

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)

ПІДТРИМКИ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ МАГАЗИНУ  
КОСМЕТИКИ. RESEARCH AND SOFTWARE IMPLEMENTATION  
OF METHODS FOR SUPPORTING INTERACTIONS WITH USERS  
OF COSMETICS STORE

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

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

УШАКОВ Д.В.

(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)**

УШАКОВА Давида Вадимовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів підтримки взаємодії з користувачами магазину косметики. Research and Software Implementation of Methods for Supporting Interactions with Users of Cosmetics Store

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, кількості 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)

\_\_\_\_\_ Давид УШАКОВ  
( підпис ) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_ Валерій ЛЬОВКІН  
( підпис ) (Ім'я ПРИЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:  
105 с., 26 рис., 8 табл., 3 дод., 10 джерел.

КОСМЕТИЧНІ ЗАСОБИ, МАГАЗИН, ЗАДАЧА ПРОГНОЗУВАННЯ,  
ОЦІНЮВАННЯ, АНСАМБЛІ ДЕРЕВ РІШЕНЬ, МАШИННЕ НАВЧАННЯ.

Об'єкт роботи – процес продажу косметики. Предмет роботи – методи підтримки взаємодії з користувачами на основі прогнозування. Мета роботи – розробити програмне забезпечення продажу косметики для підтримки взаємодії з користувачами на основі прогнозування оцінювання використання косметичних засобів на попередньому етапі.

Матеріали, методи та технічні засоби: мова Python разом з відповідними засобами машинного навчання, фреймворк веброзробки Django, система керування базами даних PostgreSQL.

Результати. Розроблена програма призначена для підтримки взаємодії з користувачами магазину косметики через надання їм інформації про наявні косметичні засоби в магазині, підбір потрібних засобів та замовлення цих засобів, надання працівникам магазину розуміння тенденцій в роботі магазину через аналіз обсягів продажів та прогнозування середньої оцінки косметичного засобу від користувачів.

Висновки. Наукова новизна роботи полягає в методі прогнозування оцінювання використання косметичних засобів покупцями, що забезпечує формування ознак, які представляють назву косметичного засобу, на основі моделі Word2Vec, ознак, які представляють склад косметичного засобу, створення моделі ансамблю дерев рішень, що дозволяє прогнозувати середню оцінку, яку отримає косметичний засіб від покупців.

Галузь використання. Підтримка роботи магазинів з продажу косметичних засобів.

## ABSTRACT

Explanatory note to the diploma qualifying work of the master: 105 pages, 26 figures, 8 tables, 3 appendixes, 10 sources.

COSMETICS, STORE, PREDICTION PROBLEM, EVALUATION, ENSEMBLES OF DECISION TREES, MACHINE LEARNING.

The object of the work is the process of selling cosmetics. The subject of the work is methods of supporting interaction with users based on prediction. The goal of the work is to develop cosmetics sales software to support interaction with users based on predictive evaluation of the use of cosmetic products at the preliminary stage.

Materials, methods and technical means: Python language together with the corresponding machine learning tools, Django web development framework, PostgreSQL database management system.

The results. The developed program is designed to support interaction with users of the cosmetics store by providing them with information about the available cosmetics in the store, selecting the necessary products and ordering these products, providing store employees with an understanding of trends in the store's work through the analysis of sales volumes and forecasting of the average evaluations of cosmetic products from users.

Conclusions. The scientific novelty of the work consists in the method of predicting the evaluation of the use of cosmetic products by buyers, which ensures the formation of features that represent the name of the cosmetic product based on the Word2Vec model, features that represent the composition of the cosmetic product, the creation of an ensemble model of decision trees that allows predicting the average rating, which the cosmetic product will receive from buyers.

Field of use. Support for the operation of stores selling cosmetics.

## ЗМІСТ

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
|                                                                                         | С. |
| Перелік скорочень та умовних познач.....                                                | 8  |
| Вступ.....                                                                              | 9  |
| 1 Аналіз проблеми та постановка завдань дослідження .....                               | 10 |
| 1.1 Огляд вебзастосунків продажу косметики .....                                        | 10 |
| 1.2 Порівняння аналогів розробки вебзастосунку продажу косметики .....                  | 13 |
| 1.3 Висновки за розділом 1 .....                                                        | 14 |
| 2 Матеріали і методи.....                                                               | 16 |
| 2.1 Задача прогнозування оцінювання використання косметичних засобів<br>покупцями ..... | 16 |
| 2.2 Метод прогнозування оцінювання використання косметичних засобів<br>покупцями .....  | 19 |
| 2.3 Висновки за розділом 2 .....                                                        | 21 |
| 3 Проєктування програмного забезпечення .....                                           | 22 |
| 3.1 Вибір мови та засобів розробки.....                                                 | 22 |
| 3.2 Моделювання роботи з програмою за функціональними вимогами .....                    | 23 |
| 3.3 Проєктування структури бази даних.....                                              | 28 |
| 3.4 Висновки за розділом 3 .....                                                        | 37 |
| 4 Основні рішення щодо реалізації програмного забезпечення .....                        | 38 |
| 4.1 Порядок взаємодії користувача з програмою магазину продажу<br>косметики .....       | 38 |
| 4.2 Рішення з реалізації вебзастосунку .....                                            | 39 |
| 4.3 Висновки за розділом 4 .....                                                        | 53 |
| 5 Експлуатація, тестування та експериментальне дослідження програми .....               | 54 |
| 5.1 Опис програми.....                                                                  | 54 |
| 5.2 Експлуатація програми.....                                                          | 54 |
| 5.3 Експериментальне дослідження .....                                                  | 59 |
| 5.4 Висновки за розділом 5 .....                                                        | 66 |
| Висновки .....                                                                          | 68 |

|                                   |    |
|-----------------------------------|----|
| Перелік джерел посилання .....    | 69 |
| Додаток А Технічне завдання ..... | 70 |
| Додаток Б Текст програми .....    | 76 |
| Додаток В Слайди презентації..... | 97 |

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

MAE – Mean Absolute Error;

MSE – Mean Squared Error;

RMSE – Root Mean Squared Error.

## ВСТУП

Магазини косметичних засобів звісно мають велику конкуренцію між собою, але з іншого боку і привертають увагу великої кількості клієнтів. При цьому розвиток засобів машинного навчання може будь-якому такому магазину давати поштовх до розвитку. Інтеграція відповідних функціональних можливостей у програмне забезпечення може дозволити отримати конкурентну перевагу в підсумку. Звісно, що для існування магазину та його успішності важливою є взаємодія з користувачами і те, як вона організована. У той же час кожен магазин має свою клієнтську базу, в жодного з них вона не співпадає. Це призводить до необхідності в тих інструментах, які б дозволяли краще розуміти свою клієнтську базу. Такі інструменти зокрема є корисними при появі нових косметичних засобів. Тоді необхідно розуміти, як клієнти будуть на них реагувати.

Одним з важливих напрямків у даному процесі є розуміння оцінок, які будуть в середньому виставлені косметичному засобу. Однак, для того щоб отримати точну інформацію про середню оцінку, необхідний достатньо тривалий період часу. Можуть минати роки, адже на початковому етапі кількість відгуків або дуже мала, або вони взагалі відсутні. Тому прогнозування середньої оцінки косметичного засобу в момент його появи дозволяє розмістити наявну аудиторію та планувати подальшу роботу з ним. Відповідно інтегрування таких засобів у програмне забезпечення підтримки магазину косметики є важливим завданням.

Враховуючи все це, метою роботи було визначено розробити програмне забезпечення продажу косметики для підтримки взаємодії з користувачами на основі прогнозування оцінювання використання косметичних засобів на попередньому етапі.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд вебзастосунків продажу косметики

Одним з найвідоміших вебзастосунків продажу косметики в Україні є вебсайт Makeup (рис. 1.1) [1]. Окрім косметики компанія здійснює також продаж і парфумерії.

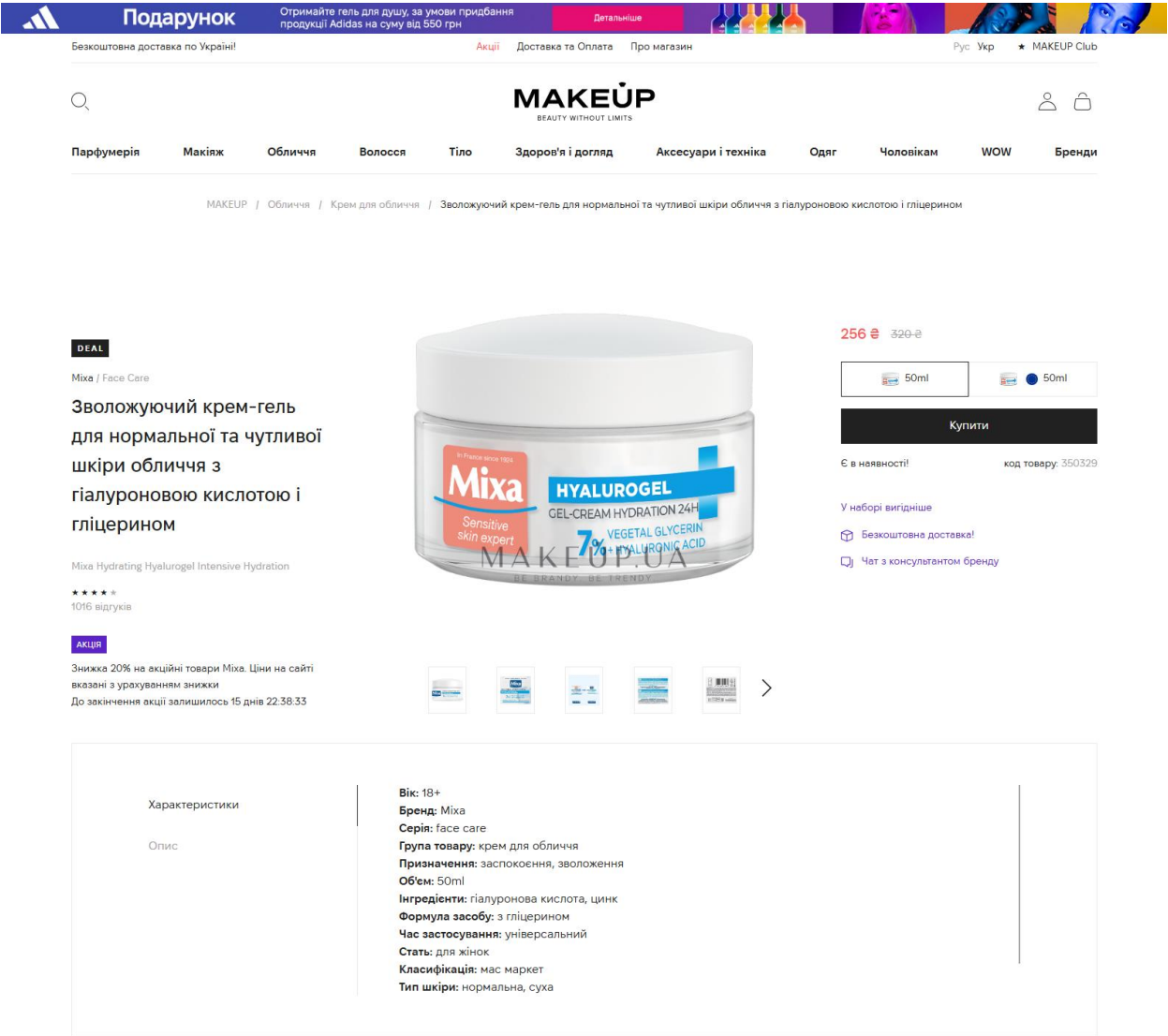


Рисунок 1.1 – Сторінка косметичного засобу на вебсайті Makeup

І пошук косметичних засобів, і вибір відповідного розділу з меню з групою косметичних засобів призводить до вибору сторінки фактично

пошуку, де є велика кількість фільтрів, які дозволяють відібрати необхідні косметичні засоби.

Кожен косметичний засіб містить на своїй сторінці доступні варіанти обсягу та відповідну вартість, назву, фотографії, відгуки, які залишили покупці, опис характеристик, що зокрема враховує вікову групу, бренд, серію, групу товару, призначення, інгредієнти, формулу, час застосування, стать, ринкову класифікацію, тип шкіри для застосування. Ці характеристики були докладно вивчені та в подальшому використані за основи при створенні власної розробки в роботі.

Кожен користувач може зареєструватися, отримуючи доступ до власного особистого кабінету, отримуючи інформацію зокрема про власні замовлення косметичних засобів таким чином.

Окрім розподілу косметичних засобів за групами, використання сторінки пошуку, доступний розділ з переліком брендів, який дозволяє фактично в пошуку обрати потрібний фільтр для відтворення косметичних засобів тільки за обраним брендом.

Компанія Brocard є однією з широковідомих в Україні за рахунок своїх мережевих магазинів, які продають засоби парфумерії та косметику. Наявний і вебсайт для підтримки роботи в мережі (рис. 1.2) [2].

Вебзастосунок також окрім розділів у каталозі товарів, що включає різні критерії групування косметичних засобів, містить розділ з брендами. При цьому вибір бренду відкриває окрему сторінку з інформацією про бренд, його косметичними засобами, тобто використовується не підхід на основі єдиної сторінки пошуку як і Makeup, а інший варіант. При цьому доступна в описі менша кількість характеристик. Вікові групи представлені конкретними проміжками, відповідно фактично кожен косметичний засіб має цілу множину невеликих вікових інтервалів.

Пошук реалізується за назвою. А отримати доступ до фільтрування косметичних засобів можна тільки, якщо обрати відповідну групу косметичних засобів з каталогу. Тоді демонструється сторінка, на якій разом з

результатами можна визначати різні критерії, фільтри, що дозволяє відбирати потрібні косметичні засоби.

Гаряча лінія 0 800 508 880

**BROCARD**  
P A R F U M S

Київ UA

Каталог товарів | Бренди | Акції | Новини | Магазини | Б'юти-експерт | SALE

Що ви шукаєте?

Назад • Brocard • Догляд за обличчям та очима • Креми для обличчя • Нічні креми • Нічний крем Bioten NutriCalcium

**BIOTEN**  
**NutriCalcium**  
Нічний крем

5.0 ★★★★★ на основі 1 відгуку

Код товару V165729548  
Штрих-код 5201314150749

Стать: для жінок

Вибір: 50 ML

Ціна: ~~653,00~~ **522,40**

Додати в кошик

Перевірити наявність в магазині

Детальніше

Тільки оригінальна продукція

Для замовлень в інтернет магазині BROCARD.UA доставка безкоштовна при онлайн оплаті від 799 грн. та при постоплаті від 999 грн. У мобільному застосунку – безкоштовна доставка без обмежень по сумі замовлення.

Оплата при отриманні або картою на сайті

**Опис** | Як використовувати | Характеристики

Антивіковий живильний нічний крем Bioten NutriCalcium містить 94% інгредієнтів природного походження. Кальцій покращує бар'єрну функцію шкіри та сприяє регенерації шкіри, тоді як активна формула, підсилена унікальною зволожувальною властивістю вітаміну B5, допомагає забезпечити здорове сяяння шкіри. FemiGen+ потужний активний екстракт натурального походження, покращує еластичність шкіри, розгладжує глибокі зморшки та підтягує контур обличчя, роблячи вашу шкіру помітно омолодженію та пружнішою.

Спеціально розроблений для жінок у період менопаузи та постменопаузи, крем допомагає зменшити ознаки гормонального старіння.

**Відгуки (1)**  
Поділіться своїми враженнями про товар

Залишити відгук

★★★★★  
Bioten  
NutriCalcium  
нічний крем  
Вибір 50 ML  
~~653,00~~  
**522,40**  
Додати в кошик

Рисунок 1.2 – Сторінка косметичного засобу на вебсайті Brocard

Користувачі вебсайту Brocard також можуть зареєструватися, авторизуватися та працювати з власним кабінетом, отримуючи доступ до виконаних замовлень. Користувачі також можуть залишати відгуки, а потім переглядати ці відгуки на сторінці косметичного засобу. Загалом робота з програмою є достатньо простою і зрозумілою. Оцінки косметичних засобів формуються на основі відгуків покупців.

Також було проаналізовано вебсайт компанії PROSTOR (рис. 1.3) [3].

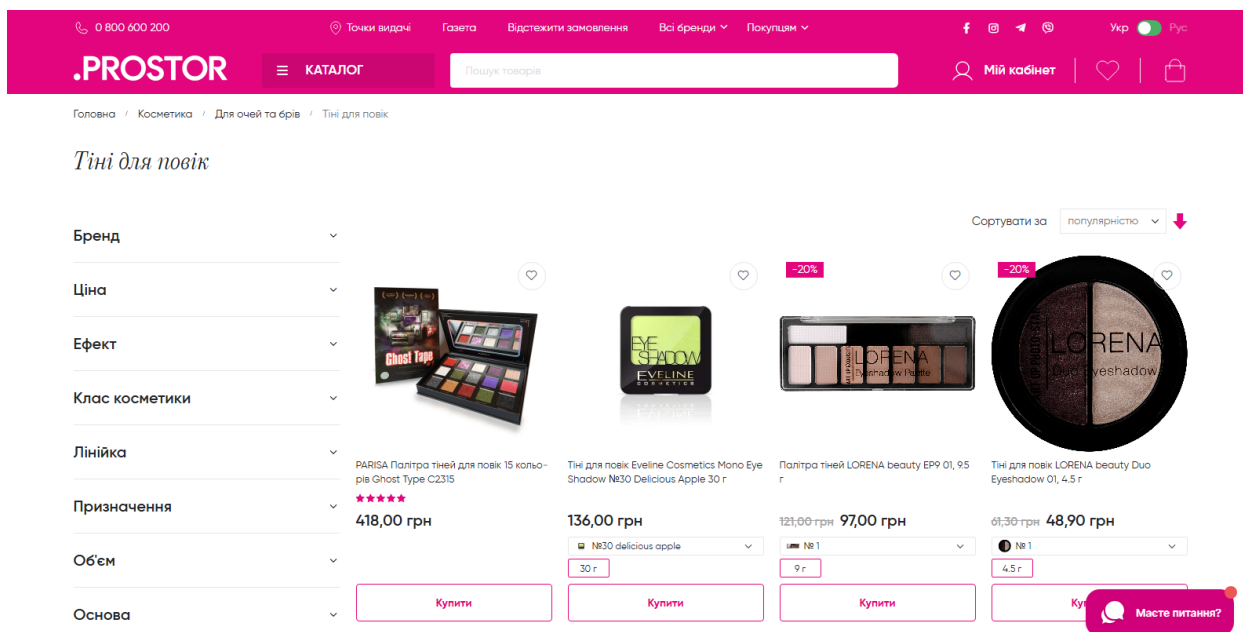


Рисунок 1.3 – Сторінка косметичних засобів за групою на вебсайті PROSTOR

Функціонально він достатньо подібний до попередніх проаналізованих вебзастосунків. Має певні відмінності в наборі характеристик косметичних засобів, серед яких найбільш повним слід вважати вебсайт Makeup.

## 1.2 Порівняння аналогів розробки вебзастосунку продажу косметики

До підсумкового порівняння винесено вебзастосунки Makeup і Brocard. Порівняння виконано з запланованою розробкою (табл. 1.1). Головною відмінністю розробки є наявність засобів прогнозування оцінки покупців для косметичного засобу. Це повинно дозволити підтримувати взаємодію з користувачами вебзастосунку на основі планування даного процесу за рахунок попереднього прогнозування. Засоби фільтрування косметичних засобів вирішено унаслідувати від Makeup як прототипу, хоча їх було в подальшому розширено додатково при реалізації пошуку косметичних засобів. У свою чергу Brocard обмежує пошук виконанням його за назвою, а сторінка з фільтрами за косметичними засобами не надає можливість задати назву косметичного засобу. Окрім того в розробці за прототип використано набір

характеристик косметичного засобу, наявний у вебзастосунку Makeup, що дозволяє в подальшому мати більше фільтрів за косметичними засобами в результаті.

Таблиця 1.1 – Порівняння аналогів розробки вебзастосунку продажу косметики

| Показник                                              | Makeup                       | Brocard                                                          | Розробка                     |
|-------------------------------------------------------|------------------------------|------------------------------------------------------------------|------------------------------|
| Пошук косметичних засобів                             | З великою кількістю фільтрів | Тільки за назвою, відбір за фільтрами доступний для груп засобів | З великою кількістю фільтрів |
| Кількість характеристик косметичних засобів           | Ширша                        | Базова                                                           | Ширша                        |
| Засоби особистого кабінету користувача                | +                            | +                                                                | +                            |
| Створення відгуків                                    | +                            | +                                                                | +                            |
| Прогнозування оцінки покупців для косметичного засобу | —                            | —                                                                | +                            |

### 1.3 Висновки за розділом 1

Завдання для виконання мети розробки програмного забезпечення продажу косметики для підтримки взаємодії з користувачами на основі прогнозування оцінювання використання косметичних засобів на попередньому етапі:

- виконати аналіз програм продажу косметики;
- розробити метод прогнозування оцінки косметичних засобів покупцями;

- виконати проєктування програми продажу косметики;
- реалізувати програму продажу косметики;
- виконати експериментальне дослідження прогнозування оцінки косметичних засобів покупцями.

У розділі 1 виконано огляд вебзастосунків продажу косметики, включаючи вебсайти Makeup, Brocard, PROSTOR. Визначено основні функції, характеристики сутностей предметної області, що було використано при проєктуванні предметної області в подальшому, сформульовано відмінності програми-розробки від аналогів. Головною відмінністю є наявність засобів прогнозування оцінки покупців для косметичного засобу.

## 2 МАТЕРІАЛИ І МЕТОДИ

### 2.1 Задача прогнозування оцінювання використання косметичних засобів покупцями

Косметичні засоби є товаром, який потребує особливої підтримки взаємодії з користувачами. Це пов'язано з тим, що покупець не може спробувати всі доступні засоби, а пошук нового косметичного засобу може бути значною проблемою. Це пов'язано з тим, що використання косметичних засобів пов'язано з індивідуальними особливостями. Наприклад, різні засоби можуть викликати подразнення шкіри або інші індивідуальні реакції. Тоді досвід інших покупців стає дуже важливим. Перед покупкою покупці нерідко аналізують досвід, про який пишуть інші покупці, а тоді вже вибирають відповідний косметичний засіб. Звісно, що в такому випадку критичними для вибору стають самі відгуки, але також і середній бал різних відгуків, таким чином підкреслюючи певне загальне рішення.

У свою чергу для магазину косметики важливо розуміти, яким може бути цей середній бал. Чим раніше є ця інформація, тим простіше реалізувати подальші дії. Ці дії можуть бути пов'язані з маркетинговою стратегією всередині самого вебзастосунку магазину продажу косметики. Окрім того такі дані можуть впливати на те, які можуть бути заплановані обсяги продажу. Отримати цю інформацію фактично можна через великий проміжок часу: можливо рік або роки. Відповідно використання прогнозування при додаванні косметичного засобу дозволяє планувати діяльність, розуміючи яким буде сприйняття покупцями даного косметичного засобу.

Тоді задача прогнозування оцінювання використання косметичних засобів покупцями полягає в тому, щоб за початковими даними, які характеризують косметичний засіб на момент отримання його початкової інформації, виконати прогнозування оцінки, яку всередньому виставлять йому покупці в майбутньому та яку слід вважати стабільною оцінкою. Стабільною слід вважати оцінку, яка підкріплена кількістю відгуків, після якої оцінка не

буде значно змінюватися. Звичайно, що у випадках зміни якості, наприклад, оцінка далі може змінюватись, однак за умов такої незмінюваності і оцінка має бути приблизно стабільною.

Відповідно задача прогнозування оцінювання використання косметичних засобів покупцями є задачею машинного навчання, в якій є вибірка даних, яка описує косметичні засоби, за якими вже отримано стабільні оцінювання від покупців. За цими косметичними засобами відомо значення ряду характеристик. Для кожного косметичного засобу набір значень цих характеристик формує вектор вхідних ознак. Для кожного такого вектору відоме значення цільової ознаки, якою є середня оцінка, отримана косметичним засобом від покупців. При цьому важливо, щоб ця оцінка була розрахована саме на основі відгуків покупців, а не будь-яких користувачів. Використання будь-яких оцінок замість оцінок виключно покупців може спотворити реальні дані.

Вибірка даних представлена екземплярами, кожен з яких представляє косметичний засіб. Для кожного екземпляра відомий набір значень вхідних ознак та значення середньої оцінки. Задача полягає у тому, щоб для косметичних засобів, для яких невідома ще середня оцінка покупців, на основі характеристик, які їх представляють, виконати прогнозування такої середньої оцінки. Така задача є задачею прогнозування або задачею регресії.

Для розв'язання задачі необхідно використати вибірку даних та на її основі навчити модель, за допомогою якої потім для нових косметичних засобів, представлених як черговий екземпляр даних, виконати прогнозування середньої оцінки.

При цьому додатковим питанням є питання визначення вхідних ознак. Множина вхідних ознак може бути уточнена на основі наявних даних під час розв'язання задачі, однак у даній постановці початково множина ознак складається з наступних:

- тип косметичного засобу;
- бренд косметичного засобу;

– назва косметичного засобу – назва, якою косметичний засіб підписаний у програмі, де відбувається взаємодія з покупцями, продаж косметичних засобів;

– вартість продажу одиниці косметичного засобу;

– перелік інгредієнтів, що входять до складу косметичного засобу, представлений у текстовому форматі на основі назв, перелічених через кому;

– призначення косметичного засобу для сухої шкіри: значення 1 вказує на призначення косметичного засобу для використання для сухої шкіри, 0 вказує на відсутність такого призначення;

– призначення косметичного засобу для нормальної шкіри: значення 1 вказує на призначення косметичного засобу для використання для нормальної шкіри, 0 вказує на відсутність такого призначення;

– призначення косметичного засобу для жирної шкіри: значення 1 вказує на призначення косметичного засобу для використання для жирної шкіри, 0 вказує на відсутність такого призначення;

– призначення косметичного засобу для одночасно сухої та жирної шкіри: значення 1 вказує на призначення косметичного засобу для використання для жирної та сухої шкіри, 0 вказує на відсутність такого призначення;

– призначення косметичного засобу для чутливої шкіри: значення 1 вказує на призначення косметичного засобу для використання для чутливої шкіри, 0 вказує на відсутність такого призначення.

Текстові дані в явному вигляді не можуть бути передані на вхід моделі, тому потребують певного перетворення. У той же час можливо деякі з цих ознак не є інформативними, а тому можуть бути відкинуті. Однак, це вже визначається рішенням цієї задачі, а не самою постановкою задачі прогнозування оцінювання використання косметичних засобів покупцями, яка представлена у цьому підрозділі.

## **2.2 Метод прогнозування оцінювання використання косметичних засобів покупцями**

Метод прогнозування оцінювання використання косметичних засобів покупцями запропоновано реалізовувати на основі такої послідовності етапів.

Спочатку має бути сформована вибірка даних, що містить екземпляри, кожен з яких представляє косметичний засіб. Кожен екземпляр містить наступні характеристики на даний момент:

- тип косметичного засобу, представлений категоріальним значенням, тобто всі можливі типи мають бути пронумеровані, а тоді саме цей номер (ціле число) встановлено у даній характеристиці;
- бренд косметичного засобу, представлений категоріальним значенням, тобто всі можливі типи мають бути пронумеровані, а тоді саме цей номер (ціле число) встановлено у даній характеристиці;
- назва косметичного засобу;
- вартість продажу одиниці косметичного засобу;
- перелік інгредієнтів;
- призначення косметичного засобу для сухої шкіри;
- призначення косметичного засобу для нормальної шкіри;
- призначення косметичного засобу для жирної шкіри;
- призначення косметичного засобу для одночасно сухої та жирної шкіри;
- призначення косметичного засобу для чутливої шкіри;
- середня оцінка, виставлена косметичному засобу покупцями (цільова ознака).

Далі має бути перетворено перелік інгредієнтів на вектор ознак. Для цього необхідно спочатку проаналізувати всі доступні інгредієнти, пройшовши по всьому доступному набору переліків інгредієнтів та виділивши з текстів окремі складові. Таким чином необхідно сформувати перелік унікальних інгредієнтів, що формують вектор. Тоді кожен екземпляр може

бути представлений таким вектором, на кожній позиції якого, що відповідає одному з інгредієнтів, визначено 1, коли цей інгредієнт входить до складу цього косметичного засобу, та 0 у протилежному випадку.

Після цього має бути перетворено назву на вектор ознак. Для цього має застосовуватися модель Word2Vec, яка дозволяє перевести текстові дані на числові значення [4]. Відповідно для кожного екземпляру формується вектор, який представляє назву у вигляді послідовності числових значень, кількість яких є заданою. У подальшому дослідженні така кількість встановлювалась рівною 50.

Після цього вибірка є повністю готовою для проведення навчання моделі та включає представлений вище набір ознак, послідовність ознак для представлення інгредієнтів зі складу косметичних засобів та послідовність ознак для представлення назви косметичного засобу.

Наступним етапом є реалізація створення моделі, яка має бути забезпечена у вигляді ансамблю дерев рішень на основі застосування методу Random Forest [5]. За результатами проведеного дослідження рекомендовано кількість дерев у ансамблі встановлювати рівною 500. Тоді така створена модель прогнозування оцінювання використання косметичних засобів покупцями має пройти навчання на основі підготовленої вище навчальної вибірки. За результатами даного етапу є навчена модель, яка далі може застосовуватися для роботи з новими екземплярами даних і фактично використовуватися в програмі для реалізації функції додавання і редагування косметичного засобу.

На останньому етапі забезпечується робота з новими косметичними засобами, кожен з яких має бути перетворений на екземпляр, який включає набір ознак, описаний вище у переліку, послідовність ознак для представлення інгредієнтів зі складу косметичного засобу та послідовність ознак для представлення назви косметичного засобу. Такий екземпляр подається на вхід навченої моделі, а на виході отримується середня оцінка від покупців для цього косметичного засобу.

### **2.3 Висновки за розділом 2**

Результати виконання розділу 2 забезпечили досягнення наукової новизни роботи, яка полягає в методі прогнозування оцінювання використання косметичних засобів покупцями, що забезпечує формування ознак, які представляють назву косметичного засобу, на основі моделі Word2Vec, ознак, які представляють склад косметичного засобу, створення моделі ансамблю дерев рішень, що дозволяє прогнозувати середню оцінку, яку отримає косметичний засіб від покупців.

### 3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Вибір мови та засобів розробки

Магазин косметики фактично влаштований за принципом організації інтернет-магазину, але не обмежений цієї функціональністю. Інтернет-магазин є типовим рішенням щодо реалізації роботи з замовленнями, а тому для реалізації може використовуватися шаблонний варіант за допомогою системи управління версіями. Зокрема це може бути Drupal (табл. 3.1) [6].

Таблиця 3.1 – Вибір вебфреймворку для розробки програми магазину косметики

|                                                                 | Django                               | Drupal                    |
|-----------------------------------------------------------------|--------------------------------------|---------------------------|
| Забезпечення високого рівня продуктивності для великих проєктів | Краще                                | Гірше                     |
| Спосіб розширення функціональності                              | Інтеграція коду до раніше створеного | Написання окремого модуля |
| Мова, якою ведеться розробка                                    | Python                               | PHP                       |
| Рівень безпеки базового рішення                                 | Вищий                                | Нижчий                    |

Однак, важливо враховувати, що функціональність програми, що розробляється, не обмежується повністю типовим варіантом інтернет-магазину. По-перше, самі сутності повинні мати цілий ряд варіантів структуризації, характеристик, а відповідно і врахування всього цього під час організації зокрема пошуку. Вже це призводить до того, що будь-яке типове рішення потрібно додатково налаштовувати, відповідно зменшує плюси його використання. Проте окрім того потрібно реалізувати роботу створеного методу прогнозування оцінювання використання косметичних засобів

покупцями, інтегрувати ці засоби до пов'язаної функціональності в програмі. А це вже значно складніше зробити в умовах необхідності реалізації мовою PHP додаткової функціональності в системі управління контентом. Тому рішення типовою мовою для реалізації таких засобів (Python) у вигляді вебфреймворку Django [7] є більш придатним у підсумку, вибір на користь якого і було зроблено.

### **3.2 Моделювання роботи з програмою за функціональними вимогами**

Покупці та працівники магазину косметики мають багато спільного у використанні відповідного програмного забезпечення, однак при цьому і ряд особливостей. При цьому набір сценаріїв працівника магазину не повторює повністю набір сценаріїв покупця в частині тих сценаріїв, що покупець може виконувати, тому що працівник магазину не має можливості реєструватися, окрім того частина можливостей стосується різних користувачів для працівника магазину та тільки власних даних для покупця.

Основний потік сценаріїв покупця магазину косметики:

- перегляд списку брендів косметики у магазині;
- перегляд інформації про бренд та його серії косметичних засобів;
- перегляд набору косметичних засобів за серією;
- пошук косметичних засобів;
- оформлення замовлення набору косметичних засобів;
- перегляд інформації про косметичний засіб;
- збереження косметичного засобу для швидкого доступу;
- перегляд панелі швидкого доступу до косметичних засобів;
- авторизація облікового запису користувача;
- коментування косметичного засобу щодо досвіду власного застосування (рис. 3.1).

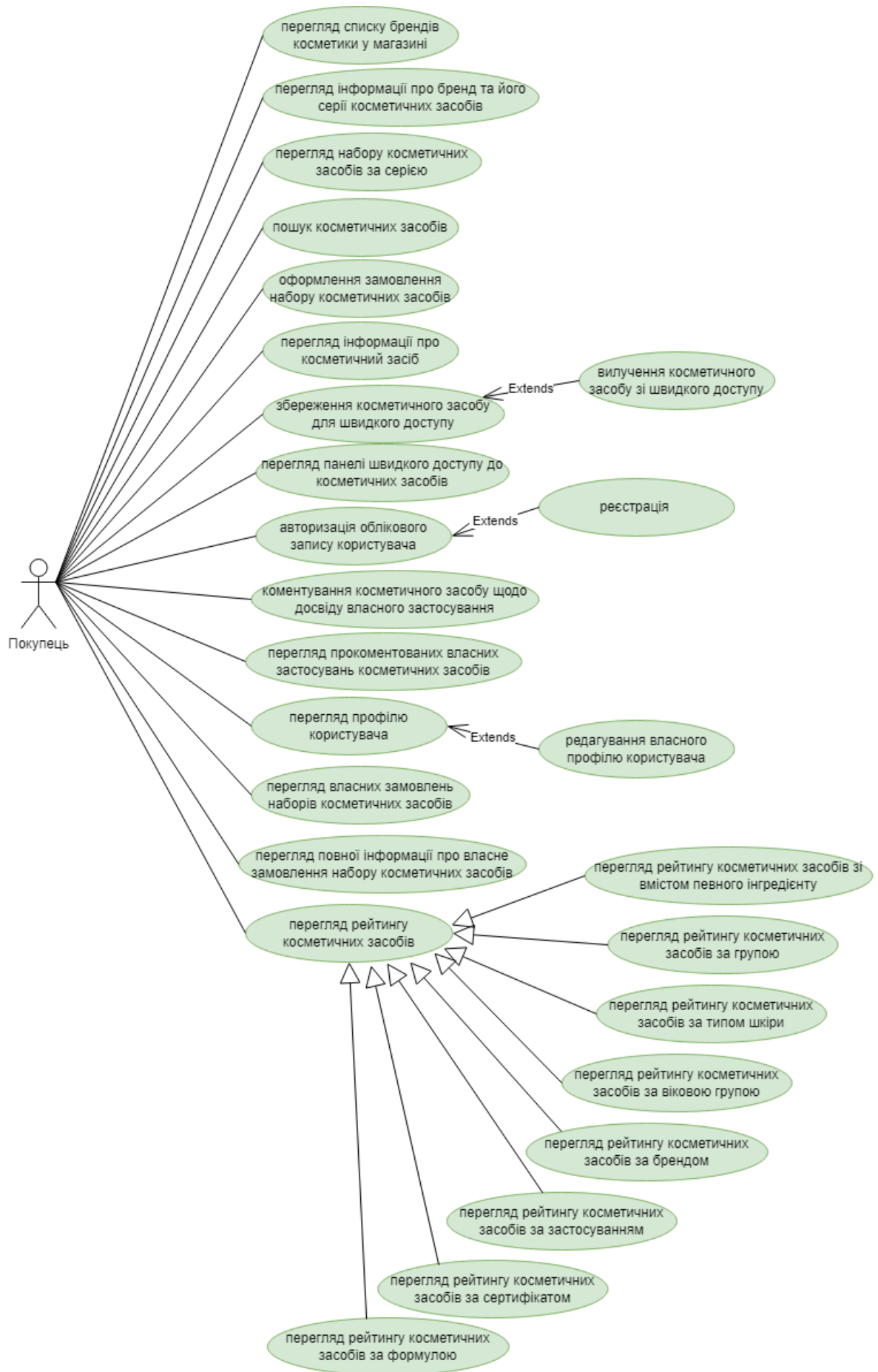


Рисунок 3.1 – Діаграма прецедентів для покупця магазину косметики

Також сценарії з основного потоку:

- перегляд прокоментованих власних застосувань косметичних засобів;
- перегляд профілю користувача;
- перегляд власних замовлень наборів косметичних засобів;
- перегляд повної інформації про власне замовлення набору косметичних засобів;
- перегляд рейтингу косметичних засобів.

Сценарій перегляду рейтингу косметичних засобів узагальнює наступні:

- перегляд рейтингу косметичних засобів зі вмістом певного інгредієнту;
- перегляд рейтингу косметичних засобів за групою;
- перегляд рейтингу косметичних засобів за типом шкіри;
- перегляд рейтингу косметичних засобів за віковою групою;
- перегляд рейтингу косметичних засобів за брендом;
- перегляд рейтингу косметичних засобів за застосуванням;
- перегляд рейтингу косметичних засобів за сертифікатом;
- перегляд рейтингу косметичних засобів за формулою.

Альтернативні сценарії:

- вилучення косметичного засобу зі швидкого доступу замість збереження косметичного засобу для швидкого доступу, коли вже на панелі швидкого доступу такий косметичний засіб є;
- редагування власного профілю користувача замість перегляду профілю користувача при потребі оновлення даних у профілі;
- реєстрація облікового запису користувача замість авторизації для забезпечення авторизації, коли спершу користувач не має зареєстрованого облікового запису.

Основний потік сценаріїв працівника магазину косметики:

- перегляд списку брендів косметики у магазині;

- додавання бренду косметики, що продається в магазині;
- перегляд інформації про бренд та його серії косметичних засобів;
- додавання нової серії косметичних засобів;
- перегляд набору косметичних засобів за серією;
- пошук косметичних засобів;
- оформлення замовлення набору косметичних засобів;
- перегляд інформації про косметичний засіб;
- додавання нового косметичного засобу, що включає сценарій прогнозування середньої оцінки косметичного засобу від користувачів;
- додавання нового доступного обсягу косметичного засобу;
- авторизація облікового запису працівника магазину косметичних засобів;
- перегляд прокоментованих користувачем застосувань косметичних засобів;
- перегляд профілю користувача;
- перегляд замовлень наборів косметичних засобів користувачем;
- перегляд всіх замовлень наборів косметичних засобів у магазині з фільтруванням;
- оновлення стану замовлення набору косметичних засобів;
- перегляд повної інформації про замовлення набору косметичних засобів;
- перегляд замовлень за косметичним засобом;
- внесення інформації про доступний обсяг косметичного засобу;
- перегляд сумарних обсягів продажу за косметичним засобом загалом або за заданим обсягом за період часу;
- перегляд рейтингу косметичних засобів зі вмістом певного інгредієнту;
- перегляд рейтингу косметичних засобів за групою;
- перегляд рейтингу косметичних засобів за типом шкіри;
- перегляд рейтингу косметичних засобів за віковою групою (рис. 3.2).

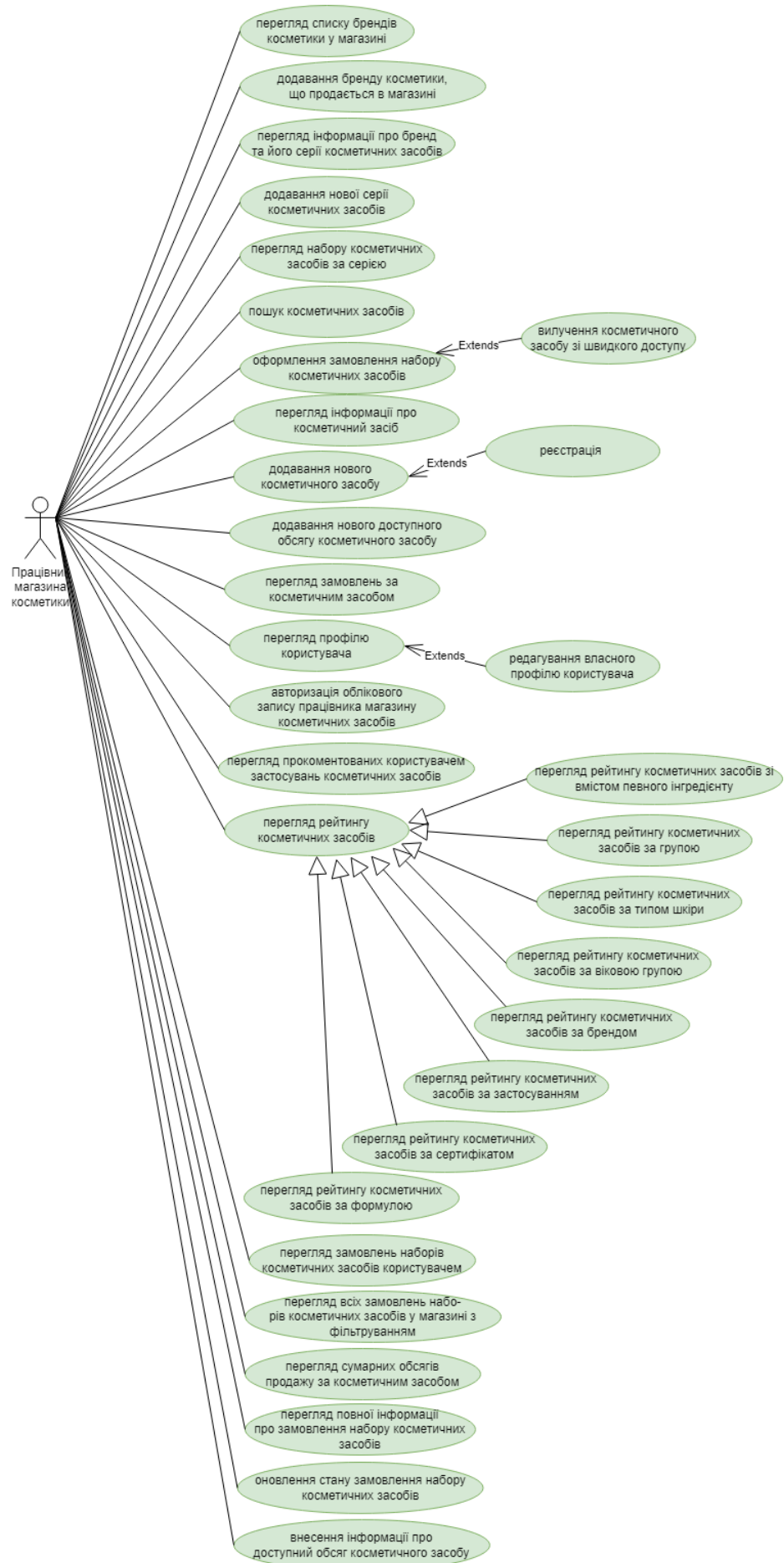


Рисунок 3.2 – Діаграма прецедентів для працівника магазину косметики

Також сценарії з основного потоку:

- перегляд рейтингу косметичних засобів за брендом;
- перегляд рейтингу косметичних засобів за застосуванням;
- перегляд рейтингу косметичних засобів за сертифікатом;
- перегляд рейтингу косметичних засобів за формулою.

Альтернативні сценарії:

- редагування даних бренду косметики, що продається в магазині, замість додавання бренду косметики, що продається в магазині, коли бренд вже додано, але дані змінилися;
- редагування інформації про косметичний засіб замість додавання нового косметичного засобу, коли косметичний засіб вже додано, але дані змінилися;
- редагування даних серії косметичних засобів замість додавання нової серії косметичних засобів, коли серію вже додано, але дані змінилися;
- редагування власного профілю користувача замість перегляду профілю користувача при потребі оновлення даних у профілі.

### **3.3 Проектування структури бази даних**

Таблиця `Cosmeticgroup` повинна містити дані стандартних груп косметичних засобів, включаючи для опису такі поля:

- `id` – номер ідентифікації групи косметичних засобів;
- `cosmeticgroupname` – назва групи косметичних засобів.

Таблиця `Cosmeticsgoodgroup` повинна містити дані відношень косметичних засобів до відповідних стандартних груп, включаючи для опису такі поля:

- `id` – номер ідентифікації відношення косметичного засобу до відповідної стандартної групи косметичних засобів;
- `cosmeticgroup` – стандартна група, до якої належить цей косметичний засіб;

– cosmeticsgood – косметичний засіб, який представляє цю стандартну групу косметичних засобів.

Таблиця Cosmeticsgoodgroup є реалізацією зв'язку багато-до-багатьох, що виникає між таблицями Cosmeticsgood і Cosmeticgroup, забезпечуючи множинність включення косметичних засобів до різних груп косметичних засобів та включаючи зовнішні ключі з таблиць Cosmeticgroup і Cosmeticsgood через однойменні поля.

Таблиця Agegroup повинна містити дані стандартних вікових груп, включаючи для опису такі поля:

- id – номер ідентифікації стандартної вікової групи;
- agefrom – числове значення мінімального віку, з якого розпочинається ця стандартна вікова група;
- agegroupname – текстова назва цієї стандартної вікової групи.

Таблиця Cosmeticsbrand повинна містити дані брендів косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації бренду косметичного засобу;
- cosbrandname – назва бренду косметичного засобу;
- brandcountry – країна, походженням з якої є цей бренд косметичного засобу, що представляє зв'язок з таблицею Country.

Таблиця Cosmeticserie повинна містити дані серій косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації серії косметичних засобів;
- cosmeticsbrand – бренд косметичних засобів, який випускає цю серію, що представляє зв'язок з таблицею Cosmeticsbrand;
- seriesname – назва серії косметичних засобів.

Таблиця Application повинна містити дані можливих призначень косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації стандартного призначення косметичних засобів;

– applicationname – назва стандартного призначення косметичних засобів.

Таблиця CosmeticsgoodApplication повинна містити дані відношень косметичних засобів до призначень, яким вони відповідають, включаючи для опису такі поля:

- id – номер ідентифікації відношення косметичного засобу до відповідного стандартного призначення косметичних засобів;
- application – призначення, за якого варто використовувати цей косметичний засіб, що представляє зв'язок з таблицею Application;
- cosmeticsgood – косметичний засіб, який має це призначення для застосування, що представляє зв'язок з таблицею Cosmeticsgood.

Таблиця CosmeticsgoodApplication є реалізацією зв'язку багато-до-багатьох, що виникає між таблицями Cosmeticsgood і Application, забезпечуючи множинність включення косметичних засобів до різних призначень косметичних засобів та включаючи зовнішні ключі з таблиць Application і Cosmeticsgood через однойменні поля.

Таблиця Cosmeticsgood повинна містити дані косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації косметичного засобу;
- cosmeticsserie – серія косметичних засобів, якій відповідає цей косметичний засіб, що представляє зв'язок з таблицею Cosmeticsserie;
- cosmeticcaption – повна назва косметичного засобу;
- agegroup – вікова група, за якої рекомендовано використовувати цей косметичний засіб;
- description – опис косметичного засобу;
- timeperiod – час застосування косметичного засобу;
- gender – стать, для якої виготовлено цей косметичний засіб (чоловіки, жінки або універсальне застосування);
- marketclass – маркетинговий клас косметичного засобу (наприклад, масмаркет);

– mark – середній бал, отриманий за коментарями покупців стосовно використання даного косметичного засобу;

– predictedmark – прогнозований середній бал косметичного засобу, отриманий за допомогою моделі прогнозування за запропонованим у роботі методом;

– applicationway – спосіб застосування косметичного засобу (текстовий опис).

Таблиця Cosmeticsgoodskin повинна містити дані відношень типів шкіри до косметичних засобів, включаючи для опису такі поля:

– id – номер ідентифікації відношення типу шкіри до косметичного засобу;

– cosmeticsgood – косметичний засіб, який може використовуватись для заданого типу шкіри, що представляє зв'язок з таблицею Cosmeticsgood;

– skintype – тип шкіри, за якого можливе використання цього косметичного засобу.

Таблиця Certificate повинна містити дані сертифікатів, які можуть видаватися косметичним засобам, включаючи для опису такі поля:

– id – номер ідентифікації сертифіката;

– certname – назва сертифіката, що може видаватися косметичному засобу;

– certdetails – опис особливостей даного сертифіката, що може видаватися косметичному засобу.

Таблиця CosmeticsgoodCertificate повинна містити дані відношень сертифікатів до конкретних косметичних засобів, на які ці сертифікати видано, включаючи для опису такі поля:

– id – номер ідентифікації відношення сертифікату до косметичного засобу;

– cosmeticsgood – косметичний засіб, на який видано цей сертифікат, що представляє зв'язок з таблицею Cosmeticsgood;

– certificate – сертифікат, який видано на цей косметичний засіб, що представляє зв'язок з таблицею Certificate.

Таблиця CosmeticsgoodCertificate є реалізацією зв'язку багато-до-багатьох, що виникає між таблицями Cosmeticsgood і Certificate, забезпечуючи множинність включення косметичних засобів до різних сертифікатів та включаючи зовнішні ключі з таблиць Cosmeticsgood і Certificate через однойменні поля.

Таблиця CosmeticsgoodFormula повинна містити дані формул косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації формули конкретного косметичного засобу;
- cosmeticsgood – косметичний засіб, який виконано за цією формулою, що представляє зв'язок з таблицею Cosmeticsgood;
- formula – формула, за якою виконано цей косметичний засіб, що представляє зв'язок з таблицею Formula.

Таблиця CosmeticsgoodFormula є реалізацією зв'язку багато-до-багатьох, що виникає між таблицями Cosmeticsgood і Formula, забезпечуючи множинність включення косметичних засобів до різних формул та включаючи зовнішні ключі з таблиць Cosmeticsgood і Formula через однойменні поля.

Таблиця Formula повинна містити дані стандартних формул засобів, включаючи для опису такі поля:

- id – номер ідентифікації стандартної формули засобу;
- formulaname – назва стандартної формули засобу.

Таблиця CosmeticsgoodIngredient повинна містити дані складу косметичного засобу на основі визначення інгредієнтів косметичного засобу, включаючи для опису такі поля:

- id – номер ідентифікації інгредієнту косметичного засобу;
- cosmeticsgood – косметичний засіб, до якого входить цей інгредієнт, що представляє зв'язок з таблицею Cosmeticsgood;
- ingredient – інгредієнт, який входить до складу цього косметичного засобу, що представляє зв'язок з таблицею Ingredient.

Таблиця Ingredient повинна містити дані стандартних інгредієнтів, включаючи для опису такі поля:

- id – номер ідентифікації інгредієнта;
- ignedname – назва інгредієнта.

Таблиця CosmeticsgoodPhoto повинна містити дані фотографій косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації фотографії косметичного засобу;
- cosmeticsgood – косметичний засіб, показаний на цій фотографії, що представляє зв'язок з таблицею Cosmeticsgood;
- photodata – дані фотографії косметичного засобу;
- topphoto – основна фотографія косметичного засобу або ні.

Таблиця CosmeticsgoodVolume повинна містити дані доступних об'ємів косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації об'єму косметичного засобу в продажу;
- cosmeticsgood – косметичний засіб, об'єм якого продається, представляє зв'язок з таблицею Cosmeticsgood;
- volume – обсяг косметичного засобу у мілілітрах;
- price – вартість, за якою зазвичай продається цей косметичний засіб;
- possdiscount – можливий максимальний дисконт у відсотках, який може бути застосований при продажу цього обсягу цього косметичного засобу;
- pricetopay – вартість, яку на поточний момент потрібно сплатити за цей обсяг косметичного засобу (акційна вартість);
- producedcountry – країна, в якій виготовлено цей варіант обсягу цього косметичного засобу, представляє зв'язок з таблицею Country;
- amount – поточна доступна кількість косметичного засобу в цьому обсязі.

Таблиця Country повинна містити дані країн, включаючи для опису такі поля:

- id – номер ідентифікації країни;

- name – назва країни.

Таблиця SoftAccount повинна містити дані облікових записів користувачів програми, включаючи для опису такі поля:

- id – номер ідентифікації облікового запису користувача;
- user – обліковий запис користувача Django, що використовується в основі, представляє зв'язок з таблицею User;
- phonenumber – номер телефону користувача;
- discount – рівень дисконту, який встановлений для користувача у відсотках;
- photo – фотографія користувача.

Таблиця CosmeticsgoodComment повинна містити дані коментарів про застосування косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації коментаря про застосування косметичного засобу;
- cosmeticsgood – косметичний засіб, до якого стосується коментар, представляє зв'язок з таблицею Cosmeticsgood;
- commentfull – сам текст коментаря з описом спостережень та висновків користувача щодо застосування цього косметичного засобу;
- commentersoftaccount – користувач, який залишив цей коментар про застосування косметичного засобу, представляє зв'язок з таблицею SoftAccount;
- commentedwhen – часова мітка, коли було створено коментар користувачем про застосування цього косметичного засобу;
- skintype – тип шкіри, який має цей користувач або за якого застосовувався цей косметичний засіб;
- mark – оцінка, яку виставлено користувачем цьому косметичному засобу після його застосування.

Таблиця CosmeticsgoodSave повинна містити дані збережень косметичних засобів, включаючи для опису такі поля:

– id – номер ідентифікації збереження косметичного засобу користувачем;

– saversoftaccount – користувач, яким збережено цей косметичний засіб, представляє зв'язок з таблицею SoftAccount;

– cosmeticsgood – косметичний засіб, який зберіг цей користувач, представляє зв'язок з таблицею Cosmeticsgood.

Таблиця SoftAccountOrder повинна містити дані замовлень наборів косметичних засобів, включаючи для опису такі поля:

– id – номер ідентифікації замовлення набору косметичних засобів;

– softaccount – користувач, який замовив цей набір косметичних засобів, представляє зв'язок з таблицею SoftAccount;

– price – вартість, що сумарно відповідає цьому замовленню набору косметичних засобів;

– discount – дисконт, який було застосовано до цього замовлення набору косметичних засобів;

– shipping – необхідність виконання доставки або потреба в тому, щоб забрати замовлення самостійно;

– status – поточний стан замовлення набору косметичних засобів;

– address – адреса доставки набору косметичних засобів за замовленням, якщо реалізується доставка;

– creation – часова мітка, коли було створено це замовлення набору косметичних засобів.

Таблиця SoftAccountOrderContent повинна містити дані вмісту набору косметичних засобів з замовлення, включаючи для опису такі поля:

– id – номер ідентифікації вмісту набору косметичних засобів у замовленні;

– softaccountorder – замовлення набору косметичних засобів, до якого відноситься цей вміст, представляє зв'язок з таблицею SoftAccountOrder;

– cosmeticsgoodvolume – обсяг косметичного засобу, який входить до складу замовлення, представляє зв'язок з таблицею CosmeticsgoodVolume;

- quantity – кількість цього косметичного засобу в замовленні;
- price – вартість одиниці цього косметичного засобу (рис. 3.3).

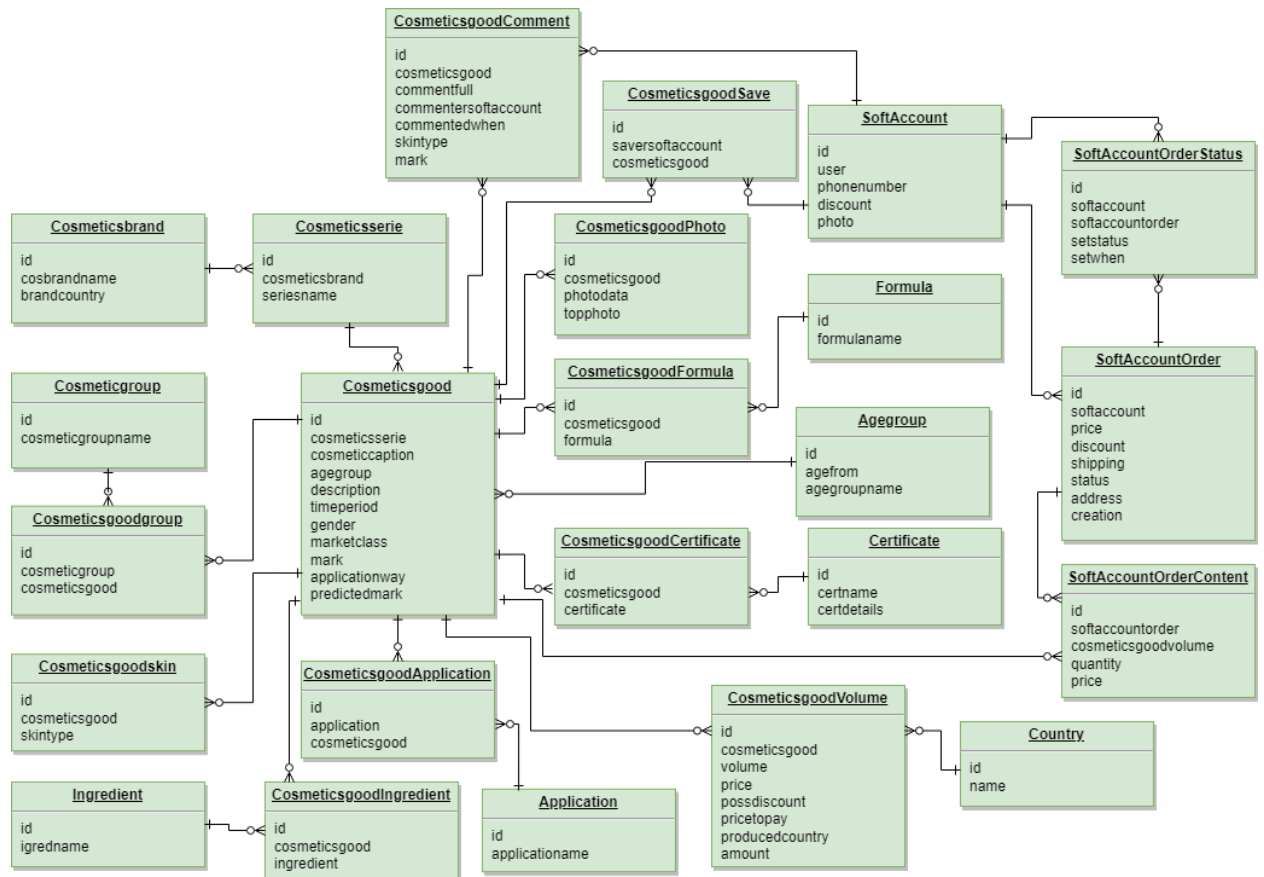


Рисунок 3.3 – Схема бази даних

Таблиця SoftAccountOrderStatus повинна містити дані виконаних змін станів замовлень наборів косметичних засобів, включаючи для опису такі поля:

- id – номер ідентифікації зміни стану замовлення набору косметичних засобів;
- softaccount – користувач, який вніс цю зміну до стану замовлення набору косметичних засобів, представляє зв'язок з таблицею SoftAccount;
- softaccountorder – замовлення набору косметичних засобів, стан якого було змінено, представляє зв'язок з таблицею SoftAccountOrder;
- setstatus – встановлений стан замовлення набору косметичних засобів, який було встановлено;

– setwhen – часова мітка виконання зміни стану замовлення набору косметичних засобів.

### **3.4 Висновки за розділом 3**

Результати виконання розділу 3 включають порівняння та подальший виконаний вибір Django та Python як вебфреймворку та основної мови програмування для розробки, проєктування структури бази даних на основі системи PostgreSQL, визначення функціональних вимог до програми з формуванням сценаріїв роботи всіх груп користувачів.

# 4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 4.1 Порядок взаємодії користувача з програмою магазину продажу косметики

Схема функціонування програми магазину продажу косметики (рис. 4.1) побудована на основі порядку взаємодії користувача з програмою.

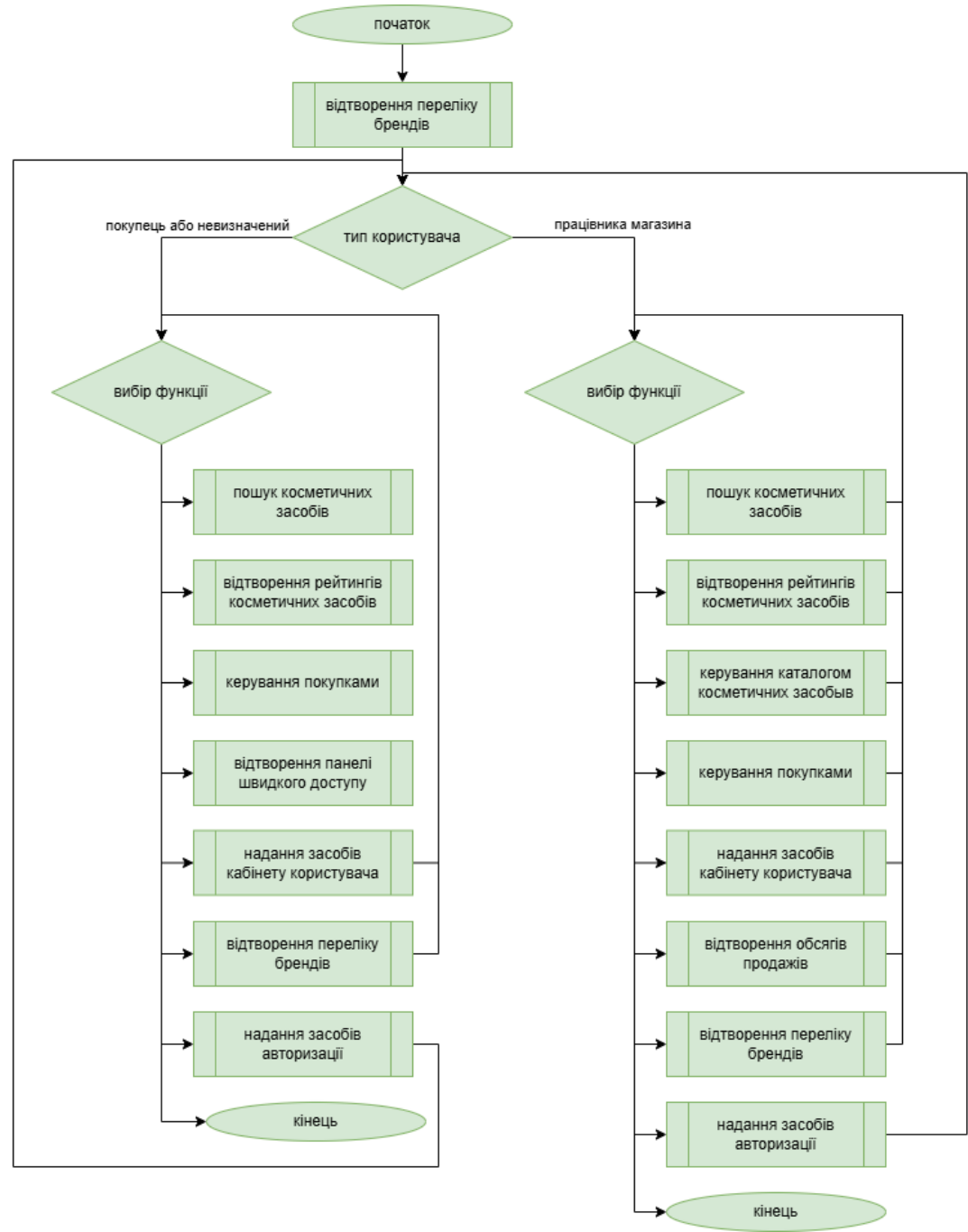


Рисунок 4.1 – Схема функціонування програми магазину продажу косметики

Робота розпочинається з відтворення переліку брендів косметики, при цьому вона включає в себе певний процес, коли далі можна обрати бренд, переглянути за ними дані, його серії косметичних засобів, продовжуючи цей процес. Спочатку користувач має невизначений тип, а засоби, які він може використовувати, доступні через меню. Далі надання засобів авторизації може призвести до того, що він після авторизації (або спочатку реєстрації, якщо обліковий запис відсутній) стане покупцем або працівником магазину, після чого йому буде доступне своє меню, а також можливість повторного звернення до засобів авторизації, після чого тип може змінитись (користувач стане неавторизованим, відповідно тим буде невизначеним, або він стане працівником магазину, або покупцем).

#### **4.2 Рішення з реалізації вебзастосунку**

Безпосередньо реалізація методу прогнозування оцінювання використання косметичних засобів покупцями потребувала цілого ряду рішень, частина з яких виділена в окремий скрипт, який запускався для створення та навчання моделі, а також підтримував процес експериментального дослідження, а інша частина пов'язана з використанням вже готової моделі та інтегрована до складу представлень, пов'язаних з додаванням та редагуванням косметичних засобів.

Для роботи з початковими даними використано пакет Pandas, що дозволив створити табличне представлення і далі працювати з ним. До цього табличного представлення було внесено розширення з додавання ознак. Для цього зокрема з пакету Gensim було використано клас Word2Vec для реалізації відповідної моделі на основі 50 елементів, що дозволило перетворити назву косметичного засобу на числову послідовність. Реалізацію самої базової моделі виконано на основі класу RandomForestRegressor з пакету scikit-learn з застосуванням 500 дерев (параметр `n_estimators`) з викликом методів `fit` для навчання та `predict` для прогнозування за навченою моделлю.

Перегляд списку брендів косметики у магазині реалізовано через представлення, виражене функцією `cosmetics_brands_view`. Перелік усіх брендів впорядковується за алфавітом та передається в шаблон сторінки `comsbrands.html`, створений як і всі інші шаблони вебсторінок на основі безкоштовного шаблону [8].

Перегляд інформації про бренд та його серії косметичних засобів реалізовано через представлення, виражене функцією `cosmetic_brand_view`. Додатковий параметр `cosmeticsbrand_id` задає номер бренду. Словник з даними самого бренду та всіх серій косметики, випущених цим брендом, передається в шаблон сторінки `cosmeticsbrand.html`.

Перегляд набору косметичних засобів за серією реалізовано через представлення, виражене функцією `cosmetic_series_view`. Додаткові параметри `cosmeticserie_id` і `cosmetics_order` задають відповідно номер серії косметичних засобів і порядок сортування косметичних засобів. Словник з даними самої серії косметичних засобів та переліком самих косметичних засобів передається в шаблон сторінки `cosmeticsserie.html`.

Пошук косметичних засобів реалізовано через представлення, виражене функцією `cosmetics_search_view`. Параметри формуються через GET-запит, виділяючи зокрема групи косметичних засобів, до яких мають входити результуючі засоби (`groups`), вікові групи (`age`), бренди (`brands`), серії косметичних засобів (`serie`), варіанти застосування (`application`), назву косметичного засобу (`costext`), стать (`gender`), ринкові групи (`markets`), мінімальну середню оцінку (`minmark`), максимальну середню оцінку (`maxmark`), потрібні сертифікати (`certificate`), формули засобу (`formula`), інгредієнти, які мають входити до складу (`ings`), та які не мають входити до складу (`notings`), порядок сортування (`order`), необхідність наявності на складі цих косметичних засобів (`available`), типи шкіри (`skins`), країни-виробники (`countries`). Перелік з результатами пошуку косметичних засобів, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `search.html`.

Оформлення замовлення набору косметичних засобів реалізовано через представлення, виражене функцією `cosmetics_order_create_view`, що забезпечується на основі форми `SoftAccountOrderForm` зі збереженням даних у базу даних.

Перегляд інформації про косметичний засіб реалізовано через представлення, виражене функцією `cosmetics_view`. Додатковий параметр `cosmetics_id` задає номер косметичного засобу. Словник з даними самого косметичного засобу, включаючи фотографії, типи шкіри, за яких можливе застосування цього косметичного засобу, наявні в продажу обсяги, формули цього засобу, склад з переліком інгредієнтів, перелік сертифікатів, застосувань, груп косметичних засобів, до яких цей засіб належить, вікову групу, серію косметичних засобів, передається в шаблон сторінки `cosmetics.html`.

Збереження або вилучення косметичного засобу для або з швидкого доступу (в залежності від попереднього внесення або невнесення туди косметичного засобу) реалізовано через представлення, виражене функцією `cosmetics_good_save_view`. Додатковий параметр `cosmetics_id` задає номер косметичного засобу.

Перегляд панелі швидкого доступу до косметичних засобів реалізовано через представлення, виражене функцією `cosmetics_good_saves_view`. Перелік з даними панелі швидкого доступу, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `saves.html`.

Коментування косметичного засобу щодо досвіду власного застосування реалізовано через представлення, виражене функцією `cosmeticsgood_comment_add_view`. Додатковий параметр `cosmeticsgood` задає номер косметичного засобу. Реалізовується перевірка, що раніше цей косметичний засіб замовлявся цим користувачем. В основі роботи лежить форма `CosmeticsgoodCommentForm`.

Перегляд прокоментованих власних застосувань косметичних засобів та прокоментованих заданим користувачем реалізовано через представлення, виражене функцією `commented_cosmetics_view`. Додатковий параметр `user_login` з логіном користувача може бути не встановлений, щоб використовувати власний логін, якщо користувач є звичайним покупцем, або має бути заданий для працівника магазину. Перелік з коментарями до використання косметичних засобів, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `comments.html`.

Перегляд профілю користувача реалізовано через представлення, виражене функцією `soft_account_view`. Додатковий параметр `user_login` з логіном користувача може бути не встановлений, щоб використовувати власний логін. Дані профілю разом з останніми коментарями передаються в шаблон сторінки `account.html`.

Перегляд власних замовлень наборів косметичних засобів реалізовано через представлення, виражене функцією `personal_cosmetics_orders_view`. Порядок сортування задається параметром `order` GET-запиту. Перелік замовлень передається в шаблон сторінки `persorders.html`.

Перегляд повної інформації про власне замовлення набору косметичних засобів реалізовано через представлення, виражене функцією `cosmetics_order_view`. Додатковий параметр `order_id` задає номер замовлення. Замовлення цього набору косметичних засобів має бути створено самим цим авторизованим користувачем, інакше дані не будуть повернуті. Дані самого замовлення, його склад та стани передаються в шаблон сторінки `order.html`.

Перегляд рейтингу косметичних засобів зі вмістом певного інгредієнту реалізовано через представлення, виражене функцією `ingredient_cosmetics_rating_view`. Додатковий параметр `ingredient_id` задає номер інгредієнта. Перелік 100 найкращих за середньою оцінкою косметичних засобів зі вмістом цього інгредієнту, розподілений на порції по 10, номер якої

задається параметром `part` GET-запиту, передається в шаблон сторінки `ingrating.html`.

Перегляд рейтингу косметичних засобів за групою реалізовано через представлення, виражене функцією `group_cosmetics_rating_view`. Додатковий параметр `cosmeticgroup_id` задає номер групи косметичних засобів. Перелік 100 найкращих за середньою оцінкою косметичних засобів, що відносяться до цієї групи, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `grrating.html`.

Перегляд рейтингу косметичних засобів за типом шкіри реалізовано через представлення, виражене функцією `skin_cosmetics_rating_view`. Додатковий параметр `skintype` задає тип шкіри. Перелік 100 найкращих за середньою оцінкою косметичних засобів, що придатні для цього типу шкіри, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `skinrating.html`.

Перегляд рейтингу косметичних засобів за віковою групою реалізовано через представлення, виражене функцією `age_cosmetics_rating_view`. Додатковий параметр `agegroup_id` задає номер вікової групи. Перелік 100 найкращих за середньою оцінкою косметичних засобів, що можуть застосовуватися для цієї вікової групи, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `agerating.html`.

Перегляд рейтингу косметичних засобів за брендом реалізовано через представлення, виражене функцією `brand_cosmetics_rating_view`. Додатковий параметр `cosmeticsbrand_id` задає номер бренда косметичних засобів. Перелік 100 найкращих за середньою оцінкою косметичних засобів цього бренда, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `brandrating.html`.

Перегляд рейтингу косметичних засобів за застосуванням реалізовано через представлення, виражене функцією `application_cosmetics_rating_view`. Додатковий параметр `application_id` задає номер застосування. Перелік 100

найкращих за середньою оцінкою косметичних засобів, що можуть використовуватися для заданого застосування, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `apprating.html`.

Перегляд рейтингу косметичних засобів за сертифікатом реалізовано через представлення, виражене функцією `certificate_cosmetics_rating_view`. Додатковий параметр `certificate_id` задає номер сертифіката, який може видаватися косметичним засобам. Перелік 100 найкращих за середньою оцінкою косметичних засобів, що мають цей сертифікат, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `certifrating.html`.

Перегляд рейтингу косметичних засобів за формулою реалізовано через представлення, виражене функцією `formula_cosmetics_rating_view`. Додатковий параметр `formula_id` задає номер формули косметичних засобів. Перелік 100 найкращих за середньою оцінкою косметичних засобів, що мають цю формулу, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `formularating.html`.

Перегляд замовлень наборів косметичних засобів користувачем реалізовано через представлення, виражене функцією `shop_user_orders_view`. Додатковий параметр `user_login` задає логін облікового засобу. Параметри GET-запиту `dp1` та `dp2` задають дати і час початку і завершення періоду, коли мали бути створені замовлення, `stat` – потрібні стани замовлень, `order` – порядок сортування замовлень. Перелік замовлень наборів косметичних засобів, розподілений на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `usorders.html`.

Перегляд всіх замовлень наборів косметичних засобів у магазині реалізовано через представлення, виражене функцією `shop_cosmetics_orders_view`. Параметри GET-запиту `dp1` та `dp2` задають дати і час початку і завершення періоду, коли мали бути створені замовлення, `stat` – потрібні стани замовлень, `order` – порядок сортування замовлень. Перелік

замовлень наборів косметичних засобів, розподілених на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `allorders.html`.

Перегляд замовлень за косметичним засобом реалізовано через представлення, виражене функцією `shop_cosmetic_orders_view`. Додатковий параметр `cosmetics_id` задає номер косметичного засобу. Параметри GET-запиту `dp1` та `dp2` задають дати і час початку і завершення періоду, коли мали бути створені замовлення, `stat` – потрібні стани замовлень, `order` – порядок сортування замовлень. Перелік замовлень наборів косметичних засобів, розподілених на порції по 10, номер якої задається параметром `part` GET-запиту, передається в шаблон сторінки `cosorders.html`.

Додавання бренду косметики, що продається в магазині, виражене функцією `cosmetics_brand_add_view` на основі форми `CosmeticsbrandForm`. За тією ж формою організовано редагування даних бренду косметики за функцією `cosmetics_brand_edit_view` для бренду з номером `cosmetics_brand_id` у додатковому параметрі функції.

Додавання нової серії косметичних засобів, виражене функцією `cosmetics_series_add_view` на основі форми `CosmeticsserieForm`. За тією ж формою організовано редагування даних серії косметичних засобів за функцією `cosmetics_series_edit_view` для серії з номером `cosmetics_series_id` у додатковому параметрі функції.

Додавання нового косметичного засобу, що включає сценарій прогнозування середньої оцінки косметичного засобу від користувачів, виражене функцією `cosmetics_good_add_view` на основі форми `CosmeticsgoodForm`. За тією ж формою організовано редагування інформації про косметичний засіб за функцією `cosmetics_good_edit_view` для косметичного засобу з номером `cosmetics_id` у додатковому параметрі функції.

Додавання нового доступного обсягу косметичного засобу виражене функцією `cosmetics_good_volume_add_view` на основі форми `CosmeticsgoodVolumeForm`. За тією ж формою організовано редагування

інформації про обсяг косметичного засобу за функцією `cosmetics_good_volume_edit_view` для обсягу косметичного засобу з номером `cosmeticsvol_id` у додатковому параметрі функції.

Внесення інформації про доступний обсяг косметичного засобу виражене функцією `cosmetics_good_volume_change_view`. Додатковий параметр `cosmeticsvol_id` задає номер обсягу косметичного засобу, `amount` – нову кількість засобу.

Оновлення стану замовлення набору косметичних засобів, виражене функцією `cosmetics_order_change_status_view`. Додатковий параметр `cosmeticsorder_id` задає номер замовлення набору косметичних засобів, `status` – новий стан. Внесення змін у базу даних виконується тільки, якщо авторизований користувач є працівником магазину.

Перегляд сумарних обсягів продажу за косметичним засобом загалом або за заданим обсягом за період часу виражений функцією `cosmetics_sale_view`. Додаткові параметри `dp1` і `dp2` задають період часу, `cosmetics_id` – номер косметичного засобу (якщо пропущений, то за всіма наявними косметичними засобами). Розподіл відбувається за місцями за вказаний період і передається в шаблон сторінки `sale.html`.

Клас `Cosmeticgroup` реалізує модель стандартних груп косметичних засобів з полями:

- `cosmeticgroupname` типу `CharField` для завдання назви групи косметичних засобів;
- `cosmeticsgoods` типу `ManyToManyField` для реалізації відношення багато-до-багатьох з визначенням косметичних засобів, які входять до цієї групи косметичних засобів.

Клас `Agegroup` реалізує модель стандартних вікових груп з такими атрибутами.

Атрибут `agefrom` типу `PositiveSmallIntegerField` визначає числове значення мінімального віку, з якого розпочинається ця стандартна вікова група.

Атрибут `agegroupname` типу `CharField` визначає текстову назву цієї стандартної вікової групи.

Клас `Cosmeticsbrand` реалізує модель брендів косметичних засобів з такими атрибутами.

Атрибут `cosbrandname` типу `CharField` визначає назву бренду косметичного засобу.

Атрибут `brandcountry` типу `ForeignKey` визначає країну, походженням з якої є цей бренд косметичного засобу, що представляє зв'язок з моделлю `Country`.

Клас `Cosmeticsserie` реалізує модель серій косметичних засобів з такими атрибутами.

Атрибут `cosmeticsbrand` типу `ForeignKey` визначає бренд косметичних засобів, який випускає цю серію, що представляє зв'язок з моделлю `Cosmeticsbrand`.

Атрибут `seriesname` типу `CharField` визначає назву серії косметичних засобів.

Клас `Application` реалізує модель можливих призначень косметичних засобів з атрибутами:

- `applicationname` типу `CharField`, що визначає назва стандартного призначення косметичних засобів.

- `cosmeticsgoods` типу `ManyToManyField` для реалізації відношення багато-до-багатьох з визначенням косметичних засобів, для яких визначено це призначення.

Клас `Cosmeticsgood` реалізує модель косметичних засобів з такими атрибутами.

Атрибут `cosmeticsserie` типу `ForeignKey` визначає серію косметичних засобів, якій відповідає цей косметичний засіб, що представляє зв'язок з моделлю `Cosmeticsserie`.

Атрибут `cosmeticcaption` типу `CharField` визначає повну назву косметичного засобу.

Атрибут `agegroup` типу `ForeignKey` визначає вікову групу, за якої рекомендовано використовувати цей косметичний засіб, що представляє зв'язок з моделлю `Agegroup`.

Атрибут `description` типу `CharField` визначає опис косметичного засобу.

Атрибут `timeperiod` типу `PositiveSmallIntegerField` визначає час застосування косметичного засобу.

Атрибут `gender` типу `PositiveSmallIntegerField` визначає стать, для якої виготовлено цей косметичний засіб.

Атрибут `marketclass` типу `PositiveSmallIntegerField` визначає маркетинговий клас косметичного засобу.

Атрибут `mark` типу `DecimalField` визначає середній бал, отриманий за коментарями покупців стосовно використання даного косметичного засобу.

Атрибут `applicationway` типу `CharField` визначає спосіб застосування косметичного засобу.

Атрибут `ingredients` типу `ManyToManyField` визначає реалізацію відношення багато-до-багатьох з визначенням інгредієнтів, які входять до складу цього косметичного засобу.

Атрибут `predictedmark` типу `DecimalField` визначає прогнозований середній бал косметичного засобу, отриманий за допомогою моделі прогнозування за запропонованим у роботі методом.

Клас `Cosmeticsgoodskin` реалізує модель відношень типів шкіри до косметичних засобів з такими атрибутами.

Атрибут `cosmeticsgood` типу `ForeignKey` визначає косметичний засіб, який може використовуватись для заданого типу шкіри, що представляє зв'язок з моделлю `Cosmeticsgood`.

Атрибут `skintype` типу `PositiveSmallIntegerField` визначає тип шкіри, за якого можливе використання цього косметичного засобу.

Клас `Certificate` реалізує модель сертифікатів з такими атрибутами.

Атрибут `certname` типу `CharField` визначає назву сертифіката, що може видаватися косметичному засобу.

Атрибут `certdetails` типу `CharField` визначає опис особливостей даного сертифіката, що може видаватися косметичному засобу.

Атрибут `cosmeticsgoods` типу `ManyToManyField` визначає реалізацію відношення багато-до-багатьох з визначенням косметичних засобів, для яких отримано цей сертифікат.

Клас `Formula` реалізує модель стандартних формул засобів з такими атрибутами:

- `formulaname` типу `CharField` визначає назву стандартної формули засобу;

- `cosmeticsgoods` типу `ManyToManyField` визначає реалізацію відношення багато-до-багатьох з визначенням косметичних засобів, для яких реалізовано цю формулу засобу.

Клас `Ingredient` реалізує модель стандартних інгредієнтів з атрибутом `igredname` типу `CharField`, що визначає назву інгредієнта.

Клас `CosmeticsgoodPhoto` реалізує модель фотографій косметичних засобів з такими атрибутами.

Атрибут `cosmeticsgood` типу `ForeignKey` визначає косметичний засіб, показаний на цій фотографії, що представляє зв'язок з моделлю `Cosmeticsgood`.

Атрибут `photodata` типу `ImageField` визначає дані фотографії косметичного засобу.

Атрибут `topphoto` типу `BooleanField` визначає встановлення таким чином цієї фотографії як основної фотографії косметичного засобу.

Клас `CosmeticsgoodVolume` реалізує модель доступних об'ємів косметичних засобів з такими атрибутами.

Атрибут `cosmeticsgood` типу `ForeignKey` визначає косметичний засіб, об'єм якого продається, представляє зв'язок з моделлю `Cosmeticsgood`.

Атрибут `volume` типу `PositiveIntegerField` визначає обсяг косметичного засобу у мілілітрах.

Атрибут `price` типу `DecimalField` визначає вартість, за якою зазвичай продається цей косметичний засіб.

Атрибут `possdiscout` типу `DecimalField` визначає можливий максимальний дисконт у відсотках, який може бути застосований при продажу цього обсягу цього косметичного засобу.

Атрибут `pricetoday` типу `DecimalField` визначає вартість, яку на поточний момент потрібно сплатити за цей обсяг косметичного засобу (акційна вартість).

Атрибут `producedcountry` типу `ForeignKey` визначає країну, в якій виготовлено цей варіант обсягу цього косметичного засобу, представляє зв'язок з моделлю `Country`.

Атрибут `amount` типу `PositiveIntegerField` визначає поточну доступну кількість косметичного засобу в цьому обсязі.

Клас `Country` реалізує модель країн з атрибутом `name` типу `CharField`, що визначає назву країни.

Клас `SoftAccount` реалізує модель облікових записів користувачів програми з такими атрибутами.

Атрибут `user` типу `ForeignKey` визначає обліковий запис користувача Django, що використовується в основі, представляє зв'язок з моделлю `User`.

Атрибут `phonenumber` типу `CharField` визначає номер телефону користувача.

Атрибут `discount` типу `DecimalField` визначає рівень дисконту, який встановлений для користувача у відсотках.

Атрибут `photo` типу `ImageField` визначає фотографія користувача.

Атрибут `cosmeticsgoods` типу `ManyToManyField` визначає реалізацію відношення багато-до-багатьох з визначенням косметичних засобів, які зберіг для себе користувач.

Клас `CosmeticsgoodComment` реалізує модель коментарів про застосування косметичних засобів з такими атрибутами.

Атрибут `cosmeticsgood` типу `ForeignKey` визначає косметичний засіб, до якого стосується коментар, представляє зв'язок з моделлю `Cosmeticsgood`.

Атрибут `commentfull` типу `CharField` визначає сам текст коментаря з описом спостережень та висновків користувача щодо застосування цього косметичного засобу.

Атрибут `commentersoftaccount` типу `ForeignKey` визначає користувача, який залишив цей коментар про застосування косметичного засобу, представляє зв'язок з моделлю `SoftAccount`.

Атрибут `commentedwhen` типу `DateTimeField` визначає часову мітку, коли було створено коментар користувачем про застосування цього косметичного засобу.

Атрибут `skintype` типу `PositiveSmallIntegerField` визначає тип шкіри, який має цей користувач або за якого застосовувався цей косметичний засіб.

Атрибут `mark` типу `PositiveSmallIntegerField` визначає оцінку, яку виставлено користувачем цьому косметичному засобу після його застосування.

Клас `SoftAccountOrder` реалізує модель замовлень наборів косметичних засобів з такими атрибутами.

Атрибут `softaccount` типу `ForeignKey` визначає користувача, який замовив цей набір косметичних засобів, представляє зв'язок з моделлю `SoftAccount`.

Атрибут `price` типу `DecimalField` визначає вартість, що сумарно відповідає цьому замовленню набору косметичних засобів.

Атрибут `discount` типу `DecimalField` визначає дисконт, який було застосовано до цього замовлення набору косметичних засобів.

Атрибут `shipping` типу `PositiveSmallIntegerField` визначає необхідність виконання доставки та варіант такої доставки або потребу в тому, щоб забрати замовлення самостійно.

Атрибут `status` типу `PositiveSmallIntegerField` визначає поточний стан замовлення набору косметичних засобів.

Атрибут `address` типу `CharField` визначає адресу доставки набору косметичних засобів за замовленням, якщо реалізується доставка.

Атрибут `creation` типу `DateTimeField` визначає часову мітку, коли було створено це замовлення набору косметичних засобів.

Клас `SoftAccountOrderContent` реалізує модель вмісту набору косметичних засобів з замовлення з такими атрибутами.

Атрибут `softaccountorder` типу `ForeignKey` визначає замовлення набору косметичних засобів, до якого відноситься цей вміст, представляє зв'язок з моделлю `SoftAccountOrder`.

Атрибут `cosmeticsgoodvolume` типу `ForeignKey` визначає косметичний засіб, який входить до складу замовлення, представляє зв'язок з моделлю `CosmeticsgoodVolume`.

Атрибут `quantity` типу `PositiveIntegerField` визначає кількість цього косметичного засобу в замовленні;

Атрибут `price` типу `DecimalField` визначає вартість одиниці цього косметичного засобу.

Клас `SoftAccountOrderStatus` реалізує модель виконаних змін станів замовлень наборів косметичних засобів з такими атрибутами.

Атрибут `softaccount` типу `ForeignKey` визначає користувача, який вніс цю зміну до стану замовлення набору косметичних засобів, представляє зв'язок з моделлю `SoftAccount`.

Атрибут `softaccountorder` типу `ForeignKey` визначає замовлення набору косметичних засобів, стан якого було змінено, представляє зв'язок з моделлю `SoftAccountOrder`.

Атрибут `setstatus` типу `PositiveIntegerField` визначає встановлений стан замовлення набору косметичних засобів, який було встановлено.

Атрибут `setwhen` типу `DateTimeField` визначає часову мітку виконання зміни стану замовлення набору косметичних засобів.

#### **4.3 Висновки за розділом 4**

Заплановані для реалізації засоби, включаючи як сам метод прогнозування оцінювання використання косметичних засобів покупцями, так і загальні функціональні можливості магазину продажу косметики, було реалізовано та представлено.

## **5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ**

### **5.1 Опис програми**

Програма призначена для підтримки взаємодії з користувачами магазину косметики через надання їм інформації про наявні косметичні засоби в магазині, підбір потрібних засобів та замовлення цих засобів, надання працівникам магазину розуміння тенденцій в роботі магазину через аналіз обсягів продажів та прогнозування середньої оцінки косметичного засобу від користувачів.

### **5.2 Експлуатація програми**

Робота з програмою спочатку за звернення за адресою вебсервера призводить для відображення для користувача сторінки з переліком брендів косметичних засобів. Одночасно користувач отримує меню, через яке може керувати подальшою роботою. З меню користувач може обрати виконання авторизації, де за необхідності можна зареєструватись, але тільки за обліковим записом покупця. Авторизуватись можна за обліковим записом і покупця, і працівника магазину продажу косметики. В залежності від того, чи це покупець, чи це працівник магазину продажу косметики, меню буде різним, а склад елементів на сторінках також може відрізнятися. Це стосується керуючих елементів, які призводять до додавання, редагування даних передовсім.

З меню можна отримати доступ до переліку брендів косметики («Бренди»), до виконання пошуку косметичних засобів («Пошук»), до набору рейтингів косметичних засобів, де потрібно обрати критерій рейтингу та потрібне значення встановити, якщо це необхідно («Рейтинги»), до власного переліку замовлень набрів косметичних засобів («Покупки»), до панелі

швидкого доступу («Швидкодоступ»), до сторінки профілю («Кабінет»), а також вийти з авторизованого облікового запису («Йти»).

На сторінці косметичного засобу (рис. 5.1) покупець бачить групу косметичних засобів, до якої належить даний, його назву, має можливість за допомогою кнопок «Характеристики», «Опис», «Склад», «Застосування» обирати різні частини інформації, що стосується конкретного косметичного засобу.

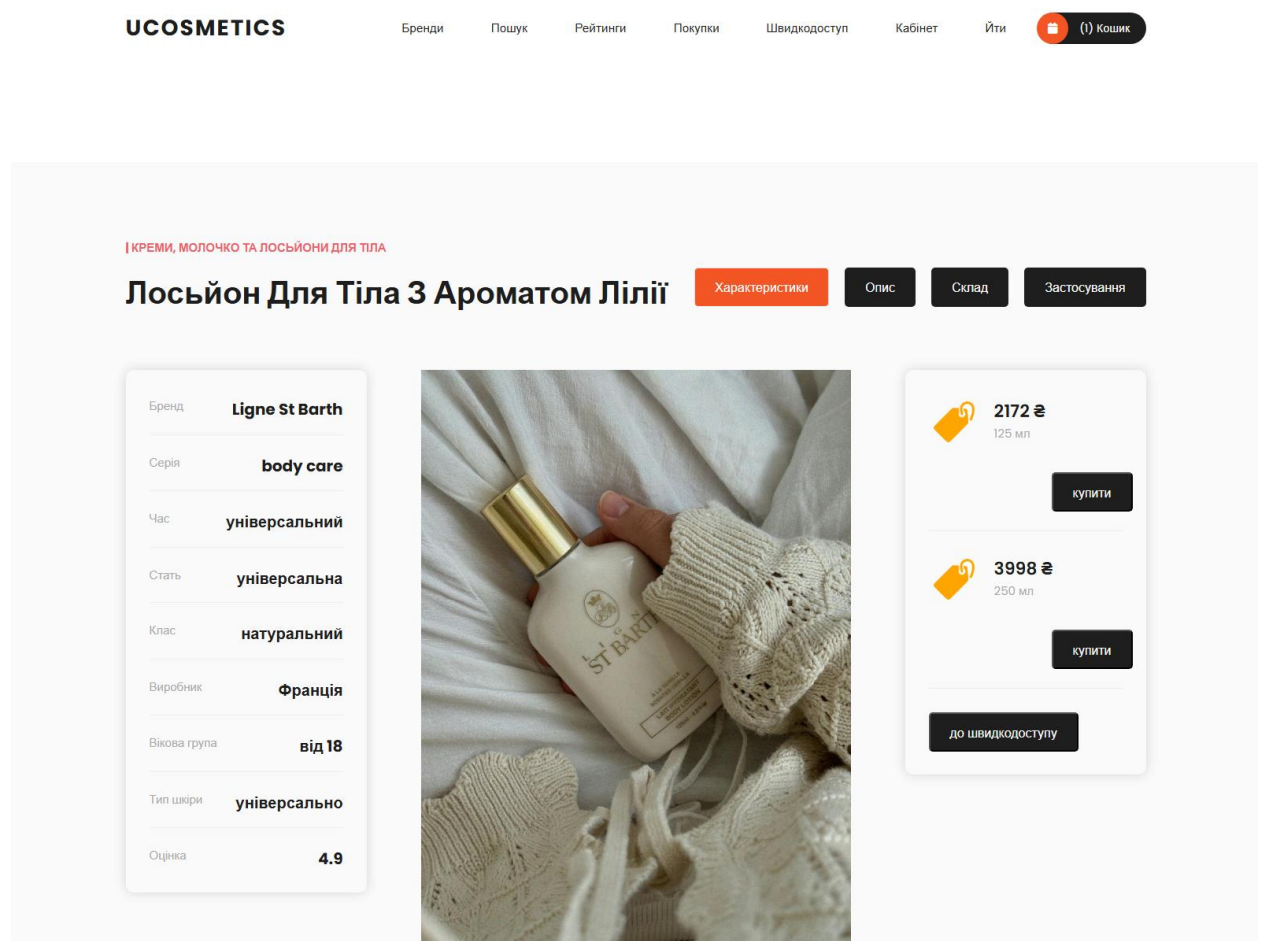


Рисунок 5.1 – Сторінка косметичного засобу для покупця

З цієї сторінки покупець може додавати косметичні засоби в кошик, натискаючи на кнопку «Купити» біля відповідного потрібного обсягу косметичного засобу. Кількість доданих засобів відображається вгорі сторінки у правому куті. Натиснувши на цю позицію, можна відкрити сторінку створення нового замовлення.

У свою чергу працівник магазину продажу косметики має ту ж саму сторінку з додатковими можливостями (рис. 5.2). До таких можливостей належить кнопка для переходу до редагування даних косметичного засобу, що відображається помаранчевою іконкою біля фотографії косметичного засобу у лівому нижньому куті. Біля кожного обсягу косметичного засобу доступна кнопка для переходу до редагування відповідних даних. За допомогою кнопки «Замовлення» можна перейти до сторінки з замовленнями за конкретним косметичним засобом.

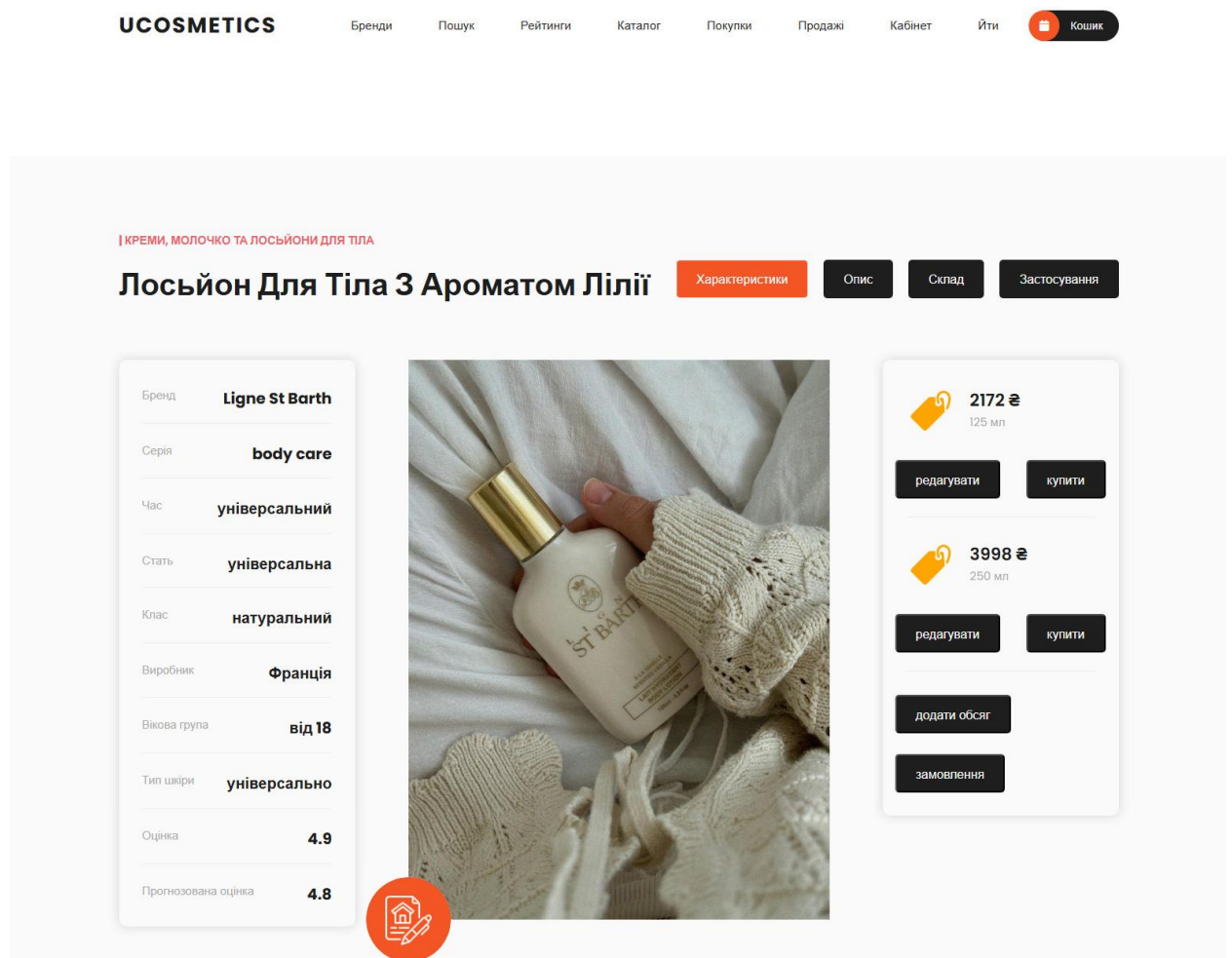


Рисунок 5.2 – Сторінка косметичного засобу для працівника магазину

Окрім того для такого користувача сторінка містить також результат прогнозування оцінки косметичного засобу, що отриманий за допомогою запропонованого методу. Ця інформація не відображається для покупця.

Меню для працівника магазину продажу косметики містить пункт для додавання даних косметичних засобів, серій косметичних засобів, обсягів косметичних засобів («Каталог»), статистику проданих засобів за період часу («Продажі»).

Пункт меню «Покупки» для працівника магазину продажу косметики містить засоби роботи зі всіма замовленнями за магазином (рис. 5.3), при чому спочатку користувач може обрати відповідну кнопку, що визначає стан замовлень, які мають відображатися, також є можливість обрати початок періоду часу, з якого потрібно відобразити замовлення, кінець цього періоду, а також спосіб сортування замовлень.

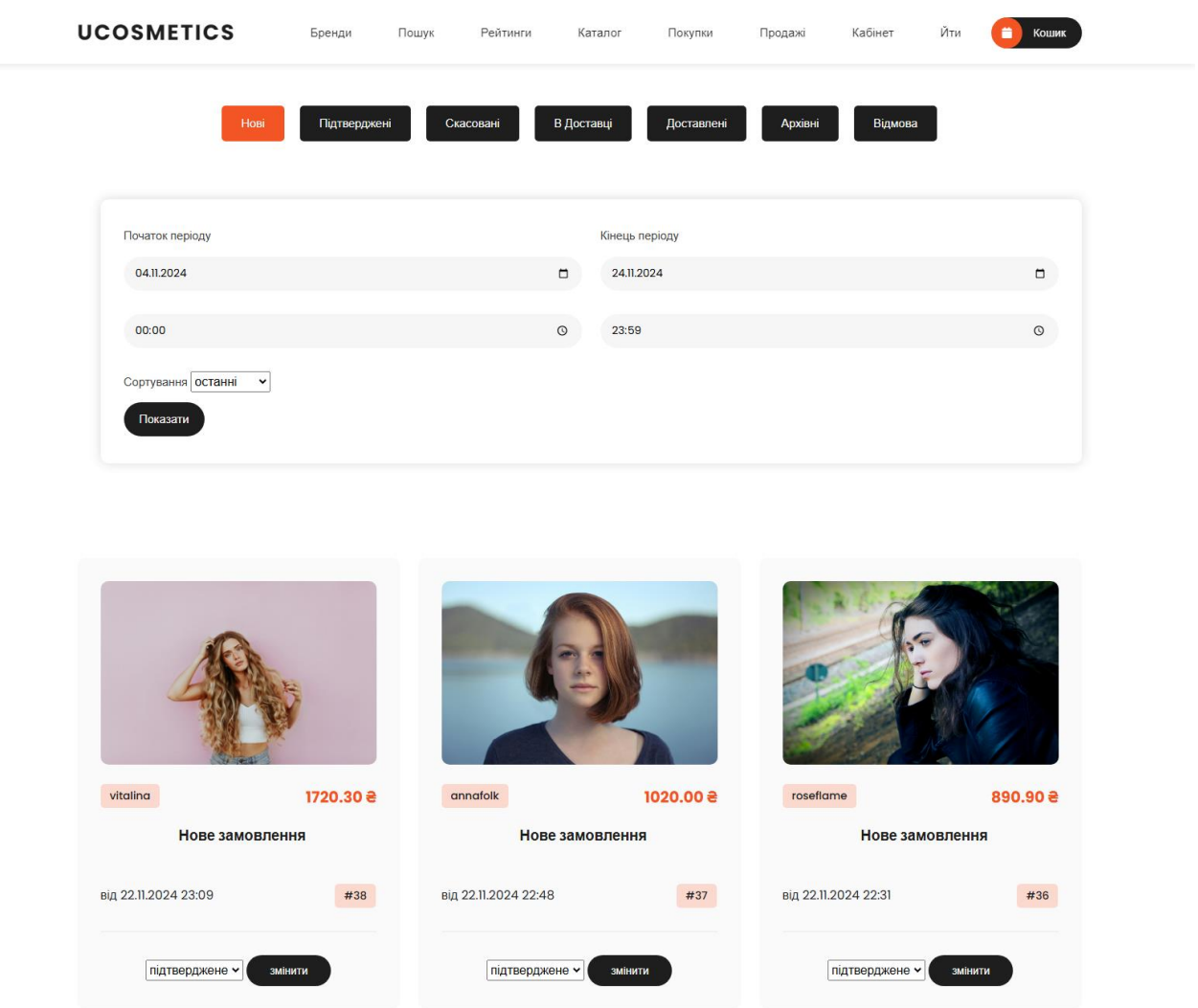


Рисунок 5.3 – Робота з замовленнями наборів косметичних засобів працівником магазину продажу косметики

Після натискання кнопки «Показати» відображається перелік відповідних замовлень. За кожним замовленням є його покупець, натискаючи на нік якого можна перейти до сторінки його профілю. За кожним замовленням можна змінити його стан, зберігши зміни. При натисканні на стан замовлення (підписаний чорним нижче вартості) відкривається сторінка з повним вмістом замовлення.

Виконання пошуку є загальною функцією для всіх користувачів, містить велику кількість фільтрів, що представлені відповідною формою, а результати відображаються нижче форми. Кожна позиція містить перелік для вибору потрібного обсягу (рис. 5.4). В залежності від обраного обсягу відображається відповідна вартість.

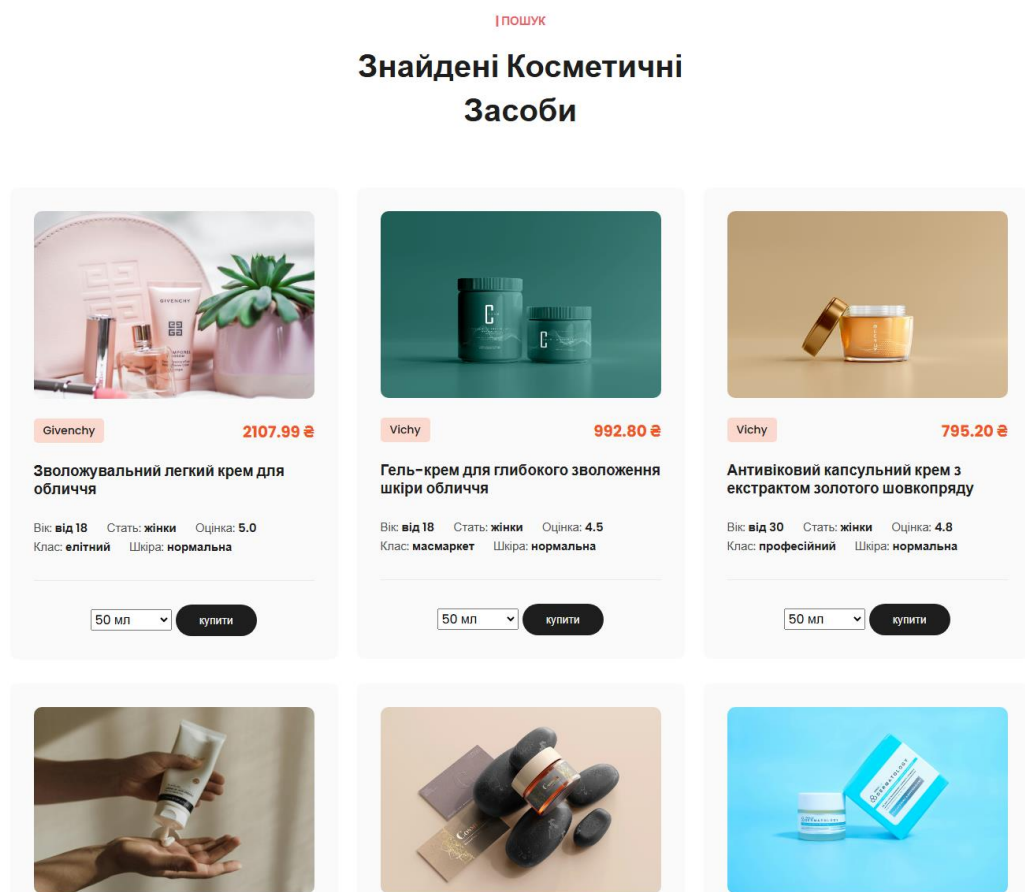


Рисунок 5.4 – Результати пошуку косметичних засобів покупцем

### 5.3 Експериментальне дослідження

Для експериментального дослідження прогнозування оцінювання використання косметичних засобів покупцями було використано вибірку даних [9].

Під час експериментального дослідження спочатку потрібно було визначити набір вхідних ознак для побудови моделі прогнозування оцінювання використання косметичних засобів покупцями.

Склад косметичного засобу потенційно є важливою вхідною ознакою. Однак проблемою є те, що цей склад представлений текстом, включаючи перелік інгредієнтів. Це потребує певного перетворення.

Спочатку було виділено всі наявні інгредієнти та кожен інгредієнт додано до вибірки як окрема ознака зі значеннями 1, якщо цей інгредієнт присутній у косметичному засобі, та 0, якщо ні.

Оскільки кількість інгредієнтів велика, то було розглянуто варіанти вхідних ознак:

- зі всіма інгредієнтами (тобто кожен представлений як окрема вхідна ознака);
- з виділенням в якості вхідних ознак тільки певної кількості інгредієнтів, які мають найвищу оцінку;
- з виділенням однієї ознаки, яка вміщує значення тільки певної кількості інгредієнтів, які мають найвищу оцінку, при цьому воно представлене як бітовий рядок (послідовність 0 та 1, що відповідають кожному інгредієнту), а далі перетворене у десяткову систему числення.

Спочатку для порівняння цих варіантів роботи з інгредієнтами було використано вибір ознак на основі функції `SelectKBest` з бібліотеки `scikit-learn` з використанням показника `f_regression`, що базується на F-тесті (табл. 5.1).

Для оцінювання отриманих результатів прогнозування у даному та всіх наступних випадках використано як показники Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE).

Таблиця 5.1 – Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів представлення переліку інгредієнтів при використанні індивідуальних оцінок для відбору ознак

| Особливості вхідних<br>ознак                            | MAE        | RMSE       | MSE        |
|---------------------------------------------------------|------------|------------|------------|
| Базовий набір                                           | 0,38286428 | 0,64323585 | 0,41375236 |
| Базовий набір та всі<br>інгредієнти                     | 0,36385763 | 0,64113305 | 0,41105158 |
| Базовий набір та 1<br>ознака для<br>інгредієнтів з 10   | 0,38274544 | 0,67020616 | 0,4491763  |
| Базовий набір та 1<br>ознака для<br>інгредієнтів з 30   | 0,40634268 | 0,77187637 | 0,59579313 |
| Базовий набір та 1<br>ознака для<br>інгредієнтів з 50   | 0,42218322 | 0,80055023 | 0,64088067 |
| Базовий набір та 1<br>ознака для<br>інгредієнтів зі 100 | 0,42085543 | 0,76022276 | 0,57793865 |
| Базовий набір та 10<br>ознак для інгредієнтів           | 0,37248929 | 0,62855266 | 0,39507845 |
| Базовий набір та 30<br>ознак для інгредієнтів           | 0,37487973 | 0,6456724  | 0,41689285 |
| Базовий набір та 50<br>ознак для інгредієнтів           | 0,37101883 | 0,6364738  | 0,4050989  |
| Базовий набір та 100<br>ознак для інгредієнтів          | 0,37839345 | 0,6646288  | 0,44173145 |

У другому варіанті використано оцінку `mutual_info_regression` для оцінювання взаємної залежності за допомогою тієї ж функції (табл. 5.2). У обох варіантах було для побудови моделі було використано метод Random Forest та 500 дерев у ансамблі.

Таблиця 5.2 – Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів представлення переліку інгредієнтів при використанні взаємних оцінок для відбору ознак

| Особливості вхідних<br>ознак                           | MAE        | RMSE       | MSE        |
|--------------------------------------------------------|------------|------------|------------|
| Базовий набір                                          | 0,38286428 | 0,64323585 | 0,41375236 |
| Базовий набір та всі<br>інгредієнти                    | 0,36385763 | 0,64113305 | 0,41105158 |
| Базовий набір та 1 озна-<br>ка для інгредієнтів з 10   | 0,38470245 | 0,64370205 | 0,41435232 |
| Базовий набір та 1 озна-<br>ка для інгредієнтів з 30   | 0,37420593 | 0,62850025 | 0,39501257 |
| Базовий набір та 1 озна-<br>ка для інгредієнтів з 50   | 0,37274539 | 0,63635288 | 0,40494499 |
| Базовий набір та 1 озна-<br>ка для інгредієнтів зі 100 | 0,36802742 | 0,62865217 | 0,39520355 |
| Базовий набір та 10<br>ознак для інгредієнтів          | 0,38509186 | 0,64551801 | 0,4166935  |
| Базовий набір та 30<br>ознак для інгредієнтів          | 0,37393303 | 0,62976939 | 0,39660948 |
| Базовий набір та 50<br>ознак для інгредієнтів          | 0,37013316 | 0,62766464 | 0,3939629  |
| Базовий набір та 100<br>ознак для інгредієнтів         | 0,38406838 | 0,6383562  | 0,40749864 |

З отриманих результатів помітно, що найкращим варіантом є використання всіх інгредієнтів як окремих ознак для оцінки MAE. При цьому за оцінок MSE та RMSE деякі інші варіанти дозволили отримати кращі результати. Окрім того додавання даних інгредієнтів дозволяє покращити отримані оцінки порівняно з варіантом без використання даних інгредієнтів.

Далі було перетворено назву на числове представлення, а отримані результати використані як окремі вхідні ознаки або як 1 ознака, обчислена піднесенням кожного значення окремої ознаки на 2 у відповідному ступені, перед тим додавши до значення 0,5, адже фактично отримані значення для кожної ознаки точно не були нижче -0,5 та точно не перевищували 0,5, дозволяючи виконати таке перетворення.

Отримані результати показали, що використання окремих ознак в якості представлення назви є кращим варіантом (табл. 5.3).

Таблиця 5.3 – Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів представлення назви

| Особливості вхідних ознак                         | MAE        | RMSE       | MSE        |
|---------------------------------------------------|------------|------------|------------|
| Базовий набір                                     | 0,38115048 | 0,64023456 | 0,40990029 |
| Базовий набір та всі складові назви               | 0,3644223  | 0,62463021 | 0,3901629  |
| Базовий набір та 1 ознака для представлення назви | 0,37717681 | 0,63950121 | 0,4089618  |

Далі було проаналізовано різні варіанти вхідних базових ознак вибірки даних:

- тип косметичного засобу, переведений у категоріальне значення (ціле число), що позначено далі як Labelcat;
- бренд косметичного засобу, переведений у категоріальне значення (ціле число), що позначено далі як Brandcat;
- вартість продажу одиниці косметичного засобу (позначено далі як Price);
- призначення косметичного засобу для сухої шкіри: значення 1 вказує на призначення косметичного засобу для використання для сухої шкіри, 0 вказує на відсутність такого призначення (позначено далі як Dry);
- призначення косметичного засобу для нормальної шкіри: значення 1 вказує на призначення косметичного засобу для використання для нормальної шкіри, 0 вказує на відсутність такого призначення (позначено далі як Normal);
- призначення косметичного засобу для жирної шкіри: значення 1 вказує на призначення косметичного засобу для використання для жирної шкіри, 0 вказує на відсутність такого призначення (позначено далі як Oily);
- призначення косметичного засобу для одночасно сухої та жирної шкіри: значення 1 вказує на призначення косметичного засобу для використання для жирної та сухої шкіри, 0 вказує на відсутність такого призначення (позначено далі як Combination);
- призначення косметичного засобу для чутливої шкіри: значення 1 вказує на призначення косметичного засобу для використання для чутливої шкіри, 0 вказує на відсутність такого призначення (позначено далі як Sensitive).

Усі вхідні ознаки були впорядковані за оцінкою  $f\_regression$ , а далі найкращі 3 були обрані як множина вхідних ознак, після чого додавались наступні ознаки (табл. 5.4). Отримані результати вказали, що варіант з використанням всіх базових ознак є найкращим у підсумку.

Таблиця 5.4 – Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів основних вхідних ознак

| Множина вхідних<br>ознак                                                                              | MAE        | RMSE       | MSE        |
|-------------------------------------------------------------------------------------------------------|------------|------------|------------|
| ['Normal',<br>'Sensitive', 'Price']                                                                   | 0,42799619 | 0,70438093 | 0,49615249 |
| ['Normal',<br>'Sensitive', 'Price',<br>'Combination',<br>'Oily', 'Dry',<br>'Brandcat']                | 0,41257128 | 0,66936651 | 0,44805153 |
| ['Normal',<br>'Sensitive', 'Price',<br>'Combination',<br>'Oily', 'Dry',<br>'Brandcat',<br>'Labelcat'] | 0,38143234 | 0,64194933 | 0,41209894 |

Далі найкращі з розглянутих варіантів були об'єднані в комбінації (табл. 5.5). Ці результати також були отримані на основі методу Random Forest та 500 дерев у ансамблі.

Найкращий результат за показником MAE був отриманий за використання всіх базових ознак, повної назви та всіх інгредієнтів.

За показниками MAE та RMSE найкращі результати отримані за використання базових ознак та 1 ознаки для інгредієнтів з 30 найкращих за взаємної оцінки.

Таблиця 5.5 – Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів вхідних ознак

| Особливості вхідних ознак                                                 | MAE        | RMSE       | MSE        |
|---------------------------------------------------------------------------|------------|------------|------------|
| Базовий набір                                                             | 0,38110588 | 0,63920134 | 0,40857835 |
| Базовий набір та всі складові назви                                       | 0,36563626 | 0,62731249 | 0,39352096 |
| Базовий набір та всі інгредієнти                                          | 0,36376684 | 0,64382925 | 0,4145161  |
| Базовий набір та 50 ознак для інгредієнтів (індивідуальна оцінка)         | 0,3729209  | 0,63829496 | 0,40742046 |
| Базовий набір та 1 ознака для інгредієнтів з 30 (взаємна оцінка)          | 0,36619323 | 0,61617152 | 0,37966735 |
| Базовий набір та 1 ознака для інгредієнтів зі 100 (взаємна оцінка)        | 0,36442174 | 0,61719343 | 0,38092773 |
| Базовий набір та 50 ознак для інгредієнтів (взаємна оцінка)               | 0,36971162 | 0,62916831 | 0,39585276 |
| Базовий набір, назва та всі інгредієнти                                   | 0,35765076 | 0,63885048 | 0,40812993 |
| Базовий набір, назва та 50 ознак для інгредієнтів (індивідуальна оцінка)  | 0,36185896 | 0,64116193 | 0,41108862 |
| Базовий набір, назва та 1 ознака для інгредієнтів з 30 (взаємна оцінка)   | 0,36458463 | 0,62284891 | 0,38794076 |
| Базовий набір, назва та 1 ознака для інгредієнтів зі 100 (взаємна оцінка) | 0,36296784 | 0,62165364 | 0,38645325 |
| Базовий набір, назва та 50 ознак для інгредієнтів (взаємна оцінка)        | 0,36082711 | 0,62347549 | 0,38872168 |

На останньому етапі було виконано порівняння різних версій ансамблів дерев рішень на основі методу Random Forest, що включали різну кількість дерев, та результатів, отриманих на основі окремого дерева рішення, створеного на основі алгоритму Classification And Regression Tree [10] (табл. 5.6). У даному випадку моделі на основі методу Random Forest були

створені з використанням всіх ознак, що представляють назву, всіх ознак, що представляють інгредієнти, та всіх базових ознак.

Таблиця 5.6 – Підсумкові результати прогнозування оцінювання використання косметичних засобів покупцями

| Особливості<br>вхідних ознак                      | MAE        | RMSE       | MSE        |
|---------------------------------------------------|------------|------------|------------|
| Random Forest (50<br>дерев)                       | 0,3629219  | 0,64312657 | 0,41361179 |
| Random Forest<br>(100 дерев)                      | 0,36191579 | 0,64227822 | 0,41252131 |
| Random Forest<br>(500 дерев)                      | 0,35676543 | 0,63566566 | 0,40407083 |
| Дерево рішень                                     | 0,47419355 | 0,82145467 | 0,67478778 |
| Дерево рішень<br>(базовий набір<br>вхідних ознак) | 0,4857446  | 0,83649795 | 0,69972883 |

Найкращі результати дозволив отримати запропонований метод при реалізації ансамблю дерев рішень на основі 500 дерев, при цьому порівняно з деревами рішень на основі тільки базового набору вхідних ознак показник MAE був зменшений на 26,55 %, показник RMSE – на 24,01 %, а показник MSE – на 42,25 %, а порівняно з деревами рішень зі всіма ознаками MAE був зменшений на 24,76 %, показник RMSE – на 22,62 %, а показник MSE – на 40,12 %.

#### 5.4 Висновки за розділом 5

За результатами розділу 5 отримано завершальні результати роботи, що включають програму магазину продажу косметики з протестованими

функціональними можливостями, що представлені за основними сценаріями в описі, результати експериментального дослідження прогнозування оцінювання використання косметичних засобів покупцями. Результати експериментального дослідження включають результати дослідження використання різних варіантів представлення переліку інгредієнтів косметичного засобу, його назви, набору базових ознак, формування підсумкового набору вхідних ознак, порівняння моделей на основі індивідуальних дерев рішень та ансамблів на основі методу Random Forest, що підтвердило практичну придатність запропонованого методу.

## ВИСНОВКИ

Розроблення програмного забезпечення продажу косметики було виконано у роботі для підтримки взаємодії з користувачами на основі прогнозування оцінювання використання косметичних засобів на попередньому етапі. Це повинно дозволити приймати рішення щодо проведення маркетингових компаній, планування роботи над продажами косметичних засобів, сприяючи ефективності відповідних процесів.

Програма створена як вебзастосунок, а для реалізації використано мову Python разом з пакетами Numpy, Pandas, Scikit-learn, Gensim, фреймворк веброзробки Django з забезпеченням повноцінного сховища на основі системи керування базами даних PostgreSQL.

Розроблена програма призначена для підтримки взаємодії з користувачами магазину косметики через надання їм інформації про наявні косметичні засоби в магазині, підбір потрібних засобів та замовлення цих засобів, надання працівникам магазину розуміння тенденцій в роботі магазину через аналіз обсягів продажів та прогнозування середньої оцінки косметичного засобу від користувачів.

Наукова новизна роботи полягає в методі прогнозування оцінювання використання косметичних засобів покупцями, що забезпечує формування ознак, які представляють назву косметичного засобу, на основі моделі Word2Vec, ознак, які представляють склад косметичного засобу, створення моделі ансамблю дерев рішень, що дозволяє прогнозувати середню оцінку, яку отримає косметичний засіб від покупців. Результати експериментального дослідження включають результати дослідження використання різних варіантів представлення переліку інгредієнтів косметичного засобу, його назви, набору базових ознак, формування підсумкового набору вхідних ознак, порівняння моделей на основі індивідуальних дерев рішень та ансамблів на основі методу Random Forest, що підтвердило практичну придатність запропонованого методу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. MAKEUP – інтернет-магазин косметики та парфумерії №1 [Електрон. ресурс]. – Режим доступу : <https://makeup.com.ua/ua/>.
2. BROCARD у Київ – інтернет-магазин оригінальної парфумерії та косметики в Україні [Електрон. ресурс]. – Режим доступу : <https://www.brocard.ua/>.
3. PROSTOR – інтернет магазин косметики в Україні [Електрон. ресурс]. – Режим доступу : <https://prostor.ua/>.
4. Suri, M. A Dummy's Guide to Word2Vec [Electronic resource] / M. Suri. – Access mode : <https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673>.
5. Random Forest Algorithm in Machine Learning [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>.
6. About. Drupal.org [Electronic resource]. – Access mode : <https://www.drupal.org/about>.
7. Django Web Framework (Python) – Learn web development. MDN [Electronic resource]. – Access mode : <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>.
8. Free Template 591 Villa Agency [Electronic resource]. – Access mode : <https://templatemo.com/tm-591-villa-agency>.
9. Awan, A. A.Cosmetics datasets [Electronic resource] / A. A. Awan. – Access mode : <https://www.kaggle.com/datasets/kingabzpro/cosmetics-datasets/data>.
10. CART (Classification And Regression Tree) in Machine Learning [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>.

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

Магазини косметичних засобів звісно мають велику конкуренцію між собою, але з іншого боку і привертають увагу великої кількості клієнтів. При цьому розвиток засобів машинного навчання може будь-якому такому магазину давати поштовх до розвитку. Інтеграція відповідних функціональних можливостей у програмне забезпечення може дозволити отримати конкурентну перевагу в підсумку. Звісно, що для існування магазину та його успішності важливою є взаємодія з користувачами і те, як вона організована. У той же час кожен магазин має свою клієнтську базу, в жодного з них вона не співпадає. Це призводить до необхідності в тих інструментах, які б дозволяли краще розуміти свою клієнтську базу. Такі інструменти зокрема є корисними при появі нових косметичних засобів. Тоді необхідно розуміти, як клієнти будуть на них реагувати.

Одним з важливих напрямків у даному процесі є розуміння оцінок, які будуть в середньому виставлені косметичному засобу. Однак, для того щоб отримати точну інформацію про середню оцінку, необхідний достатньо тривалий період часу. Можуть минати роки, адже на початковому етапі кількість відгуків або дуже мала, або вони взагалі відсутні. Тому прогнозування середньої оцінки косметичного засобу в момент його появи дозволяє розмістити наявну аудиторію та планувати подальшу роботу з ним. Відповідно інтегрування таких засобів у програмне забезпечення підтримки магазину косметики є важливим завданням.

### **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 підтримки взаємодії з

користувачами магазину косметики», що було деталізовано також виданим завданням на роботу.

## **A.2 Призначення розробки**

Розробка призначена для підтримки взаємодії з користувачами магазину косметики через надання їм інформації про наявні косметичні засоби в магазині, підбір потрібних засобів та замовлення цих засобів, надання працівникам магазину розуміння тенденцій в роботі магазину через аналіз обсягів продажів та прогнозування середньої оцінки косметичного засобу від користувачів.

## **A.3 Основні вимоги до програми**

### **A.3.1 Вимоги до функціональних характеристик**

Для реалізації комплексної роботи вебзастосунку продажу косметичних засобів необхідно забезпечити такі функціональні можливості:

- перегляд списку брендів косметики у магазині;
- додавання бренду косметики, що продається в магазині;
- редагування даних бренду косметики, що продається в магазині;
- перегляд інформації про бренд та його серії косметичних засобів;
- редагування даних серії косметичних засобів;
- додавання нової серії косметичних засобів;
- перегляд набору косметичних засобів за серією;
- пошук косметичних засобів;
- оформлення замовлення набору косметичних засобів;
- перегляд інформації про косметичний засіб;
- збереження косметичного засобу для швидкого доступу;
- вилучення косметичного засобу зі швидкого доступу;
- перегляд панелі швидкого доступу до косметичних засобів;

- додавання нового косметичного засобу;
- редагування інформації про косметичний засіб;
- додавання нового доступного обсягу косметичного засобу;
- внесення інформації про доступний обсяг косметичного засобу;
- реєстрація облікового запису користувача;
- авторизація облікового запису користувача або працівника магазину косметичних засобів;
- коментування косметичного засобу щодо досвіду власного застосування;
- перегляд прокоментованих користувачем застосувань косметичних засобів;
- перегляд профілю користувача;
- перегляд замовлень наборів косметичних засобів користувачем;
- перегляд всіх замовлень наборів косметичних засобів у магазині з фільтруванням;
- оновлення стану замовлення набору косметичних засобів;
- перегляд повної інформації про замовлення набору косметичних засобів;
- прогнозування середньої оцінки косметичного засобу від користувачів;
- перегляд замовлень за косметичним засобом;
- перегляд сумарних обсягів продажу за косметичним засобом загалом або за заданим обсягом за період часу;
- перегляд рейтингу косметичних засобів зі вмістом певного інгредієнту;
- перегляд рейтингу косметичних засобів за групою;
- перегляд рейтингу косметичних засобів за типом шкіри;
- перегляд рейтингу косметичних засобів за віковою групою;
- перегляд рейтингу косметичних засобів за брендом;
- перегляд рейтингу косметичних засобів за застосуванням;

- перегляд рейтингу косметичних засобів за сертифікатом;
- перегляд рейтингу косметичних засобів за формулою.

### **A.3.2 Умови експлуатації**

Експлуатація ПЗ підтримки роботи магазину косметики 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 PostgreSQL;
- інсталювані Python-пакети Numpy, Pandas, Scikit-learn.

### **A.3.3 Вимоги до складу та параметрів технічних засобів**

ПЗ підтримки роботи магазину косметики 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 косметичних засобів, 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 Cosmeticgroup
from .models import Agegroup
from .models import Cosmeticsbrand
from .models import Cosmeticsserie
from .models import Application
from .models import Cosmeticsgood
from .models import Certificate
from .models import Formula
from .models import Ingredient
from .models import CosmeticsgoodPhoto
from .models import CosmeticsgoodVolume
from .models import Country
from .models import SoftAccount
from .models import CosmeticsgoodComment
from .models import SoftAccountOrder
from .models import SoftAccountOrderContent
from .models import SoftAccountOrderStatus
from django.core.paginator import Paginator
from django.shortcuts import render

def ingredient_cosmetics_rating_view(request,
ingredient_id):
    ingred = None
    try:
        ingred = Ingredient.objects.get(id = ingredient_id)
    except Ingredient.DoesNotExist:
        cosmetics = None
    if ingred:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups").select_
related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(ingredients__in =
[ingred]).order_by("-mark")[:100]
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user
                cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
                for cosmetic in cosmetics:
                    if cosmetic in
cosmetics_account.cosmeticsgoods:
                        cosmetic.tobesaved = False
                    else:
                        cosmetic.tobesaved = True

```

```

        paginator_for_cosmetics = Paginator(cosmetics, 10)
        cosmetics_number = request.GET.get('part')
        cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
        return render(request, "ucosmetics/ingrating.html",
{"cosmetics": cosmetics})

def group_cosmetics_rating_view(request, cosmeticgroup_id):
    cosmeticgroup = None
    try:
        cosmeticgroup = Cosmeticgroup.objects.get(id =
cosmeticgroup_id)
    except Cosmeticgroup.DoesNotExist:
        cosmetics = None
    if cosmeticgroup:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups").select_
related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(cosmeticgroups__in =
[cosmeticgroup]).order_by("-mark")[:100]
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user
                cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
                for cosmetic in cosmetics:
                    if cosmetic in
cosmetics_account.cosmeticsgoods:
                        cosmetic.tobesaved = False
                    else:
                        cosmetic.tobesaved = True
                paginator_for_cosmetics = Paginator(cosmetics, 10)
                cosmetics_number = request.GET.get('part')
                cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
                return render(request, "ucosmetics/grrating.html",
{"cosmetics": cosmetics})

def skin_cosmetics_rating_view(request, skintype):
    cosmetics_skin =
Cosmeticsgoodskin.objects.select_related("cosmeticsgood").filter
(skintype = skintype).values_list("cosmeticsgood__id")
    cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups").select_
related("agegroup", "cosmeticsserie",

```

```

"cosmeticsserie__cosmeticsbrand").filter(id__in =
cosmetics_skin).order_by("-mark")[:100]
    for cosmetic in cosmetics:
        cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
        cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
        cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
        if request.user.is_authenticated:
            cosmetics_user = request.user
            cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
            for cosmetic in cosmetics:
                if cosmetic in
cosmetics_account.cosmeticsgoods:
                    cosmetic.tobesaved = False
                else:
                    cosmetic.tobesaved = True
            paginator_for_cosmetics = Paginator(cosmetics, 10)
            cosmetics_number = request.GET.get('part')
            cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
            return render(request, "ucosmetics/skinrating.html",
{"cosmetics": cosmetics})

def age_cosmetics_rating_view(request, agegroup_id):
    agegroup = None
    try:
        agegroup = Agegroup.objects.get(id = agegroup_id)
    except Agegroup.DoesNotExist:
        cosmetics = None
    if agegroup:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups").select_
related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(agegroup =
agegroup).order_by("-mark")[:100]
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user
                cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)

```

```

        for cosmetic in cosmetics:
            if cosmetic in
cosmetics_account.cosmeticsgoods:
                cosmetic.tobesaved = False
            else:
                cosmetic.tobesaved = True
        paginator_for_cosmetics = Paginator(cosmetics, 10)
        cosmetics_number = request.GET.get('part')
        cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
        return render(request, "ucosmetics/agerating.html",
{"cosmetics": cosmetics})

def brand_cosmetics_rating_view(request,
cosmeticsbrand_id):
    cosmeticsbrand = None
    try:
        cosmeticsbrand = Cosmeticsbrand.objects.get(id =
cosmeticsbrand_id)
    except Cosmeticsbrand.DoesNotExist:
        cosmetics = None
    if cosmeticsbrand:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups").select_
related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(cosmeticsserie__cosmeti
csbrand = cosmeticsbrand).order_by("-mark")[:100]
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user
                cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
                for cosmetic in cosmetics:
                    if cosmetic in
cosmetics_account.cosmeticsgoods:
                        cosmetic.tobesaved = False
                    else:
                        cosmetic.tobesaved = True
                paginator_for_cosmetics = Paginator(cosmetics, 10)
                cosmetics_number = request.GET.get('part')
                cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
                return render(request, "ucosmetics/brandrating.html",
{"cosmetics": cosmetics})

```

```

def application_cosmetics_rating_view(request,
application_id):
    cosmeticsapplication = None
    try:
        cosmeticsapplication = Application.objects.get(id =
application_id)
    except Application.DoesNotExist:
        cosmetics = None
    if cosmeticsapplication:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups",
"applications").select_related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(applications__in =
[cosmeticsapplication]).order_by("-mark")[:100]
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user
                cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
                for cosmetic in cosmetics:
                    if cosmetic in
cosmetics_account.cosmeticsgoods:
                        cosmetic.tobesaved = False
                    else:
                        cosmetic.tobesaved = True
                paginator_for_cosmetics = Paginator(cosmetics, 10)
                cosmetics_number = request.GET.get('part')
                cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
                return render(request, "ucosmetics/apprating.html",
{"cosmetics": cosmetics})

def certificate_cosmetics_rating_view(request,
certificate_id):
    certificate_for_cosmetics = None
    try:
        certificate_for_cosmetics =
Certificate.objects.get(id = certificate_id)
    except Certificate.DoesNotExist:
        cosmetics = None
    if certificate_for_cosmetics:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups",
"certificates").select_related("agegroup", "cosmeticsserie",

```

```

"cosmeticsserie__cosmeticsbrand").filter(certificates__in =
[certificate_for_cosmetics]).order_by("-mark")[:100]
    for cosmetic in cosmetics:
        cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
        cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
        cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
        if request.user.is_authenticated:
            cosmetics_user = request.user
            cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
            for cosmetic in cosmetics:
                if cosmetic in
cosmetics_account.cosmeticsgoods:
                    cosmetic.tobesaved = False
                else:
                    cosmetic.tobesaved = True
            paginator_for_cosmetics = Paginator(cosmetics, 10)
            cosmetics_number = request.GET.get('part')
            cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
            return render(request, "ucosmetics/certifrating.html",
{"cosmetics": cosmetics})

def formula_cosmetics_rating_view(request, formula_id):
    formula_for_cosmetics = None
    try:
        formula_for_cosmetics = Formula.objects.get(id =
formula_id)
    except Formula.DoesNotExist:
        cosmetics = None
    if formula_for_cosmetics:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups",
"formulas").select_related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(formulas__in =
[formula_for_cosmetics]).order_by("-mark")[:100]
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user

```

```

        cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
        for cosmetic in cosmetics:
            if cosmetic in
cosmetics_account.cosmeticsgoods:
                cosmetic.tobesaved = False
            else:
                cosmetic.tobesaved = True
        paginator_for_cosmetics = Paginator(cosmetics, 10)
        cosmetics_number = request.GET.get('part')
        cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
        return render(request, "ucosmetics/formularating.html",
{"cosmetics": cosmetics})

def cosmetic_brand_view(request, cosmeticsbrand_id):
    cosmetic_series = None
    cosmeticsbrand = None
    try:
        cosmeticsbrand =
Cosmeticsbrand.objects.select_related("brandcountry").get(id =
cosmeticsbrand_id)
    except Cosmeticsbrand.DoesNotExist:
        cosmetics = None
    if cosmeticsbrand:
        cosmetic_series =
Cosmeticsserie.objects.filter(cosmeticsbrand =
cosmeticsbrand).order_by("seriesname")
        return render(request,
"ucosmetics/cosmeticsbrand.html", {"cosmeticseries":
cosmetic_series, "brand": cosmeticsbrand })

def cosmetic_series_view(request, cosmeticserie_id,
cosmetics_order = 0):
    cosmetics = None
    cosmeticsserie = None
    try:
        cosmeticsserie =
Cosmeticsserie.objects.select_related("cosmeticsbrand").get(id =
cosmeticserie_id)
    except Cosmeticsserie.DoesNotExist:
        cosmetics = None
    if cosmeticsserie:
        cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups").select_
related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").filter(cosmeticsserie =
cosmeticsserie)
        if cosmetics_order == 0:
            cosmetics = cosmetics.order_by("-mark")
        elif cosmetics_order == 1:
            cosmetics = cosmetics.order_by("mark")

```

```

        elif cosmetics_order == 3:
            cosmetics =
cosmetics.order_by("cosmeticcaption")
        else:
            cosmetics = cosmetics.order_by("-
cosmeticcaption")
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            if request.user.is_authenticated:
                cosmetics_user = request.user
                cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
            for cosmetic in cosmetics:
                if cosmetic in
cosmetics_account.cosmeticsgoods:
                    cosmetic.tobesaved = False
                else:
                    cosmetic.tobesaved = True
            paginator_for_cosmetics = Paginator(cosmetics, 10)
            cosmetics_number = request.GET.get('part')
            cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
            return render(request,
"ucosmetics/cosmeticsserie.html", {"cosmeticserie":
cosmeticsserie, "cosmetics": cosmetics })

def cosmetics_view(request, cosmetics_id):
    cosmetics = None
    try:
        cosmetics =
Cosmetics.objects.prefetch_related("cosmeticgroups",
"certificates", "formulas", "applications",
"ingredients").select_related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").get(id = cosmetics_id)
    except Cosmetics.DoesNotExist:
        cosmetics = None
    cosmetics.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetics)
    cosmetics.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetics)
    cosmetics.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetics)
    if request.user.is_authenticated:
        cosmetics_user = request.user

```

```

        cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
        if cosmetics in cosmetics_account.cosmeticsgoods:
            cosmetics.tobesaved = False
        else:
            cosmetics.tobesaved = True
        cosmetics_comments =
CosmeticsgoodComment.objects.select_related("commentersoftaccoun
t", "commentersoftaccount__user").filter(cosmeticsgood =
cosmetics).order_by("-commentedwhen")
        paginator_for_cosmetics = Paginator(cosmetics_comments,
10)
        cosmetics_number = request.GET.get('part')
        cosmetics_comments =
paginator_for_cosmetics.get_page(cosmetics_number)
        return render(request, "ucosmetics/cosmetics.html",
{"cosmetics": cosmetics })

def cosmetics_search_view(request):
    cosmetic_groups =
Cosmeticgroup.objects.all().order_by("cosmeticgroupname")
    age_groups =
Agegroup.objects.all().order_by("agegroupname")
    cosmetic_brands =
Cosmeticsbrand.objects.all().order_by("cosbrandname")
    for cosmetic_brand in cosmetic_brands:
        cosmetic_brand.series =
Cosmeticsserie.objects.filter(cosmeticsbrand = cosmetic_brand)
        .order_by("seriesname")
    cosmetic_applications =
Application.objects.all().order_by("applicationname")
    cosmetic_certificates =
Certificate.objects.all().order_by("certname")
    cosmetic_formulas =
Formula.objects.all().order_by("formulaname")
    cosmetic_ingredients =
Ingredient.objects.all().order_by("igredname")
    countries = Country.objects.all().order_by("name")
    cosmetics =
Cosmeticsgood.objects.prefetch_related("cosmeticgroups",
"certificates", "formulas", "applications",
"ingredients").select_related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").all()
    search_set_cosmetic_groups = request.GET.get('groups')
    if search_set_cosmetic_groups:
        search_set_cosmetic_groups_criterion = []
        search_set_cosmetic_groups =
search_set_cosmetic_groups.split(",")
        for search_set_cosmetic_group in
search_set_cosmetic_groups:
            search_set_cosmetic_group_id =
int(search_set_cosmetic_group)

```

```

        search_set_cosmetic_group =
Cosmeticgroup.objects.get(id = search_set_cosmetic_group_id)

search_set_cosmetic_groups_criterion.append(search_set_cosmetic_
group)
        cosmetics = cosmetics.filter(cosmeticgroups__in =
search_set_cosmetic_groups_criterion)
        search_set_age_groups = request.GET.get('age')
        if search_set_age_groups:
            search_set_age_groups_criterion = []
            search_set_age_groups =
search_set_age_groups.split(",")
            for search_set_age_group in search_set_age_groups:
                search_set_age_group_id =
int(search_set_age_group)
                search_set_age_group =
Cosmeticgroup.objects.get(id = search_set_age_group_id)

search_set_age_groups_criterion.append(search_set_age_group)
        cosmetics = cosmetics.filter(agegroup__in =
search_set_age_groups_criterion)
        search_set_brands = request.GET.get('brands')
        if search_set_brands:
            search_set_brands_criterion = []
            search_set_brands = search_set_brands.split(",")
            for search_set_brand in search_set_brands:
                search_set_brand_id = int(search_set_brand)
                search_set_brand =
Cosmeticsbrand.objects.get(id = search_set_brand_id)

search_set_age_groups_criterion.append(search_set_brand)
        cosmetics =
cosmetics.filter(cosmeticsserie__cosmeticsbrand__in =
search_set_brands_criterion)
        search_set_series = request.GET.get('serie')
        if search_set_series:
            search_set_series_criterion = []
            search_set_series = search_set_series.split(",")
            for search_set_serie in search_set_series:
                search_set_serie_id = int(search_set_serie)
                search_set_serie =
Cosmeticsserie.objects.get(id = search_set_serie_id)

search_set_series_criterion.append(search_set_serie)
        cosmetics = cosmetics.filter(cosmeticsserie__in =
search_set_series_criterion)
        search_set_applications =
request.GET.get('application')
        if search_set_applications:
            search_set_applications_criterion = []
            search_set_applications =
search_set_applications.split(",")

```

```

        for search_set_application in
search_set_applications:
            search_set_application_id =
int(search_set_application)
            search_set_application =
Application.objects.get(id = search_set_application_id)

search_set_applications_criterion.append(search_set_application)
            cosmetics = cosmetics.filter(applications__in =
search_set_applications_criterion)
            search_set_caption = request.GET.get('costext')
            if search_set_caption:
                cosmetics =
cosmetics.filter(cosmeticcaption__icontains =
search_set_caption)
            search_set_time_periods =
request.GET.get('timeperiods')
            if search_set_time_periods:
                search_set_time_periods_criterion = []
                search_set_time_periods =
search_set_time_periods.split(",")
                for search_set_time_period in
search_set_time_periods:
                    search_set_time_period =
int(search_set_time_period)

search_set_time_periods_criterion.append(search_set_time_period)
                    cosmetics = cosmetics.filter(timeperiod__in =
search_set_time_periods_criterion)
                    search_set_gender = request.GET.get('gender')
                    if search_set_gender:
                        search_set_gender = int(search_set_gender)
                        cosmetics = cosmetics.filter(gender =
search_set_gender)
                    search_set_marketclasses = request.GET.get('markets')
                    if search_set_marketclasses:
                        search_set_marketclasses_criterion = []
                        search_set_marketclasses =
search_set_marketclasses.split(",")
                        for search_set_marketclass in
search_set_marketclasses:
                            search_set_marketclass =
int(search_set_marketclass)

search_set_marketclasses_criterion.append(search_set_marketclass
)
                            cosmetics = cosmetics.filter(marketclass__in =
search_set_marketclasses_criterion)
                            search_set_minmark = request.GET.get('minmark')
                            if search_set_minmark:
                                search_set_minmark = float(search_set_minmark)
                                cosmetics = cosmetics.filter(mark__gte =
search_set_minmark)

```

```

        search_set_maxmark = request.GET.get('maxmark')
        if search_set_maxmark:
            search_set_maxmark = float(search_set_maxmark)
            cosmetics = cosmetics.filter(mark__lte =
search_set_maxmark)
        for cosmetic in cosmetics:
            cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
            cosmetic.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic)
            cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
            search_set_certificates =
request.GET.get('certificate')
            if search_set_certificates:
                search_set_applications_criterion = []
                search_set_certificates =
search_set_certificates.split(",")
                for search_set_certificate in
search_set_certificates:
                    search_set_certificate_id =
int(search_set_certificate)
                    search_set_certificate =
Certificate.objects.get(id = search_set_certificate_id)

                search_set_applications_criterion.append(search_set_certificate)
                cosmetics = cosmetics.filter(certificates__in =
search_set_applications_criterion)
            search_set_formulas = request.GET.get('formula')
            if search_set_formulas:
                search_set_formulas_criterion = []
                search_set_formulas =
search_set_formulas.split(",")
                for search_set_formula in search_set_formulas:
                    search_set_formula_id = int(search_set_formula)
                    search_set_formula = Formula.objects.get(id =
search_set_formula_id)

                search_set_formulas_criterion.append(search_set_formula)
                cosmetics = cosmetics.filter(formulas__in =
search_set_formulas_criterion)
            search_set_ingredients = request.GET.get('ings')
            if search_set_ingredients:
                search_set_ingredients_criterion = []
                search_set_ingredients =
search_set_ingredients.split(",")
                for search_set_ingredient in
search_set_ingredients:
                    search_set_ingredient_id =
int(search_set_ingredient)
                    search_set_ingredient =
Ingredient.objects.get(id = search_set_ingredient)

```

```

        search_set_ingredients_criterion.append(search_set_ingredient)

        cosmetics = cosmetics.filter(ingredients__in =
search_set_ingredients_criterion)
        search_set_ingredients = request.GET.get('notings')
        if search_set_ingredients:
            search_set_ingredients_criterion = []
            search_set_ingredients =
search_set_ingredients.split(",")
            for search_set_ingredient in
search_set_ingredients:
                search_set_ingredient_id =
int(search_set_ingredient)
                search_set_ingredient =
Ingredient.objects.get(id = search_set_ingredient)

search_set_ingredients_criterion.append(search_set_ingredient)
        cosmetics = cosmetics.exclude(ingredients__in =
search_set_ingredients_criterion)
        search_set_order = request.GET.get('order')
        if search_set_order:
            search_set_order = int(search_set_order)
            if search_set_order == 0:
                cosmetics = cosmetics.order_by("-mark")
            elif search_set_order == 1:
                cosmetics = cosmetics.order_by("mark")
            elif search_set_order == 3:
                cosmetics =
cosmetics.order_by("cosmeticcaption")
            else:
                cosmetics = cosmetics.order_by("-
cosmeticcaption")
        search_set_available = request.GET.get('available')
        if search_set_available:
            cosmetics_volumed = []
            for cosmetic in cosmetics:
                cosmetic_volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic,
amount__gt = 0)
                if cosmetic_volumes:
                    cosmetics_volumed.append(cosmetic)
            cosmetics = cosmetics_volumed
        search_set_skins = request.GET.get('skins')
        if search_set_skins:
            cosmetics_skinned = []
            search_set_skins_criterion = []
            search_set_skins = search_set_skins.split(",")
            for search_set_skin in search_set_skins:
                search_set_skin = int(search_set_skin)

search_set_skins_criterion.append(search_set_skin)
            for cosmetic in cosmetics:

```

```

        cosmetic_skins =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetic,
skintype__in = search_set_skins_criterion)
        if cosmetic_skins:
            cosmetics_skinned.append(cosmetic)
        cosmetics = cosmetics_skinned
        search_set_countries = request.GET.get('countries')
        if search_set_countries:
            search_set_countries_criterion = []
            search_set_countries =
search_set_countries.split(",")
            for search_set_country in search_set_countries:
                search_set_country_id = int(search_set_country)
                search_set_country = Country.objects.get(id =
search_set_country_id)

search_set_countries_criterion.append(search_set_country)
            cosmetics_volumed = []
            for cosmetic in cosmetics:
                cosmetic_volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic,
producedcountry__in = search_set_countries_criterion)
                if cosmetic_volumes:
                    cosmetics_volumed.append(cosmetic)
                cosmetics = cosmetics_volumed
            for cosmetic in cosmetics:
                cosmetic.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetic)
                if request.user.is_authenticated:
                    cosmetics_user = request.user
                    cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
                for cosmetic in cosmetics:
                    if cosmetic in
cosmetics_account.cosmeticsgoods:
                        cosmetic.tobesaved = False
                    else:
                        cosmetic.tobesaved = True
                paginator_for_cosmetics = Paginator(cosmetics, 10)
                cosmetics_number = request.GET.get('part')
                cosmetics =
paginator_for_cosmetics.get_page(cosmetics_number)
            return render(request, "ucosmetics/search.html",
{"cosmetics": cosmetics, "groups": cosmetic_groups,
"age_groups": age_groups, "brands": cosmetic_brands,
"applications": cosmetic_applications, "certificates":
cosmetic_certificates, "formulas": cosmetic_formulas,
"ingredients": cosmetic_ingredients, "countries": countries })

def commented_cosmetics_view(request, user_login = None):
    user_to_get = None
    if request.user.is_authenticated:

```

```

        cosmetics_user = request.user
        cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
        if cosmetics_account.user.is_staff:
            user_to_get =
SoftAccount.objects.get(user__username = user_login)
        else:
            user_to_get = cosmetics_account
            cosmetics_comments =
CosmeticsgoodComment.objects.select_related("commentersoftaccoun
t", "commentersoftaccount__user",
"cosmeticsgood").filter(commentersoftaccount =
user_to_get).order_by("-commentedwhen")
            for cosmetics_comment in cosmetics_comments:
                cosmetics_comment.cosmeticsgood.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood =
cosmetics_comment.cosmeticsgood, topphoto = True)
                if not cosmetics_account.user.is_staff:
                    if cosmetics_comment.cosmeticsgood in
user_to_get.cosmeticsgoods:

cosmetics_comment.cosmeticsgood.tobesaved = False
                    else:

cosmetics_comment.cosmeticsgood.tobesaved = True
                paginator_for_cosmetics =
Paginator(cosmetics_comments, 10)
                cosmetics_number = request.GET.get('part')
                cosmetics.comments =
paginator_for_cosmetics.get_page(cosmetics_number)
                return render(request, "ucosmetics/comments.html",
{"cosmetics": cosmetics })
            else:
                return render(request, "ucosmetics/comments.html",
{"cosmetics": None })

def cosmetics_order_view(request, order_id):
    if request.user.is_authenticated:
        cosmetics_user = request.user
        cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
        access_true = False
        cosmetics_order =
SoftAccountOrder.objects.select_related("softaccount").get(id =
order_id)
        if cosmetics_account.user.is_staff:
            access_true = True
        else:
            if cosmetics_order.softaccount ==
cosmetics_account:
                access_true = True

```

```

        if not access_true:
            return render(request, "ucosmetics/order.html",
{"order": None, "cosmetics": None, "statuses": None })
        else:
            statuses =
SoftAccountOrderStatus.objects.filter(softaccountorder =
cosmetics_order).order_by("setwhen")
            cosmetics =
SoftAccountOrderContent.objects.filter(softaccountorder =
cosmetics_order)
            for cosmetic in cosmetics:
                cosmetic.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetic,
topphoto = True)
                if not cosmetics_account.user.is_staff:
                    if cosmetic in
cosmetics_account.cosmeticsgoods:
                        cosmetic.tobesaved = False
                    else:
                        cosmetic.tobesaved = True
            return render(request, "ucosmetics/order.html",
{"order": cosmetics_order, "cosmetics": cosmetics, "statuses":
statuses })
        else:
            return render(request, "ucosmetics/order.html",
{"order": None, "cosmetics": None, "statuses": None })

def personal_cosmetics_orders_view(request):
    if request.user.is_authenticated:
        cosmetics_user = request.user
        cosmetics_account =
SoftAccount.objects.prefetch_related("cosmeticsgoods").get(user
= cosmetics_user)
        cosmetics_orders =
SoftAccountOrder.objects.select_related("softaccount").filter(so
ftaccount = cosmetics_account)
        cosmetics_ord = request.GET.get('order')
        if cosmetics_ord:
            cosmetics_ord = int(cosmetics_ord)
        else:
            cosmetics_ord = 0
        if cosmetics_ord == 0:
            cosmetics_orders = cosmetics_orders.order_by("-
creation")
        elif cosmetics_ord == 1:
            cosmetics_orders =
cosmetics_orders.order_by("creation")
        elif cosmetics_ord == 2:
            cosmetics_orders = cosmetics_orders.order_by("-
price")
        elif cosmetics_ord == 3:
            cosmetics_orders =
cosmetics_orders.order_by("price")

```

```

        return render(request,
"ucosmetics/persorders.html", {"orders": cosmetics_orders })
    else:
        return render(request,
"ucosmetics/persorders.html", {"orders": None})

def shop_cosmetics_orders_view(request):
    if not request.user.is_authenticated:
        return render(request, "ucosmetics/allorders.html",
{"orders": None})
    else:
        cosmetics_user = request.user
        if not cosmetics_user.is_staff:
            return render(request,
"ucosmetics/allorders.html", {"orders": None})
        cosmetics_orders =
SoftAccountOrder.objects.select_related("softaccount",
"softaccount__user").all()
        from datetime import datetime
        cosmetics_date1 = request.GET.get('dp1')
        if cosmetics_date1:
            cosmetics_date1=
datetime.strptime(cosmetics_date1, '%d.%m.%y %H:%M')
            cosmetics_orders =
cosmetics_orders.filter(creation__gte = cosmetics_date1)
            cosmetics_date2 = request.GET.get('dp2')
            if cosmetics_date2:
                cosmetics_date2=
datetime.strptime(cosmetics_date2, '%d.%m.%y %H:%M')
                cosmetics_orders =
cosmetics_orders.filter(creation__lte = cosmetics_date2)
            cosmetics_statuses = request.GET.get('stat')
            if cosmetics_statuses:
                cosmetics_statuses =
cosmetics_statuses.split(',')
                cosmetics_statuses_criterion = []
                for cosmetics_status in cosmetics_statuses:
                    cosmetics_status = int(cosmetics_status)

cosmetics_statuses_criterion.append(cosmetics_status)
                cosmetics_orders =
cosmetics_orders.filter(status__in =
cosmetics_statuses_criterion)
            cosmetics_ord = request.GET.get('order')
            if cosmetics_ord:
                cosmetics_ord = int(cosmetics_ord)
            else:
                cosmetics_ord = 0
            if cosmetics_ord == 0:
                cosmetics_orders = cosmetics_orders.order_by("-
creation")
            elif cosmetics_ord == 1:

```

```

        cosmetics_orders =
cosmetics_orders.order_by("creation")
        elif cosmetics_ord == 2:
            cosmetics_orders = cosmetics_orders.order_by("-
price")
        elif cosmetics_ord == 3:
            cosmetics_orders =
cosmetics_orders.order_by("price")
            return render(request, "ucosmetics/allorders.html",
{"orders": cosmetics_orders })

def shop_cosmetic_orders_view(request, cosmetics_id):
    if not request.user.is_authenticated:
        return render(request, "ucosmetics/cosorders.html",
{"orders": None})
    else:
        cosmetics = None
        try:
            cosmetics =
Cosmetics.objects.prefetch_related("cosmeticgroups",
"certificates", "formulas", "applications",
"ingredients").select_related("agegroup", "cosmeticsserie",
"cosmeticsserie__cosmeticsbrand").get(id = cosmetics_id)
        except Cosmetics.DoesNotExist:
            cosmetics = None
        cosmetics.photo =
CosmeticsgoodPhoto.objects.filter(cosmeticsgood = cosmetics)
        cosmetics.skin =
Cosmeticsgoodskin.objects.filter(cosmeticsgood = cosmetics)
        cosmetics.volumes =
CosmeticsgoodVolume.objects.filter(cosmeticsgood = cosmetics)
        orders_with_cosmetics =
SoftAccountOrderContent.objects.filter(cosmeticsgood =
cosmetics)
        orders_with_cosmetics =
orders_with_cosmetics.values_list("softaccountorder__id")
        cosmetics_user = request.user
        if not cosmetics_user.is_staff:
            return render(request,
"ucosmetics/cosorders.html", {"orders": None})
        cosmetics_orders =
SoftAccountOrder.objects.select_related("softaccount",
"softaccount__user").filter(id__in = orders_with_cosmetics)
        from datetime import datetime
        cosmetics_datel = request.GET.get('dp1')
        if cosmetics_datel:
            cosmetics_datel=
datetime.strptime(cosmetics_datel, '%d.%m.%y %H:%M')
            cosmetics_orders =
cosmetics_orders.filter(creation__gte = cosmetics_datel)
            cosmetics_date2 = request.GET.get('dp2')
            if cosmetics_date2:

```

```

        cosmetics_date2=
datetime.strptime(cosmetics_date2, '%d.%m.%y %H:%M')
        cosmetics_orders =
cosmetics_orders.filter(creation__lte = cosmetics_date2)
        cosmetics_statuses = request.GET.get('stat')
        if cosmetics_statuses:
            cosmetics_statuses =
cosmetics_statuses.split(',')
            cosmetics_statuses_criterion = []
            for cosmetics_status in cosmetics_statuses:
                cosmetics_status = int(cosmetics_status)

cosmetics_statuses_criterion.append(cosmetics_status)
            cosmetics_orders =
cosmetics_orders.filter(status__in =
cosmetics_statuses_criterion)
            cosmetics_ord = request.GET.get('order')
            if cosmetics_ord:
                cosmetics_ord = int(cosmetics_ord)
            else:
                cosmetics_ord = 0
            if cosmetics_ord == 0:
                cosmetics_orders = cosmetics_orders.order_by("-
creation")
            elif cosmetics_ord == 1:
                cosmetics_orders =
cosmetics_orders.order_by("creation")
            elif cosmetics_ord == 2:
                cosmetics_orders = cosmetics_orders.order_by("-
price")
            elif cosmetics_ord == 3:
                cosmetics_orders =
cosmetics_orders.order_by("price")
            return render(request, "ucosmetics/cosorders.html",
{"orders": cosmetics_orders })

def shop_user_orders_view(request, user_login = None):
    user_to_get = None
    if request.user.is_authenticated:
        cosmetics_user = request.user
        if cosmetics_user.is_staff:
            user_to_get =
SoftAccount.objects.get(user__username = user_login)
        else:
            user_to_get = cosmetics_account
    if not request.user.is_authenticated:
        return render(request, "ucosmetics/usorders.html",
{"orders": None})
    else:
        cosmetics_orders =
SoftAccountOrder.objects.select_related("softaccount",
"softaccount__user").filter(softaccount = user_to_get)
        from datetime import datetime

```

```

        cosmetics_date1 = request.GET.get('dp1')
        if cosmetics_date1:
            cosmetics_date1 =
datetime.strptime(cosmetics_date1, '%d.%m.%y %H:%M')
            cosmetics_orders =
cosmetics_orders.filter(creation__gte = cosmetics_date1)
            cosmetics_date2 = request.GET.get('dp2')
            if cosmetics_date2:
                cosmetics_date2 =
datetime.strptime(cosmetics_date2, '%d.%m.%y %H:%M')
                cosmetics_orders =
cosmetics_orders.filter(creation__lte = cosmetics_date2)
            cosmetics_statuses = request.GET.get('stat')
            if cosmetics_statuses:
                cosmetics_statuses =
cosmetics_statuses.split(',')
                cosmetics_statuses_criterion = []
                for cosmetics_status in cosmetics_statuses:
                    cosmetics_status = int(cosmetics_status)

cosmetics_statuses_criterion.append(cosmetics_status)
                cosmetics_orders =
cosmetics_orders.filter(status__in =
cosmetics_statuses_criterion)
                cosmetics_ord = request.GET.get('order')
                if cosmetics_ord:
                    cosmetics_ord = int(cosmetics_ord)
                else:
                    cosmetics_ord = 0
                if cosmetics_ord == 0:
                    cosmetics_orders = cosmetics_orders.order_by("-
creation")
                elif cosmetics_ord == 1:
                    cosmetics_orders =
cosmetics_orders.order_by("creation")
                elif cosmetics_ord == 2:
                    cosmetics_orders = cosmetics_orders.order_by("-
price")
                elif cosmetics_ord == 3:
                    cosmetics_orders =
cosmetics_orders.order_by("price")
                return render(request, "ucosmetics/usorders.html",
{"orders": cosmetics_orders })

```

**ДОДАТОК В**  
**Слайди презентації**



Рисунок В.1 – Початковий слайд

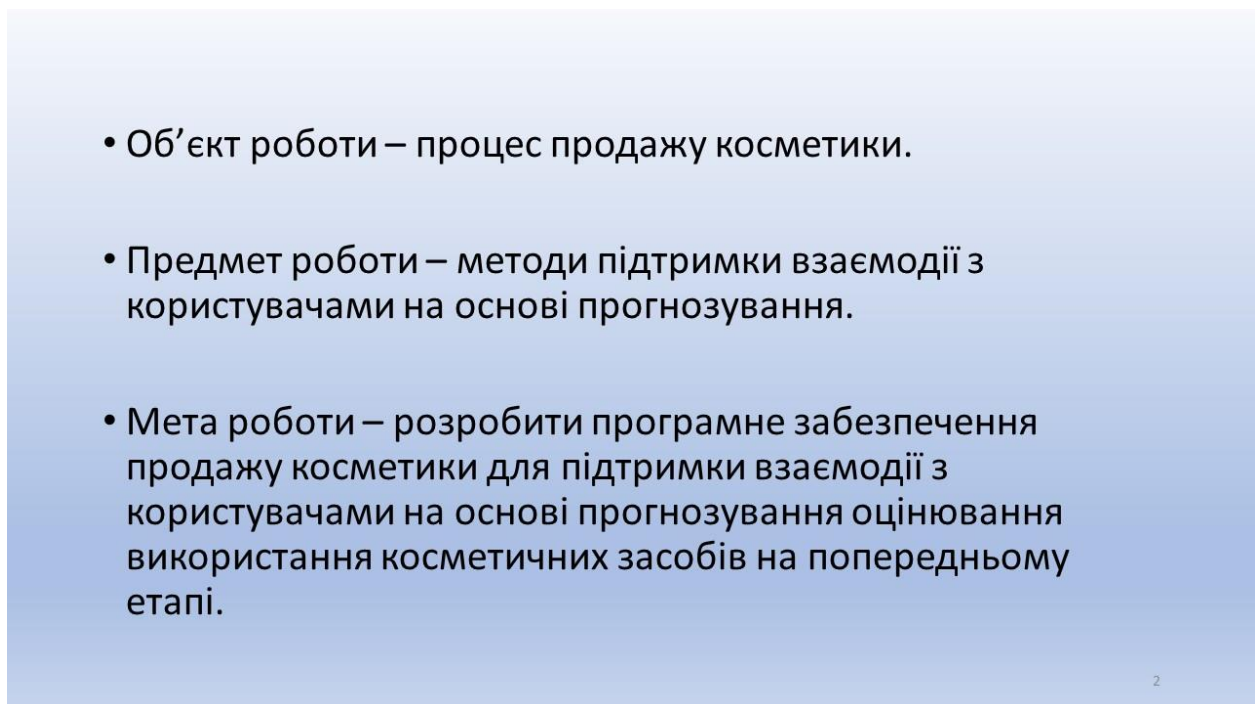


Рисунок В.2 – Об'єкт, предмет, мета роботи

## Завдання роботи

- виконати аналіз програм продажу косметики;
- розробити метод прогнозування оцінки косметичних засобів покупцями;
- виконати проектування програми продажу косметики;
- реалізувати програму продажу косметики;
- виконати експериментальне дослідження прогнозування оцінки косметичних засобів покупцями.

3

Рисунок В.3 – Завдання роботи

Порівняння аналогів розробки вебзастосунку продажу косметики

| Показник                                              | Makeup                       | Brocard                                                          | Розробка                     |
|-------------------------------------------------------|------------------------------|------------------------------------------------------------------|------------------------------|
| Пошук косметичних засобів                             | З великою кількістю фільтрів | Тільки за назвою, відбір за фільтрами доступний для груп засобів | З великою кількістю фільтрів |
| Кількість характеристик косметичних засобів           | Ширша                        | Базова                                                           | Ширша                        |
| Засоби особистого кабінету користувача                | +                            | +                                                                | +                            |
| Створення відгуків                                    | +                            | +                                                                | +                            |
| Прогнозування оцінки покупців для косметичного засобу | –                            | –                                                                | +                            |

4

Рисунок В.4 – Порівняння аналогів розробки вебзастосунку продажу косметики

Метод прогнозування оцінювання використання косметичних засобів покупцями

- формування вибірки даних, що містить екземпляри, кожен з яких представляє косметичний засіб з базовим набором ознак;
- перетворення переліку інгредієнтів зі складу косметичного засобу на вектор ознак;
- перетворення назви косметичного засобу на вектор ознак за допомогою моделі Word2Vec;
- створення моделі, яка має бути забезпечена у вигляді ансамблю дерев рішень на основі застосування методу Random Forest, та її навчання на основі сформованої навчальної вибірки;
- підготовка нових екземплярів даних та прогнозування середньої оцінки на основі навченої моделі.

5

Рисунок В.5 – Метод прогнозування оцінювання використання косметичних засобів покупцями

Вибір вебфреймворку для розробки програми магазину косметики

|                                                                 | Django                               | Drupal                    |
|-----------------------------------------------------------------|--------------------------------------|---------------------------|
| Забезпечення високого рівня продуктивності для великих проєктів | Краще                                | Гірше                     |
| Спосіб розширення функціональності                              | Інтеграція коду до раніше створеного | Написання окремого модуля |
| Мова, якою ведеться розробка                                    | Python                               | PHP                       |
| Рівень безпеки базового рішення                                 | Вищий                                | Нижчий                    |

6

Рисунок В.6 – Вибір вебфреймворку для розробки програми магазину косметики



Рисунок В.7 – Діаграми прецедентів для користувачів вебзастосунку магазину косметики

## Схема бази даних

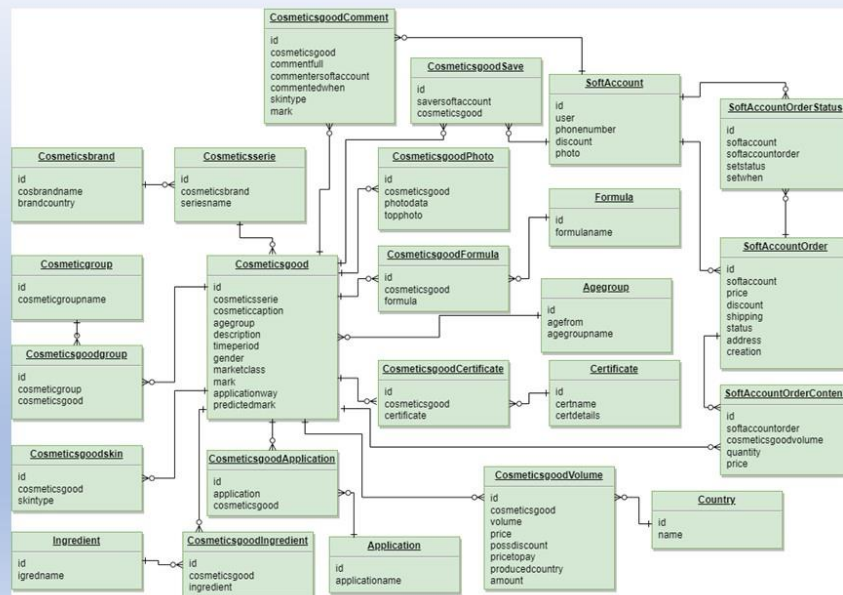


Рисунок В.8 – Схема бази даних

## Схема функціонування програми магазину продажу косметики

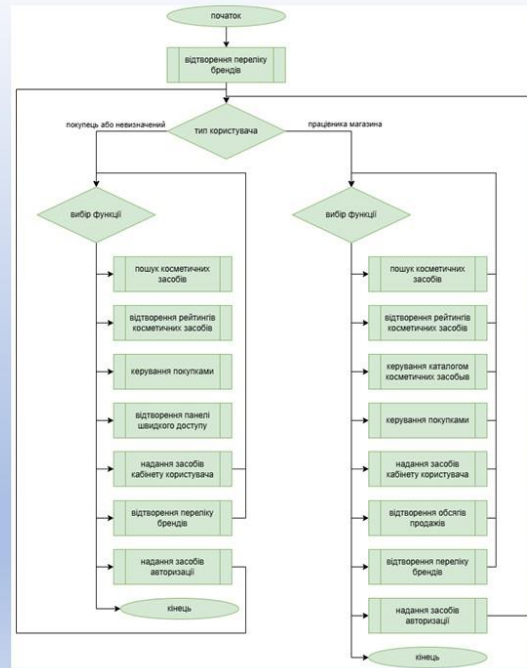
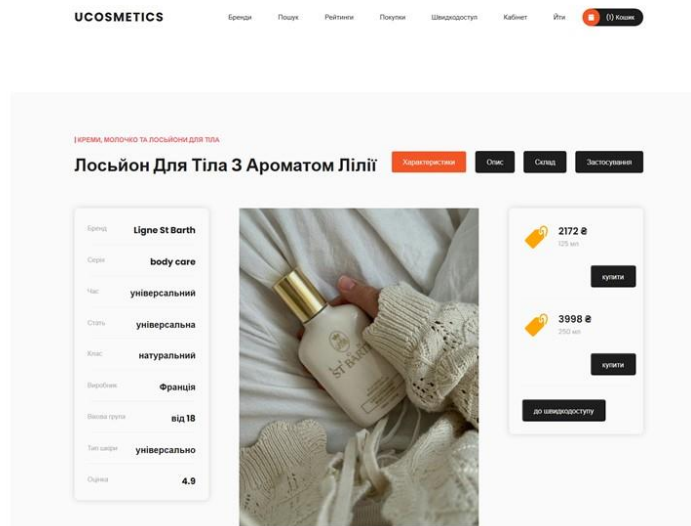


Рисунок В.9 – Схема функціонування програми магазину продажу косметики

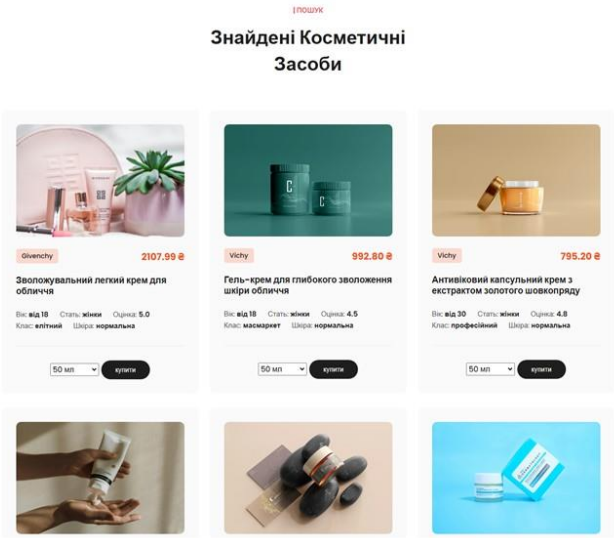
## Сторінка косметичного засобу для покупця в програмі



10

Рисунок В.10 – Сторінка косметичного засобу для покупця в програмі

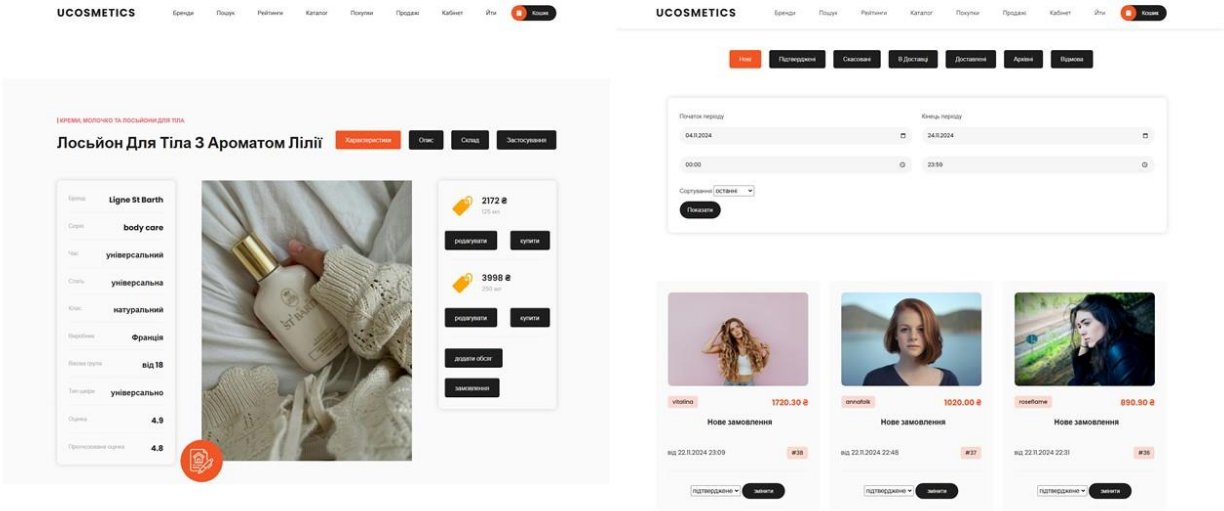
Результати пошуку косметичних засобів покупцем



11

Рисунок В.11 – Результати пошуку косметичних засобів покупцем

Робота з програмою працівника магазину продажу косметики



12

Рисунок В.12 – Робота з програмою працівника магазину продажу косметики

Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів представлення назви косметичного засобу

| Особливості<br>вхідних ознак                               | MAE        | RMSE       | MSE        |
|------------------------------------------------------------|------------|------------|------------|
| Базовий набір                                              | 0,38115048 | 0,64023456 | 0,40990029 |
| Базовий набір<br>та всі складові<br>назви                  | 0,3644223  | 0,62463021 | 0,3901629  |
| Базовий набір<br>та 1 ознака для<br>представлення<br>назви | 0,37717681 | 0,63950121 | 0,4089618  |

13

Рисунок В.13 – Результати прогнозування оцінювання використання косметичних засобів покупцями на основі різних варіантів представлення назви косметичного засобу

Результати прогнозування оцінювання використання косметичних засобів покупцями

| Особливості<br>вхідних ознак                      | MAE        | RMSE       | MSE        |
|---------------------------------------------------|------------|------------|------------|
| Random Forest (50<br>дерев)                       | 0,3629219  | 0,64312657 | 0,41361179 |
| Random Forest<br>(100 дерев)                      | 0,36191579 | 0,64227822 | 0,41252131 |
| Random Forest<br>(500 дерев)                      | 0,35676543 | 0,63566566 | 0,40407083 |
| Дерево рішень                                     | 0,47419355 | 0,82145467 | 0,67478778 |
| Дерево рішень<br>(базовий набір<br>вхідних ознак) | 0,4857446  | 0,83649795 | 0,69972883 |

14

Рисунок В.14 – Результати прогнозування оцінювання використання косметичних засобів покупцями

## Висновки

- Розроблена програма призначена для підтримки взаємодії з користувачами магазину косметики через надання їм інформації про наявні косметичні засоби в магазині, підбір потрібних засобів та замовлення цих засобів.
- Наукова новизна роботи полягає в методі прогнозування оцінювання використання косметичних засобів покупцями, що забезпечує формування ознак, які представляють назву косметичного засобу, на основі моделі Word2Vec, ознак, які представляють склад косметичного засобу, створення моделі ансамблю дерев рішень, що дозволяє прогнозувати середню оцінку, яку отримає косметичний засіб від покупців.
- Результати експериментального дослідження включають результати дослідження використання різних варіантів представлення переліку інгредієнтів косметичного засобу, його назви, набору базових ознак, формування підсумкового набору вхідних ознак, порівняння моделей на основі індивідуальних дерев рішень та ансамблів на основі методу Random Forest, що підтвердило практичну придатність запропонованого методу.

15

Рисунок В.15 – Висновки