, наявність бренду, наявність пакування, ціновий діапазон, можливість придбання з пільгами та наявність знижок.

2. Система зберігає надані відповіді, потім зчитує їх та за допомогою формування SQL запиту фільтрує всі товари, які знаходяться у базі даних та

знаходить за заданими критеріями.

3. Після цього користувач отримує набір основних рекомендованих товарів, якщо таких не знайдено через не вірно заповнені поля або відсутність у базі даних таких товарів, система буде рекомендувати користувачу пройти заново опитування.

Додатково:

1. Також на сторінці з основними пропозиціями, внизу, знаходяться додаткові пропозиції, які формуються, відштовхуючись від основних по принципу “суміжних категорій”.

2. Фільтрація відбирає товари, які попадають по цьому принципу, виключаючи товари, які знаходяться в категорії основних пропозицій.

Отже, весь цей алгоритм допоможе споживачу швидко та ефективно знайти товари, про які він знав, і відкрити щось нове для себе. Також за допомогою додаткових пропозицій можна ще обрати продукти до загальної покупки. Все що споживачу потрібно – це ввести вподобання під час опитування, а сервер вже зробить усі механізми фільтрації та висвітить результат на сторінці.

2.3 Модель вебдодатку та база даних для забезпечення основних операцій вебдодатку

Модель вебдодатку для аналізу вподобань споживача супермаркету ґрунтується на взаємодії з користувачем на основі його переваг. Основною метою є формування персональних пропозицій товарів з каталогу, які будуть максимально точно відповідати заданим критеріям від споживача. Для досягнення цієї цілі вебдодаток використовує структуровану модель даних, механізми збереження різних взаємодій із системою та алгоритм фільтрації.

Розпочнемо з моделі формування пропозицій, яка побудована, як пряма фільтрація. Вона працює таким чином, що після проходження опитування користувач отримує результати на основі того, які категорії він зазначив. Наприклад, якщо споживач вибрав категорію “м’ясні вироби” та “продовольча” система запропонує йому товари з бази даних, які відповідають цим критеріям.

Таким чином, кінцевий результат буде демонструвати користувачу саме ті продукти, на які він надіявся.

Не мало важливу роль займає формування додаткових пропозицій. Працює вона на наборі знань – суміжні категорій. Це такий набір правил, який визначає різні логічні зв'язки між різними категоріями товарів. В даному проекті використовуються суміжні категорії для того, щоб розширити загальні пропозиції споживача, тобто запропонувати йому категорії, які можуть його зацікавити, відштовхуючись від основного вибору користувача. Наприклад, людина обрала “м'ясні вироби”, тобто вона вибирає гарнір, зазвичай до такої страви також подають різні хліба, напої, тощо. Система це аналізує і пропонує споживачу товари з категоріями “напої” та “хлібобулочні вироби”, які лаконічно доповнюють основний вибір під час реєстрації.

Також, якщо система не знайде збігів між результатом опитуванням і інформації, яка збережена, то користувачеві запропонують товари, які виставлені за замовчуванням та пройти тест заново, щоб пропозиції почали працювати.

Перед тим як споживач перейде до сторінки “Пропозиції”, йому потрібно зареєструватися. Цей процес не буде займати у нього багато часу, адже все що від нього потрібно – це ім'я, пошта та пароль. Користувачу не потрібно вводити свої якісь персональні дані або номер телефону та підтверджувати його. Через таку швидку реєстрацію він не втратить цікавості до вебдодатку та не буде надавати не потрібної інформації для сайту.

Далі, система запропонує пройти опитування, що визначить його вподобання завдяки системі пропозицій, яка описана зверху. Якщо споживача не влаштують результати не влаштують або він просто захоче змінити їх, йому достатньо натиснути одну кнопку і поміняти все, що йому не подобається.

Сторінка “Улюблене” демонструє споживачу всі товари та їхній опис, якщо він не буде пам'ятає, що саме він додавав, то він запросто зможе перевірити це на цій сторінці. Також користувач може видаляти продукти, які йому вже більше не подобаються або не потрібні і система адаптується під це.

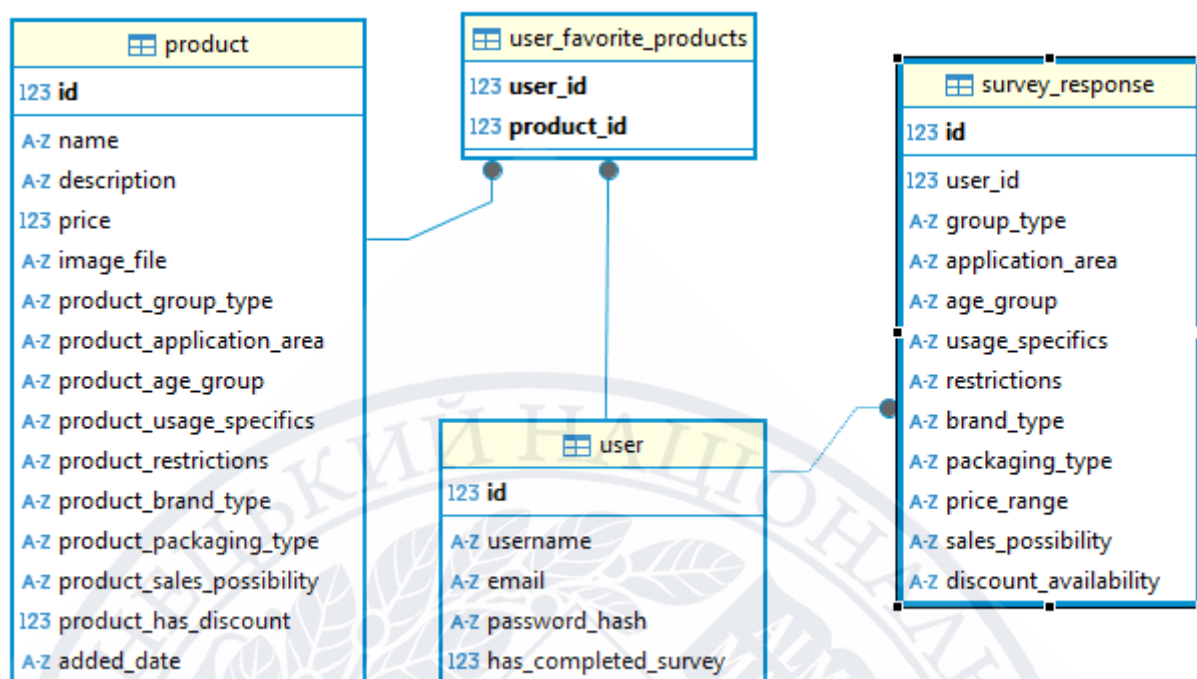


Рисунок 2.1 - Представлення бази даних зі зв'язками для вебдодатку

Слід також згадати про каталог товарів та їх характеристики в системі. До вебдодатку під'єднана база даних SQLite, яка зберігає не тільки дані користувачів, а й продукти, які вибирають користувачі. В загалом у всіх товарів є певний набір атрибутів, а саме: їх назва, опис, ціна, зображення, а також ключові характеристики, що співвідносяться з питаннями опитування. Саме ці характеристики є основою для роботи механізму пропозицій, який був описаний вище

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ ВЕБДОДАТКУ ДЛЯ АНАЛІЗУ ВПОДОБАНЬ СПОЖИВАЧІВ СУПЕРМАРКЕТУ

3.1 Проєкт дизайну вебдодатку для аналізу вподобань

Для цього проєкту весь дизайн, включаючи UI компоненти, були створені завдяки HTML, CSS та системі шаблонів Jinja2 і підібрані вручну. Розглянемо основний дизайн вебдодатку:

Так виглядає хедер, завдяки якому споживач може пересуватися по сайту, натискаючи на відповідні сторінки:



Рисунок 3.1 - Вигляд хедера

Далі слід розглянути футер, який містить в собі інформацію про рік створення вебдодатку та саму його назву:

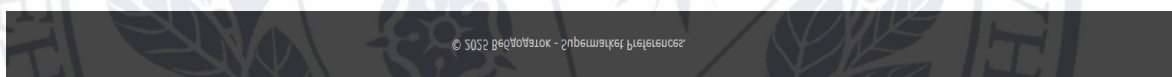


Рисунок 3.2 - Вигляд футера

Далі, коли споживач відкриє вперше вебдодаток його зустрінє привітання та просьба зареєструватися, якщо у нього немає акаунту:

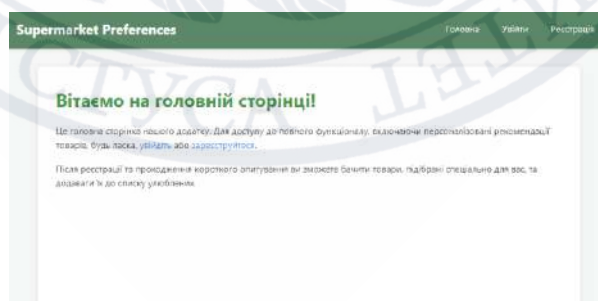


Рисунок 3.3 - Вигляд вітальної головної сторінки

Якщо користувач знаходиться у вебдодатку вперше, то йому потрібно буде зареєструватися. Ось як виглядає вікно реєстрації:

Supermarket Preferences Головна Увійти Реєстрація

Реєстрація

Ім'я користувача

Email

Пароль

Підтвердьте пароль

Зареєструватися

Рисунок 3.4 - Вигляд сторінки реєстрації нового користувача

Після реєстрації, споживача перекине на сторінку з опитуванням, яке визначить, які саме його вподобання:

Опитування

Ви можете обрати свої відповіді нижче. Це вплине на ваші майбутні пропозиції

1. Група товару

☒ Продовольча

☐ Непродовольча

2. Галузь застосування

м'ясні вироби

3. Вікова група застосування

☒ Звичайна

☐ Діти

4. Специфіка використання

☒ Без обмежень

☐ Обмеження

5. Обмеження

☐ Без хімічних

☐ Органіка виробництва

☒ Інше

Рисунок 3.5 - Вигляд сторінки першої частини опитування

6. Бренд/нейм

☒ Бренд

☐ Нейм

7. Пакування/безпакування (розсил)

☒ Пакування

☐ Без пакування (розсил)

8. Бажаний ціновий діапазон

1

Вкажіть приблизний діапазон ціни, який вас цікавить (наприклад, "100-200 грн", "до 50 грн", "від 300 грн")

9. Можливість продажу (пільги)

☐ Пільги

☒ Без пільг

10. Знижка

☒ Наявність

☐ Відсутність

Надіслати відповідь

Рисунок 3.6 - Вигляд сторінки другої частини опитування

Після того, як споживач дав відповіді на запитання в опитуванні і натиснув кнопку “Надіслати відповіді”, його перекидує на сторінку “Пропозиції”:

Supermarket Preferences

Головна Пропозиції Улюблені Пріяті, Івані

Товари, які відповідають вашим вподобанням

На основі ваших відповідей на опитування, ми підбрали наступні товари. Ви можете порівняти свої відповіді, щоб змінити список товарів.

Ковбаски гриль «ЦЕМ'ЯСО» ТМ «ЦЕМ'ЯСО» 95.00 грн Напівкопчені з м'яса для смаження на грилі. В улюблені +	Біфтекс в беконі зі спеціями, охолоджений 25.40 грн Біфтекс із молоччини та свинини, обгорнутий беконом. Додати +	Чевапчіці зі спеціями, у беконі 25.40 грн Вампелі М'ясо ковбаски з беконом, охолоджені. Додати +
Ковбаски куриці фірмові, охолоджені 25.90 грн Куриці ковбаски зі спеціями, в пакуванні.		

Рисунок 3.7 - Вигляд сторінки “Пропозиції”

Також на цій ж сторінці, внизу, можна побачити додаткові пропозиції, які працюють по принципу “суміжних категорій”:

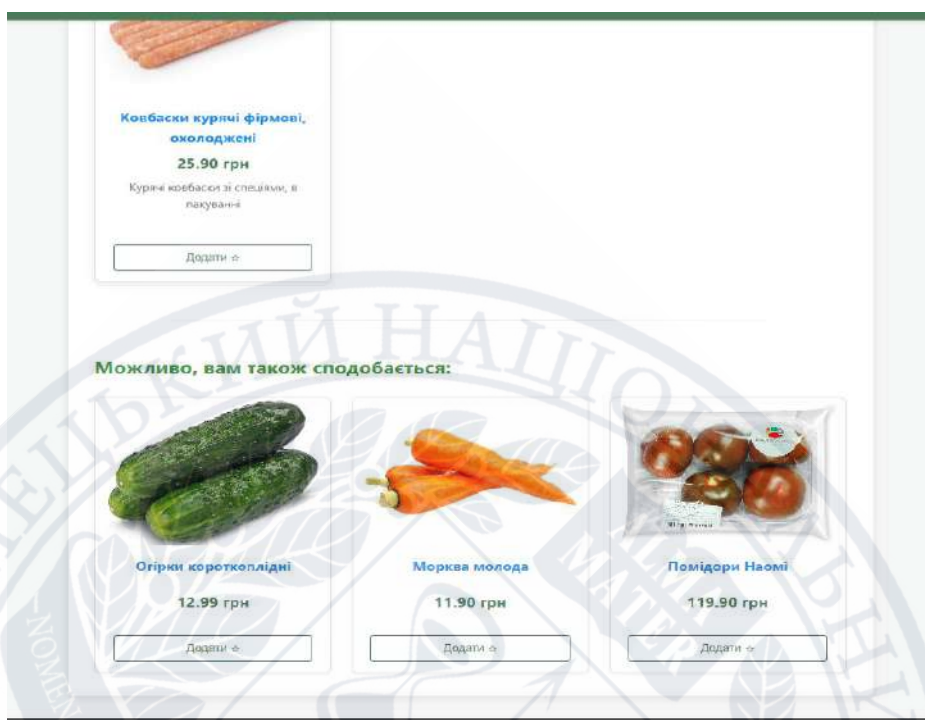


Рисунок 3.8 - Вигляд додаткових пропозицій

Ось так виглядає сторінка “Улюблені”, де споживач може переглядати додані товари:

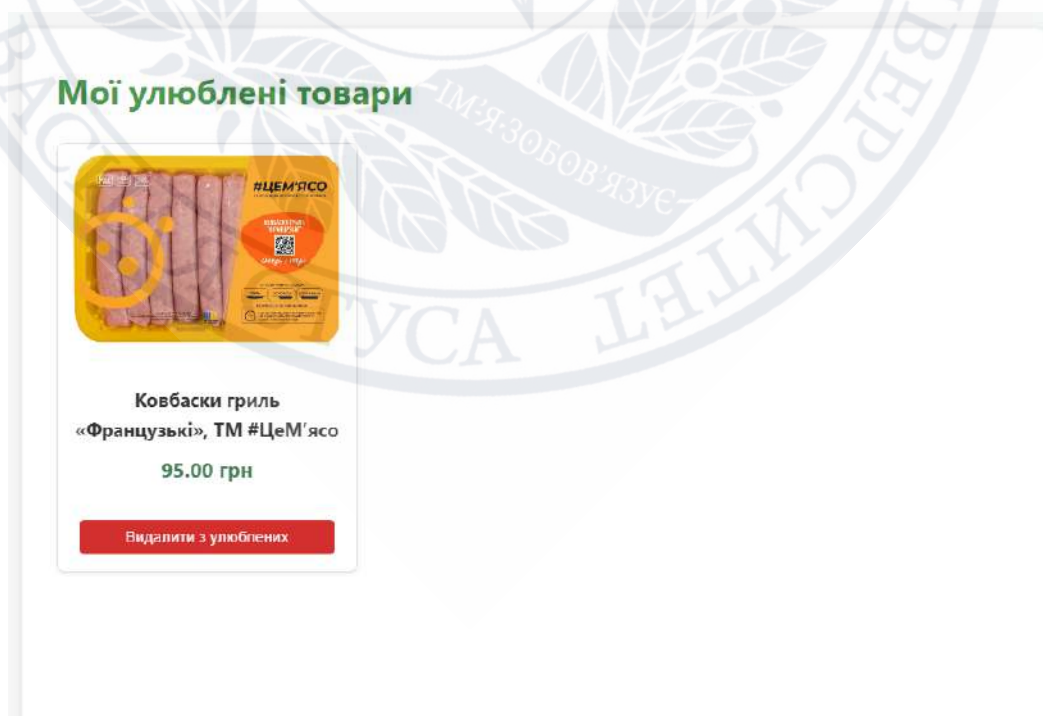


Рисунок 3.9 - Вигляд сторінки “Улюблені”

Ось так виглядає “Головна” сторінка, коли користувач авторизований у вебдодатку:

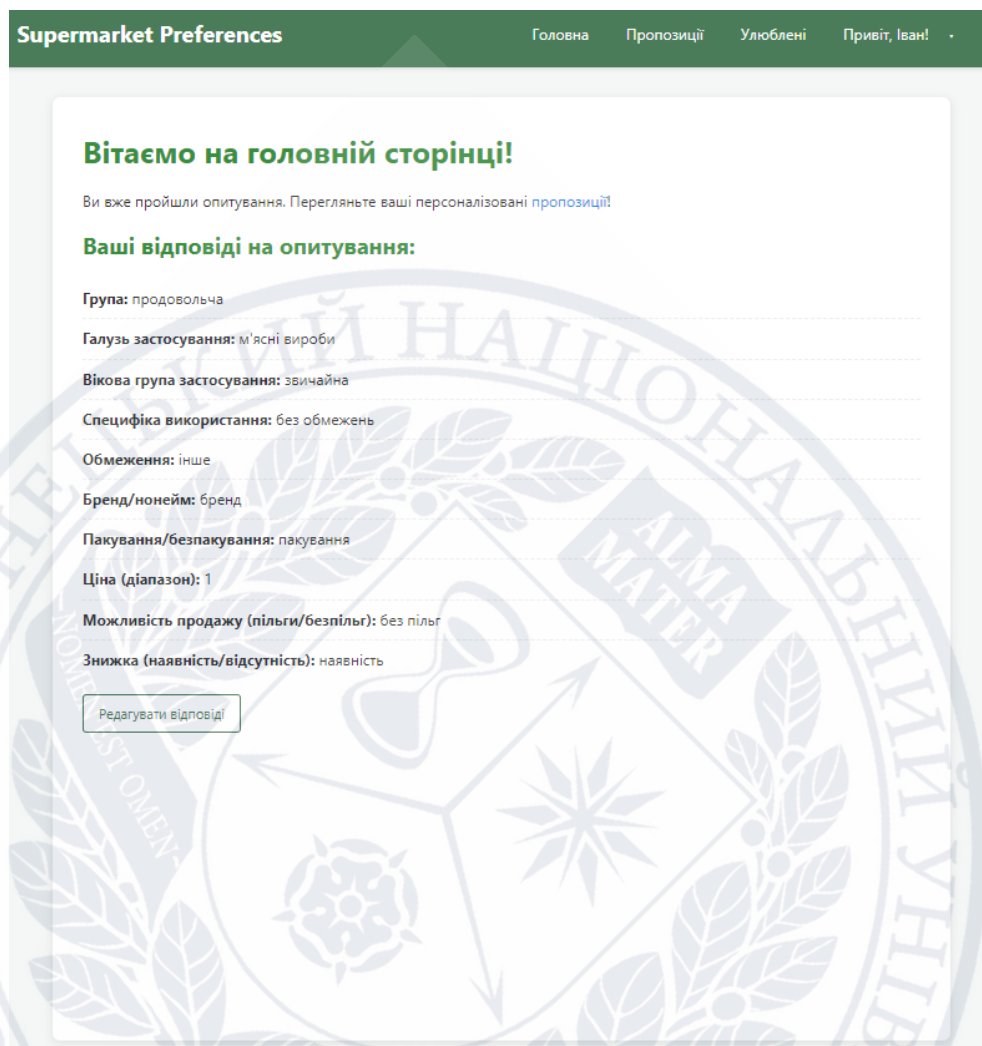


Рисунок 3.9 - Вигляд головної сторінки для авторизованих користувачів

Ось так виглядає випадаюче меню, де користувач може обрати 1 з 3 дій:

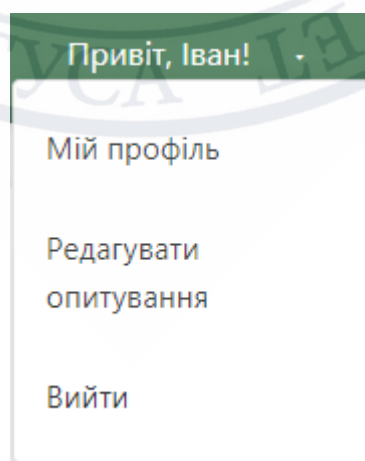


Рисунок 3.10 - Вигляд випадаючого меню

Також, якщо споживач захоче змінити дані про себе, йому достатньо перейти на сторінку “Мій профіль”, де він зможе здійснити ці операції:

Рисунок 3.11 - Вигляд сторінки “Мій профіль”

3.2 Програмна реалізація вебдодатку для аналізу вподобань

Розпочнемо з розбору архітектури вебдодатку. Ось так виглядає папка з повноцінним проектом:

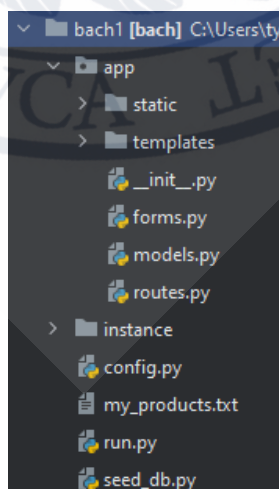


Рисунок 3.12 – Вигляд папки з проектом

Як можна побачити існує ще 3 папки, де зберігаються важливі компоненти вебдодатку. В папці static знаходяться каскадні таблиці стилів CSS, код JavaScript та зображення товарів:

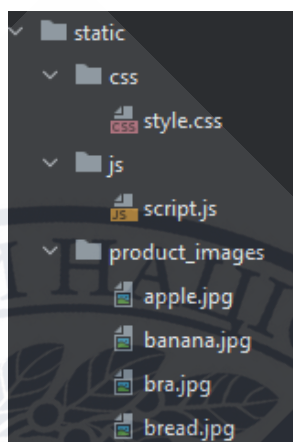


Рисунок 3.13 – Вигляд папки static

В папці templates знаходяться HTML-шаблони, які використовуються задля генерації динамічних вебсторінок:

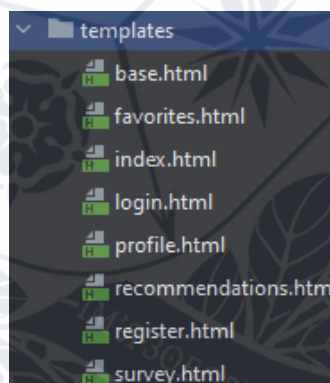


Рисунок 3.14 – Вигляд папки templates

І остання папка – instance, яка існує для того, щоб зберігати файл бази даних SQLite:

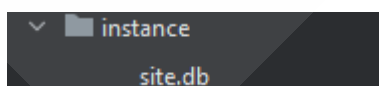


Рисунок 3.15 – Вигляд папки instance

З самого початку були створені функціональні хедер та футер у файлі base.html. Якщо футер це показ коли був створений вебдодаток та його назва, то

хедер вже більш реалізований, як навігатор по сайту. Завдяки ньому споживач може перемикатися між сторінками (Головна, Пропозиції, Улюблене, Мій профіль) та продивлятися всю інформацію, яка його цікавить.

Слід розглянути найголовніші сторінки вебдодатку, з якими споживачі будуть найчастіше взаємодіяти – це сторінка з опитуванням та “Пропозиції”.

Розпочнемо зі сторінки з опитуванням, де споживач після реєстрації повинен відповісти на 10 запитань. Вона є ключовим інструментом для збору інформації про вподобання користувача.

Після того, як на запитання дали відповідь споживач відправляє свої результати на сервер через метод POST, завдяки функції `def survey` у файлі `routes.py`. Також, якщо користувач захоче редагувати відповіді в опитуванні, то це реалізовано в GET частині логіки.

Лістинг коду, який реалізовує дану логіку:

```
@bp.route('/survey', methods=['GET', 'POST'])
@login_required
def survey():
    form = SurveyForm()
    if request.method == 'GET' and current_user.has_completed_survey and
current_user.survey_response:
        s_data = current_user.survey_response
        form.group_type.data = s_data.group_type
        form.application_area.data = s_data.application_area
        form.age_group.data = s_data.age_group
        form.usage_specifics.data = s_data.usage_specifics
        form.restrictions.data = s_data.restrictions
        form.brand_type.data = s_data.brand_type
        form.packaging_type.data = s_data.packaging_type
        form.price_range.data = s_data.price_range
        form.sales_possibility.data = s_data.sales_possibility
        form.discount_availability.data = s_data.discount_availability
```

```

if form.validate_on_submit():
    response =
SurveyResponse.query.filter_by(user_id=current_user.id).first()
    if response is None:
        response = SurveyResponse(user_id=current_user.id)
    response.group_type = form.group_type.data
    response.application_area = form.application_area.data.strip().lower() #
Зберігаємо в нижньому регістрі
    response.age_group = form.age_group.data
    response.usage_specifics = form.usage_specifics.data
    response.restrictions = form.restrictions.data
    response.brand_type = form.brand_type.data
    response.packaging_type = form.packaging_type.data
    response.price_range = form.price_range.data.strip()
    response.sales_possibility = form.sales_possibility.data
    response.discount_availability = form.discount_availability.data

    if response not in db.session:
        db.session.add(response)
    current_user.has_completed_survey = True
    db.session.commit()
    flash('Ваші відповіді на опитування оновлено/збережено!', 'success')
    return redirect(url_for('main.recommendations'))

title = "Опитування"
if current_user.has_completed_survey:
    title = "Редагувати відповіді на опитування"
return render_template('survey.html', title=title, form=form)

```

Далі дані передаються до об'єкта SurveyForm у файлі forms.py, де завдяки зазначеним правилам проходять валідацію. Використовуються такі валідатори, як DataRequired (для обов'язкових полів, тобто для всіх) та Length (для полів, де потрібно вписувати дані вручну).

Лістинг коду, який реалізовує дану логіку:

```
class SurveyForm(FlaskForm):
    group_type_choices = [('продовольча', 'Продовольча'), ('непродовольча',
'Непродовольча')]
    group_type = RadioField('1. Група', choices=group_type_choices,
validators=[DataRequired()])
    application_area = StringField('2. Галузь застосування',
validators=[DataRequired(), Length(max=200)])
    age_group_choices = [('звичайна', 'Звичайна'), ('діти', 'Діти')]
    age_group = RadioField('3. Вікова група застосування',
choices=age_group_choices, validators=[DataRequired()])
    usage_specifics_choices = [('без обмежень', 'Без обмежень'),
('обмеження', 'Обмеження')]
    usage_specifics = RadioField('4. Специфіка використання',
choices=usage_specifics_choices, validators=[DataRequired()])
    restrictions_choices = [('без хімічних', 'Без хімічних'), ('органіка
виробництво', 'Органіка виробництво'), ('інше', 'Інше')]
    restrictions = RadioField('5. Обмеження', choices=restrictions_choices,
validators=[DataRequired()])
    brand_type_choices = [('бренд', 'Бренд'), ('нонейм', 'Нонейм')] # Змінено
на "нонейм"
    brand_type = RadioField('6. Бренд/нонейм', choices=brand_type_choices,
validators=[DataRequired()])
    packaging_type_choices = [('пакування', 'Пакування'), ('без пакування
(розсип)', 'Без пакування (розсип)')]
    packaging_type = RadioField('7. Пакування/безпакування',
```



```

choices=packaging_type_choices, validators=[DataRequired()])
    price_range = StringField('8. Ціна (діапазон, напр. 100-200 грн, до 50, від
500)', validators=[DataRequired(), Length(max=100)])
    sales_possibility_choices = [('пільги', 'Пільги'), ('без пільг', 'Без пільг')]
    sales_possibility = RadioField('9. Можливість продажу',
choices=sales_possibility_choices, validators=[DataRequired()])
    discount_availability_choices = [('наявність', 'Наявність'), ('відсутність',
'Відсутність')]
    discount_availability = RadioField('10. Знижка',
choices=discount_availability_choices, validators=[DataRequired()])
    submit = SubmitField('Надіслати відповіді')

```

Після всіх цих операцій та успішної валідації, відповіді споживача будуть збережені у таблиці SurveyResponse в базі даних SQLite. Там створюється новий запис або оновлюється існуючий, якщо користувач захоче змінити свої вподобання.

Після успішного проходження опитування, споживач переходить до наступної ключової функції, а саме персональні пропозиції. Ця сторінка надає користувачу список товарів, який підібраний відповідно до його вподобань, які були заявлені під час опитування.

Працює механізм підбору персональних пропозицій після того, як отримає відповіді після опитування. Далі механізм робить запит до бази даних, де спочатку охоплює всі товари, а потім поступово звужує коло пошуку завдяки умов фільтрації .filter() для кожного критерію з опитування. Також кожен фільтр відповідає одному з 10 питань, на які споживачі давали відповідь. Цей механізм реалізований в функції def recommendation у файлі routes.py

Лістинг частини коду, який реалізовує дану логіку:

```

recommended_products_query = Product.query
if survey_data.group_type:

```

```

recommended_products_query =
recommended_products_query.filter(Product.product_group_type ==
survey_data.group_type)
user_main_category_for_related = None
if survey_data.application_area:
    user_main_category_for_related = survey_data.application_area
    search_term = f"%{survey_data.application_area}%" # Пошук підрядка
    recommended_products_query =
recommended_products_query.filter(Product.product_application_area.ilike(se
arch_term))

if survey_data.age_group:
    recommended_products_query =
recommended_products_query.filter(Product.product_age_group ==
survey_data.age_group)

... (аналогічні фільтри для інших категорій)

if survey_data.price_range:
    try:
        pr_text = survey_data.price_range.lower()
        if '-' in pr_text:
            parts = pr_text.split('-')
            min_p_str = ".join(filter(str.isdigit, parts[0]))
            max_p_str = ".join(filter(str.isdigit, parts[1]))
            if min_p_str and max_p_str:
                min_price = float(min_p_str)
                max_price = float(max_p_str)
                recommended_products_query =
recommended_products_query.filter(

```

```

        Product.price.between(min_price, max_price))
    elif pr_text.startswith('до '):
        max_p_str = ".join(filter(str.isdigit, pr_text.replace('до ', '')))
        if max_p_str:
            max_price = float(max_p_str)
            recommended_products_query =
recommended_products_query.filter(Product.price <= max_price)
    elif pr_text.startswith('від '):
        min_p_str = ".join(filter(str.isdigit, pr_text.replace('від ', '')))
        if min_p_str:
            min_price = float(min_p_str)
            recommended_products_query =
recommended_products_query.filter(Product.price >= min_price)
    except (ValueError, IndexError) as e:
        print(f"Не вдалося розпарсити діапазон цін: {survey_data.price_range},
помилка: {e}")

```

Також слід розглянути механізм “суміжних категорій”, який існує для розширення вибору потенційно цікавих товарів для споживача. Для цього механізму існує визначений словник категорій у файлі routes.py. Далі система використовує категорію “Галузь застосування”, яка була вказана споживачем під час проходження опитування і дивиться чи така є в словнику, якщо є то вебдодаток створює запит до бази даних, щоб витягнути з неї продукти, які підпадають під інші категорії. Product.product_application_area.in() допомагає обрати товари, чия “Галузь застосування” входить до списку суміжних категорій. Лістинг коду, який реалізовує дану логіку:

```

key_category_for_related = user_main_category_for_related
if not key_category_for_related and main_recommendations:
    key_category_for_related =
main_recommendations[0].product_application_area.lower()

```



```

if key_category_for_related and key_category_for_related in
RELATED_CATEGORIES_MAP:
    related_categories_list =
RELATED_CATEGORIES_MAP[key_category_for_related]
    if related_categories_list:
        related_query = Product.query.filter(
            Product.product_application_area.in_(related_categories_list),
            not_(Product.id.in_(main_recommendations_ids))
        )
        if survey_data.group_type:
            related_query = related_query.filter(Product.product_group_type ==
survey_data.group_type)

        all_possible_related = related_query.all()
        random.shuffle(all_possible_related)
        related_recommendations = all_possible_related[:3]

```

3.3 Оцінка функціональності роботи вебдодатку для аналізу вподобань споживачів супермаркету

Після того, як були розглянуті програмна та дизайнерська сторони проекту, слід оцінити функціональну роботу цього вебдодатку для аналізу вподобань споживача супермаркету.

Розпочнемо з реєстрації, де користувач вписує своє ім'я, пошту та пароль. Після цього дані записуються до бази даних SQLite в таблицю user, де є username (збереження вказаного імені користувача), email (збереження вказаної пошти), password_hash (збереження хешованого пароля користувача) та has_complete_survey (вказує чи користувач пройшов опитування чи ні).

	id	username	email	password_hash	has_completed_survey
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр
1	1				
2	2	Іван	buraha.i@donnu.edu.ua	script:...	1

Рисунок 3.16 – Приклад запису даних споживача

Як можна бачити, дані, які були вказані під час реєстрації були успішно внесені до бази даних, що може означати цей аспект вебдодатку працює успішно. Перейдемо до оцінки функціоналу опитування та системи персональних пропозицій.

Оцінімо роботу опитування, де споживачу потрібно відповісти на 10 запитань, які визначають його вподобання та система дасть йому список персональних пропозицій завдяки механізмам, які були розглянуті у попередньому підрозділі. Для прикладу заповнимо ці запитання, так щоб система запропонувала молочні продукти.

Опитування

Ви можете оновити свої відповіді нижче. Це вплине на ваші майбутні рекомендації.

- Група товару**
 - ☒ Продовольча
 - ☐ Непродовольча
- Галузь застосування**
 - молочні продукти
- Вікова група застосування**
 - ☒ Звичайна
 - ☐ Діти
- Специфіка використання**
 - ☒ Без обмежень
 - ☐ Обмеження
- Обмеження**
 - ☐ Без хімічних
 - ☐ Органіка виробництво
 - ☒ Інше

Рисунок 3.17 – Приклад заповнення першої частини опитування

user_id	group_type	application_area	age_group	usage_specifcs	restrictions
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1					
2	продовольча	молочні продукти	звичайна	без обмежень	інше

Рисунок 3.18 – Занесення першої частини відповідей до бази даних

Рисунок 3.19 – Приклад заповнення другої частини опитування

rand_type	packaging_type	price_range	sales_possibility	discount_availability
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
бренд	пакування	1	без пільг	відсутність

Рисунок 3.20 – Занесення другої частини відповідей до бази даних

Як можна бачити відповіді, які були дані під час опитування без проблем з'явилися в базі даних, що може означати успішне виконання механізму, перейдемо далі.

Оцінемо функціональність системи персональних пропозицій, яка підбирається завдяки відповідям в опитуванні. Завдяки механізму підбору пропозицій, який був описаний в попередньому підрозділі, система показала такі пропозиції:



Рисунок 3.21 – Результати пропозицій, які відповідають вказаним вподобанням

Можна побачити, що пропозиції підібрані вдало та відповідають вказаним відповідям, як тут можна побачити продукти відповідають галузі застосування, ціновому діапазону, тощо.

Також слід оцінити функціональність додаткових пропозицій, які доповнюють основні. Працює вона по принципу “суміжних категорій”, тобто до молочних продуктів споживачів може зацікавити ще чай, кава, солодоці або фрукти, тощо. Тоді система має запропонувати щось з цього списку, що в майбутньому може спонукати користувача переглянути ще інші категорії.

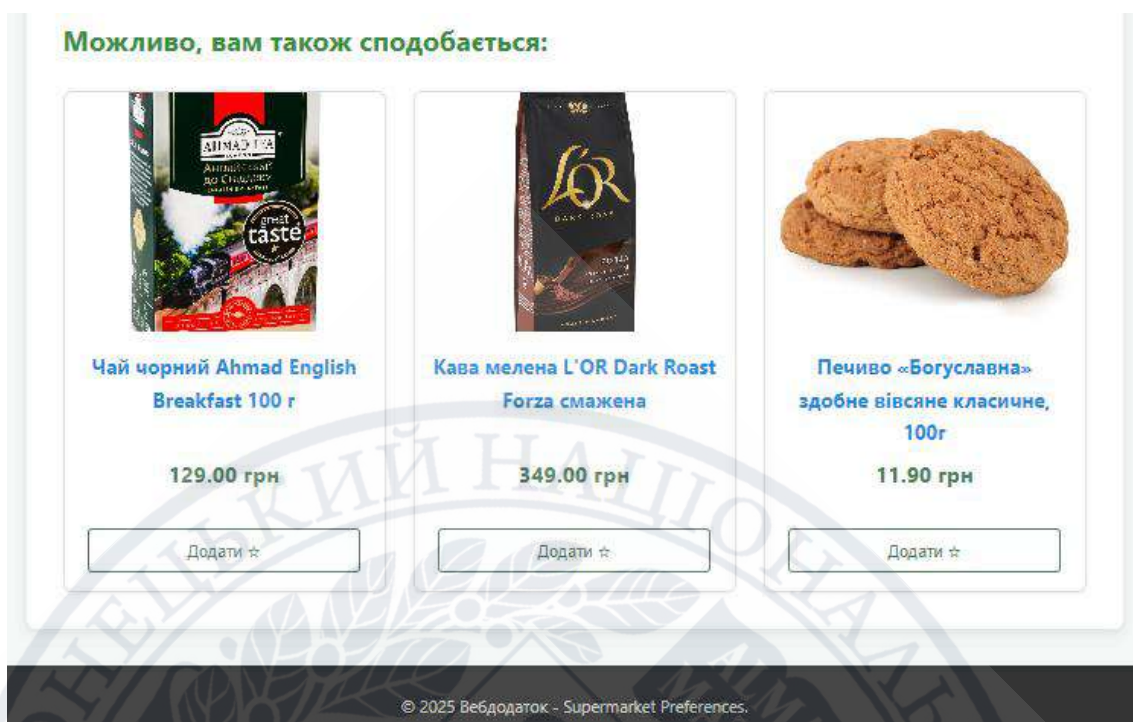


Рисунок 3.21 – Результати додаткових пропозицій, які відштовхуються від
ОСНОВНИХ

Далі слід оцінити функціонал додавання товару до сторінки “Улюблені”. Після того як споживач натисне кнопку “Додати” на сторінці “Пропозиції”, цей товар попадає в базу даних у таблицю user_favorite_products та на сторінку, де користувач може переглядати додані продукти та якщо йому вони вже не цікаві – прибрати їх.



Рисунок 3.22 – Додані продукти до сторінки “Улюблені”

<u>user_id</u>	<u>product_id</u>
Фільтр	Фільтр
1	2
2	2
3	2

20	20	Ковбаса варена «Любительська», в/Г,...
21	21	Ковбаса варена «Любительська», в/Г,...
22	22	Ковбаски гриль «Французькі», ТМ ...
23	23	Віфштекс в беконі зі спеціями, ...
24	24	Чевапчічі зі спеціями, у беконі
25	25	Ковбаски курячі фірмові, охолоджені
26	26	Яблука Голден Делішес
27	27	Апельсини
28	28	Груші Конференція
29	29	Лимони
30	30	Помідори Наомі
31	31	Морква молода
32	32	Буряк "Грінвіль" варений
33	33	Капуста білотолова
34	34	Молоко «Селянське» 2.5%, 900 мл
35	35	Йогурт питний «Галичина» полуниця ...
36	36	Сир кисломолочний «President зі ...
37	37	Масло солодковершкове «ферма» екстр...
38	38	Кефір «ПростоНаше» 2.5%, 870 г

Рисунок 3.23 – Демонстрація запису доданих товарів до бази даних

Як можна бачити додані товари були успішно записані до бази даних, отже цей функціонал є працездатним. І остання оцінка функціональності буде зміна даних користувача.

Якщо користувач захоче змінити свою пошту або пароль, чи можливо використане ім'я, то йому достатньо зайти на сторінку “Мій профіль”, де він може все це змінити. Дані, які змінені також заміняться в базі даних, але ніяка інформація, типу товари, які були додадні до “Улюблені”, не пропадуть.

Мій профіль

Оновити інформацію профілю

Ім'я користувача

Ivan

Email

bedboxlivechannel@gmail.com

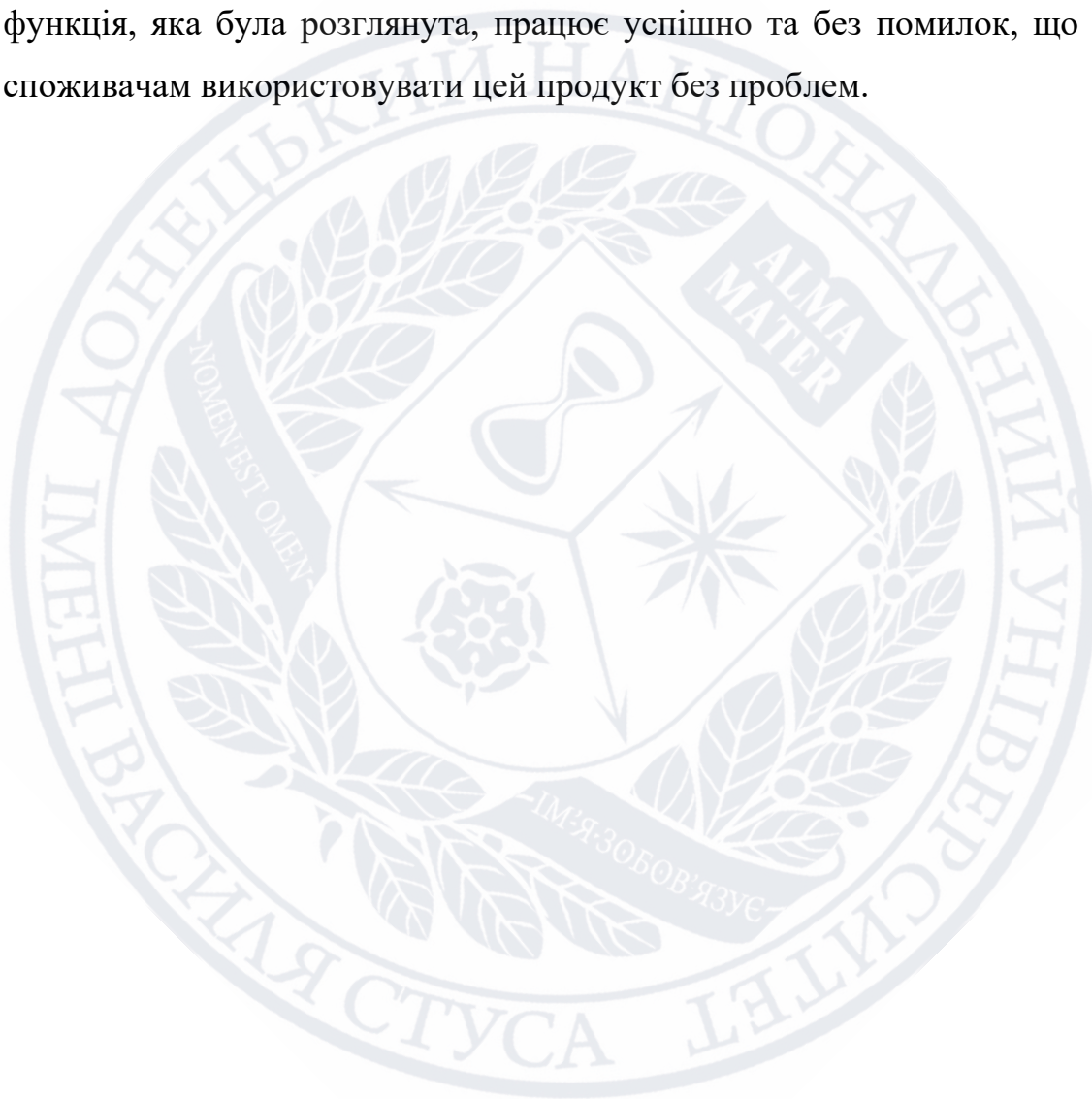
Оновити профіль

Рисунок 3.24 – Демонстрація зміни даних користувача

id	username	email	password_hash
Фи...	Фильтр	Фильтр	Фильтр
1			
2	Ivan	bedboxlivechannel@gmail.com	script:...

Рисунок 3.25 – Демонстрація зміни даних користувача в базі даних

Отже, оцінити функціонал вебдодатку можна як робочий, адже кожна функція, яка була розглянута, працює успішно та без помилок, що дозволить споживачам використовувати цей продукт без проблем.



ВИСНОВКИ

Сучасний світ швидко адаптується під нові технології і тому ринок просить використовувати персональний підхід до кожного споживача. Тоді у цьому контексті вебдодаток для аналізу вподобань споживача супермаркету набуде більшої популярності. Його успішність забезпечить: простий та зручний дизайн, персональні пропозиції, які базуються на вподобаннях споживача та їх швидкість прийому та обробки цих даних. По даній бакалаврській роботі можна зробити наступні висновки:

1. Проведено аналіз теоретичних основ та концептуальні підходів, що надало можливість окреслити напрями розробки архітектури вебдодатку споживача супермаркету, його дизайну та програмної реалізації. Проаналізовано основні характеристики додатків за наступними класифікаційними групами: MPA (односторінкові додати), SPA (багатосторінкові додатки), PWA (Прогресивні вебдодатки).
2. На основі аналізу сучасних технологій програмування обґрунтовано вибір інструментів розробки вебдодатку для аналізу вподобань споживача супермаркету. основними серед яких є використання: Flask, HTML, CSS, JavaScript, SQLite.
3. Визначено сутність сподобань споживача та основні критерії, які їх формують, що надало змогу окреслити вимоги та механізми їх аналізу, також сформулювати інші потреби користувача при взаємодії з вебдодатком. Окреслено, що вподобанням споживача можна вважати деякий набір ознак, за якими споживач може проводити суб'єктивну оцінку цінності товару в супермаркеті. Основними критеріями оцінювання обрано: група товару, галузь застосування, вікова група, специфіка використання, обмеження, наявність бренду, наявність пакування, ціновий діапазон, можливість придбання з пільгами та наявність знижок.
4. Розроблено структуру бази даних, яка є одним із основних елементів в роботі вебдодатка по забезпеченню операцій зберігання, обробки та аналізу

даних по визначенню вподобань споживача супермаркету.

5. Розроблено дизайн та програмно реалізовано вебдодаток для аналізу вподобань споживача супермаркету. Аналіз вподобань реалізовано за алгоритмом обробки результатів опитування та формування пропозицій, на основі відповідей споживача. Основними етапами обробки є: реєстрація споживача в системі, опитування, формування SQL запиту та фільтрація обраних товарів, видача рекомендованих товарів.

6. Проведена перевірка працездатності вебдодатку та оцінено функціональність його роботи, на підставі чого можна стверджувати про відповідність розробки заявленим функціональним характеристикам.

Отже, завдяки розробленому вебдодатку для аналізу вподобань споживача супермаркету та алгоритму обробки результатів опитування та формування пропозицій, користувачі мають отримувати майже точні результати своїх майбутніх покупок.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Web Application. URL: <https://www.sciencedirect.com/topics/computer-science/web-application>
2. Client-Server Architecture: Introduction to Web Development Concepts. URL: <https://dev.to/kihuni/client-server-introduction-to-web-development-concepts-2g2m?>
3. Single Page Application (SPA) vs Multi Page Application (MPA): Which Is The Best? URL: <https://cleancommit.io/blog/spa-vs-mpa-which-is-the-king/>
4. What Are Single Page Applications and Why Do People Like Them So Much? URL: <https://www.bloomreach.com/en/blog/what-is-a-single-page-application>
5. Що являє собою Progressive Web App. URL: <https://foxminded.ua/progressive-web-app/>
6. Frontend, Backend і Fullstack: що це таке та в чому відмінності. URL: <https://goit.global.ua/articles/frontend-backend-i-fullstack-shcho-tse-take-ta-v-chomu-vidminnosti/>
7. Introduction to HTML. URL: https://www.w3schools.com/html/html_head.asp
8. CSS Introduction. URL: https://www.w3schools.com/css/css_intro.asp
9. What is JavaScript? URL: <https://umbraco.com/knowledge-base/javascript/>
10. Introduction to Node.js. URL: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
11. Що таке Django? URL: https://w3schoolsua.github.io/django/django_intro.html#gsc.tab=0
12. Ruby Programming Language. URL: <https://www.geeksforgeeks.org/ruby-programming-language>
13. Що таке база даних? URL: <https://hostiq.ua/wiki/ukr/database/>
14. MySQL Overview: Key Features, Benefits, and Use Cases. URL: <https://www.openlogic.com/blog/mysql-overview>
15. About PostgreSQL. URL: <https://www.postgresql.org/about/>
16. What Is MongoDB? URL: <https://www.purestorage.com/knowledge/what-is-mongodb.html>

17. What Is Cloud Computing? URL: <https://www.investopedia.com/terms/c/cloud-computing.asp>
18. What is DevOps? URL: <https://learn.microsoft.com/en-us/devops/what-is-devops>
19. How to Set the Right DevOps Goals and Align Them with Business Objectives? URL: <https://brainhub.eu/library/how-to-set-the-right-devops-goals>
20. What is Git version control? URL: <https://about.gitlab.com/topics/version-control/what-is-git-version-control/>
21. Grafana: Powerful Metrics Analytics and Visualization. URL: <https://muditmathur121.medium.com/grafana-powerful-metrics-analytics-and-visualization-276457150594>
22. What is Docker? URL: <https://docs.docker.com/get-started/docker-overview/>
23. Jenkins User Documentation. URL: <https://www.jenkins.io/doc/>
24. What is an API? URL: <https://www.ibm.com/think/topics/api>
25. What is a RESTful API? URL: <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>
26. Introduction to JSON Web Tokens. URL: <https://jwt.io/introduction/>
27. OAuth 2.0. URL: <https://oauth.net/2/>
28. Digital Footprint: Concept, Examples, and How to Protect It. URL: <https://vida.id/en/blog/digital-footprint>
29. What Is First-Party Data (+ Second-Party and Third-Party Data)? URL: <https://blog.hubspot.com/service/first-party-data>
30. First-Party Data vs Third-Party Data: Navigating the Digital Marketing Landscape. URL: <https://www.storyly.io/post/first-party-data-vs-third-party-data>
31. What is Data Preparation? Process & Examples. URL: <https://www.matillion.com/learn/blog/data-preparation>
32. What is Data Preparation? A 6-Step Guide to Clean, Transform, and Optimize Data for Analysis URL: <https://firsteigen.com/blog/data-preparation/>
33. Apriori Algorithm. URL: <https://www.geeksforgeeks.org/apriori-algorithm>
34. What is Time Series Analysis? URL: <https://www.sigmacomputing.com/blog/what-is-time-series-analysis>

35. What Is Recency, Frequency, Monetary Value (RFM) in Marketing? URL: <https://www.investopedia.com/terms/r/rfm-recency-frequency-monetary-value.asp>
36. K means Clustering – Introduction. URL: <https://www.geeksforgeeks.org/k-means-clustering-introduction/>
37. What is collaborative filtering? URL: <https://www.ibm.com/think/topics/collaborative-filtering>
38. Understanding Flask Framework: Installation & features. URL: <https://www.analyticsvidhya.com/blog/2021/10/flask-python/>
39. Flask-SQLAlchemy Documentation. URL: <https://flask-sqlalchemy.readthedocs.io/en/stable/quickstart/>
40. SQLite Documentation. URL: <https://www.sqlite.org/about.html>

ДОДАТКИ

ДОДАТОК А

Таблиця: survey_response												
	id	user_id	group_type	application_area	age_group	usage_specifics	restrictions	brand_type	packaging_type	price_range	sales_possibility	discount_availability
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1	1	продовольча	хлібобулочні вироби	звичайна	без обмежень	інше	бренд	пакування	1	без пільг	наявність
2	2	2	продовольча	молочні продукти	звичайна	без обмежень	інше	бренд	пакування	1	без пільг	відсутність
3	3	4	продовольча	молочні продукти	звичайна	без обмежень	інше	бренд	пакування	1	без пільг	наявність
4	4	5	продовольча	молочні продукти	звичайна	без обмежень	інше	бренд	пакування	1	без пільг	наявність
5	5	6	продовольча	м'ясні вироби	звичайна	без обмежень	інше	бренд	пакування	1	без пільг	наявність

Таблица: user

id	username	email	password_hash	has_completed_survey
Фи...	Фильтр	Фильтр	Фильтр	Фильтр
1	1 BestJokeer	belarax1406@gmail.com	script:...	1
2	2 ІванН	bedboxlivechannel@gmail.com	script:...	1
3	3 Іванн	belarax1406@gmail.com	script:...	0
4	4 Ваняя	belarax1406@gmail.com	script:...	1
5	5 Абобаф	belarax1406@gmail.com	script:...	1
6	6 Іван	beylarax1406@gmail.com	script:...	1

id	name	description	price	image_file	product_group_type	product_application_area
Фи...	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1 Ковбаса Casademont «Фуєт» ...	Сиров'ялена іспанська ...	289.9	sausage.jpg	продовольча	м'ясні вироби
2	2 Сир твердий Zveny Gora ...	Класичний твердий сир ...	59.99	cheese.jpg	продовольча	молочні продукти
3	3 Молоко "ПростоНаше" 2.5% 900 мл	Пастеризоване молоко у ...	44.9	milk.jpg	продовольча	молочні продукти
4	4 Яйця курячі СО 10 шт	Столові курячі яйця ...	68.0	eggs.jpg	продовольча	яйця
5	5 Макарони Barilla Spaghetti №5	Тверді сорти пшениці, ...	54.99	pasta.jpg	продовольча	бакалія
6	6 Хліб Дарницький нарізаний 500г	Традиційний житньо-...	27.9	bread.jpg	продовольча	хлібобулочні вироби
7	7 Олія соняшникова Щедрий Дар 85...	Рафінована дезодорована...	81.9	oil.jpg	продовольча	кулінарія
8	8 Огірки короткоплідні	Свіжі українські огірки	12.99	cucu.jpg	продовольча	овочі
9	9 Банани	Стигли банани з Еквадору	52.9	banana.jpg	продовольча	фрукти
10	10 Вода негазована Моршинська 1.5...	Мінеральна природна ...	27.0	water.jpg	продовольча	напої
11	11 Шоколад чорний Millennium 80% ...	Гіркий шоколад для ...	100.9	chock.jpg	продовольча	солодощі
12	12 Чай чорний Ahmad English ...	Класичний чорний чай в ...	129.0	tea.jpg	продовольча	чай
13	13 Кава мелена L'OR Dark Roast ...	Мелена кава темного ...	349.0	coffee.jpg	продовольча	кава
14	14 Пюре Gerber яблуко-банан дитяче	Натуральне фруктове пюр...	49.0	mash.jpg	продовольча	дитяче харчування
15	15 Сік Rich яблучний 1л	Натуральний 100% ...	95.9	juice.jpg	продовольча	напої
16	16 Мило рідке Dove крем-мило 500 ...	Рідке мило з кремовою ...	129.0	soap.jpg	непродовольча	гігієна
17	17 Пральний порошок Persil автома...	Засіб для прання з ...	349.0	powder.jpg	непродовольча	побутова хімія
18	18 Зубна паста Colgate Total	Комплексний догляд за ...	193.99	paste.jpg	непродовольча	гігієна
19	19 Серветки вологі Smile Baby 100...	Вологі серветки для ...	120.0	paper.jpg	непродовольча	дитяча гігієна
20	20 Батон Київхліб Пшеничний	Свіжоспечений білий хлі...	29.9	breadd.jpg	продовольча	хлібобулочні вироби
21	21 Ковбаса варена «Любительська»,...	Варена ковбаса зі ...	29.99	sauprem.jpg	продовольча	м'ясні вироби
22	22 Ковбаски гриль «Французькі», Т...	Напівфабрикат з м'яса ...	95.0	saumeat.jpg	продовольча	м'ясні вироби
23	23 Біфштекс в беконі зі спеціями,...	Біфштекс із яловичини т...	25.4	stale.jpg	продовольча	м'ясні вироби
24	24 Чевапчічі зі спеціями, у беконі	Балканські м'ясні ...	25.4	tube.jpg	продовольча	м'ясні вироби
25	25 Ковбаски курячі фірмові, ...	Курячі ковбаски зі ...	25.9	tube2.jpg	продовольча	м'ясні вироби
26	26 Яблука Голден Делішес	Зелені солодкі яблука, ...	8.9	apple.jpg	продовольча	фрукти
27	27 Апельсини	Апельсини середнього ...	7.9	orange.jpg	продовольча	фрукти
28	28 Груші Конференція	Соковиті груші світло-...	16.9	noapple.jpg	продовольча	фрукти
29	29 Лимони	Свіжі лимони, без ...	13.9	lemon.jpg	продовольча	фрукти

product_age_group	product_usage_specifics	product_restrictions	product_brand_type	product_packaging_type	product_sales_possibility	product_has_discount
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
звичайна	без обмежень	інше	бренд	пакування	без пільг	0
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	без хімічних	бренд	пакування	без пільг	1
звичайна	без обмежень	без хімічних	ноунейм	пакування	без пільг	0
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	0
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	органіка виробництво	ноунейм	без пакування (розсип)	без пільг	0
звичайна	без обмежень	інше	ноунейм	без пакування (розсип)	без пільг	0
звичайна	без обмежень	без хімічних	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	0
звичайна	без обмежень	без хімічних	бренд	пакування	без пільг	1
звичайна	без обмежень	без хімічних	бренд	пакування	без пільг	1
діти	обмеження	органіка виробництво	бренд	пакування	пільги	0
звичайна	без обмежень	без хімічних	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	обмеження	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
діти	обмеження	без хімічних	бренд	пакування	пільги	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	0
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	бренд	пакування	без пільг	1
звичайна	без обмежень	інше	ноунейм	без пакування (розсип)	без пільг	0
звичайна	без обмежень	інше	ноунейм	без пакування (розсип)	без пільг	0
звичайна	без обмежень	інше	ноунейм	без пакування (розсип)	без пільг	0
звичайна	без обмежень	інше	ноунейм	без пакування (розсип)	без пільг	0

ДОДАТОК Б

ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загальноновизнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»; що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації; що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів; що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)

(підпис)