

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ

(назва теми)

РЕКОМЕНДУВАННЯ ШОКОЛАДУ

RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS

FOR RECOMMENDING THE CHOCOLATE

Виконав(ла): студент(ка) 2 курсу, групи КНТ-123м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ДЕМЧЕНКО М.Ю.

(ПРІЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В.М.

(ПРІЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРІЗВИЩЕ та ініціали)

2024_____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет _____ КНТ _____
Кафедра _____ програмних засобів _____
Ступінь вищої освіти _____ магістр _____
Спеціальність _____ 121 Інженерія програмного забезпечення _____
(код і найменування)
Освітня програма (спеціалізація) _____ Інженерія програмного забезпечення _____
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф. _____
Сергій СУББОТІН
« _____ » _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ДЕМЧЕНКА Микити Юрійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів
рекомендування шоколаду. Research and Software Implementation of Methods for
Recommending the Chocolate

керівник проєкту (роботи) к.т.н., доцент ЛЬОВКІН Валерій Миколайович _____,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 7 » жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 01 грудня 2024 року _____
3. Вихідні дані до проєкту (роботи) _____ рекомендована література, технічне
завдання _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали
і методи. 3. Проєктування програмного забезпечення. 4. Основні рішення
щодо реалізації програмного забезпечення. 5. Експлуатація, тестування та
експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість
слайдів, плакатів)
Слайди презентації _____

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ЛЬОВКІН В.М., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання « 06 » вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Дослідження проблеми	3-5 тижні	Розділ 2
4	Проектування та розробка архітектури програми	6 тиждень	Розділ 3
5	Розробка програми	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження	9-10 тижні	Розділ 5
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	11 тиждень	Додатки
8	Захист роботи	12 тиждень	

Студент(ка)

_____ Микита ДЕМЧЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Валерій ЛЬОВКІН
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
103 с., 26 рис., 3 табл., 3 дод., 11 джерел.

СОРТ ШОКОЛАДУ, ПЛИТКА ШОКОЛАДУ, ВИРОБНИК,
РЕКОМЕНДУВАННЯ, ВЕБЗАСТОСУНОК, ЗАМОВЛЕННЯ.

Об'єкт роботи – процес рекомендування шоколаду.

Предмет роботи – методи рекомендування шоколаду.

Метою роботи є розробка програмного забезпечення підтримки продажу та рекомендування шоколаду для інформування користувачів на різних етапах введення шоколаду в продаж.

Матеріали, методи та технічні засоби: мова Python разом з екосистемою Python-пакетів для розробки, фреймворк Django, система керування базами даних MySQL.

Результати. Створена програма призначена для виконання замовлення як окремих сортів шоколаду, так і плиток шоколаду, шоколадних цукерок або інших виробів з шоколаду, оцінювання власних вражень від споживання цих сортів та плиток шоколаду, використання цих даних для рекомендування шоколаду користувачам. Оцінювання здійснюється з врахуванням року врожаю.

Висновки. Наукова новизна роботи полягає в методі рекомендування шоколаду, який виконує прогнозування відсутніх оцінок клієнтами сортів шоколаду за допомогою моделі XGBoost, визначає перетворення основних характеристик сорту шоколаду на конкретні вхідні ознаки, що дозволяє виконувати рекомендування шоколаду за допомогою виконання кластеризації на основі методу k-середніх.

Галузь використання. Компанії, які займаються продажем шоколаду певних сортів або різних шоколадних виробів.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 103 pages, 26 figures, 3 tables, 3 appendixes, 11 sources.

SORT OF CHOCOLATE, CHOCOLATE BAR, MANUFACTURER, RECOMMENDATION, WEBSITE, ORDER.

The object of the work is the chocolate recommendation process.

The subject of the work is chocolate recommendation methods.

The purpose of the work is the development of software for sales support and chocolate recommendation to inform users at various stages of introducing chocolate into the market.

Materials, methods and technical means: the Python language together with the ecosystem of Python packages for development, the Django framework, the MySQL database management system.

The results. The created program is designed for ordering both individual sorts of chocolate and chocolate bars, chocolate candies or other chocolate products, evaluating one's own impressions of consuming these chocolate sorts and bars, using this data to recommend chocolate to users. The assessment is carried out taking into account the year of harvest.

Conclusions. The scientific novelty of the work is in the method of recommending chocolate, which performs the prediction of missing ratings of chocolate sorts by customers using the XGBoost model, determines the transformation of the main characteristics of the chocolate sorts into specific input features, which allows to recommend chocolate by performing clustering based on the k-means method.

Field of use. Companies that sell chocolate of certain sorts or various chocolate products.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Огляд вебзастосунків продажу шоколаду	10
1.2 Порівняння аналогів розробки вебзастосунку продажу шоколаду	13
1.3 Висновки за розділом 1	15
2 Матеріали і методи.....	16
2.1 Задача рекомендації шоколаду	16
2.2 Метод рекомендації шоколаду.....	17
2.3 Висновки за розділом 2	19
3 Проектування програмного забезпечення	21
3.1 Вибір мови та засобів розробки.....	21
3.2 Моделювання роботи з програмою за функціональними вимогами	22
3.3 Проектування структури бази даних.....	29
3.4 Висновки за розділом 3	39
4 Основні рішення щодо реалізації програмного забезпечення	41
4.1 Порядок взаємодії користувача з програмою підтримки продажу та рекомендації шоколаду	41
4.2 Рішення з реалізації вебзастосунку	42
4.3 Висновки за розділом 4	58
5 Експлуатація, тестування та експериментальне дослідження програми	59
5.1 Опис програми.....	59
5.2 Експлуатація програми.....	59
5.3 Експериментальне дослідження	65
5.4 Висновки за розділом 5	67
Висновки	68
Перелік джерел посилання	69
Додаток А Технічне завдання	70

Додаток Б Текст програми	75
Додаток В Слайди презентації.....	96

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- MAE – Mean Absolute Error;
MSE – Mean Squared Error;
RMSE – Root Mean Squared Error.
XGBoost – eXtreme Gradient Boosting.

ВСТУП

Ринок шоколаду величезний на цей момент, зокрема в 2024 році цей ринок був оцінений у 130,72 мільярди американських доларів, при цьому до 2030 року ринок має досягти 172,89 мільярдів американських доларів [1]. У той же час активно розвиваються ніші цього ринку: світовий ринок шоколаду преміум-класу має розмір, який оцінювався в 38,74 мільярди американських доларів в 2024 році, але має досягти 42,30 мільярдів американських доларів до 2025 року та 85,61 мільярдів американських доларів до 2033 року [2].

Крафтове виробництво, використання преміальних зерен какао є прикладами розвитку ринку шоколаду. Як будь-які процеси зараз у світі, цей також має забезпечуватися відповідним програмним забезпеченням. Однак просто існування програмного забезпечення для продажу шоколаду недостатньо, адже кількість сортів, різних варіантів шоколаду у вигляді плиток, цукерок є багато. Окрім того рік врожаю може впливати на результати, тому важливо створити відповідну систему оцінювання, що дозволило б орієнтуватися всім користувачам серед наявних сортів, варіантів виробів з шоколаду. Відповідно в залежності від характеристик шоколаду, який тільки з'являється в самій програмі, тобто у продавця шоколаду, необхідно оцінити його сприйняття клієнтами. Для цього необхідно забезпечити метод рекомендації шоколаду, який би це підтримував зокрема і в складі програмного забезпечення.

Метою роботи є розробка програмного забезпечення підтримки продажу та рекомендації шоколаду для інформування користувачів на різних етапах введення шоколаду в продаж. Це пов'язано з тим, що під час появи шоколаду в магазині він ще не оцінений користувачами, відповідно орієнтуватися при придбанні на оцінку, на відгуки клієнтів неможливо. Тому необхідно виконати прогнозування, а тоді такий шоколад разом з шоколадом, який був вже високо оцінений, може бути рекомендований клієнтам.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд вебзастосунків продажу шоколаду

Велика кількість різних продавців шоколаду призводить до наявності величезної кількості вебзастосунків, які такий продаж шоколаду клієнтам підтримують. Під час огляду вебзастосунків продажу шоколаду розглянуто велику кількість різних застосунків з різних країн, тільки частину з них винесено до опису нижче.

Neuhaus (рис. 1.1) є бельгійським продавцем шоколаду, який присутній на ринку з 1857 року, надаючи різні варіанти шоколадних виробів для продажу [3]. Вебзастосунок містить розділ магазин, який дозволяє виконувати пошук потрібних шоколадних виробів. Пошук реалізується за типом шоколаду, типом виробу з шоколаду, вартістю, кількістю. Додаткових фільтрів під час пошуку шоколаду немає. Кожен шоколадний виріб має свою сторінку, що містить вартість, опис, перелік інгредієнтів, харчову цінність шоколаду, фотографії. При цьому ні інгредієнти, ні харчова цінність під час пошуку не задіяні.

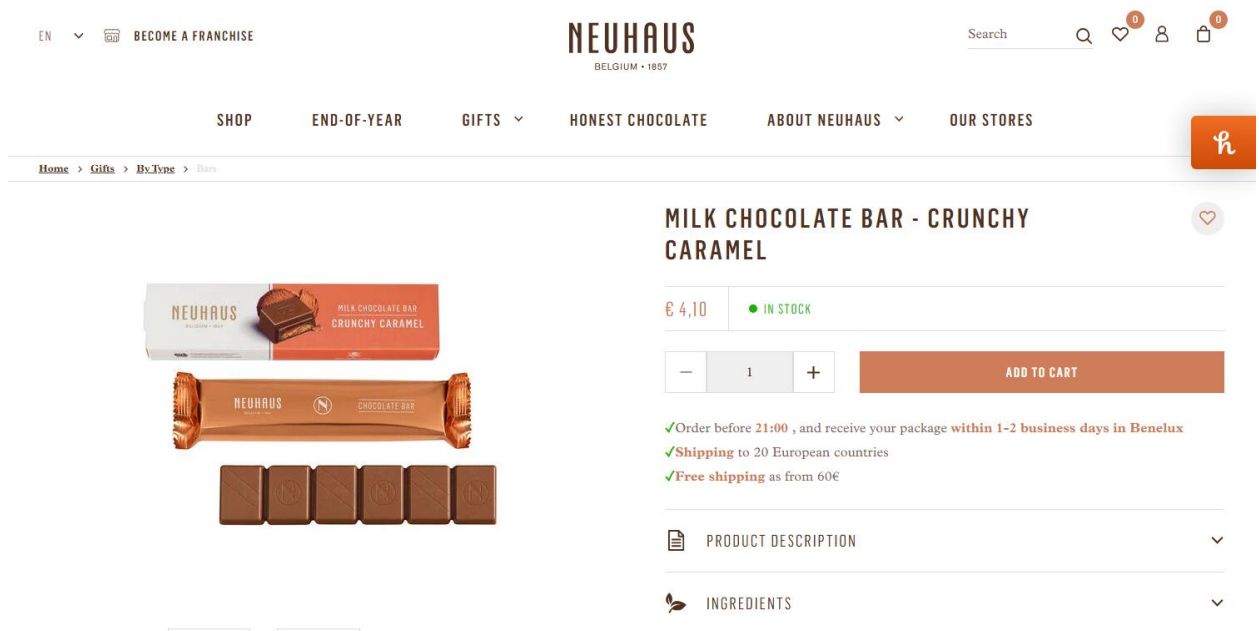


Рисунок 1.1 – Вебсторінка шоколадної плитки у вебзастосунку Neuhaus

Belgian Chocolate Makers (рис. 1.2) також представляють бельгійських виробників шоколаду, що є одними з найавторитетніших у світі, тому і представлені активно в результатах огляду [4]. Вебзастосунок має ряд пунктів меню, які певним чином за типами виробів групують вироби з шоколаду. При виборі пункту відображається відповідний набір виробів з шоколаду з можливістю фільтрації цього набору. Окрім того є засіб пошуку шоколадних виробів, який дозволяє задати назву та фільтрувати результати за переліком інгредієнтів, за вартістю, за вагою, за типом шоколаду, за відсотком вмісту какао, за країною походження зерна, за особливостями (відсутність алкоголю, глютену, придатність для веганів, вміст барвників).

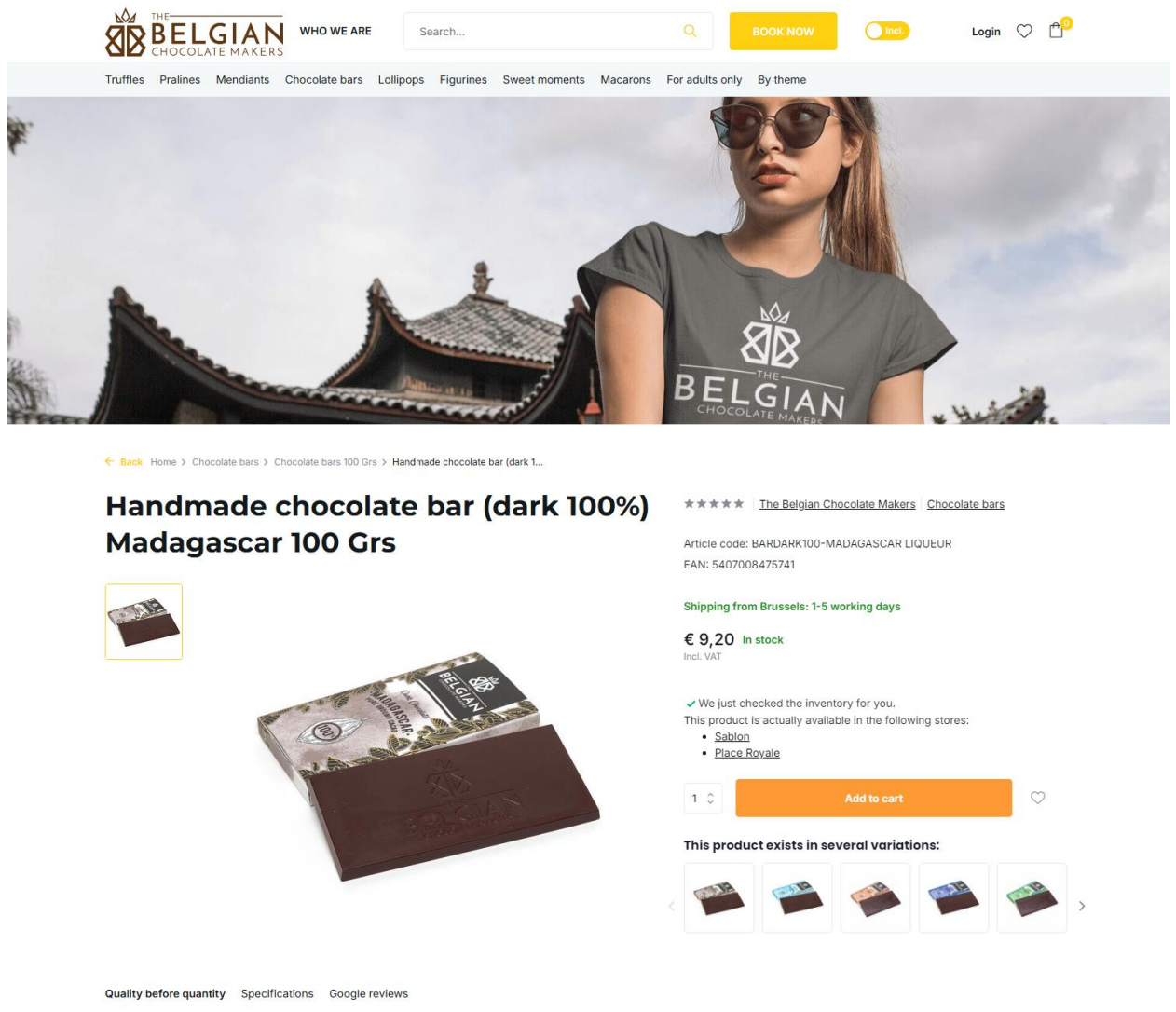


Рисунок 1.2 – Вебсторінка шоколадної плитки у вебзастосунку Belgian Chocolate Makers

Відгуки можуть бути створені тільки через Google, власної системи оцінювання немає, відповідно оцінити, наприклад, шоколад на основі зерна певного року збору врожаю неможливо.

За кожним шоколадним виробом визначається набір тегів, що цей виріб характеризують.

Вебзастосунок компанії Pierre Marcollini (рис. 1.3) містить інформацію про конкретні магазини з продажу шоколаду, перелік виробів, розділених на розділи шоколаду, випічки з шоколаду, подарунки [5].



Рисунок 1.3 – Вебсторінка шоколадної плиточки у вебзастосунку Pierre Marcollini

Вебзастосунок Pierre Marcollini відображає шоколадні вироби за відповідним розділом меню або дозволяє виконувати пошук потрібного виробу. Під час пошуку можна обрати категорію, вартість, аромат, призначення.

Залишити відгуки про конкретні вироби з шоколаду неможливо.

Вебзастосунок Mike & Besky (рис. 1.4) містить до засобів продажу додаткові засоби, що включають новини, відображення інформації про сорти

какао, події [6]. Пошук здійснюється за назвою та з визначенням категорії. Додаткових фільтрів пошук не містить.

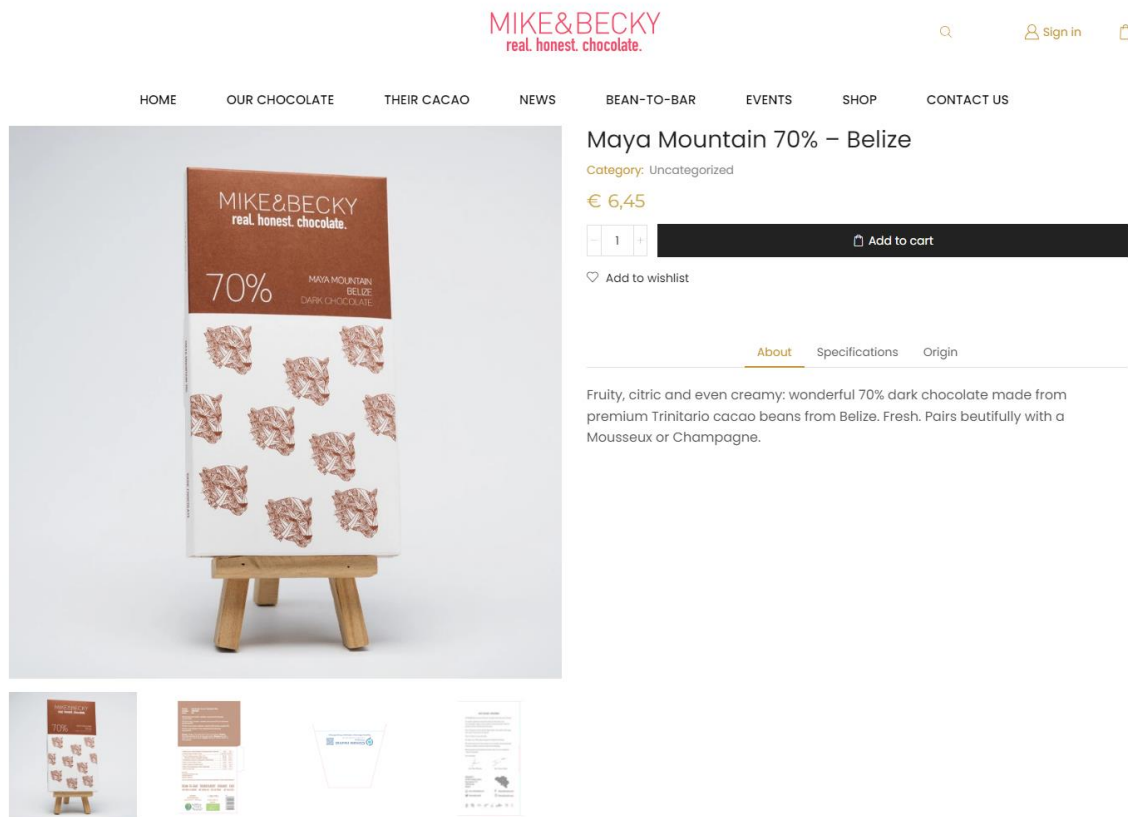


Рисунок 1.4 – Вебсторінка шоколадної плитки у вебзастосунку Mike & Becky

За кожним виробом з шоколаду визначено його опис, вартість, походження, особливості, в тому числі перелік інгредієнтів.

Кожен з виробів з шоколаду можна додати до улюблених, визначивши власне бажання. Ці засоби достатні в абсолютно кожному з розглянутих вебзастосунків.

1.2 Порівняння аналогів розробки вебзастосунку продажу шоколаду

До фінального порівняння аналогів розробки вебзастосунку продажу шоколаду внесено вебзастосунки Belgian Chocolate Makers, Pierre Marcollini та

саму розробку (табл. 1.1).

Таблиця 1.1 – Порівняння аналогів розробки вебзастосунку продажу шоколаду

Показник	Belgian Chocolate Makers	Pierre Marcollini	Розробка
Замовлення шоколадних виробів	Так	Так	Так
Збереження шоколадних виробів у бажання	Так	Так	Так
Пошук шоколадних виробів	Вартість, тип, вміст какао, країна походження, особливості	Вартість, тип, смак, призначення	Вартість, тип шоколаду, виробу, харчова цінність, країна походження, вміст какао, сорт, рейтинг, інгредієнти
Визначення харчової цінності та складу інгредієнтів шоколадного виробу	Ні	Так	Так
Оцінювання шоколадних виробів у вебзастосунку з врахуванням вражаю	Ні	Ні	Так
Рекомендування шоколадних виробів	Ні	Ні	Так
Прогнозування оцінки шоколадного виробу	Ні	Ні	Так

Розробка відрізняється від наявних аналогів найголовніше наявністю можливостей оцінювання шоколадних виробів у вебзастосунку з врахуванням врожаю, рекомендування шоколадних виробів та включеної в процес рекомендування можливості прогнозування оцінки шоколадного виробу. Окрім того успадковано від вебзастосунку Pierre Marcollini можливість визначення харчової цінності та складу інгредієнтів шоколадного виробу. Також забезпечено організацію пошуку шоколадних виробів у тому числі на основі інгредієнтів, харчової цінності продукту.

1.3 Висновки за розділом 1

Завдання для реалізації в роботі:

- виконати аналіз вебзастосунків продажу шоколаду;
- розробити метод рекомендування шоколаду;
- виконати проектування підтримки продажу та рекомендування шоколаду;
- реалізувати програму підтримки продажу та рекомендування шоколаду;
- виконати експериментальне дослідження рекомендування шоколаду на основі прогнозування оцінок шоколаду.

Розділ 1 містить вже виконаний аналіз вебзастосунків продажу шоколаду, виділення можливостей існуючих вебзастосунків, аналіз цих можливостей стосовно виділення особливостей розробки.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Задача рекомендації шоколаду

Задача рекомендації шоколаду полягає в тому, щоб визначити подібні шоколадні сорти або шоколадні плитки для того набору шоколадних сортів або шоколадних плиток, які сподобались користувачу-клієнту. При цьому кожен сорт шоколаду або шоколадна плитка представлена такими характеристиками:

- номер запису;
- компанія-виробник шоколаду;
- країна, в якій розташовано виробника шоколаду;
- рік створення відгуків або врожай шоколаду, за яким створено відгуки;
- походження зерна какао або назва плитки шоколаду;
- відсоток вмісту какао в шоколаді;
- базовий набір інгредієнтів;
- найбільш визначні характеристики шоколаду: значення перелічені через кому та визначають смакові характеристики шоколаду;
- рейтинг (оцінка) шоколаду.

Існує набір даних, в якому за кожним екземпляром відомо значення кожної характеристики. Відповідно вхідними характеристиками є всі окрім рейтингу (оцінки) шоколаду. Вихідною ознакою є рейтинг (оцінка) шоколаду. Розглянемо, чому рейтинг є саме вихідною ознакою, незважаючи на загальну задачу.

Відповідно задача рекомендації шоколаду пов'язана з формуванням певних наборів шоколадних сортів або плиток і визначенні того, до якого набору схильний за своїм смаком клієнт. Проте для того, щоб всі шоколадні сорти та плитки розподілити, потрібно мати оцінку для кожного з цих сортів або плиток шоколаду. Усі інші характеристики мають початково встановлені та наперед відомі значення. Відповідно для тих екземплярів даних, для яких

може виконуватися рекомендація, але для яких невідомою є оцінка, потрібно виконати прогнозування цієї оцінки (рейтинг). Тобто фактично задача рекомендації шоколаду містить в собі підзадачу прогнозування оцінки шоколаду для тих випадків, коли сорт шоколаду або шоколадний виріб не має ще середньої оцінки визначеної користувачами вебзастосунку, або цій оцінці ще не можна довіряти, оскільки вона не є стабільною. Оцінка не може вважатися стабільною, коли передбачає оцінювання декількома клієнтами. Якщо це так, то будь-яка наступна оцінка може значно вплинути на середню оцінку, яка фактично тоді є проміжною та потребує уточнення. Тому навіть якщо шоколадний виріб тільки з'явився на ринку, а він вже має оцінку від клієнта, це не робить таку середню оцінку стабільним показником, на якому можна ґрунтуватись в подальшому для створення системи рекомендацій.

2.2 Метод рекомендації шоколаду

Для розв'язання задачі рекомендації шоколаду та підзадачі прогнозування оцінки шоколаду запропоновано цілісний метод рекомендації шоколаду, що передбачає роботу над такою послідовністю етапів.

На першому етапі має реалізуватись збір даних про шоколадні вироби або сорти шоколаду (в залежності від того, на що саме направлена ця система рекомендацій шоколаду). При цьому не потрібно перемішувати в одній вибірці даних одночасно і сорти шоколаду, в шоколадні вироби.

Кожен запис про шоколадний виріб або сорт повинен характеризуватись щонайменше наступними характеристиками:

- компанія-виробник шоколаду;
- країна, в якій розташовано виробника шоколаду;
- рік створення відгуків або врожай шоколаду, за яким створено відгуки;
- походження зерна какао або назва плитки шоколаду;

- відсоток вмісту какао в шоколаді;
- базовий набір інгредієнтів;
- найбільш визначні характеристики шоколаду: значення перелічені через кому та визначають смакові характеристики шоколаду;
- рейтинг (оцінка) шоколаду.

Після цього для перетворення цього набору записів на навчальну вибірку даних необхідно виконати наступні трансформації:

- визначити числові значення для компаній-виробників шоколаду, для чого сформувати набір всіх виробників, представити кожного виробника числом, а відповідне значення характеристики у екземпляра даних у навчальній вибірці визначити таким числом;

- визначити числові значення для країни, в якій розташовано виробника шоколаду, для чого сформувати набір всіх країн, представити кожну країну числом, а відповідне значення характеристики у екземпляра даних у навчальній вибірці визначити таким числом;

- проаналізувати характеристику походження зерна какао або назви плитки шоколаду щодо виділення року врожаю, після чого порівняти його з роком відгуку, а якщо вони не співпадають, то задати саме рік врожаю;

- з походження зерна залишити в характеристиці саме регіон;

- базовий набір інгредієнтів, зважаючи на типовість складу, перетворити на числові значення, для чого сформувати набір всіх можливих інгредієнтів, представити кожний набір інгредієнтів числом, а відповідне значення характеристики у екземпляра даних у навчальній вибірці визначити таким числом;

- найбільш визначні характеристики шоколаду розподілити на окремі характеристики, після чого сформувати набір можливих (фактично смакових) характеристик шоколаду, а для кожного елемента такого набору для кожного шоколадного виробу встановити, чи входить ця характеристика в опис шоколаду, чи ні, відповідно встановивши значення або 0, або 1;

– виконати нормування оцінок шоколаду зі зведенням значень в інтервал від 0 до 1.

На другому етапі створити модель на основі eXtreme Gradient Boosting (XGBoost) [7], яка дозволяє забезпечити ансамбль дерев рішень для застосування його для прогнозування оцінки шоколаду. Виходячи з експериментального дослідження прогнозування оцінки шоколаду, варто встановити кількість дерев рівною 500, проте варто враховувати також реальний обсяг навчальної вибірки. Для створеної моделі прогнозування оцінки необхідно виконати її навчання на сформованій на першому етапі навчальній вибірці, а тоді застосувати навчену модель для визначення оцінок шоколаду, для якого така оцінка невідома або не є надійною, а значення потрібних вхідних характеристик при цьому задані.

На третьому етапі необхідно виконати кластеризацію всіх даних, які входять до початкової навчальної вибірки та тих даних, які були додатково до неї внесені на другому етапі, тобто екземплярів, для яких не була відома оцінка. Кластеризація виконується на основі методу k-середніх.

На четвертому етапі користувач визначає той шоколад, який відповідає його смакам. Для цього або він робить це безпосередньо, або використовуються виставлені оцінки. Якщо оцінка не менше ніж 4,5, то слід вважати, що такий шоколад оцінено користувачем позитивно. Це мають бути саме персональні оцінки, виставлені користувачем. Після цього визначається, до якого саме зі сформованих кластерів найчастіше належить шоколад, який обрав користувач. Відповідно той шоколад, який до вподобаного користувачем набору не належить, але належить до цього кластеру, який обрано, відноситься до рекомендацій та може демонструватися користувачу.

2.3 Висновки за розділом 2

Наукова новизна роботи полягає в методі рекомендування шоколаду, який виконує прогнозування відсутніх оцінок клієнтами сортів шоколаду за

допомогою моделі XGBoost, визначає перетворення основних характеристик сорту шоколаду на конкретні вхідні ознаки, що дозволяє виконувати рекомендації шоколаду за допомогою виконання кластеризації на основі методу k-середніх. У розділі 2 розглянуто постановку задачі рекомендації шоколаду та підзадачі прогнозування оцінки шоколаду, описано запропонований метод рекомендації шоколаду.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір мови та засобів розробки

JavaScript є актуальною і активно застосовуваною мовою для веброзробки, в тому числі при створенні рішень на сервері у вигляді надання бекенд-фреймворку для веброзробки, одним з яких є Express.js [8]. Альтернативою є мова Python, яка також застосовується для веброзробки, але є також і потужним рішенням для машинного навчання і роботи з даними, що характерно для розробки в частині рекомендації шоколаду, разом з фреймворком Django [9] (табл. 3.1).

Таблиця 3.1 – Вибір вебфреймворку для розробки програми підтримки продажу та рекомендації шоколаду

	Django	Express.js
Тип фреймворку	Бекенд-фреймворк для веброзробки	Бекенд-фреймворк для веброзробки
Мова програмування	Python	JavaScript
Архітектура та структура застосунку	Чітка структура згідно з Model-View-Template як варіацією Model-View-Controller	Model-View-Controller без вимоги суворого дотримання
Взаємодія з базою даних	Шар об'єктно-реляційного відображення, інтерфейс адміністратора	Тільки інтерфейс для зв'язку з системою управління базою даних
Машинне навчання та робота з даними	Мова Python має більш розгалужену систему засобів	Мова JavaScript має меншу систему засобів

У підсумку обрано фреймворк Django та мову програмування Python, враховуючи, що Django надає шар об'єктно-реляційного відображення, інтерфейс адміністратора окрім просто інтерфейс для зв'язку з системою управління базою даних у Express.js, має чітку структуру на основі шаблону Model-View-Template, що стандартизує розробку, мова Python має більш розгалужену систему засобів, що сприяє реалізації методу рекомендації шоколаду.

3.2 Моделювання роботи з програмою за функціональними вимогами

Користувачами програми є звичайні клієнти, що виступають покупцями шоколаду, та оператори і менеджери компанії з продажу шоколаду, для підтримки діяльності якої і створюється програмне забезпечення.

Користувач-клієнт, який є покупцем шоколаду, потребує для забезпечення роботи з програмою сценаріїв:

- відтворення переліку сортів шоколаду за регіоном або країною;
- відтворення даних про виробника шоколаду;
- відтворення переліку всіх сортів шоколаду виробника;
- відтворення переліку всіх плиток шоколаду виробника;
- відтворення даних сорту шоколаду;
- відтворення даних плитки шоколаду;
- пошук сорту шоколаду;
- пошук плиток шоколаду;
- відтворення переліку плиток шоколаду за типом;
- створення відгуку про сорт шоколаду;
- створення відгуку про плитку шоколаду;
- відтворення переліку відгуків про сорт шоколаду;
- відтворення переліку відгуків про плитку шоколаду;
- відтворення даних користувача-покупця (рис. 3.1).

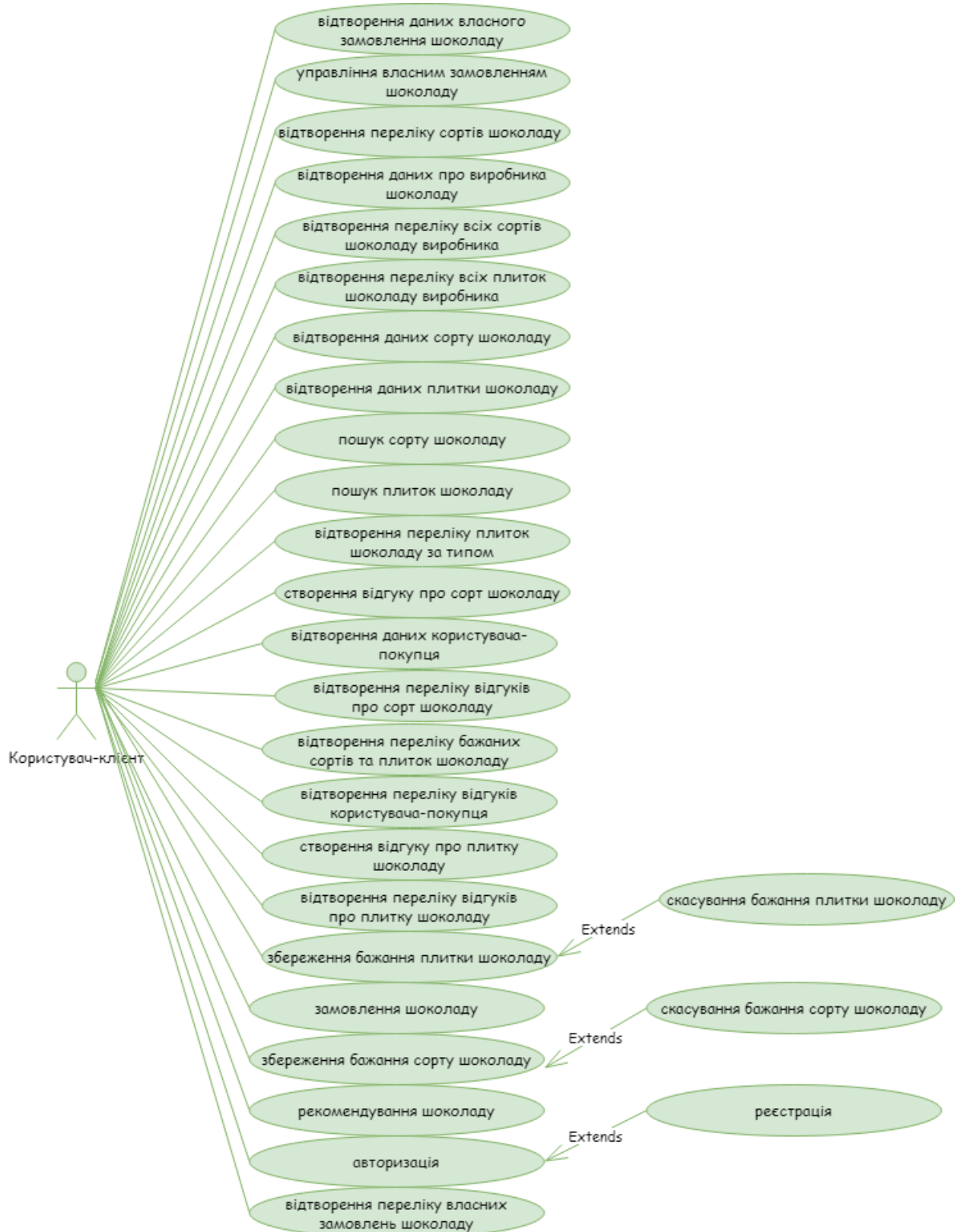


Рисунок 3.1 – Діаграма варіантів використання програми підтримки продажу та рекомендування шоколаду користувачем-клієнтом

Додаткові сценарії користувача-клієнта:

- відтворення переліку відгуків користувача-покупця;
- збереження бажання плитки шоколаду;
- збереження бажання сорту шоколаду;
- відтворення переліку бажаних сортів та плиток шоколаду;
- замовлення шоколаду;
- рекомендування шоколаду;
- відтворення переліку власних замовлень шоколаду;
- управління власним замовленням шоколаду;
- відтворення даних власного замовлення шоколаду;
- авторизація.

Альтернативні сценарії користувача-клієнта:

- скасування бажання плитки шоколаду, що є альтернативою для збереження бажання плитки шоколаду для випадку наявності такого бажання цієї плитки шоколаду у цього клієнта в базі даних;

- скасування бажання сорту шоколаду, що є альтернативою для збереження бажання сорту шоколаду для випадку наявності такого бажання цього сорту шоколаду у цього клієнта в базі даних;

- реєстрація, що є альтернативою для авторизації, коли авторизуватись неможливо через відсутність у клієнта облікового запису.

Оператор компанії з продажу шоколаду потребує для забезпечення роботи з програмою сценаріїв:

- відтворення переліку сортів шоколаду за регіоном або країною;
- відтворення даних про виробника шоколаду;
- відтворення переліку всіх сортів шоколаду виробника;
- відтворення переліку всіх плиток шоколаду виробника;
- відтворення даних сорту шоколаду;
- відтворення даних плитки шоколаду;
- відтворення переліку пропозицій вартості за сортом шоколаду;
- відтворення переліку пропозицій вартості за плиткою шоколаду;
- відтворення даних замовлення шоколаду (рис. 3.2).

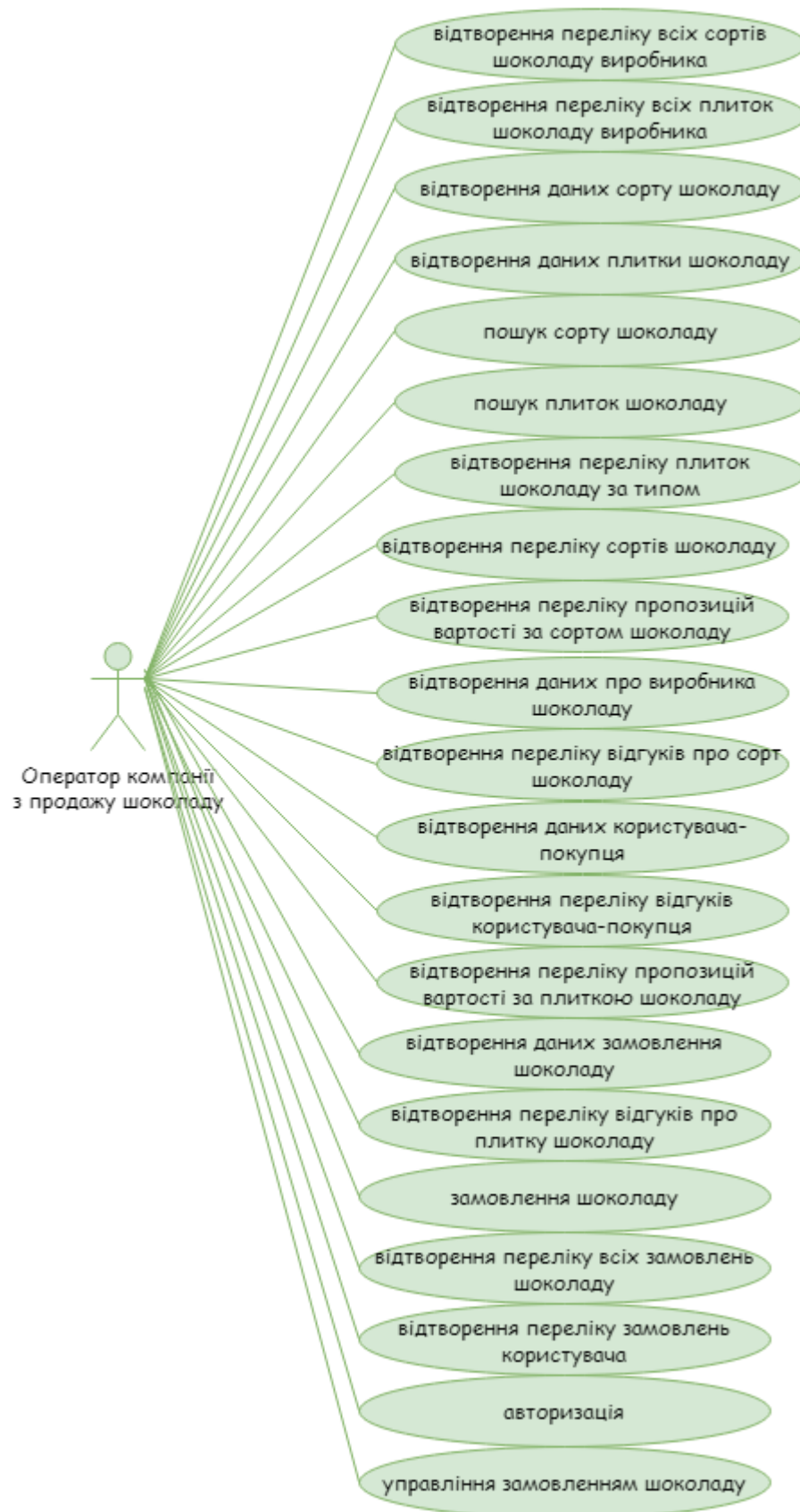


Рисунок 3.2 – Діаграма варіантів використання програми підтримки продажу та рекомендування шоколаду оператором компанії з продажу шоколаду

Також сценарії оператора компанії з продажу шоколаду:

- відтворення переліку всіх замовлень шоколаду;
- відтворення переліку замовлень користувача;
- відтворення переліку відгуків про сорт шоколаду;
- відтворення переліку відгуків про плитку шоколаду;
- відтворення даних користувача-покупця;
- відтворення переліку відгуків користувача-покупця;
- замовлення шоколаду;
- пошук сорту шоколаду;
- пошук плиток шоколаду;
- відтворення переліку плиток шоколаду за типом;
- управління замовленням шоколаду;
- авторизація.

Менеджер компанії з продажу шоколаду потребує для забезпечення роботи з програмою сценаріїв:

- відтворення переліку сортів шоколаду за регіоном або країною;
- відтворення даних про виробника шоколаду;
- відтворення переліку всіх сортів шоколаду виробника;
- відтворення переліку всіх плиток шоколаду виробника;
- відтворення даних сорту шоколаду;
- відтворення даних плитки шоколаду;
- пошук сорту шоколаду;
- пошук плиток шоколаду;
- відтворення переліку плиток шоколаду за типом;
- додавання даних плитки шоколаду;
- додавання даних сорту шоколаду;
- додавання варіанта плитки шоколаду за вагою;
- додавання пропозиції вартості плитки шоколаду за вагою;
- додавання пропозиції вартості сорту шоколаду;
- управління замовленням шоколаду (рис. 3.3).

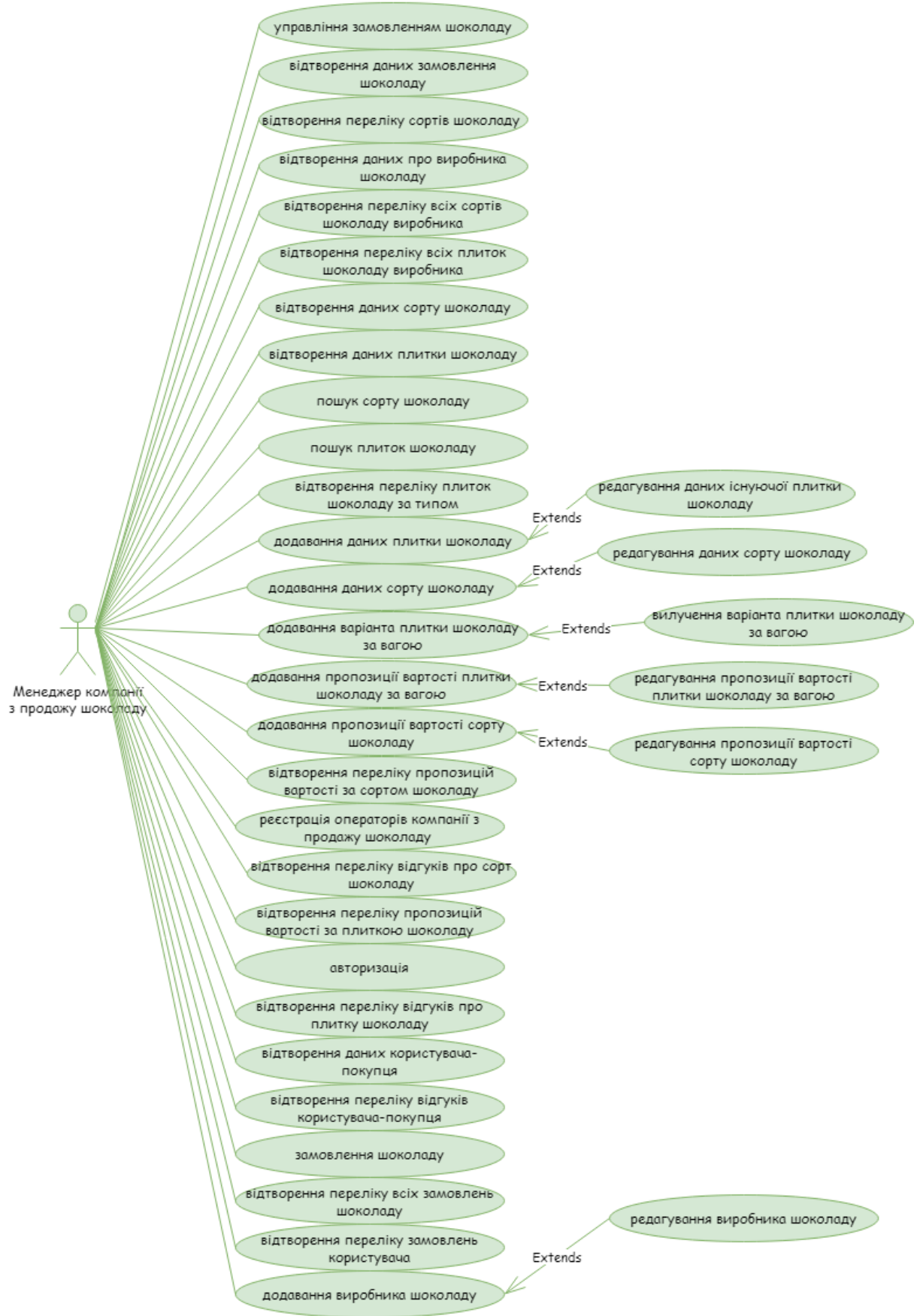


Рисунок 3.3 – Діаграма варіантів використання програми підтримки продажу та рекомендування шоколаду менеджером компанії з продажу шоколаду

Також сценарії менеджера компанії з продажу шоколаду:

- відтворення переліку пропозицій вартості за сортом шоколаду;
- відтворення даних замовлення шоколаду;
- відтворення переліку всіх замовлень шоколаду;
- відтворення переліку замовлень користувача;
- додавання виробника шоколаду;
- відтворення переліку пропозицій вартості за плиткою шоколаду;
- відтворення переліку відгуків про сорт шоколаду;
- відтворення переліку відгуків про плитку шоколаду;
- відтворення даних користувача-покупця;
- відтворення переліку відгуків користувача-покупця;
- замовлення шоколаду;
- авторизація;
- реєстрація операторів компанії з продажу шоколаду.

Альтернативні сценарії менеджера компанії з продажу шоколаду:

– редагування даних існуючої плитки шоколаду, що є альтернативою для додавання даних плитки шоколаду для випадку, коли дані плитки шоколаду внесено в базу даних, але вони потребують змін;

– редагування даних сорту шоколаду, що є альтернативою для додавання даних сорту шоколаду для випадку, коли дані сорту шоколаду внесено в базу даних, але вони потребують змін;

– вилучення варіанта плитки шоколаду за вагою, що є альтернативою для додавання варіанта плитки шоколаду за вагою для випадку, коли цей варіант плитки шоколаду за вагою був внесений помилково;

– редагування пропозиції вартості плитки шоколаду за вагою, що є альтернативою для додавання пропозиції вартості плитки шоколаду за вагою для випадку, коли дані пропозиції вартості плитки шоколаду внесено в базу даних, але вони потребують змін;

– редагування пропозиції вартості сорту шоколаду, що є альтернативою для додавання пропозиції вартості сорту шоколаду для

випадку, коли дані пропозиції вартості сорту шоколаду внесено в базу даних, але вони потребують змін;

– редагування виробника шоколаду, що є альтернативою для додавання виробника шоколаду для випадку, коли дані виробника шоколаду внесено в базу даних, але вони потребують змін

3.3 Проєктування структури бази даних

MySQL є обраною для реалізації та проєктування бази даних в роботі в складі вебзастосунку підтримки продажу та рекомендування шоколаду. Враховуючи потужність, безкоштовність, це рішення є логічним.

Країни, з яких може походити шоколад або його виробники, зведені до таблиці Country з полями:

– id – ідентифікатор країни, з якої може походити шоколад або його виробники;

– name – назва країни, з якої може походити шоколад або його виробники.

Конкретні місця, які фактично є регіонами, де вирощуються кавові зерна, що використовуються для створення шоколаду, зведені до таблиці Chocolateregion з полями:

– id – ідентифікатор регіону, де вирощуються кавові зерна, що використовуються для створення шоколаду;

– country – країна, в якій знаходиться регіон, де вирощуються кавові зерна, що використовуються для створення шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Country;

– region_name – назва регіону, де вирощуються кавові зерна, що використовуються для створення шоколаду.

Виробники шоколаду зведені до таблиці Chocolatemanufacturer з полями:

– id – ідентифікатор виробника шоколаду;

– country – країна, в якій знаходиться основний штаб виробника шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Country;

– company_name – назва компанії як виробника шоколаду;
 – company_website – вебсайт компанії як виробника шоколаду;
 – company_logo – логотип або певна фотографія, яка асоціюється з виробником шоколаду.

Сорти шоколаду зведені до таблиці Chocolate з полями:

– id – ідентифікатор сорту шоколаду;
 – chocolatemanufacturer – виробник, який виготовляє цей сорт шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatemanufacturer;

– sort_name – назва сорту шоколаду;
 – cacao_percent – відсоток вмісту какао в шоколаді цього сорту;
 – base_chocolate_type – базовий тип шоколаду (молочний, чорний, білий), до якого належить цей сорт;

– base_ingredient_formula – базова формула інгредієнтів (набір позначок, який визначає основні інгредієнти створення цього сорту шоколаду);

– review_average – середня оцінка, яку користувачі-клієнти виставили цьому сорту шоколаду;

– predicted_review_average – прогнозована середня оцінка сорту шоколаду;

– bean_region – регіон, де вирощується цей сорт какао, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolateregion;

– description – опис сорту шоколаду текстом.

Типи плиток шоколаду (шоколадних цукерок, батончиків) зведені до таблиці Chocolatetype з полями:

– id – ідентифікатор типу плиток шоколаду;
 – type_name – назва типу плиток шоколаду.

Конкретні плитки шоколаду (шоколадні цукерки, батончики) зведені до таблиці Chocolatebar з полями:

- id – ідентифікатор плитки шоколаду;
- chocolatetype – тип плиток шоколаду, до якого належить ця плитка, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatetype;
- bar_name – назва плитки шоколаду;
- chocolatemanufacturer – виробник плитки шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatemanufacturer;
- cacao_percent – відсоток вмісту какао в плитці шоколаду;
- bardescription – опис плитки шоколаду;
- review_average – середня оцінка, яку клієнти виставили цій плитці шоколаду;
- predicted_review_average – прогнозована середня оцінка плитки шоколаду;
- energy_kcalories – харчова енергія в 100 грамах плитки шоколаду, виражена в кількості кілокалорій;
- total_fat_gram – сумарна кількість жиру в грамах, що міститься в цій плитці шоколаду в 100 грамах;
- saturated_fat_gram – кількість насичених жирів в 100 грамах плитки шоколаду в грамах;
- trans_fat_gram – кількість трансжирів в 100 грамах плитки шоколаду в грамах;
- cholesterol_mgram – кількість холестерину в 100 грамах плитки шоколаду у плитці шоколаду;
- carbohydrate_gram – кількість вуглеводів у грамах в 100 грамах плитки шоколаду;
- dietary_fiber_gram – кількість клітковини в грамах в 100 грамах плитки шоколаду;
- sugar_gram – кількість цукру в 100 грамах плитки шоколаду в грамах;

- protein_gram – кількість білку в 100 грамах плитки шоколаду в грамах;
- sodium_mgram – кількість натрію в 100 грамах плитки шоколаду в міліграмах;
- calcium_mgram – кількість кальцію в 100 грамах плитки шоколаду в міліграмах;
- iron_mgram – кількість заліза в 100 грамах плитки шоколаду в міліграмах;
- salt_gram – кількість солі в 100 грамах плитки шоколаду в грамах;
- vitaminD_ngram – кількість вітаміну Д в 100 грамах плитки шоколаду в нанограмах;
- potassium_mgram – кількість калію в 100 грамах плитки шоколаду в міліграмах;
- added_sugar_gram – кількість цукру в 100 грамах плитки шоколаду в грамах.

Фотографії плитки шоколаду зведені до таблиці Chocolatebarphoto з полями:

- id – ідентифікатор фотографії плитки шоколаду;
- chocolatebar – плитка шоколаду, що представлена на фотографії, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebar;
- photo – фотографія плитки шоколаду.

Фотографії сорту шоколаду зведені до таблиці Chocolatephoto з полями:

- id – ідентифікатор фотографії сорту шоколаду;
- chocolate – сорт шоколаду, який показаний на фотографії, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolate;
- photo – фотографія шоколаду цього сорту.

Варіанти продажу плитки шоколаду за вагою зведені до таблиці Chocolatebarmass з полями:

- id – ідентифікатор варіанта продажу плитки шоколаду за вагою;

- chocolatebar – плитка шоколаду, варіант якої представлений, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebar;
- mass – вага в грамах варіанта продажу плитки шоколаду.

Вміст сортів шоколаду в плитці шоколаду зведені до таблиці Chocolatebarchocolate з полями:

- id – ідентифікатор сорту шоколаду, який міститься в плитці шоколаду;
- chocolatebar – плитка шоколаду, в якій міститься цей сорт шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebar;
- chocolate – сорт шоколаду, який міститься в цій плитці шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolate.

Вартість плитки шоколаду у варіанті за вагою зведено до таблиці Chocolatebarmassprice з полями:

- id – ідентифікатор вартості плитки шоколаду у варіанті за вагою;
- chocolatebarmass – варіант продажу плитки шоколаду за вагою, для якого визначена ця вартість, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebarmass;
- price_value – вартість в гривнях варіанта продажу плитки шоколаду за вагою;
- proposition_start – дата і час, з яких діє ця вартість даного варіанта продажу плитки шоколаду за вагою;
- proposition_end – дата і час, до яких діє ця вартість даного варіанта продажу плитки шоколаду за вагою;
- active – активність застосування цього тарифу для продажу даного варіанта продажу плитки шоколаду за вагою (використовується замість вилучення тарифу).

Вартість сорту шоколаду зведено до таблиці Chocolateprice з полями:

- id – ідентифікатор вартості сорту шоколаду;
- chocolate – сорт шоколаду, для якого визначена ця вартість, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolate;

- price_value – вартість в гривнях сорту шоколаду;
- proposition_start – дата і час, з яких діє ця вартість даного варіанта продажу сорту шоколаду;
- proposition_end – дата і час, до яких діє ця вартість даного варіанта продажу сорту шоколаду;
- active – активність застосування цього тарифу для продажу даного сорту шоколаду (використовується замість вилучення тарифу).

Інгредієнти, з яких виготовляються плитки шоколаду, зведені до таблиці Ingredient з полями:

- id – ідентифікатор інгредієнта, з якого можуть виготовлятися плитки шоколаду;
- name – назва інгредієнта, з якого можуть виготовлятися плитки шоколаду.

Інгредієнти, що входять до складу плитки шоколаду, зведені до таблиці Chocolatebaringredient з полями:

- id – ідентифікатор входження інгредієнта до складу плитки шоколаду;
- chocolatebar – плитка шоколаду, до якої входить цей інгредієнт, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebar;
- ingredient_group – група входження інгредієнта до складу плитки шоколаду (начинка, сам шоколад);
- ingredient – інгредієнт, який входить до складу цієї плитки шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Ingredient.

Врожаї сортів шоколаду зведені до таблиці Chocolateyear з полями:

- id – ідентифікатор врожаю сорту шоколаду;
- harvest_year – рік збору врожаю сорту шоколаду;
- review_average – середня оцінка, яку клієнти виставили сорту шоколаду цього року збору врожаю.

Відгуки про сорти шоколаду за роком збору врожаю зведені до таблиці Chocolatereview з полями:

– id – ідентифікатор відгуку клієнта про сорт шоколаду за роком збору врожаю;

– chocolateyear – рік збору врожаю конкретного сорту шоколаду, за яким створено відгук, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolateyear;

– clientaccount – обліковий запис клієнта, який створив цей відгук про сорт шоколаду за роком збору врожаю, передбачає створення зовнішнього ключа для зв'язку з таблицею Clientaccount;

– review_date – дата створення відгуку клієнта про сорт шоколаду за роком збору врожаю;

– review_mark – оцінка відгуку клієнта про сорт шоколаду за роком збору врожаю.

Відгуки про плитки шоколаду зведені до таблиці Chocolatebarreview з полями:

– id – ідентифікатор відгуку клієнта про плитку шоколаду;

– chocolatebar – плитка шоколаду, за якою створено відгук, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebar;

– clientaccount – обліковий запис клієнта, який створив цей відгук про плитку шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Clientaccount;

– review_date – дата створення відгуку клієнта про плитку шоколаду;

– review_mark – оцінка відгуку клієнта про плитку шоколаду.

Бажання клієнтів щодо сортів шоколаду зведені до таблиці Chocolatewish з полями:

– id – ідентифікатор бажання клієнта щодо сорту шоколаду;

– chocolate – сорт шоколаду, бажання стосовно якого зберіг клієнт, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolate;

– clientaccount – клієнт, який створив це бажання сорту шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Clientaccount;

– wish_date – дата, коли було створено це бажання сорту шоколаду.

Бажання клієнтів щодо плиток шоколаду зведені до таблиці Chocolatebarwish з полями:

– id – ідентифікатор бажання клієнта щодо плиток шоколаду;

– chocolatebar – плитка шоколаду, бажання стосовно якої зберіг клієнт, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebar;

– clientaccount – клієнт, який створив це бажання плитки шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Clientaccount;

– wish_date – дата, коли було створено це бажання плитки шоколаду.

Замовлення шоколаду клієнтами зведені до таблиці Chocolateorder з полями:

– id – ідентифікатор замовлення шоколаду клієнтами;

– clientaccount – клієнт, який створив замовлення шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Clientaccount;

– order_date – дата замовлення шоколаду клієнтами;

– supply_date – дата, коли має бути доставлено замовлення шоколаду клієнту;

– delivery – адреса для доставки замовлення шоколаду;

– price – загальна вартість, яка відповідає замовленню шоколаду клієнтом.

Вміст замовлення шоколаду стосовно плиток шоколаду зведено до таблиці Chocolateorderchocolatebar з полями:

– id – ідентифікатор запису про вміст замовлення шоколаду стосовно плиток шоколаду;

– chocolateorder – замовлення шоколаду, до якого входить ця плитка шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolateorder;

– chocolatebarmass – ваговий варіант плитки шоколаду, яка входить до складу цього замовлення, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebarmass;

– quantity – кількість цих плиток шоколаду, що входить до цього замовлення шоколаду;

– price – вартість сумарно цих плиток шоколаду, що входять до складу цього замовлення шоколаду.

Вміст замовлення шоколаду стосовно сортів шоколаду зведено до таблиці Chocolateorderchocolate з полями:

– id – ідентифікатор запису про вміст замовлення шоколаду стосовно сорту шоколаду;

– chocolateorder – замовлення шоколаду, до якого входить цей сорт шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolateorder;

– chocolate – сорт шоколаду, який входить до складу цієї плитки шоколаду, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolate;

– quantity – кількість в грамах цього сорту шоколаду, яку було замовлено у складі цього замовлення шоколаду;

– price – вартість сумарно цього сорту шоколаду, що входить до складу цього замовлення шоколаду.

Можливі смаки, які можуть асоціюватися з шоколадом, зведені до таблиці Taste з полями:

– id – ідентифікатор можливого смаку, який може асоціюватися з шоколадом;

– short_name – коротка назва можливого смаку, який може асоціюватися з шоколадом;

– full_name – повна назва можливого смаку, який може асоціюватися з шоколадом (рис. 3.4).

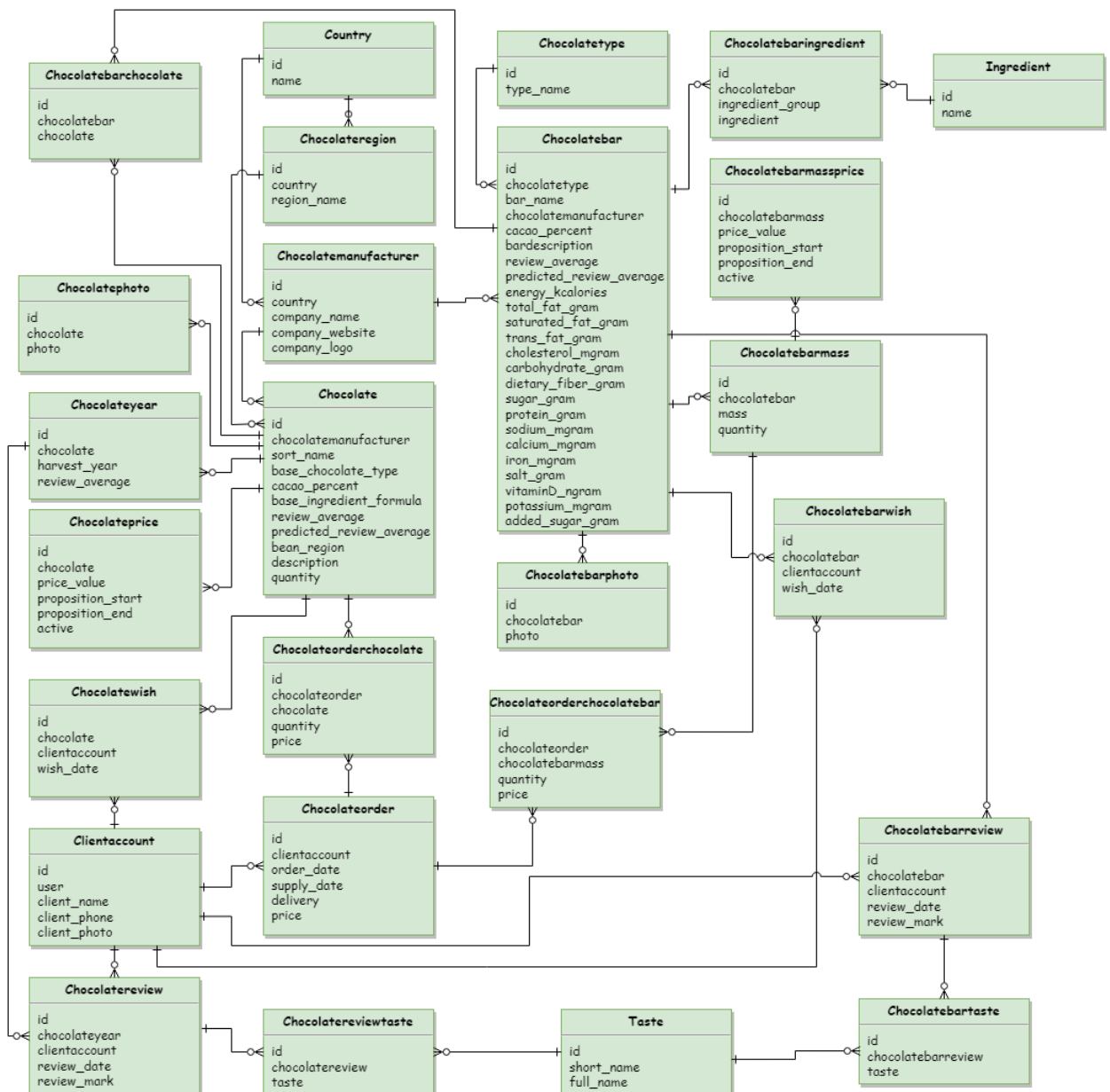


Рисунок 3.4 – Схема бази даних

Смаки, які асоціював зі своїм відгуком про сорт шоколаду клієнт, зведені до таблиці *Chocolatereviewtaste* з полями:

- *id* – ідентифікатор смаку, який асоціював зі своїм відгуком про сорт шоколаду клієнт;
- *chocolatereview* – відгук клієнта про сорт шоколаду за роком збору врожаю, за яким визначено смаки, передбачає створення зовнішнього ключа для зв'язку з таблицею *Chocolatereview*;

– taste – смак, який визначено за цим сортом шоколаду за роком збору врожаю у відгуку клієнта, передбачає створення зовнішнього ключа для зв'язку з таблицею Taste.

Смаки, які асоціював зі своїм відгуком про плитку шоколаду клієнт, зведені до таблиці Chocolatebartaste з полями:

– id – ідентифікатор смаку, який асоціював зі своїм відгуком про плитку шоколаду клієнт;

– chocolatebarreview – відгук клієнта про плитку шоколаду, за яким визначено смаки, передбачає створення зовнішнього ключа для зв'язку з таблицею Chocolatebarreview;

– taste – смак, який визначено за цією плиткою шоколаду у відгуку клієнта, передбачає створення зовнішнього ключа для зв'язку з таблицею Taste.

Додаткові дані облікових записів клієнтів зведені до таблиці Clientaccount з полями:

– id – ідентифікатор додаткових даних облікового запису клієнта;

– user – користувач Django, який працює в програмі як клієнт, його обліковий запис;

– client_name – ім'я клієнта вебзастосунку підтримки продажу та рекомендування шоколаду;

– client_phone – номер телефону клієнта вебзастосунку підтримки продажу та рекомендування шоколаду;

– client_photo – фотографія клієнта вебзастосунку підтримки продажу та рекомендування шоколаду.

3.4 Висновки за розділом 3

У розділі 3 реалізовано вибір для програмного забезпечення підтримки продажу та рекомендування шоколаду засобів реалізації, що включають в основі мову програмування Python, фреймворк Django, систему MySQL,

моделювання сценаріїв роботи користувача-клієнта, оператора компанії з продажу шоколаду, менеджера компанії з продажу шоколаду, визначення структури бази даних.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Порядок взаємодії користувача з програмою підтримки продажу та рекомендування шоколаду

Порядок взаємодії користувача з програмою підтримки продажу та рекомендування шоколаду визначений порядком роботи з меню на основі відповідної схеми функціонування (рис. 4.1).

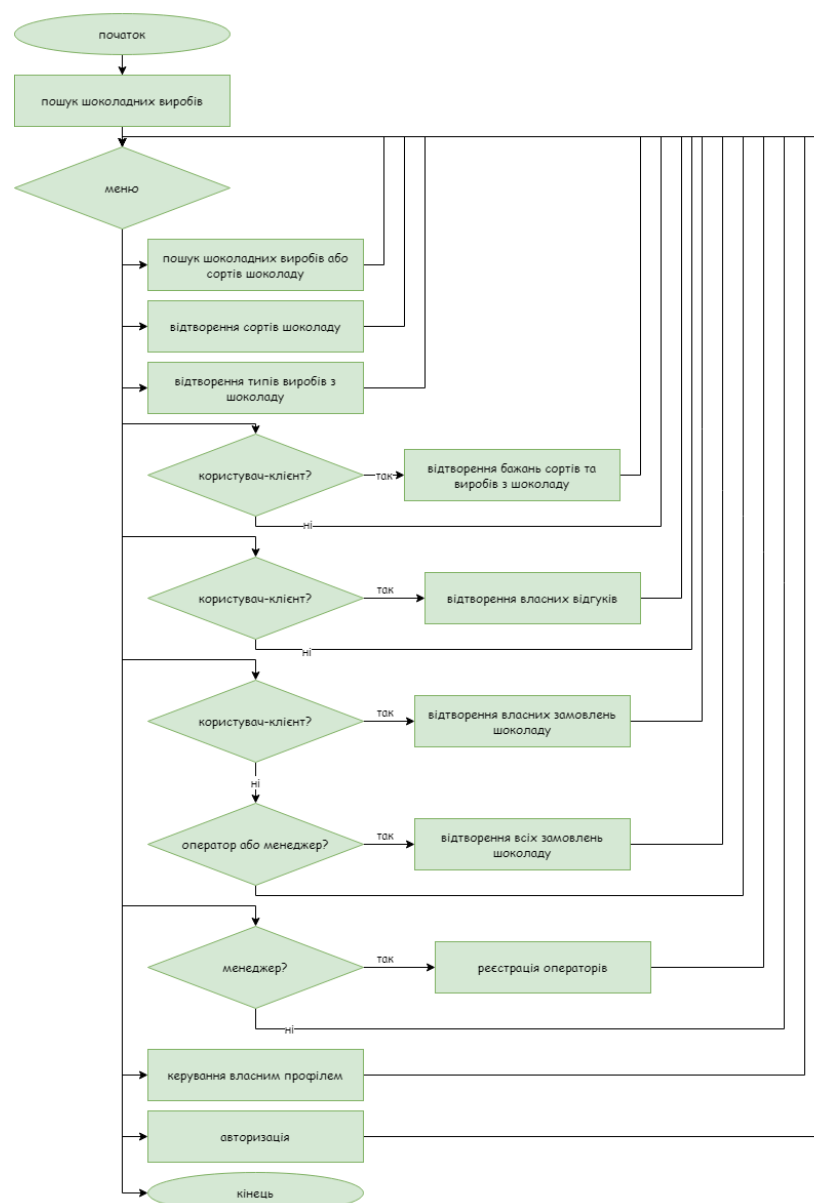


Рисунок 4.1 – Схема функціонування програми підтримки продажу та рекомендування шоколаду

4.2 Рішення з реалізації вебзастосунку

Реалізацію запропонованого методу рекомендування шоколаду виконано з застосуванням пакету `pandas` для роботи з даними у вигляді вибірки даних як об'єкта класу `DataFrame`, з застосуванням пакету `xgboost` для реалізації відповідно роботи з моделлю `XGBoost`, з якої застосовано клас `XGBRegressor`, а потім відповідно викликані методи `fit` та `predict` для роботи з моделлю стосовно навчання та прогнозування відповідно на основі навченої моделі, з застосуванням пакету `scikit-learn` для реалізації кластеризації на основі методу `k-середніх`.

Представлення відтворення переліку сортів шоколаду за регіоном реалізовано функцією `RegionChocolateView`, де параметр `chocolateregion` задає ідентифікатор регіону, де вирощують шоколад, а параметр `sortvar` задає напрямок сортування: 0 – за зменшенням рейтингу, 1 – за його збільшенням, 2 – за збільшенням відсотку вмісту какао в шоколаді, 3 – за його зменшенням, 4 – за збільшенням вартості, 5 – за зменшенням вартості. Сформовані дані передаються в шаблон вебсторінки `chocoregion.html`. Шаблони всіх сторінок розроблені з використанням вільного шаблону [10].

Представлення відтворення переліку сортів шоколаду за країною реалізовано функцією `CountryChocolateView`, де параметр `country_name` задає назву країни, де вирощують шоколад, а параметр `sortvar` задає напрямок сортування: 0 – за зменшенням рейтингу, 1 – за його збільшенням, 2 – за збільшенням відсотку вмісту какао в шоколаді, 3 – за його зменшенням, 4 – за збільшенням вартості, 5 – за зменшенням вартості. Сформовані дані передаються в шаблон вебсторінки `chococountry.html`.

Представлення відтворення даних про виробника шоколаду реалізовано функцією `ChocolatemanufacturerView`, де параметр `chocolate_manufacturer` задає ідентифікатор виробника шоколаду. Сформовані дані передаються в шаблон вебсторінки `chocomanuf.html`, включаючи дані самого виробника шоколаду за ключем `manufacturer`, 5 найкращих за

рейтингом сортів шоколаду за ключем `chocolates`, 5 найкращих за рейтингом плиток шоколаду за ключем `chocolatebars`.

Представлення відтворення переліку всіх сортів шоколаду виробника реалізовано функцією `ManufacturerChocolatesView`, де параметр `chocolate_manufacturer` задає ідентифікатор виробника шоколаду, а параметр `sortvar` задає напрямок сортування: 0 – за зменшенням рейтингу, 1 – за його збільшенням, 2 – за збільшенням відсотку вмісту какао в шоколаді, 3 – за його зменшенням, 4 – за збільшенням вартості, 5 – за зменшенням вартості. Сформовані дані передаються в шаблон вебсторінки `chocomanufchocs.html`.

Представлення відтворення переліку всіх плиток шоколаду виробника реалізовано функцією `ManufacturerChocolateBarsView`, де параметр `chocolate_manufacturer` задає ідентифікатор виробника шоколаду, а параметр `sortvar` задає напрямок сортування: 0 – за зменшенням рейтингу, 1 – за його збільшенням, 2 – за збільшенням відсотку вмісту какао в шоколаді, 3 – за його зменшенням, 4 – за збільшенням вартості, 5 – за зменшенням вартості. Сформовані дані передаються в шаблон вебсторінки `chocomanufchocbars.html`.

Представлення відтворення даних сорту шоколаду реалізовано функцією `ChocolateView`, де параметр `chocolate` задає ідентифікатор сорту шоколаду. До даних сорту включається поточна вартість, роки збору врожаю, 5 останніх відгуків. Сформовані дані передаються в шаблон вебсторінки `choco.html`.

Представлення відтворення даних плитки шоколаду реалізовано функцією `ChocolatebarView`, де параметр `chocolate` задає ідентифікатор плитки шоколаду. До даних плитки шоколаду включається поточна вартість всіх варіантів ваги плитки шоколаду, варіанти ваги плитки шоколаду, фотографії, 5 останніх відгуків, інгредієнти. Сформовані дані передаються в шаблон вебсторінки `chocobar.html`.

Представлення пошуку сорту шоколаду реалізовано функцією `ChocolateSearchView`. З GET-запиту витягаються параметри пошуку сорту шоколаду:

- назва сорту шоколаду `sort`;
- назва базового типу шоколаду `type`;
- максимальний відсоток вмісту какао в шоколаді `cacao`max;
- мінімальний відсоток вмісту какао в шоколаді `cacao`min;
- загальний склад інгредієнтів сорту шоколаду `ingredientformula`;
- мінімальна оцінка сорту шоколаду `review`min;
- максимальна оцінка сорту шоколаду `review`max;
- назва регіону, де вирощувався цей сорт шоколаду, `beanregion`;
- напрямок сортування результатів `sorting`;
- мінімальна вартість сорту шоколаду `price`min;
- максимальна вартість сорту шоколаду `price`max;
- номер сторінки результатів `page`.

Результати пошуку сорту шоколаду повертаються в шаблоні вебсторінки `chocosearch.html`.

Представлення пошуку плиток шоколаду реалізовано функцією `ChocolatebarSearchView`. З GET-запиту витягаються параметри пошуку плиток шоколаду:

- ідентифікатор типу плитки шоколаду `type`;
- максимальний відсоток вмісту какао в шоколаді `cacao`max;
- мінімальний відсоток вмісту какао в шоколаді `cacao`min;
- назва плитки шоколаду `name`;
- мінімальна оцінка плитки шоколаду `review`min;
- максимальна оцінка плитки шоколаду `review`max;
- мінімальна харчова енергія в 100 грамах плитки шоколаду `energy`min;
- максимальна харчова енергія в 100 грамах плитки шоколаду `energy`max;
- мінімальна сумарна кількість жиру в грамах, що міститься в плитці шоколаду, `fat`min;
- максимальна сумарна кількість жиру в грамах, що міститься в плитці шоколаду, `fat`max;

- мінімальна кількість насичених жирів в 100 грамах плити шоколаду
satfatmin;
- максимальна кількість насичених жирів в 100 грамах плити шоколаду
satfatmax;
- мінімальна кількість трансжирів в 100 грамах плити шоколаду
transfatmin;
- максимальна кількість трансжирів в 100 грамах плити шоколаду
transfatmax;
- мінімальна кількість холестерину в 100 грамах плити шоколаду
cholmin;
- максимальна кількість холестерину в 100 грамах плити шоколаду
cholmax;
- мінімальна кількість вуглеводів у грамах в 100 грамах плити шоколаду
carbomin;
- максимальна кількість вуглеводів у грамах в 100 грамах плити шоколаду
carbomax;
- мінімальна кількість клітковини в грамах в 100 грамах плити шоколаду
fibermin;
- максимальна кількість клітковини в грамах в 100 грамах плити шоколаду
fibermax;
- мінімальна кількість цукру в 100 грамах плити шоколаду в грамах
sugarmin;
- максимальна кількість цукру в 100 грамах плити шоколаду в грамах
sugarmax;
- мінімальна кількість білку в 100 грамах плити шоколаду в грамах
proteinmin;
- максимальна кількість білку в 100 грамах плити шоколаду в грамах
proteinmax;
- мінімальна кількість натрію в 100 грамах плити шоколаду в міліграмах
sodiummin;

- максимальна кількість натрію в 100 грамах плитки шоколаду в міліграмах sodiummax;
- мінімальна кількість кальцію в 100 грамах плитки шоколаду в міліграмах calciummin;
- максимальна кількість кальцію в 100 грамах плитки шоколаду в міліграмах calciummax;
- мінімальна кількість заліза в 100 грамах плитки шоколаду в міліграмах ironmin;
- максимальна кількість заліза в 100 грамах плитки шоколаду в міліграмах ironmax;
- мінімальна кількість солі в 100 грамах плитки шоколаду в грамах saltmin;
- максимальна кількість солі в 100 грамах плитки шоколаду в грамах saltmax;
- мінімальна кількість вітаміну Д в 100 грамах плитки шоколаду в нанограмах vitmin;
- максимальна кількість вітаміну Д в 100 грамах плитки шоколаду в нанограмах vitmax;
- мінімальна кількість калію в 100 грамах плитки шоколаду в міліграмах potassiummin;
- максимальна кількість калію в 100 грамах плитки шоколаду в міліграмах potassiummax;
- мінімальна кількість цукру в 100 грамах плитки шоколаду в грамах addsugarmin;
- максимальна кількість цукру в 100 грамах плитки шоколаду в грамах addsugarmax;
- напрямок сортування результатів sorting;
- назва регіону, де вирощувався сорт шоколаду зі складу плитки шоколаду, beanregion;
- мінімальна вартість плитки шоколаду pricemin;

- максимальна вартість плитки шоколаду `pricemax`;
- номер сторінки результатів `page`;
- назва сорту шоколаду зі складу плитки шоколаду `sort`;
- загальний склад інгредієнтів сорту шоколаду зі складу плитки шоколаду `ingredientformula`.

Результати пошуку плиток шоколаду повертаються в шаблоні вебсторінки `chocobarsearch.html`.

Представлення відтворення переліку плиток шоколаду за типом реалізовано функцією `TypeChocolateBarsView`, де параметр `chocolate_type` задає тип плиток шоколаду, а параметр `sortvar` задає напрямок сортування: 0 – за зменшенням рейтингу, 1 – за його збільшенням, 2 – за збільшенням відсотку вмісту какао в шоколаді, 3 – за його зменшенням, 4 – за збільшенням вартості, 5 – за зменшенням вартості. Сформовані дані передаються в шаблон вебсторінки `typeschocobars.html`.

Представлення відтворення переліку відгуків про сорт шоколаду реалізовано функцією `ChocolatereviewsView`, де параметр `chocolate` задає ідентифікатор сорту шоколаду, а параметр `sortvar` задає напрямок сортування: 0 – за датою публікації відгуку з останніх, 1 – за датою публікації відгуку з перших. Сформовані дані передаються в шаблон вебсторінки `chocoreviews.html`.

Представлення відтворення переліку відгуків про плитку шоколаду реалізовано функцією `ChocolatebarreviewsView`, де параметр `chocolatebar` задає ідентифікатор плитки шоколаду, а параметр `sortvar` задає напрямок сортування: 0 – за датою публікації відгуку з останніх, 1 – за датою публікації відгуку з перших. Сформовані дані передаються в шаблон вебсторінки `chocobarreviews.html`.

Представлення відтворення переліку відгуків користувача-покупця реалізовано функцією `ClientaccountreviewsView`, де параметр `reviewtype` задає відгуки про сорт шоколаду (значення 0), про плитки шоколаду (значення 1).

Параметр `clientaccount` задає логін користувача-покупця, але якщо він не заданий, то визначається він за даними авторизації.

Представлення збереження бажання плитки шоколаду суміщено зі скасуванням бажання плитки шоколаду та реалізовано функцією `ChocolatebarWishView`, де параметр `chocolatebar` задає ідентифікатор плитки шоколаду. Якщо таке бажання цієї плитки шоколаду таким користувачем створено, то воно скасовується і вилучається з бази даних.

Представлення збереження бажання сорту шоколаду суміщено зі скасуванням бажання сорту шоколаду та реалізовано функцією `ChocolateWishView`, де параметр `chocolate` задає ідентифікатор сорту шоколаду. Якщо таке бажання цього сорту шоколаду таким користувачем створено, то воно скасовується і вилучається з бази даних.

Представлення відтворення переліку бажаних сортів та плиток шоколаду користувачем-покупцем реалізовано функцією `WishesView`.

Представлення створення відгуку про сорт шоколаду реалізовано функцією `ChocolateReviewAddView` на основі класу форми `ChocolateReviewForm`.

Представлення створення відгуку про плитку шоколаду реалізовано функцією `ChocolatebarReviewAddView` на основі класу форми `ChocolatebarReviewForm`.

Представлення замовлення шоколаду реалізовано функцією `ChocolateorderAddView`, а склад замовлення передається через GET запит, який розшифровується у функції. Представлення ж відтворення даних замовлення шоколаду реалізовано функцією `ChocolateorderView`, де параметр `chocolateorder` задає ідентифікатор замовлення шоколаду.

Представлення управління замовленням шоколаду реалізовано функцією `ChocolateorderManageView`, де параметр `chocolateorder` задає ідентифікатор замовлення шоколаду, а параметр `state` задає стан замовлення, до якого його потрібно перевести. У функції перевіряється, чи це власник цього замовлення, якщо він є користувачем-клієнтом, чи це оператор або

менеджер. Окрім того перевіряється, чи може цей користувач це замовлення шоколаду перевести у цей стан з поточного, чи ні. Якщо це можливо, то зберігаються зміни у базі даних. При початковому підтвердженні замовлення зокрема змінюється кількість доступних одиниць шоколаду.

Представлення відтворення переліку замовлень шоколаду реалізовано функцією `ChocolateordersView`, де параметр `clientaccount` задає логін користувача-клієнта. Якщо цей параметр заданий, то замовлення шоколаду вибираються за користувачем. Якщо не заданий, то якщо це користувач-клієнт, то вибираються замовлення шоколаду за клієнтом, а якщо він оператор або менеджер, то вибираються всі замовлення шоколаду. Параметр `sortvar` задає напрямок сортування. Параметр `state` задає стан потрібних замовлень шоколаду. Параметр `startday` задає початкову дату замовлень шоколаду, а `endday` – кінцеву.

Представлення відтворення переліку пропозицій вартості за сортом шоколаду реалізовано функцією `ChocolatePricesView`, де параметр `chocolate` задає ідентифікатор сорту шоколаду.

Представлення відтворення переліку пропозицій вартості за плиткою шоколаду реалізовано функцією `ChocolatebarPricesView`, де параметр `chocolatebar` задає ідентифікатор плитки шоколаду.

Представлення відтворення даних користувача-покупця реалізовано функцією `ClientaccountView`, де параметр `clientaccount` задає логін користувача-клієнта. Якщо цей параметр не задано, то беруться дані авторизації.

Представлення додавання даних плитки шоколаду до програми реалізовано функцією `ChocolatebarAddView` на основі класу форми `ChocolatebarForm`. Представлення редагування даних існуючої плитки шоколаду реалізовано функцією `ChocolatebarEditView` на основі того ж класу форми з параметром `chocolatebar`, який задає ідентифікатор плитки шоколаду. Аналогічно влаштовано інші представлення додавання і редагування, які необхідно було створити.

Модель країн, з яких може походити шоколад або його виробники, реалізована класом `Country`. Поле `name` визначає назву країни, з якої може походити шоколад або його виробники, за допомогою типу `CharField`.

Модель конкретних місць, які фактично є регіонами, де вирощуються кавові зерна, що використовуються для створення шоколаду, реалізована класом `Chocolateregion`.

Поле `country` визначає країну, в якій знаходиться регіон, де вирощуються кавові зерна, що використовуються для створення шоколаду, за допомогою типу `ForeignKey` на основі моделі `Country`.

Поле `region_name` визначає назву регіону, де вирощуються кавові зерна, що використовуються для створення шоколаду, за допомогою типу `CharField`.

Модель виробників шоколаду реалізована класом `Chocolatemanufacturer`.

Поле `country` визначає країну, в якій знаходиться основний штаб виробника шоколаду, за допомогою типу `ForeignKey` на основі моделі `Country`.

Поле `company_name` визначає назву компанії як виробника шоколаду за допомогою типу `CharField`.

Поле `company_website` визначає вебсайт компанії як виробника шоколаду за допомогою типу `URLField`.

Поле `company_logo` визначає логотип або певну фотографію, яка асоціюється з виробником шоколаду, за допомогою типу `ImageField`.

Модель сортів шоколаду реалізована класом `Chocolate`.

Поле `chocolatemanufacturer` визначає виробника, який виготовляє цей сорт шоколаду, за допомогою типу `ForeignKey` на основі моделі `Chocolatemanufacturer`.

Поле `sort_name` визначає назву сорту шоколаду за допомогою типу `CharField`.

Поле `cacao_percent` визначає відсоток вмісту какао в шоколаді цього сорту за допомогою типу `DecimalField`.

Поле `base_chocolate_type` визначає базовий тип шоколаду (молочний, чорний, білий), до якого належить цей сорт, за допомогою типу `CharField`.

Поле `base_ingredient_formula` визначає базову формулу інгредієнтів за допомогою типу `CharField`.

Поле `review_average` визначає середню оцінку, яку користувачі-клієнти виставили цьому сорту шоколаду, за допомогою типу `DecimalField`.

Поле `predicted_review_average` визначає прогнозовану середню оцінку сорту шоколаду за допомогою типу `DecimalField`.

Поле `bean_region` визначає регіон, де вирощується цей сорт какао, за допомогою типу `ForeignKey` на основі моделі `Chocolateregion`.

Поле `description` визначає опис сорту шоколаду текстом за допомогою типу `TextField`.

Модель типів плиток шоколаду (шоколадних цукерок, батончиків) реалізована класом `Chocolatetype`. Поле `type_name` визначає назву типу плиток шоколаду за допомогою типу `CharField`.

Модель конкретних плиток шоколаду реалізована класом `Chocolatebar`.

Поле `chocolatetype` визначає тип плиток шоколаду, до якого належить ця плитка, за допомогою типу `ForeignKey` на основі моделі `Chocolatetype`.

Поле `bar_name` визначає назву плитки шоколаду за допомогою типу `CharField`.

Поле `chocolatemanufacturer` визначає виробника плитки шоколаду, за допомогою типу `ForeignKey` на основі моделі `Chocolatemanufacturer`.

Поле `cacao_percent` визначає відсоток вмісту какао в плитці шоколаду за допомогою типу `DecimalField`.

Поле `bardescription` визначає опис плитки шоколаду за допомогою типу `TextField`.

Поле `review_average` визначає середню оцінку, яку клієнти виставили цій плитці шоколаду, за допомогою типу `DecimalField`.

Поле `predicted_review_average` визначає прогнозовану середню оцінку плитки шоколаду за допомогою типу `DecimalField`.

Поле `energy_kcalories` визначає харчову енергію в 100 грамах плиточки шоколаду, виражену в кількості кілокалорій, за допомогою типу `DecimalField`.

Поле `total_fat_gram` визначає сумарну кількість жиру в грамах, що міститься в цій плитці шоколаду в 100 грамах, за допомогою типу `DecimalField`.

Поле `saturated_fat_gram` визначає кількість насичених жирів в 100 грамах плиточки шоколаду в грамах за допомогою типу `DecimalField`.

Поле `trans_fat_gram` визначає кількість трансжирів в 100 грамах плиточки шоколаду в грамах за допомогою типу `DecimalField`.

Поле `cholesterol_mgram` визначає кількість холестерину в 100 грамах плиточки шоколаду у плитці шоколаду за допомогою типу `DecimalField`.

Поле `carbohydrate_gram` визначає кількість вуглеводів у грамах в 100 грамах плиточки шоколаду за допомогою типу `DecimalField`.

Поле `dietary_fiber_gram` визначає кількість клітковини в грамах в 100 грамах плиточки шоколаду за допомогою типу `DecimalField`.

Поле `sugar_gram` визначає кількість цукру в 100 грамах плиточки шоколаду в грамах за допомогою типу `DecimalField`.

Поле `protein_gram` визначає кількість білку в 100 грамах плиточки шоколаду в грамах за допомогою типу `DecimalField`.

Поле `sodium_mgram` визначає кількість натрію в 100 грамах плиточки шоколаду в міліграмах за допомогою типу `DecimalField`.

Поле `calcium_mgram` визначає кількість кальцію в 100 грамах плиточки шоколаду в міліграмах за допомогою типу `DecimalField`.

Поле `iron_mgram` визначає кількість заліза в 100 грамах плиточки шоколаду в міліграмах за допомогою типу `DecimalField`.

Поле `salt_gram` визначає кількість солі в 100 грамах плиточки шоколаду в грамах за допомогою типу `DecimalField`.

Поле `vitaminD_ngram` визначає кількість вітаміну Д в 100 грамах плиточки шоколаду в нанограмах за допомогою типу `DecimalField`.

Поле `potassium_mgram` визначає кількість калію в 100 грамах плити шоколаду в міліграмах за допомогою типу `DecimalField`.

Поле `added_sugar_gram` визначає кількість цукру в 100 грамах плити шоколаду в грамах за допомогою типу `DecimalField`.

Поле `chocolates` визначає сорти шоколаду, які містяться в цій плитці шоколаду, за допомогою типу `ManyToManyField` на основі моделі `Chocolate`.

Модель фотографії плити шоколаду реалізована класом `Chocolatebarphoto`.

Поле `chocolatebar` визначає плитку шоколаду, що представлена на фотографії, за допомогою типу `ForeignKey` на основі моделі `Chocolatebar`.

Поле `photo` визначає фотографію плити шоколаду за допомогою типу `ImageField`.

Модель фотографії сорту шоколаду реалізована класом `Chocolatephoto`.

Поле `chocolate` визначає сорт шоколаду, який показаний на фотографії, за допомогою типу `ForeignKey` на основі моделі `Chocolate`.

Поле `photo` визначає фотографію шоколаду цього сорту за допомогою типу `ImageField`.

Модель варіантів продажу плити шоколаду за вагою реалізована класом `Chocolatebarmass`.

Поле `chocolatebar` визначає плитку шоколаду, варіант якої представлений, за допомогою типу `ForeignKey` на основі моделі `Chocolatebar`.

Поле `mass` визначає вагу в грамах варіанта продажу плити шоколаду за допомогою типу `DecimalField`.

Модель вартості плити шоколаду у варіанті за вагою реалізована класом `Chocolatebarmassprice`.

Поле `chocolatebarmass` визначає варіант продажу плити шоколаду за вагою, для якого визначена ця вартість, за допомогою типу `ForeignKey` на основі моделі `Chocolatebarmass`.

Поле `price_value` визначає вартість в гривнях варіанта продажу плити шоколаду за вагою за допомогою типу `DecimalField`.

Поле `proposition_start` визначає дату і час, з яких діє ця вартість даного варіанта продажу плитки шоколаду за вагою, за допомогою типу `DateTimeField`.

Поле `proposition_end` визначає дату і час, до яких діє ця вартість даного варіанта продажу плитки шоколаду за вагою, за допомогою типу `DateTimeField`.

Поле `active` визначає активність застосування цього тарифу для продажу даного варіанта продажу плитки шоколаду за вагою за допомогою типу `BooleanField`.

Модель вартості сортів шоколаду реалізована класом `Chocolateprice`.

Поле `chocolate` визначає сорт шоколаду, для якого визначена ця вартість, за допомогою типу `ForeignKey` на основі моделі `Chocolate`.

Поле `price_value` визначає вартість в гривнях сорту шоколаду за допомогою типу `DecimalField`.

Поле `proposition_start` визначає дату і час, з яких діє ця вартість даного варіанта продажу сорту шоколаду за допомогою типу `DateTimeField`.

Поле `proposition_end` визначає дату і час, до яких діє ця вартість даного варіанта продажу сорту шоколаду за допомогою типу `DateTimeField`.

Поле `active` визначає активність застосування цього тарифу для продажу даного сорту шоколаду за допомогою типу `BooleanField`.

Модель інгредієнтів, з яких виготовляються плитки шоколаду, реалізована класом `Ingredient`. Поле `name` визначає назву інгредієнта, з якого можуть виготовлятися плитки шоколаду, за допомогою типу `CharField`.

Модель інгредієнтів, що входять до складу плитки шоколаду, реалізована класом `Chocolatebaringredient`.

Поле `chocolatebar` визначає плитку шоколаду, до якої входить цей інгредієнт, за допомогою типу `ForeignKey` на основі моделі `Chocolatebar`.

Поле `ingredient_group` визначає групу входження інгредієнта до складу плитки шоколаду за допомогою типу `CharField`.

Поле `ingredient` визначає інгредієнт, який входить до складу цієї плити шоколаду, за допомогою типу `ForeignKey` на основі моделі `Ingredient`.

Модель врожаїв сортів шоколаду реалізована класом `Chocolateyear`.

Поле `harvest_year` визначає рік збору врожаю сорту шоколаду за допомогою типу `PositiveIntegerField`.

Поле `review_average` визначає середню оцінку, яку клієнти виставили сорту шоколаду цього року збору врожаю, за допомогою типу `DecimalField`.

Модель відгуків про сорти шоколаду за роком збору врожаю реалізована класом `Chocolatereview`.

Поле `chocolateyear` визначає рік збору врожаю конкретного сорту шоколаду, за яким створено відгук, за допомогою типу `ForeignKey` на основі моделі `Chocolateyear`.

Поле `clientaccount` визначає обліковий запис клієнта, який створив цей відгук про сорт шоколаду за роком збору врожаю, за допомогою типу `ForeignKey` на основі моделі `Clientaccount`.

Поле `review_date` визначає дату створення відгуку клієнта про сорт шоколаду за роком збору врожаю за допомогою типу `DateField`.

Поле `review_mark` визначає оцінку відгуку клієнта про сорт шоколаду за роком збору врожаю за допомогою типу `PositiveSmallIntegerField`.

Поле `tastes` визначає смаки, які визначено за цим сортом шоколаду за роком збору врожаю у відгуку клієнта, за допомогою типу `ManyToManyField` на основі моделі `Taste`.

Модель відгуків про плити шоколаду реалізована класом `Chocolatebarreview`.

Поле `chocolatebar` визначає плитку шоколаду, за якою створено відгук, за допомогою типу `ForeignKey` на основі моделі `Chocolatebar`.

Поле `clientaccount` визначає обліковий запис клієнта, який створив цей відгук про плитку шоколаду, за допомогою типу `ForeignKey` на основі моделі `Clientaccount`.

Поле `review_date` визначає дату створення відгуку клієнта про плитку шоколаду за допомогою типу `DateField`.

Поле `review_mark` визначає оцінку відгуку клієнта про плитку шоколаду за допомогою типу `PositiveSmallIntegerField`.

Поле `tastes` визначає смаки, які визначено за цією плиткою шоколаду у відгуку клієнта, за допомогою типу `ManyToManyField` на основі моделі `Taste`.

Модель бажання клієнтів щодо сортів шоколаду реалізована класом `Chocolatewish`.

Поле `chocolate` визначає сорт шоколаду, бажання стосовно якого зберіг клієнт, за допомогою типу `ForeignKey` на основі моделі `Chocolate`.

Поле `clientaccount` визначає клієнта, який створив це бажання сорту шоколаду, за допомогою типу `ForeignKey` на основі моделі `Clientaccount`.

Поле `wish_date` визначає дату, коли було створено це бажання сорту шоколаду, за допомогою типу `DateField`.

Модель бажання клієнтів щодо плиток шоколаду реалізована класом `Chocolatebarwish`.

Поле `chocolatebar` визначає плитку шоколаду, бажання стосовно якої зберіг клієнт, за допомогою типу `ForeignKey` на основі моделі `Chocolatebar`.

Поле `clientaccount` визначає клієнта, який створив це бажання плитки шоколаду, за допомогою типу `ForeignKey` на основі моделі `Clientaccount`.

Поле `wish_date` визначає дату, коли було створено це бажання плитки шоколаду, за допомогою типу `DateField`.

Модель замовлення шоколаду клієнтами реалізована класом `Chocolateorder`.

Поле `clientaccount` визначає клієнта, який створив замовлення шоколаду, за допомогою типу `ForeignKey` на основі моделі `Clientaccount`.

Поле `order_date` визначає дату замовлення шоколаду клієнтами за допомогою типу `DateField`.

Поле `supply_date` визначає дату, коли має бути доставлено замовлення шоколаду клієнту, за допомогою типу `DateField`.

Поле `delivery` визначає адресу для доставки замовлення шоколаду за допомогою типу `CharField`.

Поле `price` визначає загальну вартість, яка відповідає замовленню шоколаду клієнтом, за допомогою типу `DecimalField`.

Модель вмісту замовлення шоколаду стосовно плиток шоколаду реалізована класом `Chocolateorderchocolatebar`.

Поле `chocolateorder` визначає замовлення шоколаду, до якого входить ця плитка шоколаду, за допомогою типу `ForeignKey` на основі моделі `Chocolateorder`.

Поле `chocolatebarmass` визначає вагову пропозицію плитки шоколаду, яка входить до складу цього замовлення, за допомогою типу `ForeignKey` на основі моделі `Chocolatebarmass`.

Поле `quantity` визначає кількість цих плиток шоколаду, що входить до цього замовлення шоколаду, за допомогою типу `PositiveIntegerField`.

Поле `price` визначає вартість сумарно цих плиток шоколаду, що входять до складу цього замовлення шоколаду, за допомогою типу `DecimalField`.

Модель вмісту замовлення шоколаду стосовно сортів шоколаду реалізована класом `Chocolateorderchocolate`.

Поле `chocolateorder` визначає замовлення шоколаду, до якого входить цей сорт шоколаду, за допомогою типу `ForeignKey` на основі моделі `Chocolateorder`.

Поле `chocolate` визначає сорт шоколаду, який входить до складу цієї плитки шоколаду, за допомогою типу `ForeignKey` на основі моделі `Chocolate`.

Поле `quantity` визначає кількість в грамах цього сорту шоколаду, яку було замовлено у складі цього замовлення шоколаду, за допомогою типу `DecimalField`.

Поле `price` визначає вартість сумарно цього сорту шоколаду, що входить до складу цього замовлення шоколаду, за допомогою типу `DecimalField`.

Модель можливих смаків, які можуть асоціюватися з шоколадом, реалізована класом Taste.

Поле short_name визначає коротку назву можливого смаку, який може асоціюватися з шоколадом, за допомогою типу CharField.

Поле full_name визначає повну назву можливого смаку, який може асоціюватися з шоколадом, за допомогою типу CharField.

Модель додаткових даних облікових записів клієнтів реалізована класом Clientaccount.

Поле user визначає користувача Django, який працює в програмі як клієнт, за допомогою типу ForeignKey на основі моделі User.

Поле client_name визначає ім'я клієнта вебзастосунку підтримки продажу та рекомендування шоколаду за допомогою типу CharField.

Поле client_phone визначає номер телефону клієнта вебзастосунку підтримки продажу та рекомендування шоколаду за допомогою типу CharField.

Поле client_photo визначає фотографію клієнта вебзастосунку підтримки продажу та рекомендування шоколаду за допомогою типу ImageField.

4.3 Висновки за розділом 4

Програмне забезпечення підтримки продажу та рекомендування шоколаду реалізовано повністю у відповідності з запланованими сценаріями, а отримані результати описано.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Опис програми

Програма призначена для виконання замовлення як окремих сортів шоколаду, так і плиток шоколаду, шоколадних цукерок або інших виробів з шоколаду, оцінювання власних вражень від споживання цих сортів та плиток шоколаду, використання цих даних для рекомендування шоколаду користувачам.

5.2 Експлуатація програми

Користувач-клієнт після звернення до вебзастосунку підтримки продажу та рекомендування шоколаду отримує під час роботи меню, яке включає пункти «Пошук» для пошуку сортів шоколаду або виробів з шоколаду (плиток шоколаду), вибираючи необхідний варіант шляхом переключення за допомогою кнопки, «Сорти» для відтворення переліку всіх сортів шоколаду, «Вироби» для відтворення типів виробів з шоколаду з можливістю відкриття конкретних виробів за визначеним типом, «Замовлення» для відтворення переліку всіх власних замовлень шоколаду, «Бажання» для відтворення тих сортів шоколаду або виробів з шоколаду, які відніс до своїх бажань авторизований користувач (рис. 5.1), «Рекомендації» для відтворення рекомендованих сортів шоколаду або виробів з шоколаду для авторизованого користувача, «Відгуки» для відтворення створених користувачем, що авторизувався, відгуків про сорти шоколаду та вироби з нього, «Профіль» для відтворення даних профілю користувача з можливістю оновлення даних профілю. Кожен користувач, який тільки звернувся до вебзастосунку і не виконав ще авторизацію, працює як користувач-клієнт.

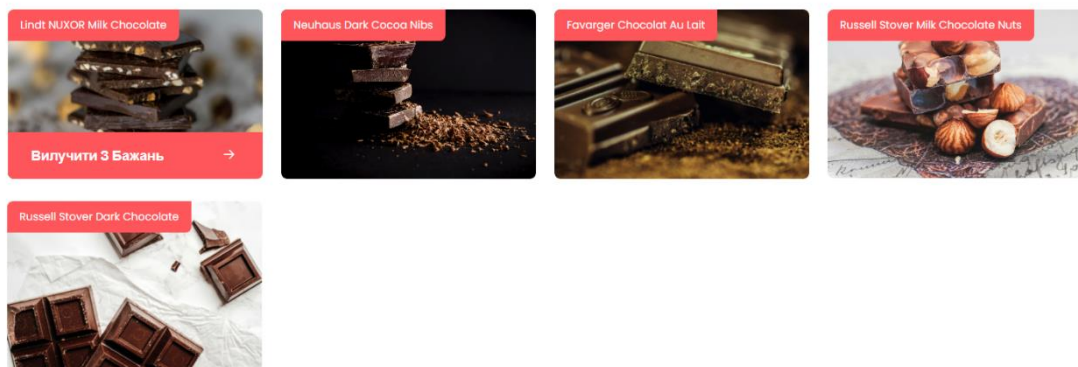
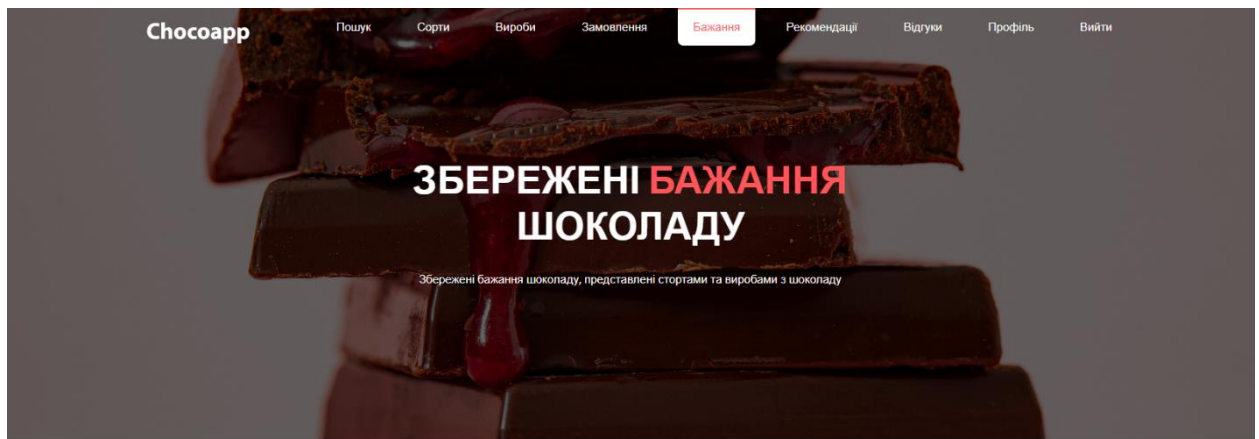


Рисунок 5.1 – Відтворення переліку бажаних сортів та плиток шоколаду

Пошук сортів шоколаду включає:

- назву сорту шоколаду;
- назву базового типу шоколаду;
- максимальний відсоток вмісту какао в шоколаді;
- мінімальний відсоток вмісту какао в шоколаді;
- загальний склад інгредієнтів сорту шоколаду;
- мінімальну оцінку сорту шоколаду;
- максимальну оцінку сорту шоколаду;
- назву регіону, де вирощувався цей сорт шоколаду;
- напрямок сортування результатів;
- мінімальну вартість сорту шоколаду;
- максимальну вартість сорту шоколаду.

Пошук виробів з шоколаду (плиток з шоколаду) включає параметри:

- тип плитки шоколаду;

- максимальний відсоток вмісту какао в шоколаді;
- мінімальний відсоток вмісту какао в шоколаді;
- назва плити шоколаду;
- мінімальна оцінка плити шоколаду;
- максимальна оцінка плити шоколаду;
- мінімальна харчова енергія в 100 грамах плити шоколаду;
- максимальна харчова енергія в 100 грамах плити шоколаду;
- мінімальна сумарна кількість жиру в грамах, що міститься в плитці шоколаду;
- максимальна сумарна кількість жиру в грамах, що міститься в плитці шоколаду;
- мінімальна кількість насичених жирів в 100 грамах плити шоколаду;
- максимальна кількість насичених жирів в 100 грамах плити шоколаду;
- мінімальна кількість трансжирів в 100 грамах плити шоколаду;
- максимальна кількість трансжирів в 100 грамах плити шоколаду;
- мінімальна кількість холестерину в 100 грамах плити шоколаду;
- максимальна кількість холестерину в 100 грамах плити шоколаду;
- мінімальна кількість вуглеводів у грамах в 100 грамах плити шоколаду;
- максимальна кількість вуглеводів у грамах в 100 грамах плити шоколаду;
- мінімальна кількість клітковини в грамах в 100 грамах плити шоколаду;
- максимальна кількість клітковини в грамах в 100 грамах плити шоколаду;
- мінімальна кількість цукру в 100 грамах плити шоколаду в грамах;
- максимальна кількість цукру в 100 грамах плити шоколаду в грамах;
- мінімальна кількість білку в 100 грамах плити шоколаду в грамах;
- максимальна кількість білку в 100 грамах плити шоколаду в грамах;

- мінімальна кількість натрію в 100 грамах плити шоколаду в міліграмах;
- максимальна кількість натрію в 100 грамах плити шоколаду в міліграмах;
- мінімальна кількість кальцію в 100 грамах плити шоколаду в міліграмах;
- максимальна кількість кальцію в 100 грамах плити шоколаду в міліграмах;
- мінімальна кількість заліза в 100 грамах плити шоколаду в міліграмах;
- максимальна кількість заліза в 100 грамах плити шоколаду в міліграмах;
- мінімальна кількість солі в 100 грамах плити шоколаду в грамах;
- максимальна кількість солі в 100 грамах плити шоколаду в грамах;
- мінімальна кількість вітаміну Д в 100 грамах плити шоколаду в нанограмах;
- максимальна кількість вітаміну Д в 100 грамах плити шоколаду в нанограмах;
- мінімальна кількість калію в 100 грамах плити шоколаду в міліграмах;
- максимальна кількість калію в 100 грамах плити шоколаду в міліграмах;
- мінімальна кількість цукру в 100 грамах плити шоколаду в грамах;
- максимальна кількість цукру в 100 грамах плити шоколаду в грамах;
- напрямок сортування результатів;
- назва регіону, де вирощувався сорт шоколаду зі складу плити шоколаду;
- мінімальна вартість плити шоколаду;
- максимальна вартість плити шоколаду;
- назва сорту шоколаду зі складу плити шоколаду;

– загальний склад інгредієнтів сорту шоколаду зі складу плитки шоколаду.

Під час рекомендування шоколаду на сторінці відображаються ті сорти та виробу з шоколаду, які належать до того самого кластера, до якого належить найбільша кількість сортів та виробів з шоколаду, які оцінив користувач, що авторизувався, достатньо високо (рис. 5.2).

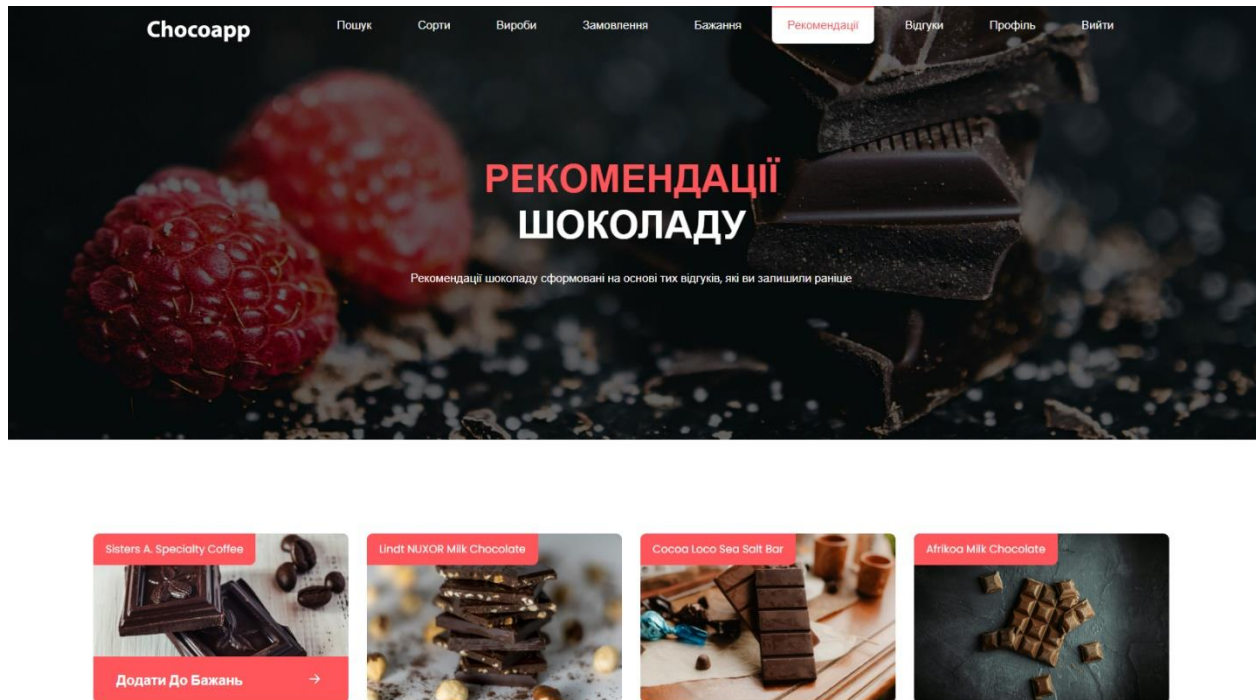


Рисунок 5.2 – Рекомендування шоколаду

Менеджер компанії з продажу шоколаду після звернення до вебзастосунку та авторизації отримує під час роботи меню, яке включає пункти «Пошук» для пошуку сортів шоколаду або виробів з шоколаду (плиток шоколаду), вибираючи необхідний варіант шляхом переключення за допомогою кнопки, «Сорти» для відтворення переліку всіх сортів шоколаду та додавання цих сортів, редагування існуючих, «Вироби» для відтворення типів виробів з шоколаду з можливістю відкриття конкретних виробів за визначеним типом та додавання, редагування виробів, «Замовлення» для відтворення переліку всіх замовлень шоколаду, «Профіль» для відтворення даних профілю користувача з можливістю оновлення даних профілю. Оператор компанії з

продажу шоколаду не має пункту меню «Оператори». Оскільки тільки менеджер компанії з продажу шоколаду може реєструвати операторів.

При зверненні до сторінки конкретної плити шоколаду менеджер або оператор компанії з продажу шоколаду має можливість переходити до редагування даних, переглядати фотографії (переключення можна виконувати, наводячи курсор на фотографію, тоді з'являються стрілки, так само як на сторінці бажань та рекомендацій при наведенні на фотографію з'являється посилання для додавання або видалення з бажань), переглядати основні характеристики (з можливістю перейти до переліку відгуків, даних виробника шоколаду), дані про вартість з вибором конкретної ваги, склад інгредієнтів, харчову цінність (рис. 5.3).

Chocoapp

Пошук

Сорти

Вироби

Замовлення

Оператори

Профіль

Вийти

SISTERS A. SPECIALTY COFFEE

ПЛИТКА ШОКОЛАДУ

Редагувати



Основні характеристики

Виробник	Sisters A. →
Вміст какао	55 %
Оцінка	4.8 →

Дані про вартість

Вага, грам	70 ▾
Вартість, грн.	258
Кількість	1 ▾

Замовити

Склад

Основні інгредієнти	какао-боби терті кокосовий цукор какао-масло кеш'ю толокно вівсяне кава
---------------------	--

Харчова цінність

Енергетична цінність, кКал	545,26
Жири, г	48,06
Вуглеводи, г	33,41
Білки, г	7,39

Рисунок 5.3 – Відтворення даних плити шоколаду

5.3 Експериментальне дослідження

Вибірка даних [11] є основою проведеного експериментального дослідження. Саме на її основі виконано прогнозування оцінок шоколаду. Дані у вибірці сформовані на 3 березня 2024 року.

Порівняння результатів прогнозування оцінок шоколаду реалізовано за показниками, що включають Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE). 60 % вибірки даних віднесено до навчальної вибірки, а 40 % – для тестової. Експериментальне дослідження прогнозування оцінок шоколаду виконано на основі навченої моделі, але з застосуванням виключено даних з тестової вибірки.

Отримані результати вказують на те, що використання всіх виділених вхідних ознак разом з повним переліком значимих характеристик шоколаду в якості додаткових ознак продемонструвало найкращі результати (табл. 5.1) за показниками MAE, MSE та RMSE. При цьому найбільш точні результати отримано саме за запропонованим методом, який використовує модель XGBoost.

Порівняння запропонованого методу рекомендування шоколаду відбувалось з деревом рішень, методом k-середніх, методом опорних векторів. Усі ці моделі порівняно у варіантах зі всіма ознаками, з обмеженим числом, коли задається певна кількість та визначаються найбільш інформативні ознаки, які потрапляють в це число. Останній варіант включає тільки 6 ознак. Це ознаки, до яких не включено значимі характеристики шоколаду в якості додаткових ознак. Запропонований метод рекомендування шоколаду навіть за використання тільки 6 ознак на вхід призвів до отримання результатів, які є точнішими порівняно і з деревом рішень, і з методом k-середніх, і методом опорних векторів за застосування абсолютно повного набору ознак, коли результати кращі порівняно з меншою кількістю ознак.

Таблиця 5.1 – Прогнозування оцінок шоколаду

Модель або метод	Кіль- кість ознак	MAE	RMSE	MSE
1	2	3	4	5
Дерево рішень	всі	0,00208076	0,00275025	7,5638574E-6
Метод k-найближчих сусідів	всі	0,00218866	0,0027651	7,64577029E-6
Метод опорних векторів	всі	0,00428611	0,0048854	2,38671622E-5
Метод, який запропоновано (XGBoost)	всі	0,00165445	0,00213002	4,53700512E-6
Дерево рішень	100	0,00229257	0,00301987	9,11960299E-6
Метод k-найближчих сусідів	100	0,00220007	0,00277957	7,72600949E-6
Метод, який запропоновано (XGBoost)	100	0,00172055	0,00222864	4,96684586E-6
Дерево рішень	50	0,0023078	0,00307646	9,46463555E-6
Метод k-найближчих сусідів	50	0,00219830	0,00277694	7,71138809E-6
Метод, який запропоновано (XGBoost)	50	0,0017553	0,00228987	5,24348549E-6
Дерево рішень	20	0,00243447	0,00324566	1,05343398E-5
Метод k-найближчих сусідів	20	0,00221119	0,00278734	7,76926425E-6
Метод, який запропоновано (XGBoost)	20	0,00187075	0,00240641	5,79082722E-6
Дерево рішень	10	0,00264144	0,00345301	1,19232508E-5
Метод k-найближчих сусідів	10	0,00221996	0,00279584	7,81673049E-6

Продовження таблиці 5.1

1	2	3	4	5
Метод, який запропоновано (XGBoost)	10	0,00195754	0,00248817	6,19097045E-6
Дерево рішень	6	0,00271689	0,00354446	1,25632138E-5
Метод k-найближчих сусідів	6	0,00225141	0,00283312	8,02654151E-6
Метод, який запропоновано (XGBoost)	6	0,00201609	0,00253937	6,4484029E-6

Якщо порівнювати результати різних моделей за однакової кількості ознак, що включає весь можливий набір, то запропонований метод рекомендації шоколаду дозволив покращити результат дерева рішень на 20,49 % за показником MAE, на 22,55 % за показником RMSE, на 40,02 % за показником MSE, результат методу k-найближчих сусідів на 24,41 % за показником MAE, на 22,97 % за показником RMSE, на 40,66 % за показником MSE, результат методу опорних векторів на 61,4 % за показником MAE, на 56,4 % за показником RMSE, на 80,99 % за показником MSE.

5.4 Висновки за розділом 5

Процес експлуатації програмного забезпечення підтримки продажу та рекомендації шоколаду розглянуто зі сторони користувачів у різних ролях, включаючи користувача-клієнта, оператора та менеджера компанії з продажу шоколаду, зі сторони організації процесу роботи над рекомендаціями шоколаду в частині прогнозування оцінки шоколаду на основі існуючої вибірки даних з застосуванням відповідного експериментального дослідження.

ВИСНОВКИ

У роботі виконана розробка програмного забезпечення підтримки продажу та рекомендування шоколаду для інформування користувачів на різних етапах введення шоколаду в продаж.

Реалізовано вибір для програмного забезпечення підтримки продажу та рекомендування шоколаду засобів реалізації, що включають в основі мову програмування Python, фреймворк Django, систему MySQL, моделювання сценаріїв роботи користувача-клієнта, оператора, менеджера компанії з продажу шоколаду, визначення структури бази даних, описано створені програмні одиниці.

Створена програма призначена для виконання замовлення як окремих сортів шоколаду, так і плиток шоколаду, шоколадних цукерок або інших виробів з шоколаду, оцінювання власних вражень від споживання цих сортів та плиток шоколаду, використання цих даних для рекомендування шоколаду користувачам. Оцінювання здійснюється з врахуванням року врожаю.

Наукова новизна роботи полягає в методі рекомендування шоколаду, який виконує прогнозування відсутніх оцінок клієнтами сортів шоколаду за допомогою моделі XGBoost, визначає перетворення основних характеристик сорту шоколаду на конкретні вхідні ознаки, що дозволяє виконувати рекомендування шоколаду за допомогою виконання кластеризації на основі методу k-середніх.

Запропонований метод рекомендування шоколаду дозволив покращити результат дерева рішень стосовно прогнозування оцінок шоколаду на 20,49 % за показником MAE, на 22,55 % за показником RMSE, на 40,02 % за показником MSE, результат методу k-найближчих сусідів на 24,41 % за показником MAE, на 22,97 % за показником RMSE, на 40,66 % за показником MSE, результат методу опорних векторів на 61,4 % за показником MAE, на 56,4 % за показником RMSE, на 80,99 % за показником MSE.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chocolate Market Size, Share, Analysis and Industry Trend to 2030 [Electronic resource]. – Access mode : <https://www.marknteladvisors.com/research-library/chocolate-market.html#:~:text=The%20Global%20Chocolate%20Market%20size,%2C%20i.e.%2C%202025%2D30>.
2. Premium Chocolate Market Size to Expand US\$85.61 billion by 2033 [Electronic resource]. – Access mode : <https://straitsresearch.com/report/premium-chocolate-market>
3. Neuhaus Chocolatesю Inventor of the Belgian praline [Electronic resource]. – Access mode : https://www.neuhauschocolates.com/en_BE/home.
4. Chokolaterie Brussels – The Belgian Chocolate Makers – The Belgian Chocolate Makers [Electronic resource]. – Access mode : <https://www.chocolaterie.brussels/>.
5. Pierre Marcolini [Electronic resource] : Discover the chocolates of eponymous House. – Access mode : <https://eu.marcolini.com/en/?force>.
6. MIKE&BECKY – Real. Honest. Chocolate. From the Bean [Electronic resource]. – Access mode : <https://mikeandbecky.be/>.
7. Kavlakoglu, E. What is XGBoost? [Electronic resource] / E. Kavlakoglu, E. Russi. – Access mode : <https://www.ibm.com/think/topics/xgboost>.
8. Express – Node.js web application framework [Electronic resource]. – Access mode : <https://expressjs.com/>.
9. Django introduction – Learn web development. MDN [Electronic resource]. – Access mode : https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Django/Introduction.
10. Free Template 572 Designer [Electronic resource]. – Access mode : <https://templatemo.com/tm-572-designer>.
11. Flavors of Cacao – Chocolate Database [Electronic resource]. – Access mode : https://flavorsofcacao.com/chocolate_database.html.

ДОДАТОК А
Технічне завдання

Вступ

Ринок шоколаду величезний на цей момент, у той же час активно розвиваються ніші цього ринку. Крафтове виробництво, використання преміальних зерен какао є прикладами такого розвитку. Як будь-які процеси зараз у світі, цей також має забезпечуватися відповідним програмним забезпеченням. Однак просто існування програмного забезпечення для продажу шоколаду недостатньо, адже кількість сортів, різних варіантів шоколаду у вигляді плиток, цукерок є багато. Окрім того рік врожаю може впливати на результати, тому важливо створити відповідну систему оцінювання, що дозволило б орієнтуватися всім користувачам серед наявних сортів, варіантів виробів з шоколаду. Відповідно в залежності від характеристик шоколаду, який тільки з'являється в самій програмі, тобто у продавця шоколаду, необхідно оцінити його сприйняття клієнтами. Для цього необхідно забезпечити метод рекомендування шоколаду, який би це підтримував зокрема і в складі програмного забезпечення.

A.1 Підстави для розробки

На підставі наказу № 398 від 7 жовтня 2024 р., який був виданий у Національному університеті «Запорізька політехніка», де і ведеться розробка в межах даної дипломної кваліфікаційної роботи магістра, визначена тема роботи «Дослідження та програмна реалізація методів рекомендування шоколаду», що було деталізовано також виданим завданням на роботу.

A.2 Призначення розробки

Розробка виконується для того, щоб забезпечити виконання замовлення як окремих сортів шоколаду, так і плиток шоколаду, шоколадних цукерок або інших виробів з шоколаду, оцінювання власних вражень від

споживання цих сортів та плиток шоколаду, використання цих даних для рекомендування шоколаду користувачам.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Для реалізації комплексної роботи програмного забезпечення підтримки продажу та рекомендування шоколаду необхідно забезпечити такі функціональні можливості:

- відтворення переліку сортів шоколаду за регіоном або країною;
- відтворення даних про виробника шоколаду;
- відтворення переліку всіх сортів шоколаду виробника;
- відтворення переліку всіх плиток шоколаду виробника;
- відтворення даних сорту шоколаду;
- відтворення даних плитки шоколаду;
- пошук сорту шоколаду;
- пошук плиток шоколаду;
- відтворення переліку плиток шоколаду за типом;
- додавання даних плитки шоколаду до програми;
- редагування даних існуючої плитки шоколаду;
- додавання даних сорту шоколаду;
- редагування даних сорту шоколаду;
- додавання варіанта плитки шоколаду за вагою;
- вилучення варіанта плитки шоколаду за вагою;
- додавання пропозиції вартості плитки шоколаду за вагою;
- редагування пропозиції вартості плитки шоколаду за вагою;
- додавання пропозиції вартості сорту шоколаду;
- редагування пропозиції вартості сорту шоколаду;
- рекомендування шоколаду;
- відтворення переліку пропозицій вартості за сортом шоколаду;

- відтворення переліку пропозицій вартості за плиткою шоколаду;
- створення відгуку про сорт шоколаду;
- створення відгуку про плитку шоколаду;
- відтворення переліку відгуків про сорт шоколаду;
- відтворення переліку відгуків про плитку шоколаду;
- відтворення даних користувача-покупця;
- відтворення переліку відгуків користувача-покупця;
- збереження бажання плитки шоколаду;
- збереження бажання сорту шоколаду;
- скасування бажання плитки шоколаду;
- скасування бажання сорту шоколаду;
- відтворення переліку бажаних сортів та плиток шоколаду користувачем-покупцем;
- авторизація, реєстрація;
- замовлення шоколаду;
- відтворення переліку замовлень шоколаду загалом, користувача або власних;
- додавання виробника шоколаду;
- редагування виробника шоколаду;
- управління замовленням шоколаду;
- відтворення даних замовлення шоколаду.

A.3.2 Умови експлуатації

Експлуатація програмного забезпечення підтримки продажу та рекомендації шоколаду має забезпечуватися виконанням ряду вимог до серверної частини:

- забезпечення вебсерверу з підтримкою інтерпретатора мови програмування Python 3;
- налаштування Django-застосунку для виконання на сервері з

реалізацією міграції для забезпечення необхідної структури бази даних за допомогою підтримки системи керування базою даних MySQL;

- інсталювані Python-пакети Numpy, Pandas, Scikit-learn.

A.3.3 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення підтримки продажу та рекомендування шоколаду має забезпечуватися серверною частиною з такими мінімальними параметрами технічних засобів:

- двоядерний процесор;
- 4 Гб оперативної пам'яті;
- 10 Гб простору на жорсткому диску, що необхідно для збереження даних самого вебсайту та фотографій шоколаду, файлів вебзастосунку, бази даних: в залежності від якості, кількості фотографій можливе подальше збільшення мінімальних вимог.

A.4 Порядок контролю та приймання

Робота має виконуватися у відповідності з затвердженим завданням і контролюється керівником. Отримані результати роботи представляються в пояснювальній записці.

Завершальним етапом виконання та відповідно контролю і приймання дипломної кваліфікаційної роботи магістра є її захист, який виконується перед екзаменаційною комісією.

ДОДАТОК Б
Текст програми

```

from .models import Country
from .models import Chocolateregion
from .models import Chocolatemanufacturer
from .models import Chocolate
from .models import Chocolatetype
from .models import Chocolatebar
from .models import Chocolatebarphoto
from .models import Chocolatebarmass
from .models import Chocolatebarchocolate
from .models import Chocolatebarmassprice
from .models import Chocolateprice
from .models import Ingredient
from .models import Chocolatebaringredient
from .models import Chocolateyear
from .models import Chocolatereview
from .models import Chocolatebarreview
from .models import Chocolatewish
from .models import Chocolatebarwish
from .models import Chocolateorder
from django.core.paginator import Paginator
from .models import Chocolateorderchocolatebar
from .models import Chocolateorderchocolate
from .models import Taste
from .models import Clientaccount
from .models import Chocolatephoto
from django.shortcuts import render
from datetime import datetime

def RegionChocolateView(request, chocolateregion, sortvar =
0):
    try:
        chocolate_region = Chocolateregion.objects.get(id =
chocolateregion)
    except Chocolateregion.DoesNotExist:
        return render(request, "chocoapp/chocoregion.html",
                        {"region": None,
                         "chocolates": []
                        })

    chocolates =
Chocolate.objects.select_related("chocolatemanufacturer",
"bean_region").filter(bean_region = chocolate_region)
    time_for_search = datetime.now()
    for chocolate in chocolates:
        chocolate.price =
Chocolateprice.objects.filter(chocolate = chocolate,
proposition_start__lte = time_for_search, proposition_end__gte =
time_for_search, active = True)[0]
        if sortvar == 0:
            chocolates = chocolates.order_by("-review_average")
        elif sortvar == 1:
            chocolates = chocolates.order_by("review_average")
        elif sortvar == 2:

```

```

        chocolates = chocolates.order_by("cacao_percent")
    elif sortvar == 3:
        chocolates = chocolates.order_by("-cacao_percent")
    elif sortvar == 4:
        chocolates = chocolates.order_by("price")
    elif sortvar == 5:
        chocolates = chocolates.order_by("-price")
    chocolate_paginator = Paginator(chocolates, 20)
    chocolate_page = request.GET.get("page")
    chocolates =
chocolate_paginator.get_page(chocolate_page)
    if request.user.is_authenticated:
        user_for_clientaccount = request.user
        client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
        for chocolate in chocolates:
            wishes = Chocolatewish.objects.filter(chocolate
= chocolate, clientaccount = client_account)
            if wishes:
                chocolate.wish = True
            else:
                chocolate.wish = False
            chocolate.photo =
Chocolatephoto.objects.filter(chocolate = chocolate)[0]
        return render(request, "chocoapp/chocoregion.html",
                        {"region": chocolate_region,
                         "chocolates": chocolates
                        })

def CountryChocolateView(request, country_name, sortvar =
0):
    try:
        chocolate_country = Country.objects.get(name =
country_name)
    except Chocolateregion.DoesNotExist:
        return render(request,
"chocoapp/chococountry.html",
                        {"country": None,
                         "chocolates": []
                        })

    chocolates =
Chocolate.objects.select_related("chocolatemanufacturer",
"bean_region").filter(bean_region__country = chocolate_country)
    time_for_search = datetime.now()
    for chocolate in chocolates:
        chocolate.price =
Chocolateprice.objects.filter(chocolate = chocolate,
proposition_start__lte = time_for_search, proposition_end__gte =
time_for_search, active = True)[0]
    if sortvar == 0:
        chocolates = chocolates.order_by("-review_average")
    elif sortvar == 1:

```

```

        chocolates = chocolates.order_by("review_average")
    elif sortvar == 2:
        chocolates = chocolates.order_by("cacao_percent")
    elif sortvar == 3:
        chocolates = chocolates.order_by("-cacao_percent")
    elif sortvar == 4:
        chocolates = chocolates.order_by("price")
    elif sortvar == 5:
        chocolates = chocolates.order_by("-price")
    chocolate_paginator = Paginator(chocolates, 20)
    chocolate_page = request.GET.get("page")
    chocolates =
chocolate_paginator.get_page(chocolate_page)
    if request.user.is_authenticated:
        user_for_clientaccount = request.user
        client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
        for chocolate in chocolates:
            wishes = Chocolatewish.objects.filter(chocolate
= chocolate, clientaccount = client_account)
            if wishes:
                chocolate.wish = True
            else:
                chocolate.wish = False
            chocolate.photo =
Chocolatephoto.objects.filter(chocolate = chocolate)[0]
        return render(request, "chocoapp/chococountry.html",
                        {"country": chocolate_country,
                         "chocolates": chocolates
                        })

def ChocolatemanufacturerView(request,
chocolate_manufacturer):
    try:
        chocolate_manufacturer =
Chocolatemanufacturer.objects.select_related("country").get(id =
chocolate_manufacturer)
    except Chocolatemanufacturer.DoesNotExist:
        return render(request, "chocoapp/chocomanuf.html",
                        {"manufacturer": None,
                         "chocolates": [],
                         "chocolatebars": []
                        })

    chocolates =
Chocolate.objects.select_related("bean_region").filter(chocolate
manufacturer = chocolate_manufacturer)
    chocolates = chocolates.order_by("-review_average")[:5]
    time_for_search = datetime.now()
    for chocolate in chocolates:
        chocolate.price =
Chocolateprice.objects.filter(chocolate = chocolate,

```

```

proposition_start__lte = time_for_search, proposition_end__gte =
time_for_search, active = True)[0]
    chocolate_bars =
Chocolatebar.objects.select_related("chocolatetype").filter(choc
olatemanufacturer = chocolate_manufacturer)
    chocolate_bars = chocolate_bars.order_by("-
review_average")[:5]
    time_for_search = datetime.now()
    for chocolate in chocolate_bars:
        chocolate.masses =
Chocolatebarmass.objects.filter(chocolatebar = chocolate)
        chocolate.prices =
Chocolatebarmassprice.objects.filter(chocolatebarmass__chocolate
bar = chocolate, proposition_start__lte = time_for_search,
proposition_end__gte = time_for_search, active = True)[0]
        if request.user.is_authenticated:
            user_for_clientaccount = request.user
            client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
            for chocolate in chocolates:
                wishes = Chocolatewish.objects.filter(chocolate
= chocolate, clientaccount = client_account)
                if wishes:
                    chocolate.wish = True
                else:
                    chocolate.wish = False
                chocolate.photo =
Chocolatephoto.objects.filter(chocolate = chocolate)[0]
                for chocolate in chocolate_bars:
                    wishes =
Chocolatebarwish.objects.filter(chocolatebar = chocolate,
clientaccount = client_account)
                    if wishes:
                        chocolate.wish = True
                    else:
                        chocolate.wish = False
                    chocolate.photo =
Chocolatebarphoto.objects.filter(chocolatebar = chocolate)[0]
                return render(request, "chocoapp/chocomanuf.html",
                    {"manufacturer": chocolate_manufacturer,
                     "chocolates": chocolates,
                     "chocolatebars": chocolate_bars
                    })

```

```

def ManufacturerChocolatesView(request,
chocolate_manufacturer, sortvar = 0):
    try:
        chocolate_manufacturer =
Chocolatemanufacturer.objects.select_related("country").get(id =
chocolate_manufacturer)
    except Chocolatemanufacturer.DoesNotExist:

```

```

        return render(request,
            "chocoapp/chocomanufchocs.html",
            {
                "manufacturer": None,
                "chocolates": []
            })

        chocolates =
Chocolate.objects.select_related("bean_region").filter(chocolate
manufacturer = chocolate_manufacturer)
        time_for_search = datetime.now()
        for chocolate in chocolates:
            chocolate.price =
Chocolateprice.objects.filter(chocolate = chocolate,
proposition_start__lte = time_for_search, proposition_end__gte =
time_for_search, active = True)[0]
            if sortvar == 0:
                chocolates = chocolates.order_by("-review_average")
            elif sortvar == 1:
                chocolates = chocolates.order_by("review_average")
            elif sortvar == 2:
                chocolates = chocolates.order_by("cacao_percent")
            elif sortvar == 3:
                chocolates = chocolates.order_by("-cacao_percent")
            elif sortvar == 4:
                chocolates = chocolates.order_by("price")
            elif sortvar == 5:
                chocolates = chocolates.order_by("-price")
        chocolate_paginator = Paginator(chocolates, 20)
        chocolate_page = request.GET.get("page")
        chocolates =
chocolate_paginator.get_page(chocolate_page)
        if request.user.is_authenticated:
            user_for_clientaccount = request.user
            client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
            for chocolate in chocolates:
                wishes = Chocolatewish.objects.filter(chocolate
= chocolate, clientaccount = client_account)
                if wishes:
                    chocolate.wish = True
                else:
                    chocolate.wish = False
                chocolate.photo =
Chocolatephoto.objects.filter(chocolate = chocolate)[0]
        return render(request, "chocoapp/chocomanufchocs.html",
            {
                "manufacturer": chocolate_manufacturer,
                "chocolates": chocolates
            })

def ManufacturerChocolateBarsView(request,
chocolate_manufacturer, sortvar = 0):
    try:

```

```

        chocolate_manufacturer =
Chocolatemanufacturer.objects.select_related("country").get(id =
chocolate_manufacturer)
        except Chocolatemanufacturer.DoesNotExist:
            return render(request,
"chocoapp/chocomanufchocbars.html",
                        {"manufacturer": None,
                        "chocolates": []
                        })
        chocolate_bars =
Chocolatebar.objects.select_related("chocolatetype").filter(choc
olatemanufacturer = chocolate_manufacturer)
        time_for_search = datetime.now()
        for chocolate in chocolate_bars:
            chocolate.masses =
Chocolatebarmass.objects.filter(chocolatebar = chocolate)
            chocolate.prices =
Chocolatebarmassprice.objects.filter(chocolatebarmass__chocolate
bar = chocolate, proposition_start__lte = time_for_search,
proposition_end__gte = time_for_search, active = True)[0]
            if sortvar == 0:
                chocolate_bars = chocolate_bars.order_by("-
review_average")
            elif sortvar == 1:
                chocolate_bars =
chocolate_bars.order_by("review_average")
            elif sortvar == 2:
                chocolate_bars =
chocolate_bars.order_by("cacao_percent")
            elif sortvar == 3:
                chocolate_bars = chocolate_bars.order_by("-
cacao_percent")
            elif sortvar == 4:
                chocolate_bars = chocolate_bars.order_by("price")
            elif sortvar == 5:
                chocolate_bars = chocolate_bars.order_by("-price")
            chocolate_paginator = Paginator(chocolate_bars, 20)
            chocolate_page = request.GET.get("page")
            chocolate_bars =
chocolate_paginator.get_page(chocolate_page)
            if request.user.is_authenticated:
                user_for_clientaccount = request.user
                client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
                for chocolate in chocolate_bars:
                    wishes =
Chocolatebarwish.objects.filter(chocolatebar = chocolate,
clientaccount = client_account)
                    if wishes:
                        chocolate.wish = True
                    else:
                        chocolate.wish = False

```

```

        chocolate.photo =
Chocolatebarphoto.objects.filter(chocolatebar = chocolate)[0]

    return render(request,
"chocoapp/chocomanufchocbars.html",
                    {"manufacturer": chocolate_manufacturer,
                     "chocolates": chocolate_bars
                    })

def ChocolateView(request, chocolate):
    try:
        chocolate =
Chocolate.objects.select_related("chocolatemanufacturer",
"chocolatemanufacturer__country", "bean_region").get(id =
chocolate)
    except Chocolate.DoesNotExist:
        return render(request, "chocoapp/choco.html",
                        {"chocolate": None
                        })
        chocolate.price =
Chocolateprice.objects.filter(chocolate = chocolate,
proposition_start__lte = time_for_search, proposition_end__gte =
time_for_search, active = True)
        chocolate.photos =
Chocolatephoto.objects.filter(chocolate = chocolate)
        chocolate.years =
Chocolateyear.objects.filter(chocolate = chocolate).order_by("-
review_average")
        chocolate.reviews =
Chocolatereview.objects.prefetch_related("tastes").filter(chocol
ateyear__chocolate = chocolate)
        chocolate.reviews = chocolate.reviews.order_by("-
review_date")[:5]
        if request.user.is_authenticated:
            user_for_clientaccount = request.user
            client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
            wishes = Chocolatewish.objects.filter(chocolate =
chocolate, clientaccount = client_account)
            if wishes:
                chocolate.wish = True
            else:
                chocolate.wish = False
        return render(request, "chocoapp/choco.html",
                        {"chocolate": chocolate
                        })

def ChocolatebarView(request, chocolatebar):
    try:
        chocolate_bar =
Chocolatebar.objects.select_related("chocolatemanufacturer",

```

```

"chocolatemanufacturer__country", "chocolatetype").get(id =
chocolatebar)
    except Chocolatebar.DoesNotExist:
        return render(request, "chocoapp/chocobar.html",
                        {"chocolate": None
                        })
    chocolate_bar.chocolates =
Chocolatebarchocolate.objects.select_related("chocolate",
        "chocolate__bean_region",
        "chocolate__bean_region__country",

"chocolate__chocolatemanufacturer").filter(chocolatebar =
chocolate_bar)
        chocolate_bar.masses =
Chocolatebarmass.objects.filter(chocolatebar = chocolate_bar)
        for chocolate in chocolate_bar.masses:
            chocolate.prices =
Chocolatebarmassprice.objects.filter(chocolatebarmass =
chocolate, proposition_start__lte = time_for_search,
proposition_end__gte = time_for_search, active = True)
            chocolate_bar.photos =
Chocolatebarphoto.objects.filter(chocolatebar = chocolate_bar)
            chocolate_bar.reviews =
Chocolatebarreview.objects.prefetch_related("tastes").filter(choc
olatebar = chocolate_bar)
            chocolate_bar.reviews =
chocolate_bar.reviews.order_by("-review_date")[:5]
            chocolate_bar.ingredients =
Chocolatebaringredient.objects.select_related("ingredient").filt
er(chocolatebar = chocolate_bar)
            if request.user.is_authenticated:
                user_for_clientaccount = request.user
                client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
                wishes =
Chocolatebarwish.objects.filter(chocolatebar = chocolate_bar,
clientaccount = client_account)
                if wishes:
                    chocolate_bar.wish = True
                else:
                    chocolate_bar.wish = False
        return render(request, "chocoapp/chocobar.html",
                        {"chocolate": chocolate_bar
                        })

def ChocolateSearchView(request):
    chocolates = Chocolate.objects.all()
    chocolate_manufacturers =
Chocolatemanufacturer.objects.select_related("country").all()
    countries = Country.objects.all()
    countries = countries.order_by("name")

```

```

        chocolate_manufacturers =
chocolate_manufacturers.order_by("country__name")
        chocolate_manufacturer =
request.GET.get("manufacturer")
        if chocolate_manufacturer:
            chocolate_manufacturer =
int(chocolate_manufacturer)
            chocolates =
chocolates.filter(chocolatemanufacturer =
chocolate_manufacturer)
            chocolate_sort = request.GET.get("sort")
            if chocolate_sort:
                chocolates = chocolates.filter(sort_name =
chocolate_sort)
            chocolate_type = request.GET.get("type")
            if chocolate_type:
                chocolates = chocolates.filter(base_chocolate_type
= chocolate_type)
            cacao_percent_max = request.GET.get("cacaomax")
            if cacao_percent_max:
                cacao_percent_max = float(cacao_percent_max)
                chocolates = chocolates.filter(cacao_percent__lte =
cacao_percent_max)
            cacao_percent_min = request.GET.get("cacaomin")
            if cacao_percent_min:
                cacao_percent_min = float(cacao_percent_min)
                chocolates = chocolates.filter(cacao_percent__gte =
cacao_percent_min)
            ingredient_formula =
request.GET.get("ingredientformula")
            if ingredient_formula:
                chocolates =
chocolates.filter(base_ingredient_formula = ingredient_formula)
            chocolate_review_min = request.GET.get("reviewmin")
            if chocolate_review_min:
                chocolate_review_min = float(chocolate_review_min)
                chocolates = chocolates.filter(review_average__gte
= chocolate_review_min)
            chocolate_review_max = request.GET.get("reviewmax")
            if chocolate_review_max:
                chocolate_review_max = float(chocolate_review_max)
                chocolates = chocolates.filter(review_average__lte
= chocolate_review_max)
            bean_region = request.GET.get("beanregion")
            if bean_region:
                chocolates =
chocolates.filter(bean_region__region_name = bean_region)
            time_for_search = datetime.now()
            for chocolate in chocolates:
                chocolate.price =
Chocolateprice.objects.filter(chocolate = chocolate,
proposition_start__lte = time_for_search, proposition_end__gte =
time_for_search, active = True)[0]

```

```

sortvar = request.GET.get("sorting")
if sortvar:
    sortvar = int(sortvar)
if sortvar == 0:
    chocolates = chocolates.order_by("-review_average")
elif sortvar == 1:
    chocolates = chocolates.order_by("review_average")
elif sortvar == 2:
    chocolates = chocolates.order_by("cacao_percent")
elif sortvar == 3:
    chocolates = chocolates.order_by("-cacao_percent")
elif sortvar == 4:
    chocolates = chocolates.order_by("price")
elif sortvar == 5:
    chocolates = chocolates.order_by("-price")
chocolate_pricemin = request.GET.get("pricemin")
if chocolate_pricemin:
    chocolate_pricemin = float(chocolate_pricemin)
chocolate_pricemax = request.GET.get("pricemax")
if chocolate_pricemax:
    chocolate_pricemax = float(chocolate_pricemax)
chocolates_results = []
for chocolate in chocolates:
    if (chocolate_pricemin and chocolate.price >=
chocolate_pricemin) or (not chocolate_pricemin):
        if (chocolate_pricemax and chocolate.price <=
chocolate_pricemax) or (not chocolate_pricemax):
            chocolates_results.append(chocolate)
chocolates = chocolates_results
chocolate_paginator = Paginator(chocolates, 20)
chocolate_page = request.GET.get("page")
chocolates =
chocolate_paginator.get_page(chocolate_page)
if request.user.is_authenticated:
    user_for_clientaccount = request.user
    client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
    for chocolate in chocolates:
        wishes = Chocolatewish.objects.filter(chocolate
= chocolate, clientaccount = client_account)
        if wishes:
            chocolate.wish = True
        else:
            chocolate.wish = False
        chocolate.photo =
Chocolatephoto.objects.filter(chocolate = chocolate)[0]
    return render(request, "chocoapp/chocosearch.html",
        {"chocolates": chocolates,
        "manufacturers":
chocolate_manufacturers,
        "countries": countries
    })

```

```

def ChocolatebarSearchView(request):
    chocolatebars = Chocolatebar.objects.all()
    chocolate_manufacturers =
Chocolatemanufacturer.objects.select_related("country").all()
    countries = Country.objects.all()
    countries = countries.order_by("name")
    chocolate_types = Chocolatetype.objects.all()
    chocolate_types = chocolate_types.order_by("type_name")
    chocolate_manufacturers =
chocolate_manufacturers.order_by("country__name")
    chocolate_manufacturer =
request.GET.get("manufacturer")
    if chocolate_manufacturer:
        chocolate_manufacturer =
int(chocolate_manufacturer)
        chocolatebars =
chocolatebars.filter(chocolatemanufacturer =
chocolate_manufacturer)
        chocolate_type = request.GET.get("type")
        if chocolate_type:
            chocolate_type = int(chocolate_type)
            chocolate_type = Chocolatetype.objects.get(id =
chocolate_type)
            chocolatebars = chocolatebars.filter(chocolatetype
= chocolate_type)
            cacao_percent_max = request.GET.get("cacaomax")
            if cacao_percent_max:
                cacao_percent_max = float(cacao_percent_max)
                chocolatebars =
chocolatebars.filter(cacao_percent__lte = cacao_percent_max)
                cacao_percent_min = request.GET.get("cacaomin")
                if cacao_percent_min:
                    cacao_percent_min = float(cacao_percent_min)
                    chocolatebars =
chocolatebars.filter(cacao_percent__gte = cacao_percent_min)
                chocolate_name = request.GET.get("name")
                if chocolate_name:
                    chocolatebars =
chocolatebars.filter(bar_name__icontains = chocolate_sort)
                chocolate_review_min = request.GET.get("reviewmin")
                if chocolate_review_min:
                    chocolate_review_min = float(chocolate_review_min)
                    chocolatebars =
chocolatebars.filter(review_average__gte = chocolate_review_min)
                chocolate_review_max = request.GET.get("reviewmax")
                if chocolate_review_max:
                    chocolate_review_max = float(chocolate_review_max)
                    chocolatebars =
chocolatebars.filter(review_average__lte = chocolate_review_max)
                energy_kcalories_min = request.GET.get("energymin")
                if energy_kcalories_min:
                    energy_kcalories_min = float(energy_kcalories_min)

```

```

        chocolatebars =
chocolatebars.filter(energy_kcalories__gte =
energy_kcalories_min)
        energy_kcalories_max = request.GET.get("energymax")
        if energy_kcalories_max:
            energy_kcalories_max = float(energy_kcalories_max)
            chocolatebars =
chocolatebars.filter(energy_kcalories__lte =
energy_kcalories_max)
        total_fat_gram_min = request.GET.get("fatmin")
        if total_fat_gram_min:
            total_fat_gram_min = float(total_fat_gram_min)
            chocolatebars =
chocolatebars.filter(total_fat_gram__gte = total_fat_gram_min)
        total_fat_gram_max = request.GET.get("fatmax")
        if total_fat_gram_max:
            total_fat_gram_max = float(total_fat_gram_max)
            chocolatebars =
chocolatebars.filter(total_fat_gram__lte = total_fat_gram_max)
        saturated_fat_gram_min = request.GET.get("satfatmin")
        if saturated_fat_gram_min:
            saturated_fat_gram_min =
float(saturated_fat_gram_min)
            chocolatebars =
chocolatebars.filter(saturated_fat_gram__gte =
saturated_fat_gram_min)
        saturated_fat_gram_max = request.GET.get("satfatmax")
        if saturated_fat_gram_max:
            saturated_fat_gram_max =
float(saturated_fat_gram_max)
            chocolatebars =
chocolatebars.filter(saturated_fat_gram__lte =
saturated_fat_gram_max)
        trans_fat_gram_min = request.GET.get("transfatmin")
        if trans_fat_gram_min:
            trans_fat_gram_min = float(trans_fat_gram_min)
            chocolatebars =
chocolatebars.filter(trans_fat_gram__gte = trans_fat_gram_min)
        trans_fat_gram_max = request.GET.get("transfatmax")
        if trans_fat_gram_max:
            trans_fat_gram_max = float(trans_fat_gram_max)
            chocolatebars =
chocolatebars.filter(trans_fat_gram__lte = trans_fat_gram_max)
        cholesterol_mgram_min = request.GET.get("cholmin")
        if cholesterol_mgram_min:
            cholesterol_mgram_min =
float(cholesterol_mgram_min)
            chocolatebars =
chocolatebars.filter(cholesterol_mgram__gte =
cholesterol_mgram_min)
        cholesterol_mgram_max = request.GET.get("cholmax")
        if cholesterol_mgram_max:

```

```

        cholesterol_mgram_max =
float(cholesterol_mgram_max)
        chocolatebars =
chocolatebars.filter(cholesterol_mgram__lte =
cholesterol_mgram_max)
        carbohydrate_gram_min = request.GET.get("carbomin")
        if carbohydrate_gram_min:
            carbohydrate_gram_min =
float(carbohydrate_gram_min)
            chocolatebars =
chocolatebars.filter(carbohydrate_gram__gte =
carbohydrate_gram_min)
            carbohydrate_gram_max = request.GET.get("carbomax")
            if carbohydrate_gram_max:
                carbohydrate_gram_max =
float(carbohydrate_gram_max)
                chocolatebars =
chocolatebars.filter(carbohydrate_gram__lte =
carbohydrate_gram_max)
                dietary_fiber_gram_min = request.GET.get("fibermin")
                if dietary_fiber_gram_min:
                    dietary_fiber_gram_min =
float(dietary_fiber_gram_min)
                    chocolatebars =
chocolatebars.filter(dietary_fiber_gram__gte =
dietary_fiber_gram_min)
                    dietary_fiber_gram_max = request.GET.get("fibermax")
                    if dietary_fiber_gram_max:
                        dietary_fiber_gram_max =
float(dietary_fiber_gram_max)
                        chocolatebars =
chocolatebars.filter(dietary_fiber_gram__lte =
dietary_fiber_gram_max)
                        sugar_gram_min = request.GET.get("sugarmin")
                        if sugar_gram_min:
                            sugar_gram_min = float(sugar_gram_min)
                            chocolatebars =
chocolatebars.filter(sugar_gram__gte = sugar_gram_min)
                            sugar_gram_max = request.GET.get("sugarmax")
                            if sugar_gram_max:
                                sugar_gram_max = float(sugar_gram_max)
                                chocolatebars =
chocolatebars.filter(sugar_gram__lte = sugar_gram_max)
                                protein_gram_min = request.GET.get("proteinmin")
                                if protein_gram_min:
                                    protein_gram_min = float(protein_gram_min)
                                    chocolatebars =
chocolatebars.filter(protein_gram__gte = protein_gram_min)
                                    protein_gram_max = request.GET.get("proteinmax")
                                    if protein_gram_max:
                                        protein_gram_max = float(protein_gram_max)
                                        chocolatebars =
chocolatebars.filter(protein_gram__lte = protein_gram_max)

```

```

sodium_mgram_min = request.GET.get("sodiummin")
if sodium_mgram_min:
    sodium_mgram_min = float(sodium_mgram_min)
    chocolatebars =
chocolatebars.filter(sodium_mgram__gte = sodium_mgram_min)
sodium_mgram_max = request.GET.get("sodiummax")
if sodium_mgram_max:
    sodium_mgram_max = float(sodium_mgram_max)
    chocolatebars =
chocolatebars.filter(sodium_mgram__lte = sodium_mgram_max)
calcium_mgram_min = request.GET.get("calciummin")
if calcium_mgram_min:
    calcium_mgram_min = float(calcium_mgram_min)
    chocolatebars =
chocolatebars.filter(calcium_mgram__gte = calcium_mgram_min)
calcium_mgram_max = request.GET.get("calciummax")
if calcium_mgram_max:
    calcium_mgram_max = float(calcium_mgram_max)
    chocolatebars =
chocolatebars.filter(calcium_mgram__lte = calcium_mgram_max)
iron_mgram_min = request.GET.get("ironmin")
if iron_mgram_min:
    iron_mgram_min = float(iron_mgram_min)
    chocolatebars =
chocolatebars.filter(iron_mgram__gte = iron_mgram_min)
iron_mgram_max = request.GET.get("ironmax")
if iron_mgram_max:
    iron_mgram_max = float(iron_mgram_max)
    chocolatebars =
chocolatebars.filter(iron_mgram__lte = iron_mgram_max)
salt_gram_min = request.GET.get("saltmin")
if salt_gram_min:
    salt_gram_min = float(salt_gram_min)
    chocolatebars = chocolatebars.filter(salt_gram__gte
= salt_gram_min)
    salt_gram_max = request.GET.get("saltmax")
    if salt_gram_max:
        salt_gram_max = float(salt_gram_max)
        chocolatebars = chocolatebars.filter(salt_gram__lte
= salt_gram_max)
    vitaminD_ngram_min = request.GET.get("vitmin")
    if vitaminD_ngram_min:
        vitaminD_ngram_min = float(vitaminD_ngram_min)
        chocolatebars =
chocolatebars.filter(vitaminD_ngram__gte = vitaminD_ngram_min)
    vitaminD_ngram_max = request.GET.get("vitmax")
    if vitaminD_ngram_max:
        vitaminD_ngram_max = float(vitaminD_ngram_max)
        chocolatebars =
chocolatebars.filter(vitaminD_ngram__lte = vitaminD_ngram_max)
    potassium_mgram_min = request.GET.get("potassiummin")
    if potassium_mgram_min:
        potassium_mgram_min = float(potassium_mgram_min)

```

```

        chocolatebars =
chocolatebars.filter(potassium_mgram__gte = potassium_mgram_min)
        potassium_mgram_max = request.GET.get("potassiummax")
        if potassium_mgram_max:
            potassium_mgram_max = float(potassium_mgram_max)
            chocolatebars =
chocolatebars.filter(potassium_mgram__lte = potassium_mgram_max)
        added_sugar_gram_min = request.GET.get("addsugarmin")
        if added_sugar_gram_min:
            added_sugar_gram_min = float(added_sugar_gram_min)
            chocolatebars =
chocolatebars.filter(added_sugar_gram__gte =
added_sugar_gram_min)
        added_sugar_gram_max = request.GET.get("addsugarmax")
        if added_sugar_gram_max:
            added_sugar_gram_max = float(added_sugar_gram_max)
            chocolatebars =
chocolatebars.filter(added_sugar_gram__lte =
added_sugar_gram_max)
        chocolate_sort = request.GET.get("sort")
        ingredient_formula =
request.GET.get("ingredientformula")
        bean_region = request.GET.get("beanregion")
        time_for_search = datetime.now()
        for chocolate in chocolatebars:
            chocolate.prices =
Chocolatebarmassprice.objects.filter(chocolatebarmass__chocolate
bar = chocolate, proposition_start__lte = time_for_search,
proposition_end__gte = time_for_search, active = True)[0]
            sortvar = request.GET.get("sorting")
            if sortvar:
                sortvar = int(sortvar)
            if sortvar == 0:
                chocolatebars = chocolatebars.order_by("-
review_average")
            elif sortvar == 1:
                chocolatebars =
chocolatebars.order_by("review_average")
            elif sortvar == 2:
                chocolatebars =
chocolatebars.order_by("cacao_percent")
            elif sortvar == 3:
                chocolatebars = chocolatebars.order_by("-
cacao_percent")
            elif sortvar == 4:
                chocolatebars = chocolatebars.order_by("price")
            elif sortvar == 5:
                chocolatebars = chocolatebars.order_by("-price")
            chocolate_pricemin = request.GET.get("pricemin")
            if chocolate_pricemin:
                chocolate_pricemin = float(chocolate_pricemin)
            chocolate_pricemax = request.GET.get("pricemax")
            if chocolate_pricemax:

```

```

        chocolate_pricemax = float(chocolate_pricemax)
        chocolates_results = []
        for chocolate in chocolatebars:
            for price in chocolate.prices:
                if (chocolate_pricemin and chocolate.price >=
chocolate_pricemin) or (not chocolate_pricemin):
                    if (chocolate_pricemax and chocolate.price
<= chocolate_pricemax) or (not chocolate_pricemax):
                        if not (chocolate in
chocolates_results):

chocolates_results.append(chocolate)
        chocolates_results_new = []
        for chocolate in chocolates_results:
            to_add = True
            if chocolate_sort:
                chocolates =
Chocolatebarchocolate.objects.filter(chocolatebar = chocolate,
chocolate__sort_name = chocolate_sort)
                if not chocolates:
                    to_add = False
            if ingredient_formula:
                chocolates =
Chocolatebarchocolate.objects.filter(chocolatebar = chocolate,
chocolate__base_ingredient_formula = ingredient_formula)
                if not chocolates:
                    to_add = False
            if bean_region:
                chocolates =
Chocolatebarchocolate.objects.filter(chocolatebar = chocolate,
chocolate__bean_region = bean_region)
                if not chocolates:
                    to_add = False
            if to_add:
                chocolates_results_new.append(chocolate)
        chocolatebars = chocolates_results_new
        chocolate_paginator = Paginator(chocolatebars, 20)
        chocolate_page = request.GET.get("page")
        chocolatebars =
chocolate_paginator.get_page(chocolate_page)
        if request.user.is_authenticated:
            user_for_clientaccount = request.user
            client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
            for chocolate in chocolatebars:
                wishes =
Chocolatebarwish.objects.filter(chocolatebar = chocolate,
clientaccount = client_account)
                if wishes:
                    chocolate.wish = True
                else:
                    chocolate.wish = False

```

```

        chocolate.photo =
Chocolatebarphoto.objects.filter(chocolatebar = chocolate)[0]
        return render(request, "chocoapp/chocobarsearch.html",
                        {"chocolatebars": chocolatebars,
                         "manufacturers":
chocolate_manufacturers,
                         "countries": countries,
                         "types": chocolate_types
                        })

def TypeChocolateBarsView(request, chocolate_type, sortvar
= 0):
    try:
        chocolate_type = Chocolatetype.objects.get(id =
chocolate_type)
    except Chocolatetype.DoesNotExist:
        return render(request,
"chocoapp/typechocobars.html",
                        {"type": None,
                         "chocolates": []
                        })
    chocolateBars =
Chocolatebar.objects.filter(chocolatetype = chocolate_type)
    time_for_search = datetime.now()
    for chocolate in chocolateBars:
        chocolate.masses =
Chocolatebarmass.objects.filter(chocolatebar = chocolate)
        chocolate.prices =
Chocolatebarmassprice.objects.filter(chocolatebarmass__chocolate
bar = chocolate, proposition_start__lte = time_for_search,
proposition_end__gte = time_for_search, active = True)[0]
        if sortvar == 0:
            chocolateBars = chocolateBars.order_by("-
review_average")
        elif sortvar == 1:
            chocolateBars =
chocolateBars.order_by("review_average")
        elif sortvar == 2:
            chocolateBars =
chocolateBars.order_by("cacao_percent")
        elif sortvar == 3:
            chocolateBars = chocolateBars.order_by("-
cacao_percent")
        elif sortvar == 4:
            chocolateBars = chocolateBars.order_by("price")
        elif sortvar == 5:
            chocolateBars = chocolateBars.order_by("-price")
        chocolate_paginator = Paginator(chocolateBars, 20)
        chocolate_page = request.GET.get("page")
        chocolateBars =
chocolate_paginator.get_page(chocolate_page)
        if request.user.is_authenticated:

```

```

        user_for_clientaccount = request.user
        client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
        for chocolate in chocolate_bars:
            wishes =
Chocolatebarwish.objects.filter(chocolatebar = chocolate,
clientaccount = client_account)
            if wishes:
                chocolate.wish = True
            else:
                chocolate.wish = False
            chocolate.photo =
Chocolatebarphoto.objects.filter(chocolatebar = chocolate)[0]
        return render(request, "chocoapp/typechocobars.html",
                        {"type": chocolate_type,
                        "chocolates": chocolate_bars
                        })

def ChocolatebarreviewsView(request, chocolatebar, sortvar
= 0):
    try:
        chocolate_bar =
Chocolatebar.objects.select_related("chocolatemanufacturer",
"chocolatemanufacturer__country", "chocolatetype").get(id =
chocolatebar)
        except Chocolatebar.DoesNotExist:
            return render(request,
"chocoapp/chocobarreviews.html",
                        {"chocolate": None
                        })
        chocolate_bar.photos =
Chocolatebarphoto.objects.filter(chocolatebar =
chocolate_bar)[0]
        chocolate_bar.reviews =
Chocolatebarreview.objects.prefetch_related("tastes").filter(cho
colatebar = chocolate_bar)
        if sortvar == 0:
            chocolate_bar.reviews =
chocolate_bar.reviews.order_by("-review_date")
        elif sortvar == 1:
            chocolate_bar.reviews =
chocolate_bar.reviews.order_by("review_date")
        chocolate_paginator = Paginator(chocolate_bar.reviews,
20)

        chocolate_page = request.GET.get("page")
        chocolate_bar.reviews =
chocolate_paginator.get_page(chocolate_page)
        if request.user.is_authenticated:
            user_for_clientaccount = request.user
            client_account = Clientaccount.objects.get(user =
user_for_clientaccount)

```

```

        wishes =
Chocolatebarwish.objects.filter(chocolatebar = chocolate_bar,
clientaccount = client_account)
        if wishes:
            chocolate_bar.wish = True
        else:
            chocolate_bar.wish = False
    return render(request, "chocoapp/chocobarreviews.html",
                  {"chocolate": chocolate_bar
                  })

def ChocolatereviewsView(request, chocolate, sortvar = 0):
    try:
        chocolate =
Chocolate.objects.select_related("chocolatemanufacturer",
"chocolatemanufacturer__country", "bean_region").get(id =
chocolate)
        except Chocolate.DoesNotExist:
            return render(request,
"chocoapp/chocoreviews.html",
                        {"chocolate": None
                        })
        chocolate.photo =
Chocolatephoto.objects.filter(chocolate = chocolate)[0]
        chocolate.reviews =
Chocolatereview.objects.prefetch_related("tastes").filter(chocol
ateyear__chocolate = chocolate)
        if sortvar == 0:
            chocolate.reviews = chocolate.reviews.order_by("-
review_date")
        elif sortvar == 1:
            chocolate.reviews =
chocolate.reviews.order_by("review_date")
            chocolate_paginator = Paginator(chocolate.reviews, 20)
            chocolate_page = request.GET.get("page")
            chocolate.reviews =
chocolate_paginator.get_page(chocolate_page)
            if request.user.is_authenticated:
                user_for_clientaccount = request.user
                client_account = Clientaccount.objects.get(user =
user_for_clientaccount)
                wishes = Chocolatewish.objects.filter(chocolate =
chocolate, clientaccount = client_account)
                if wishes:
                    chocolate.wish = True
                else:
                    chocolate.wish = False
    return render(request, "chocoapp/chocoreviews.html",
                  {"chocolate": chocolate
                  })

```

```

def ClientaccountreviewsView(request, reviewtype = 0,
clientaccount = None):
    if not clientaccount:
        if request.user.is_authenticated:
            user_for_clientaccount = request.user
            client_account = Clientaccount.objects.get(user
= user_for_clientaccount)
        else:
            return render(request,
"chocoapp/clientreviews.html",
                        {"client": None,
                        "reviews": []
                        })
    else:
        try:
            client_account =
Clientaccount.objects.get(user__username = clientaccount)
        except Clientaccount.DoesNotExist:
            return render(request,
"chocoapp/clientreviews.html",
                        {"client": None,
                        "reviews": []
                        })
        if reviewtype == 0:
            reviews =
Chocolatereview.objects.prefetch_related("tastes").filter(client
account = client_account)
        elif reviewtype == 0:
            reviews =
Chocolatebarreview.objects.prefetch_related("tastes").filter(cli
entaccount = client_account)
        reviews = reviews.order_by("-review_date")
        chocolate_paginator = Paginator(reviews, 20)
        chocolate_page = request.GET.get("page")
        reviews = chocolate_paginator.get_page(chocolate_page)
        return render(request, "chocoapp/chocoreviews.html",
                        {"client": client_account,
                        "reviews": reviews
                        })

```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
кафедра програмних засобів

Дипломна кваліфікаційна робота магістра на тему:
«Дослідження та програмна реалізація методів рекомендування
шоколаду. Research and Software Implementation of Methods for
Recommending the Chocolate»

Виконав: студент групи КНТ-123м

Микита ДЕМЧЕНКО

Керівник: к.т.н., доцент

Валерій ЛЬОВКІН

2024



Рисунок В.1 – Початковий слайд

Об'єкт, предмет і мета роботи

Об'єкт роботи – процес рекомендування шоколаду.

Предмет роботи – методи рекомендування шоколаду.

Метою роботи є розробка програмного забезпечення підтримки продажу та рекомендування шоколаду для інформування користувачів на різних етапах введення шоколаду в продаж.

Рисунок В.2 – Об'єкт, предмет і мета роботи

Завдання роботи

- виконати аналіз вебзастосунків продажу шоколаду;
- розробити метод рекомендування шоколаду;
- виконати проектування підтримки продажу та рекомендування шоколаду;
- реалізувати програму підтримки продажу та рекомендування шоколаду;
- виконати експериментальне дослідження рекомендування шоколаду на основі прогнозування оцінок шоколаду.

3

Рисунок В.3 – Завдання роботи

Порівняння аналогів розробки вебзастосунку продажу шоколаду

Показник	Belgian Chocolate Makers	Pierre Marcolini	Розробка
Замовлення шоколадних виробів	Так	Так	Так
Збереження шоколадних виробів у бажання	Так	Так	Так
Пошук шоколадних виробів	Вартість, тип, вміст какао, країна походження, особливості	Вартість, тип, смак, призначення	Вартість, тип шоколаду, виробу, харчова цінність, країна походження, вміст какао, сорт, рейтинг, інгредієнти
Визначення харчової цінності та складу інгредієнтів шоколадного виробу	Ні	Так	Так
Оцінювання шоколадних виробів у вебзастосунку з врахуванням врожаю	Ні	Ні	Так
Рекомендування шоколадних виробів	Ні	Ні	Так
Прогнозування оцінки шоколадного виробу	Ні	Ні	Так

4

Рисунок В.4 – Порівняння аналогів розробки вебзастосунку продажу шоколаду

Метод рекомендування шоколаду

- збір даних про шоколадні вироби або сорти шоколаду на основі заданого набору характеристик, перетворення набору значень характеристик на навчальну вибірку;
- створення моделі на основі XGBoost, її навчання на сформованій навчальній вибірці, застосування навченої моделі для визначення оцінок шоколаду, для якого така оцінка невідома або не є надійною, а значення потрібних вхідних характеристик при цьому задані;
- кластеризація всіх даних, які входять до початкової навчальної вибірки та тих даних, які були додатково до неї внесені на другому етапі, тобто екземплярів, для яких не була відома оцінка, на основі методу k-середніх;
- визначення користувачем того шоколаду, який відповідає його смакам, співвіднесення цих сортів або виробів з основним кластером та визначення рекомендацій на основі визначеного кластера.

5

Рисунок В.5 – Метод рекомендування шоколаду

Вибір вебфреймворку для розробки програми підтримки продажу та рекомендування шоколаду

	Django	Express.js
Тип фреймворку	Бекенд-фреймворк для веброзробки	Бекенд-фреймворк для веброзробки
Мова програмування	Python	JavaScript
Архітектура та структура застосунку	Чітка структура згідно з Model-View-Template як варіацією Model-View-Controller	Model-View-Controller без вимоги суворого дотримання
Взаємодія з базою даних	Шар об'єктно-реляційного відображення, інтерфейс адміністратора	Тільки інтерфейс для зв'язку з системою управління базою даних
Машинне навчання та робота з даними	Мова Python має більш розгалужену систему засобів	Мова JavaScript має меншу систему засобів

6

Рисунок В.6 – Вибір вебфреймворку для розробки програми підтримки продажу та рекомендування шоколаду

Діаграми варіантів використання програми підтримки продажу та рекомендування шоколаду



Рисунок В.7 – Діаграми варіантів використання програми підтримки продажу та рекомендування шоколаду

Схема бази даних

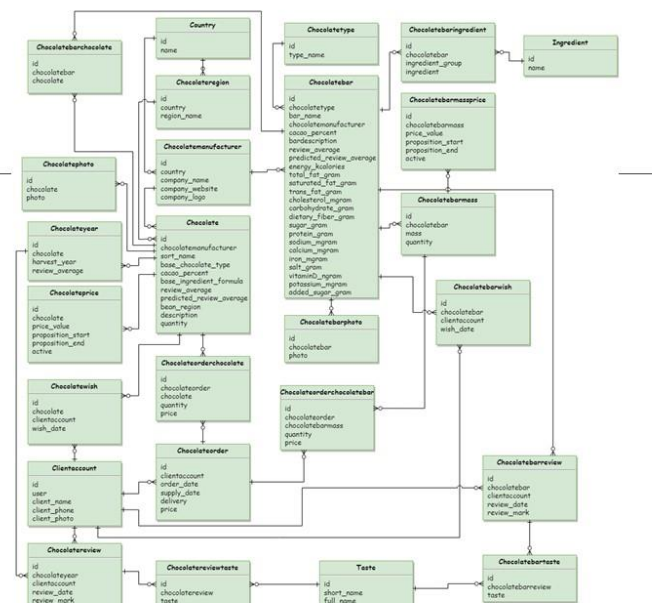
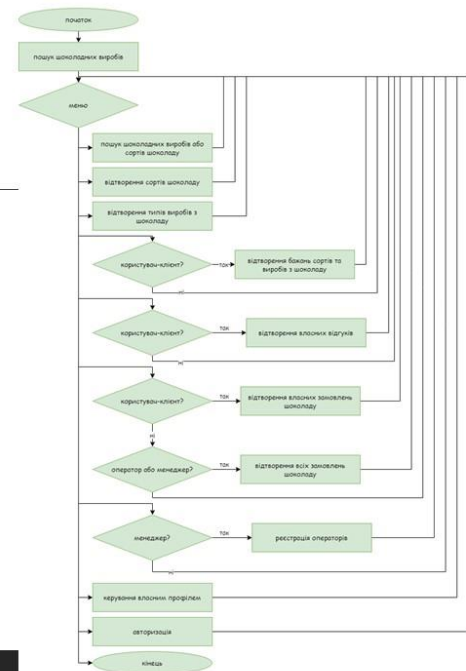


Рисунок В.8 – Схема бази даних

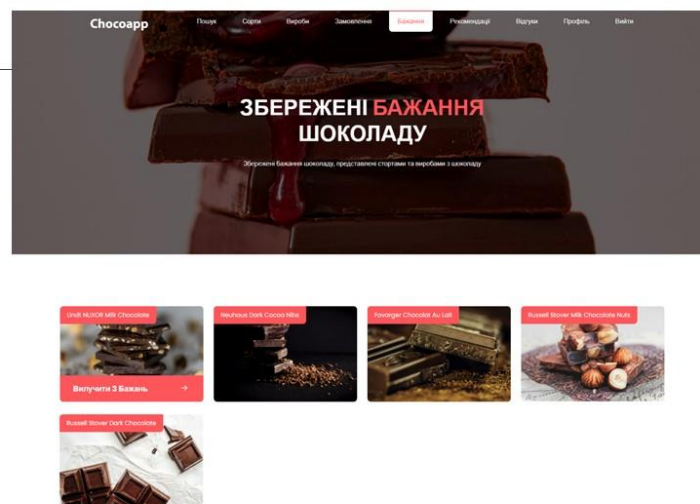
Схема функціонування програми підтримки продажу та рекомендації шоколаду



9

Рисунок В.9 – Схема функціонування програми підтримки продажу та рекомендації шоколаду

Відтворення переліку бажаних сортів та плиток шоколаду



10

Рисунок В.10 – Відтворення переліку бажаних сортів та плиток шоколаду

Рекомендування шоколаду

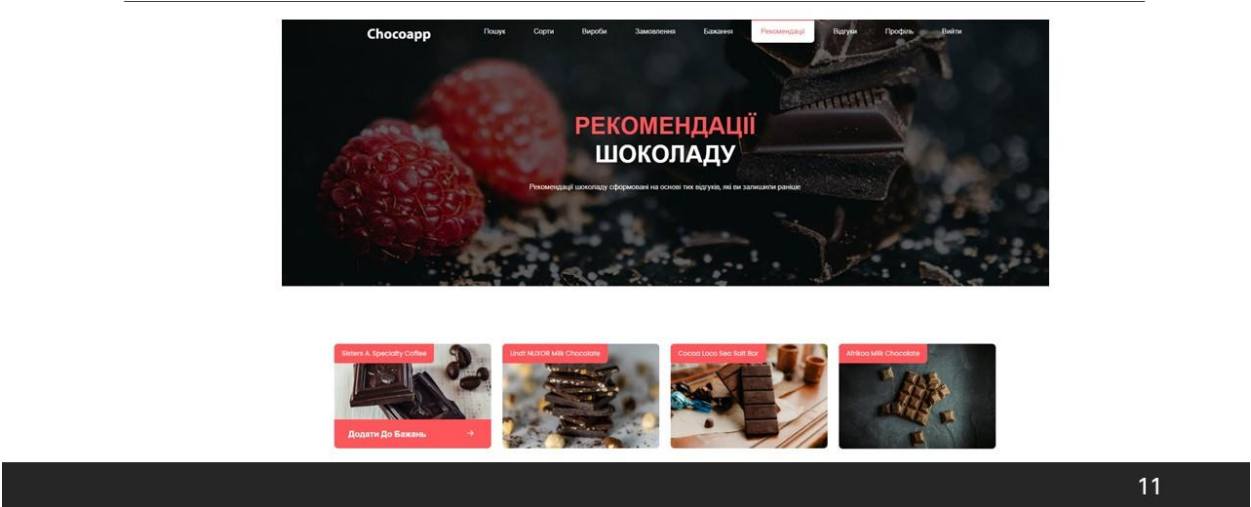


Рисунок В.11 – Рекомендування шоколаду

Відтворення даних плитки шоколаду

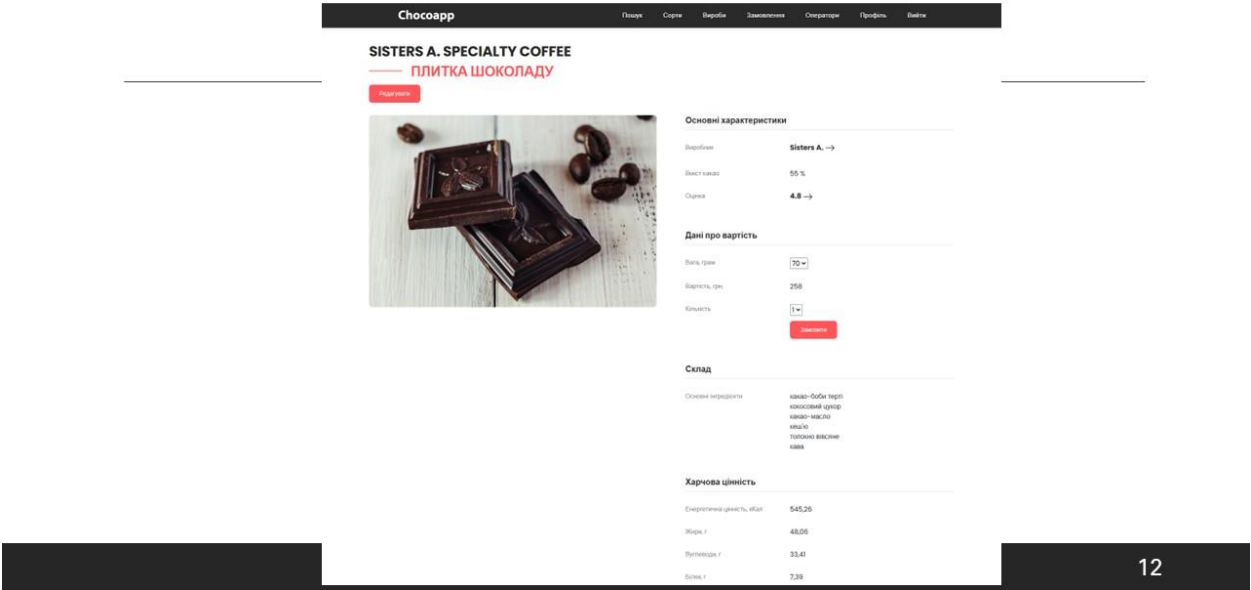


Рисунок В.12 – Відтворення даних плитки шоколаду

Результати експериментального дослідження

Модель або метод	Кількість ознак	MAE	RMSE	MSE
1	2	3	4	5
Дерево рішень	всі	0,00208076	0,00275025	7,5638574E-6
Метод k-найближчих сусідів	всі	0,00218866	0,0027651	7,64577029E-6
Метод опорних векторів	всі	0,00428611	0,0048854	2,38671622E-5
Метод, який запропоновано (XGBoost)	всі	0,00165445	0,00213002	4,53700512E-6
Дерево рішень	100	0,00229257	0,00301987	9,11960299E-6
Метод k-найближчих сусідів	100	0,00220007	0,00277957	7,72600949E-6
Метод, який запропоновано (XGBoost)	100	0,00172055	0,00222864	4,96684586E-6
Дерево рішень	50	0,0023078	0,00307646	9,46463555E-6
Метод k-найближчих сусідів	50	0,00219830	0,00277694	7,71138809E-6
Метод, який запропоновано (XGBoost)	50	0,0017553	0,00228987	5,24348549E-6
Дерево рішень	20	0,00243447	0,00324566	1,05343398E-5
Метод k-найближчих сусідів	20	0,00221119	0,00278734	7,76926425E-6
Метод, який запропоновано (XGBoost)	20	0,00187075	0,00240641	5,79082722E-6
Дерево рішень	10	0,00264144	0,00345301	1,19232508E-5
Метод k-найближчих сусідів	10	0,00221996	0,00279584	7,81673049E-6

1	2	3	4	5
Метод, який запропоновано (XGBoost)	10	0,00195754	0,00248817	6,19097045E-6
Дерево рішень	6	0,00271689	0,00354446	1,25632138E-5
Метод k-найближчих сусідів	6	0,00225141	0,00283312	8,02654151E-6
Метод, який запропоновано (XGBoost)	6	0,00201609	0,00253937	6,4484029E-6

Рисунок В.13 – Результати експериментального дослідження

Висновки

У роботі виконана розробка програмного забезпечення підтримки продажу та рекомендування шоколаду для інформування користувачів на різних етапах введення шоколаду в продаж.

Реалізовано вибір для програмного забезпечення підтримки продажу та рекомендування шоколаду засобів реалізації, що включають в основі мову програмування Python, фреймворк Django, систему MySQL, моделювання сценаріїв роботи користувача-клієнта, оператора, менеджера компанії з продажу шоколаду, визначення структури бази даних, описано створені програмні одиниці.

Створена програма призначена для виконання замовлення як окремих сортів шоколаду, так і плиток шоколаду, шоколадних цукерок або інших виробів з шоколаду, оцінювання власних вражень від споживання цих сортів та плиток шоколаду, використання цих даних для рекомендування шоколаду користувачам. Оцінювання здійснюється з врахуванням року врожаю.

Наукова новизна роботи полягає в методі рекомендування шоколаду, який виконує прогнозування відсутніх оцінок клієнтами сортів шоколаду за допомогою моделі XGBoost, визначає перетворення основних характеристик сорту шоколаду на конкретні вхідні ознаки, що дозволяє виконувати рекомендування шоколаду за допомогою виконання кластеризації на основі методу k-середніх.

Рисунок В.14 – Висновки