

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ІНЖЕНЕРНИЙ НАВЧАЛЬНО – НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

**КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Кваліфікаційна робота
перший (бакалаврський)

(рівень вищої освіти)

на тему **Рекомендаційна система продажу товарів на основі колаборативної
фільтрації**

Виконав: студент 4 курсу, групи 6.1210 – пзс
спеціальності 121 Інженерія програмного
забезпечення

(код і назва спеціальності)

освітньої програми Програмне забезпечення
систем

(код і назва освітньої програми)

В. В. Юр'єв

(ініціали та прізвище)

Керівник доцент кафедри ЕІС та ПЗ

А.І. Безверхий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дискус»

Р.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2024

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

Кафедра Електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти перший(бакалаврський)
Спеціальність 121 Інженерія програмного забезпечення
(код та назва)
Освітня програма Програмне забезпечення систем
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Тетяна КРИТСЬКА
“ 01 ” березня 2024 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Юрьєву Володимиру Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Рекомендаційна система продажу товарів на основі
колаборативної фільтрації

керівник роботи Безверхий Анатолій Ігорович, доцент
(прізвище ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від 26.12.2023 № 2215-с

2. Строк подання студентом кваліфікаційної роботи _____
3. Вихідні дані кваліфікаційної роботи бакалавра
 - комплект нормативних документів ;
 - технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - огляд та збір літератури стосовно теми кваліфікаційної роботи;

- огляд та аналіз існуючих рішень та аналогів;
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) слайдів презентації

6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		завдання прийняв

7. Дата видачі завдання 01.03.2024

КАЛЕНДАРНИЙ ПЛАН

з/п	№	Назва етапів бакалаврської роботи	Строк виконання етапів бакалаврської роботи	Примітка
1		Аналіз існуючих рішень	03.01 – 25.01.24	виконано
2		Постановка завдання	26.01 – 29.01.24	виконано
3		Огляд мов програмування	01.01 – 07.02.24	виконано
4		Аналіз сучасних фреймворків	08.02 – 22.02.24	виконано
5		Огляд архітектурних патернів	23.02 – 28.02.24	виконано
6		Огляд управлінь залежностями	01.03 – 07.03.24	виконано
7		Огляд СКБД	08.03 – 13.03.24	виконано
8		Огляд API	14.03 – 18.03.24	виконано
9		Розробка архітектури системи	19.03 – 26.03.24	виконано
10		Розробка фронтенд-частини	27.03 – 23.04.24	виконано
11		Розробка бекенд-частини	24.04 – 10.05.24	виконано
12		Тестування та оптимізація	11.05 – 29.05.24	виконано

Студент _____ Юрьєв В.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Безверхий А.І.
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ Скрипник І.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок — 67

Рисунків — 19

Джерел — 19

Юрьєв В.В. Рекомендаційна система продажу товарів на основі колаборативної фільтрації: кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2024. 67 с.

Колаборативна фільтрація є однією з провідних технологій, що дозволяє створювати персоналізовані рекомендації, враховуючи вподобання та поведінку користувачів. Це сприяє підвищенню задоволеності клієнтів, збільшенню обсягу продажів та лояльності до бренду.

Метою кваліфікаційної роботи є розробка ефективної рекомендаційної системи продажу товарів на основі колаборативної фільтрації, яка здатна надавати користувачам персоналізовані пропозиції, збільшуючи їх задоволеність та покращуючи показники продажів.

Практичне значення одержаних результатів полягає у створенні рекомендаційної системи, яка дозволяє надавати користувачам персоналізовані пропозиції, що сприяє підвищенню їх задоволеності та збільшенню обсягів продажів. Результати дослідження можуть бути використані для подальшого вдосконалення рекомендаційних систем у сфері електронної комерції.

Ключові слова: Рекомендаційна система, Колаборативна фільтрація, Електронна комерція, Алгоритм, Персоналізація, Користувач

ABSTRACT

Pages — 67

Drawings — 19

Source — 19

Yuriev V.V. Recommendation system for the sale of goods based on collaborative filtering: qualification work of the bachelor of specialty 121 "Software engineering" / Science. manager A. I. Bezverkhy. Zaporizhzhia : ZNU, 2024. 67 p.

Collaborative filtering is one of the leading technologies that allows you to create personalized recommendations based on user preferences and behavior. This helps increase customer satisfaction, increase sales and brand loyalty.

The goal of the qualification work is to develop an effective recommender system for the sale of goods based on collaborative filtering, which is able to provide users with personalized offers, increasing their satisfaction and improving sales performance.

The practical significance of the obtained results lies in the creation of a recommender system that allows providing users with personalized offers, which contributes to increasing their satisfaction and increasing sales volumes. The results of the study can be used for further improvement of recommender systems in the field of electronic commerce.

Keywords: Recommender system, Collaborative filtering, E-commerce, Algorithm, Personalization, User.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ З РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНИХ СИСТЕМ.....	11
1.1 Опис проблеми	11
1.1.1 User-Based Collaborative Filtering	12
1.1.2 Item-Based Collaborative Filtering	13
1.2 Огляд літературних джерел	15
1.3 Аналіз програмних продуктів-аналогів	16
1.4 Постановка завдання.....	20
РОЗДІЛ 2 АЛГОРИТМИ І ЗАСОБИ РОЗРОБКИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ	22
2.1 Алгоритм колаборативної фільтрації на основі item-based підходу.....	22
2.2 Аналіз сучасних технологій для розробки рекомендаційних систем.....	24
2.3 Аналіз вимог до користувачів рекомендаційної системи.....	26
2.3.1 Кінцеві споживачі (покупці).....	26
2.3.2 Адміністратори системи.....	27
2.3.3 Загальні вимоги до всіх користувачів системи.....	27
РОЗДІЛ 3 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ ТОВАРІВ.....	28
3.1 Опис предметної області	28
3.2 Вимоги до системи.....	29
3.2.1 Функціональні вимоги.....	29
3.2.2 Нефункціональні вимоги.....	30
3.3 Архітектура системи.....	31
3.3.1 База даних	31
3.3.2 API	38
3.3.3 Клієнтська частина	45

3.3.4 Реалізація рекомендаційної системи на основі item-based	57
3.4 Висновки з 3 розділу	64
ВИСНОВКИ	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66

ВСТУП

Актуальність теми

Рекомендаційні системи відіграють важливу роль у сучасному торговельному середовищі. Зі зростанням кількості доступних товарів та послуг, споживачі стикаються з проблемою вибору, що призводить до потреби в ефективних інструментах для прийняття рішень. Колаборативна фільтрація є однією з провідних технологій, що дозволяє створювати персоналізовані рекомендації, враховуючи вподобання та поведінку користувачів. Це сприяє підвищенню задоволеності клієнтів, збільшенню обсягу продажів та лояльності до бренду.

Технологічний прогрес та широкий доступ до Інтернету відкривають нові можливості для впровадження рекомендаційних систем у різні галузі торгівлі. Вони дозволяють автоматизувати процес аналізу споживацьких даних та надання персоналізованих рекомендацій у режимі реального часу. Це не лише спрощує процес покупки для користувачів, а й допомагає бізнесам краще розуміти своїх клієнтів та адаптувати свої маркетингові стратегії відповідно до їх потреб.

Мета дослідження

Метою дослідження є розробка ефективної рекомендаційної системи продажу товарів на основі колаборативної фільтрації, яка здатна надавати користувачам персоналізовані пропозиції, збільшуючи їх задоволеність та покращуючи показники продажів.

Завдання дослідження

- Розглянути основні принципи та технології колаборативної фільтрації.

- Проаналізувати існуючі рішення у сфері рекомендаційних систем.
- Розробити алгоритм колаборативної фільтрації для рекомендаційної системи.
- Реалізувати прототип рекомендаційної системи та провести його тестування.
- Оцінити ефективність розробленої системи та визначити шляхи її подальшого вдосконалення.

Об'єкт дослідження

Об'єктом дослідження є процес розробки рекомендаційної системи на основі колаборативної фільтрації для сфери електронної комерції.

Предмет дослідження

Предметом дослідження є методи та технології колаборативної фільтрації, які використовуються для створення персоналізованих рекомендацій у системах електронної комерції.

Методи дослідження

У процесі дослідження використовувалися наступні методи:

- Теоретичний аналіз літератури та існуючих рішень у сфері рекомендаційних систем.
- Порівняльний аналіз методів колаборативної фільтрації.
- Розробка та моделювання алгоритму колаборативної фільтрації.
- Експериментальне тестування розробленої системи.

Практичне значення одержаних результатів

Практичне значення одержаних результатів полягає у створенні рекомендаційної системи, яка дозволяє надавати користувачам персоналізовані пропозиції, що сприяє підвищенню їх задоволеності та збільшенню обсягів продажів. Результати дослідження можуть бути використані для подальшого вдосконалення рекомендаційних систем у сфері електронної комерції.

Глосарій

Рекомендаційна система – інформаційна система, яка надає персоналізовані рекомендації користувачам на основі аналізу їх вподобань та поведінки.

Колаборативна фільтрація – метод рекомендацій, що використовує інформацію про вподобання та поведінку користувачів або характеристик товарів для надання індивідуальних рекомендацій.

Електронна комерція – процес купівлі та продажу товарів і послуг через Інтернет.

Алгоритм – послідовність дій або інструкцій, які виконуються для досягнення певного результату.

Персоналізація – процес адаптації продуктів або послуг під індивідуальні потреби та вподобання користувача.

Користувач – особа, яка використовує рекомендаційну систему для отримання персоналізованих пропозицій.

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ З РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Опис проблеми

У сучасному світі електронна комерція швидко розвивається, і кількість доступних товарів та послуг на ринку зростає з кожним днем. Це створює нові виклики для користувачів, які стикаються з проблемою вибору серед величезної кількості варіантів. Відповідно, проблема полягає у тому, як надати користувачам персоналізовані рекомендації, щоб полегшити процес прийняття рішень та покращити їх задоволеність покупками.

Основна проблема полягає в ефективному аналізі великих обсягів даних про поведінку та вподобання користувачів. Колаборативна фільтрація (Collaborative Filtering) є одним з найбільш поширених підходів для розв'язання цієї проблеми. Вона дозволяє створювати персоналізовані рекомендації, враховуючи схожість між користувачами (user-based) або товарами (item-based).

Проте колаборативна фільтрація має свої обмеження та проблеми. Однією з найбільших проблем є проблема "холодного старту", яка виникає, коли новий користувач або товар додається до системи, але ще не має достатньо даних для створення рекомендацій.

Іншою важливою проблемою є масштабованість системи. Зі зростанням кількості користувачів та товарів обчислювальні витрати на обробку даних значно збільшуються. Це призводить до необхідності розробки ефективних алгоритмів, які можуть працювати з великими обсягами даних в режимі реального часу.

Цей опис проблеми зосереджується на необхідності розробки ефективної рекомендаційної системи, яка здатна вирішувати вищезазначені виклики та

надавати користувачам персоналізовані пропозиції, що сприяють підвищенню їх задоволеності та збільшенню обсягів продажів.

1.1.1 User-Based Collaborative Filtering

User-based підхід полягає у знаходженні схожих користувачів на основі їхніх оцінок товарів і створенні рекомендацій на основі цих схожостей. Основні характеристики цього підходу:

- User-based підхід може бути менш масштабованим, особливо при великій кількості користувачів. Це пов'язано з необхідністю частого перерахунку схожостей між користувачами.
- User-based підхід може надавати хороші рекомендації, але це залежить від наявності великої кількості схожих користувачів. У випадку рідкісних або нових товарів цей підхід може бути менш ефективним.
- Реалізація user-based підходу може бути складнішою через необхідність динамічного обчислення схожостей між користувачами, що може збільшувати обчислювальні витрати.

Етапи реалізації user-based рекомендаційної системи:

- У матриці строки – це користувачі, а стовпці - товари. Кожна комірка містить оцінку, яку користувач надав конкретному товару (див. Рис.1).
- Використовуються метрики, такі як косинусна схожість або коефіцієнт кореляції Пірсона для обчислення схожості між користувачами на основі їхніх оцінок товарів.
- Визначення найбільш схожих користувачів для кожного користувача.

- Використання оцінок «сусідів» для генерації рекомендацій для користувача.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	4
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	4

Рисунок 1 – User-based матриця

1.1.2 Item-Based Collaborative Filtering

Item-based підхід зосереджується на знаходженні схожих товарів на основі їхніх оцінок користувачами. Основні характеристики цього підходу:

- Item-based підхід зазвичай вважається більш масштабованим, особливо в системах з великою кількістю користувачів та меншою кількістю товарів. Схожість між товарами можна обчислити заздалегідь і зберігати, що прискорює процес рекомендацій.
- Item-based підхід часто надає більш точні рекомендації, оскільки поведінка користувачів з конкретними товарами є більш

передбачуваною та стабільною. Це забезпечує високу точність рекомендацій навіть для нових користувачів.

- Реалізація item-based підходу часто є простішою через можливість попереднього обчислення схожостей між товарами і зберігання цих даних.

Етапи реалізації item-based рекомендаційної системи

- У матриці строки – це товари, а стовпці - користувачі. Кожна комірка містить оцінку, яку користувач надав конкретному товару (див. Рис.2).
- Використовуються метрики, такі як косинусна схожість або коефіцієнт кореляції Пірсона для обчислення схожості між товарами на основі оцінок користувачів.
- Визначення найбільш схожих товарів для кожного товару.
- Використання оцінок користувачів для схожих товарів для генерації рекомендацій.

	User 1	User 2	User 3	User 4
Item 1	1	1	0	0
Item 2	1	1	1	1
Item 3	0	1	0	0
Item 4	0	0	1	1
Item 5	1	1	1	1

Рисунок 2 – Item-based матриця

З огляду на наведені характеристики, було обрано item-based підхід для реалізації рекомендаційної системи. Цей підхід забезпечує кращу масштабованість та швидкість обробки даних, що є критично важливим при роботі з великими обсягами даних в режимі реального часу. Крім того, item-based підхід дозволяє надавати більш точні та стабільні рекомендації, що підвищує задоволеність користувачів та сприяє збільшенню обсягів продажів.

Таким чином, item-based підхід є оптимальним вибором рекомендаційної системи, оскільки він відповідає основним вимогам щодо ефективного аналізу даних та надання високоякісних рекомендацій користувачам.

1.2 Огляд літературних джерел

Під час підготовки кваліфікаційної роботи з теми "Рекомендаційна система на основі колаборативної фільтрації" було проведено детальний огляд літературних джерел, щоб розширити розуміння теми та виявити існуючі розробки у галузі рекомендаційних систем.

Одним з основних методів створення персоналізованих рекомендацій є колаборативна фільтрація. Колаборативна фільтрація дозволяє враховувати схожість між користувачами (user-based) або товарами (item-based) для генерації рекомендацій. У кваліфікаційній роботі бакалавра було реалізовано item-based підхід, використовуючи метрику Косинусна відстань (Cosine Distance) для визначення схожості між товарами.

Під час огляду літературних джерел були розглянуті основні принципи та технології, що застосовуються в колаборативній фільтрації. Важливими аспектами є методи обчислення схожості, такі як косинусна відстань, які дозволяють точно визначати, наскільки схожі товари між собою на основі оцінок користувачів.

Також було приділено увагу проблемам, які виникають при використанні колаборативної фільтрації. Однією з таких проблем є "холодний старт", яка виникає, коли новий користувач або товар додається до системи, але ще не має достатньо даних для створення рекомендацій. Ця проблема є актуальною для багатьох рекомендаційних систем і вимагає розробки спеціальних алгоритмів для її вирішення.

Іншою важливою проблемою є масштабованість системи. Зі збільшенням кількості користувачів та товарів обчислювальні витрати значно зростають. Це вимагає використання ефективних алгоритмів та оптимізації процесів для забезпечення швидкої обробки даних та надання рекомендацій у режимі реального часу.

Літературні джерела також пропонують різні підходи до покращення ефективності рекомендаційних систем. Наприклад, деякі дослідження зосереджуються на використанні гібридних методів, які поєднують колаборативну фільтрацію з іншими підходами, такими як контентна фільтрація. Це дозволяє подолати деякі обмеження колаборативної фільтрації та покращити точність рекомендацій.

Аналіз літературних джерел показує, що колаборативна фільтрація активно використовується у різних сферах, включаючи електронну комерцію, потокові медіа-сервіси та соціальні мережі. Це підкреслює важливість досліджень у цій галузі та можливість подальшого вдосконалення методів і технологій для створення персоналізованих рекомендацій.

1.3 Аналіз програмних продуктів-аналогів

Проблематика надання персоналізованих рекомендацій є надзвичайно актуальною у сучасному світі електронної комерції. З розвитком технологій та зростанням обсягів даних, на ринку з'являються різноманітні рішення, що

дозволяють користувачам отримувати індивідуальні рекомендації на основі їхніх вподобань та поведінки. Розглянемо деякі з них.

Amazon (див. Рис.3) використовує один із найбільш відомих і успішних алгоритмів колаборативної фільтрації для надання персоналізованих рекомендацій. Система аналізує історію покупок, переглядів та оцінок користувачів, що дозволяє пропонувати товари, які можуть зацікавити конкретного користувача. Цей підхід довів свою ефективність і дозволяє Amazon значно збільшувати обсяги продажів завдяки точним рекомендаціям.

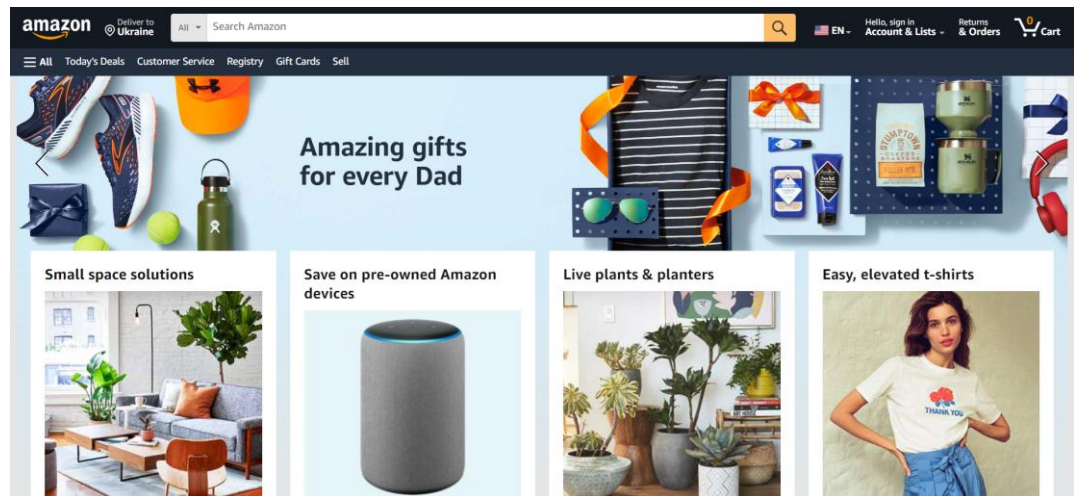


Рисунок 3 – Головна сторінка Amazon

Netflix (див. Рис.4) також використовує колаборативну фільтрацію для рекомендації фільмів та серіалів. Їх алгоритм враховує не лише оцінки та перегляди користувачів, але й подібність між користувачами (user-based) та подібність між контентом (item-based). Це дозволяє надавати високоякісні рекомендації, що сприяє утриманню аудиторії та підвищенню задоволеності користувачів.

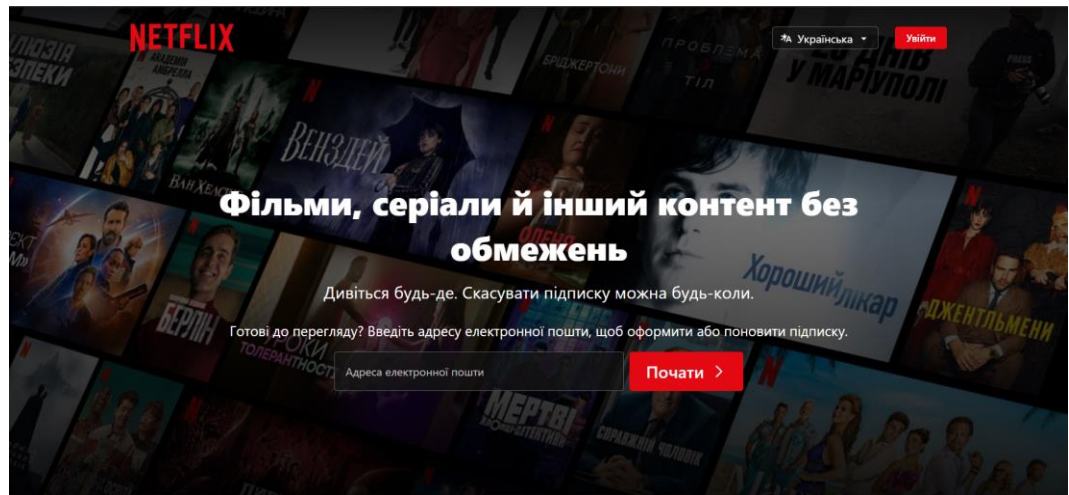


Рисунок 4 – Головна сторінка Netflix

Spotify (див. Рис.5) застосовує колаборативну фільтрацію для створення персоналізованих плейлистів та рекомендацій музичних треків. Використовуючи дані про прослуховування та вподобання користувачів, Spotify здатен формувати плейлисти, які відповідають музичним смакам кожного користувача. Це допомагає утримувати користувачів та збільшувати тривалість їхньої взаємодії з сервісом.

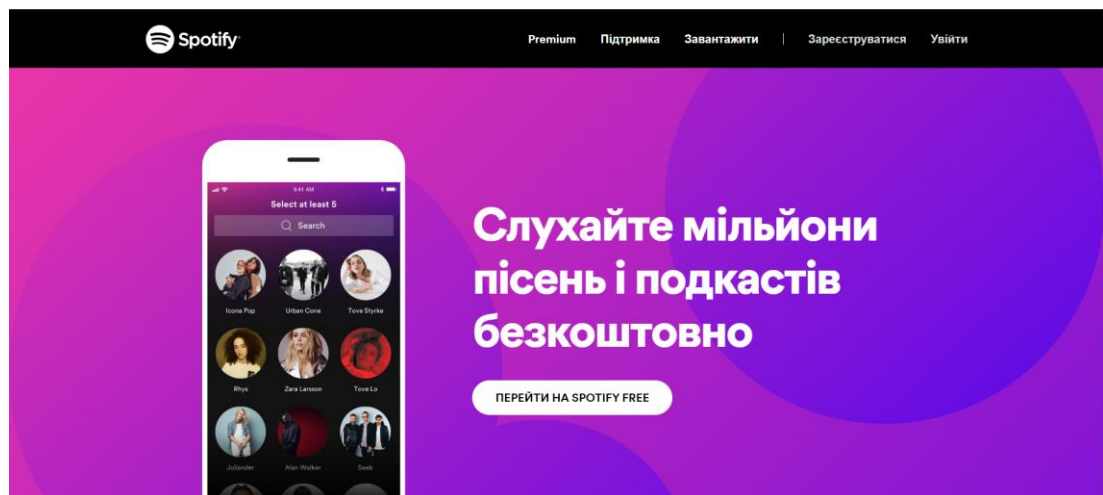


Рисунок 5 – Головна сторінка Spotify

YouTube (див. Рис.6) також використовує колаборативну фільтрацію для рекомендації відео. Аналізуючи перегляди, оцінки та інші дії користувачів, система пропонує відео, які можуть зацікавити конкретного користувача. Це сприяє збільшенню часу, проведеного на платформі, та підвищенню рівня взаємодії користувачів з контентом.

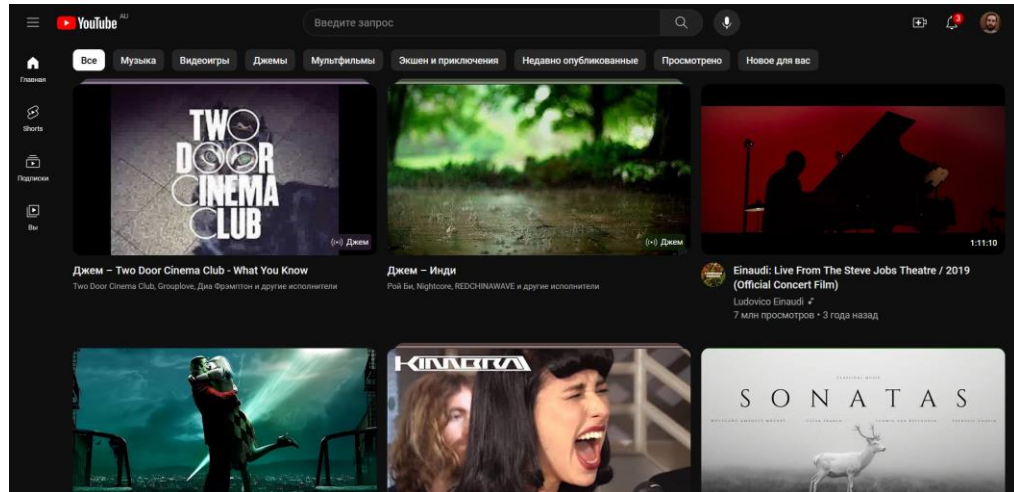


Рисунок 6 – Головна сторінка YouTube

MovieLens (див. Рис.7) – це дослідницький проект, що також використовує колаборативну фільтрацію для надання рекомендацій фільмів. Він дозволяє користувачам оцінювати фільми та отримувати персоналізовані рекомендації на основі їхніх оцінок та вподобань. MovieLens є важливим інструментом для досліджень у сфері рекомендаційних систем та аналізу ефективності різних алгоритмів.

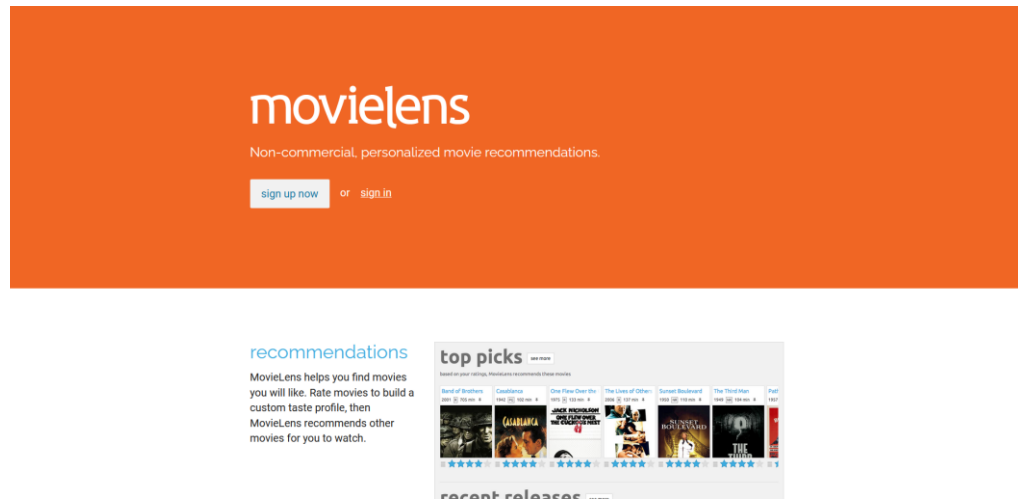


Рисунок 7 – Головна сторінка MovieLens

Аналіз існуючих рішень дозволяє зробити висновок про високу ефективність колаборативної фільтрації у різних сферах електронної комерції та медіа. Ці системи довели свою здатність підвищувати задоволеність користувачів та збільшувати обсяги продажів або взаємодії з контентом. Розробка рекомендаційної системи для продажу товарів на основі колаборативної фільтрації може стати перспективним напрямком, що дозволить надавати користувачам персоналізовані пропозиції та сприяти зростанню бізнесу, збільшуючи їх задоволеність та покращуючи показники продажів.

1.4 Постановка завдання

Мета кваліфікаційної роботи – це розробка ефективної рекомендаційної системи для продажу товарів на основі колаборативної фільтрації. Ця система повинна надавати користувачам персоналізовані пропозиції, підвищуючи їх задоволеність та покращуючи показники продажів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Аналіз основних принципів та технологій колаборативної фільтрації:

- Вивчення існуючих методів та алгоритмів колаборативної фільтрації.
 - Аналіз переваг та недоліків різних підходів до колаборативної фільтрації.
2. Розробка алгоритму колаборативної фільтрації:
 - Вибір підходу для колаборативної фільтрації (item-based).
 - Реалізація алгоритму колаборативної фільтрації з використанням метрики Косинусна відстань.
 3. Проектування та створення бази даних для зберігання інформації про користувачів, товари та оцінки:
 - Вибір відповідної СУБД (MS SQL Server).
 - Розробка структури бази даних.
 - Створення таблиць для зберігання даних про користувачів, товари, категорії, виробників та оцінки.
 4. Реалізація прототипу рекомендаційної системи:
 - Розробка серверної частини системи для обробки запитів та надання рекомендацій (на основі .NET Core API для передачі даних з бази даних в Unity3D).
 - Створення інтерфейсу для взаємодії з користувачами.
 5. Тестування та оцінка ефективності розробленої системи:
 - Проведення тестування прототипу на даних.

Поставлене завдання включає розробку надійної та зручної рекомендаційної системи, яка допоможе користувачам отримувати персоналізовані пропозиції товарів. Система має забезпечувати швидкий та точний доступ до рекомендацій, що сприятиме підвищенню задоволеності користувачів та збільшенню обсягів продажів.

РОЗДІЛ 2 АЛГОРИТМИ І ЗАСОБИ РОЗРОБКИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

2.1 Алгоритм колаборативної фільтрації на основі item-based підходу

Колаборативна фільтрація – це техніка рекомендаційних систем, яка використовує минулі дані про переваги користувачів для прогнозування їхніх майбутніх інтересів. Як вже було сказано, існує два основні підходи до колаборативної фільтрації: на основі користувачів (user-based) і на основі елементів (item-based). В даній роботі використовується item-based підхід.

Item-based колаборативна фільтрація передбачає виявлення взаємозв'язків між предметами на основі оцінок користувачів. Вона надає рекомендації, ґрунтуючись на предметах, які були оцінені подібно до предметів, якими зацікавлений користувач. Проте, ці базові методи мають свої недоліки, як-от проблема "нового користувача" або "нового товару".

Основні етапи алгоритму item-based колаборативної фільтрації включають:

1. Збір та підготовка даних:

Спочатку збираються дані про оцінки або взаємодії користувачів з предметами. Наприклад, у вигляді матриці користувач-елемент, де кожен елемент представляє оцінку, яку користувач дав конкретному предмету. Ці дані очищуються та готуються до подальшого аналізу, видаляються шуми та заповнюються відсутні значення.

2. Обчислення подібності між елементами:

Використовується метрика подібності, така як косинусна подібність, для визначення ступеня схожості між різними предметами. Косинусна подібність дозволяє визначити, наскільки схожі два предмети на основі оцінок користувачів.

Косинусна подібність (див. Рис.8) вимірює кут між двома векторами в просторі характеристик, не враховуючи їхню величину. Це особливо корисно для

рекомендаційних систем, де важливо порівнювати патерни оцінок користувачів, а не самі оцінки.

$$\text{sim}(i, j) = \frac{\sum_{u \in U} R_{ui} \times R_{uj}}{\sqrt{\sum_{u \in U} (R_{ui})^2} \times \sqrt{\sum_{u \in U} (R_{uj})^2}}$$

Рисунок 7 – Формула косинусної подібності

У контексті рекомендаційної системи, де матриця взаємодій користувачів з товарами представляється у вигляді двовимірного масиву, кожен вектор А або В відповідає рядку (предмету) в матриці.

Суть цього методу полягає в наступному:

- Для кожної пари предметів обчислюється скалярний добуток їхніх векторів оцінок. Це дозволяє визначити, наскільки сильно пов'язані ці предмети на основі оцінок користувачів.
- Довжина кожного вектора обчислюється як квадратний корінь суми квадратів його елементів. Це нормалізує вектори, дозволяючи порівнювати їхню орієнтацію, а не масштаб.
- Використовуючи скалярний добуток і довжини векторів, обчислюється косинус кута між двома векторами. Це значення нормалізується до діапазону від -1 до 1, де 1 означає ідеальну схожість, 0 – відсутність схожості, а -1 – протилежну схожість.

Результатом є міра схожості між кожною парою предметів, яка використовується для генерації рекомендацій. Наприклад, якщо користувач оцінив певний товар, система може знайти товари з високою косинусною подібністю до цього товару і рекомендувати їх користувачу.

3. Формування рекомендацій:

На основі обчисленої подібності між предметами, система формує список рекомендацій для кожного користувача. Це робиться шляхом вибору найсхожіших предметів до тих, які вже були оцінені користувачем. Таким чином, користувачу пропонуються предмети, які, ймовірно, його зацікавлять на основі схожості з тими, що він вже оцінив.

4. Покращення алгоритму:

На основі результатів оцінки, алгоритм можна покращити шляхом налаштування параметрів, використання різних методів обчислення подібності або інтеграції додаткових підходів. Це може включати факторизацію матриць, гібридні методи та інші техніки, які допомагають підвищити точність і релевантність рекомендацій.

Таким чином, item-based колаборативна фільтрація є потужним інструментом для створення персоналізованих рекомендацій на основі аналізу схожості між предметами, використовуючи дані про оцінки користувачів.

2.2 Аналіз сучасних технологій для розробки рекомендаційних систем

Розробка рекомендаційної системи є складним процесом, який включає вибір та інтеграцію різних технологій для забезпечення ефективної роботи системи. Рекомендаційна система повинна надавати користувачам персоналізовані пропозиції на основі їхньої поведінки та вподобань. Для цього проекту було обрано наступні технології:

Microsoft SQL Server: MS SQL Server використовується для створення та управління базою даних, що зберігає інформацію про користувачів, товари, категорії, виробників та оцінки. Це реляційна система управління базами даних, яка забезпечує високий рівень безпеки та надійності зберігання даних, а також підтримує складні запити і транзакції. MS SQL Server було обрано за його

надійність та продуктивність у роботі з великими обсягами даних. Він дозволяє швидко і ефективно виконувати складні запити, що є критично важливим для рекомендаційних систем.

.NET Core: Для створення серверної частини системи було використано .NET Core. Це крос-платформенний фреймворк з відкритим вихідним кодом, який забезпечує високу продуктивність і масштабованість. Використовуючи .NET Core, було розроблено API для передачі даних з бази даних у Unity3D. Це дозволяє клієнтському додатку отримувати доступ до актуальної інформації в режимі реального часу. .NET Core надає розробникам інструменти для створення масштабованих і продуктивних додатків. Його крос-платформенність дозволяє розгорнути API на різних операційних системах, що підвищує гнучкість розробки.

Unity3D: Unity3D обрано як основну платформу для створення клієнтського додатку-магазину. Це потужний ігровий рушій, який надає широкі можливості для розробки інтерактивних додатків з високоякісною графікою. Unity3D підтримує крос-платформенну розробку, що дозволяє запускати додаток на різних пристроях. Unity3D обрано через його можливості для створення візуально привабливих ігор і додатків. Він підтримує інтеграцію з різними системами, включаючи серверні API, що дозволяє створювати комплексні додатки.

Postman: Postman використовувався для тестування API. Це інструмент, який забезпечує зручний інтерфейс для створення, відправки та аналізу HTTP-запитів. З його допомогою було проведено перевірку роботи API, що дозволило виявити і виправити можливі помилки на етапі розробки. Postman є незамінним інструментом для тестування API. Він дозволяє швидко перевіряти правильність роботи серверної частини та виявляти помилки ще на етапі розробки.

Мова програмування C#: Мова програмування C# забезпечує можливість ефективної роботи з алгоритмами та структурами даних, що дозволяє досягти

високої продуктивності системи. C# є однією з основних мов програмування для розробки на платформі .NET Core, що забезпечує її інтеграцію з іншими компонентами системи та високу швидкість при виконанні алгоритмів колаборативної фільтрації.

Загалом, вибір цих технологій дозволив створити ефективну та надійну рекомендаційну систему, що відповідає сучасним вимогам до таких систем у сфері електронної комерції.

2.3 Аналіз вимог до користувачів рекомендаційної системи

Головною метою цього аналізу було з'ясування потреб та очікувань цільової аудиторії, а також визначення вимог, які повинні відповідати розроблюваній рекомендаційній системі. Було враховано, що користувачі системи можуть бути як кінцевими споживачами, так і адміністраторами системи.

2.3.1 Кінцеві споживачі (покупці).

Функції:

- Пошук і перегляд товарів.
- Отримання персоналізованих рекомендацій.
- Оцінювання товарів.
- Оформлення замовлень.

Можливості системи:

- Зручний інтерфейс для навігації та пошуку товарів.
- Відображення персоналізованих рекомендацій на основі вподобань.
- Можливість залишати оцінки на товари.
- Зручна форма для оформлення замовлень.

Вимоги:

- Базові навички роботи з інтернет-магазинами.
- Розуміння принципів роботи з веб-додатками.
- Наявність облікового запису в системі.

2.3.2 Адміністратори системи.

Функції:

- Управління каталогом товарів.
- Аналіз даних про продажі та вподобання користувачів.

Можливості системи:

- Доступ до панелі адміністрування для управління товарами та категоріями.

Вимоги:

- Розуміння принципів роботи рекомендаційних систем.
- Досвід роботи з базами даних та інтерфейсами адміністрування.

2.3.3 Загальні вимоги до всіх користувачів системи.

- Базове розуміння принципів роботи з інтернет-магазинами та рекомендаційними системами.
- Наявність стабільного інтернет-з'єднання для роботи з веб-додатком.

Вимоги до кінцевих споживачів та адміністраторів були визначені на основі аналізу цільової аудиторії та функціональних можливостей системи, що дозволяє забезпечити її ефективно та зручне використання.

РОЗДІЛ 3 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОДАЖУ ТОВАРІВ

3.1 Опис предметної області

Предметна область даної роботи охоплює розробку та впровадження рекомендаційної системи для електронної комерції на основі колаборативної фільтрації, зокрема item-based підходу. Система спрямована на покращення користувацького досвіду шляхом надання персоналізованих рекомендацій товарів, які враховують індивідуальні вподобання та поведінкові дані користувачів.

В умовах сучасного ринку, де кількість товарів і послуг постійно зростає, споживачі часто стикаються з труднощами при виборі продуктів. Рекомендаційні системи допомагають вирішити цю проблему, надаючи індивідуальні пропозиції, що базуються на аналізі історії покупок, оцінок та іншої активності користувачів. Це дозволяє не лише спростити процес вибору для клієнтів, але й збільшити задоволеність від покупок, що, в свою чергу, сприяє підвищенню лояльності до бренду та збільшенню обсягів продажів.

Основою розробленої системи є item-based колаборативна фільтрація. Цей підхід передбачає аналіз схожості між товарами на основі оцінок користувачів. В результаті користувачам пропонуються товари, подібні до тих, які вони вже високо оцінили або придбали. Однією з головних переваг цього підходу є його масштабованість, що дозволяє ефективно обробляти великі обсяги даних та надавати рекомендації в режимі реального часу.

Метою даної кваліфікаційної роботи бакалавра є створення ефективної рекомендаційної системи, яка забезпечує високу точність і швидкість надання рекомендацій, враховуючи індивідуальні потреби користувачів, що сприятиме підвищенню їх задоволеності та збільшенню обсягів продажів. Це дозволяє

підвищити якість обслуговування клієнтів та збільшити конкурентоспроможність бізнесу в умовах зростаючого ринку електронної комерції. Розробка охоплює всі етапи від створення бази даних до впровадження алгоритмів фільтрації та розробки клієнтського додатку.

В рамках цієї роботи детально розглянуто процес розробки рекомендаційної системи, включаючи проектування структури бази даних, реалізацію алгоритмів фільтрації, створення API для передачі даних та розробку клієнтського додатку на платформі Unity3D.

3.2 Вимоги до системи

Розробка рекомендаційної системи для продажу товарів передбачає визначення чітких вимог, які забезпечать її ефективне функціонування. Вимоги до системи поділяються на функціональні та нефункціональні. Ці вимоги служать основою для проектування та розробки системи, забезпечуючи її відповідність потребам користувачів і бізнес-цілям. Вимоги дозволяють встановити критерії для оцінки успішності системи та задоволення всіх зацікавлених сторін.

3.2.1 Функціональні вимоги

- Система повинна надавати користувачам індивідуальні рекомендації товарів на основі їх попередніх оцінок і поведінки. Рекомендації мають бути релевантними та точно відповідати вподобанням користувачів.
- Система повинна підтримувати реєстрацію нових користувачів, авторизацію та автентифікацію, забезпечуючи збереження особистих даних та історії покупок.
- Користувачі повинні мати можливість оцінювати товари, а система - обробляти ці оцінки для покращення точності рекомендацій.

- Система повинна містити детальний каталог товарів з інформацією про кожен товар, включаючи категорію, ціну, та виробника.
- Користувачі повинні мати можливість шукати товари за різними критеріями, такими як назва, категорія, ціна тощо.
- Система повинна мати інтерфейс для адміністраторів, де можна керувати каталогом товарів, аналізувати дані про продажі та вподобання користувачів.
- Система повинна забезпечувати збереження та отримання даних з бази даних MS SQL Server, що містить інформацію про користувачів, товари та їхні оцінки.
- Необхідно розробити API на основі .NET Core, який дозволяє клієнтському додатку Unity3D отримувати дані з бази даних та надавати рекомендації в режимі реального часу.

3.2.2 Нефункціональні вимоги

- Система повинна забезпечувати високу швидкість обробки даних та надання рекомендацій, навіть при значному навантаженні. Рекомендації повинні надаватися в межах кількох секунд.
- Система повинна бути здатна обробляти великі обсяги даних та підтримувати збільшення кількості користувачів та товарів без значного погіршення продуктивності.
- Система повинна забезпечувати захист особистих даних користувачів, включаючи шифрування даних, безпечну авторизацію та захист від несанкціонованого доступу.
- Інтерфейс системи повинен бути інтуїтивно зрозумілим та зручним у використанні для кінцевих користувачів і адміністраторів. Важливо забезпечити простоту навігації та доступ до основних функцій.

- Система повинна працювати на різних операційних системах та пристроях, забезпечуючи стабільність та коректну роботу незалежно від платформи.

Визначені вимоги забезпечують створення ефективної та надійної рекомендаційної системи, яка відповідає потребам користувачів та забезпечує високу якість обслуговування в сфері електронної комерції.

3.3 Архітектура системи

Архітектура розробленої рекомендаційної системи для продажу товарів складається з двох основних підсистем: клієнтської частини та серверної частини. Кожна з цих підсистем виконує певні функції та має свої особливості.

3.3.1 База даних

База даних є центральним компонентом рекомендаційної системи, оскільки вона зберігає всю необхідну інформацію для функціонування системи та забезпечує доступ до неї. База даних містить інформацію про користувачів, товари, категорії товарів, виробників, рейтинги товарів, відвідування сторінок, вибране та покупки (див. Рис.8). Це дозволяє системі здійснювати рекомендації на основі аналізу даних про поведінку користувачів та їхні вподобання.

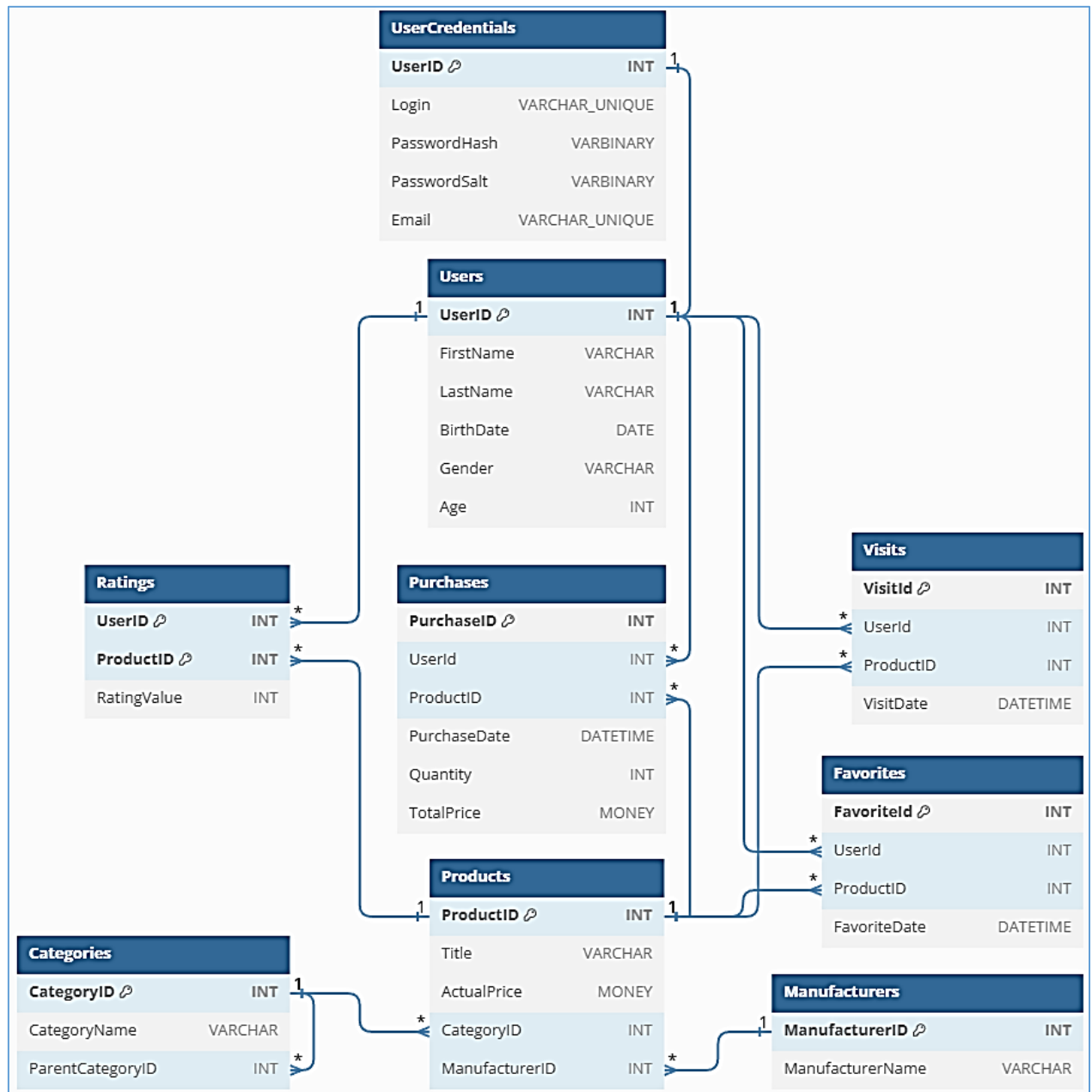


Рисунок 8 – Схема бази даних

Опис таблиць:

1. Таблиця Users (Користувачі)

Ця таблиця зберігає інформацію про користувачів системи, включаючи їхні імена, дату народження, стать та вік. Вона є ключовою для ідентифікації користувачів та надання персоналізованих рекомендацій (див. Лістинг 1).

Лістинг 1 Створення таблиці Users (Користувачі)

```

IF OBJECT_ID('Users') IS NOT NULL
    DROP TABLE Users;

GO

CREATE TABLE Users (
    UserID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    BirthDate DATE,
    Gender VARCHAR(255),
    Age AS
    CASE
        WHEN MONTH(BirthDate) > MONTH(GETDATE())
            OR (MONTH(BirthDate) = MONTH(GETDATE()) AND
DAY(BirthDate) > DAY(GETDATE()))
        THEN DATEDIFF(YEAR, BirthDate, GETDATE()) - 1
        ELSE DATEDIFF(YEAR, BirthDate, GETDATE())
    END
);

GO

```

2. Таблиця UserCredentials (Облікові дані користувачів)

Таблиця зберігає дані для авторизації користувачів, такі як логін, хеш пароля, сіль пароля та електронну пошту. Це забезпечує безпеку та контроль доступу до системи (див. Лістинг 2).

Лістинг 2 Створення таблиці UserCredentials (Облікові дані користувачів)

```

IF OBJECT_ID('UserCredentials') IS NOT NULL
    DROP TABLE UserCredentials;

GO

CREATE TABLE UserCredentials (
    UserID INT PRIMARY KEY FOREIGN KEY REFERENCES
Users(UserID),
    Login VARCHAR(255) NOT NULL UNIQUE,
    PasswordHash VARBINARY(MAX) NOT NULL,
    PasswordSalt VARBINARY(MAX) NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE
);

GO

```

3. Таблиця Categories (Категорії)

Ця таблиця містить інформацію про категорії товарів, що дозволяє організувати товари у логічні групи для полегшення їх пошуку та рекомендацій (див. Лістинг 3).

Лістинг 3 Створення таблиці Categories (Категорії)

```

IF OBJECT_ID('Categories') IS NOT NULL
    DROP TABLE Categories;

GO

CREATE TABLE Categories (
    CategoryID INT IDENTITY(1,1) PRIMARY KEY,
    CategoryName VARCHAR(255),
    ParentCategoryID INT NULL,
    FOREIGN KEY (ParentCategoryID) REFERENCES
Categories(CategoryID) );GO

```

4. Таблиця Manufacturers (Виробники)

Таблиця зберігає дані про виробників товарів, що дозволяє користувачам отримувати інформацію про бренди та їх продукцію (див. Лістинг 4).

Лістинг 4 Створення таблиці Manufacturers (Виробники)

```
IF OBJECT_ID('Manufacturers') IS NOT NULL
    DROP TABLE Manufacturers;
GO
CREATE TABLE Manufacturers (
    ManufacturerID INT IDENTITY(1,1) PRIMARY KEY,
    ManufacturerName VARCHAR(255)
);
GO
```

5. Таблиця Products (Товари)

Таблиця містить інформацію про товари, включаючи їхні назви, ціни, категорії та виробників. Це основний набір даних для формування рекомендацій (див. Лістинг 5).

Лістинг 5 Створення таблиці Products (Товари)

```
IF OBJECT_ID('Products') IS NOT NULL
    DROP TABLE Products;
GO
CREATE TABLE Products (
    ProductID INT IDENTITY(1,1) PRIMARY KEY,
    Title VARCHAR(255),
    ActualPrice MONEY,
    CategoryID INT FOREIGN KEY REFERENCES
Categories(CategoryID),
```

```

        ManufacturerID    INT    FOREIGN    KEY    REFERENCES
Manufacturers (ManufacturerID)
    );
GO

```

6. Таблиця Ratings (Рейтинги)

Таблиця зберігає оцінки товарів, надані користувачами. Ця інформація використовується для визначення уподобань користувачів і формування персоналізованих рекомендацій (див. Лістинг 6).

Лістинг 6 Створення таблиці Ratings (Рейтинги)

```

IF OBJECT_ID('Ratings') IS NOT NULL
    DROP TABLE Ratings;
GO
CREATE TABLE Ratings (
    UserID INT FOREIGN KEY REFERENCES Users (UserID) ON
DELETE CASCADE,
    ProductID INT FOREIGN KEY REFERENCES
Products (ProductID) ON DELETE CASCADE,
    RatingValue INT,
    PRIMARY KEY (UserID, ProductID)
);
GO

```

7. Таблиця Visits (Відвідування сторінок)

Ця таблиця зберігає дані про відвідування сторінок товарів користувачами. Це дозволяє аналізувати інтереси користувачів та покращувати рекомендації (див. Лістинг 7).

Лістинг 7 Створення таблиці *Visits* (Відвідування сторінок)

```

IF OBJECT_ID('Visits') IS NOT NULL
    DROP TABLE Visits;
GO
CREATE TABLE Visits (
    VisitId INT PRIMARY KEY IDENTITY,
    UserId INT,
    ProductId INT,
    VisitDate DATETIME,
    FOREIGN KEY (UserId) REFERENCES Users(UserId),
    FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
);
GO

```

8. Таблиця Favorites (Вибране)

Таблиця зберігає інформацію про товари, додані користувачами до вибраного. Це допомагає відстежувати вподобання користувачів і створювати відповідні рекомендації (див. Лістинг 8).

Лістинг 8 Створення таблиці *Favorites* (Вибране)

```

IF OBJECT_ID('Favorites') IS NOT NULL
    DROP TABLE Favorites;
GO
CREATE TABLE Favorites (
    FavoriteId INT PRIMARY KEY IDENTITY,
    UserId INT,
    ProductId INT,
    FavoriteDate DATETIME,
    FOREIGN KEY (UserId) REFERENCES Users(UserId),
    FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
);
GO

```

```
);
GO
```

9. Таблиця Purchases (Покупки)

Таблиця містить дані про покупки, включаючи ідентифікатори користувачів і товарів, дату покупки, кількість та загальну вартість. Це дозволяє аналізувати купівельну поведінку та покращувати рекомендації на основі історії покупок (див. Лістинг 9).

Лістинг 9 Створення таблиці Purchases (Покупки)

```
IF OBJECT_ID('Purchases') IS NOT NULL
    DROP TABLE Purchases;
GO
CREATE TABLE Purchases (
    PurchaseID INT IDENTITY(1,1) PRIMARY KEY,
    UserID INT FOREIGN KEY REFERENCES Users(UserID) ON
DELETE CASCADE,
    ProductID INT FOREIGN KEY REFERENCES
Products(ProductID) ON DELETE CASCADE,
    PurchaseDate DATETIME NOT NULL DEFAULT GETDATE(),
    Quantity INT NOT NULL,
    TotalPrice MONEY NOT NULL
);
GO
```

3.3.2 API

API (Application Programming Interface) є ключовим компонентом, що забезпечує взаємодію між клієнтською та серверною частинами системи. API дозволяє клієнтському додатку в Unity обмінюватися даними з серверною базою

даних, надаючи необхідні дані для роботи рекомендаційної системи. Оскільки Unity використовує обмежену версію .NET, яка не підтримує прямий доступ до баз даних, було необхідно створити API для забезпечення зв'язку між клієнтською частиною та сервером (див. Рис.9).

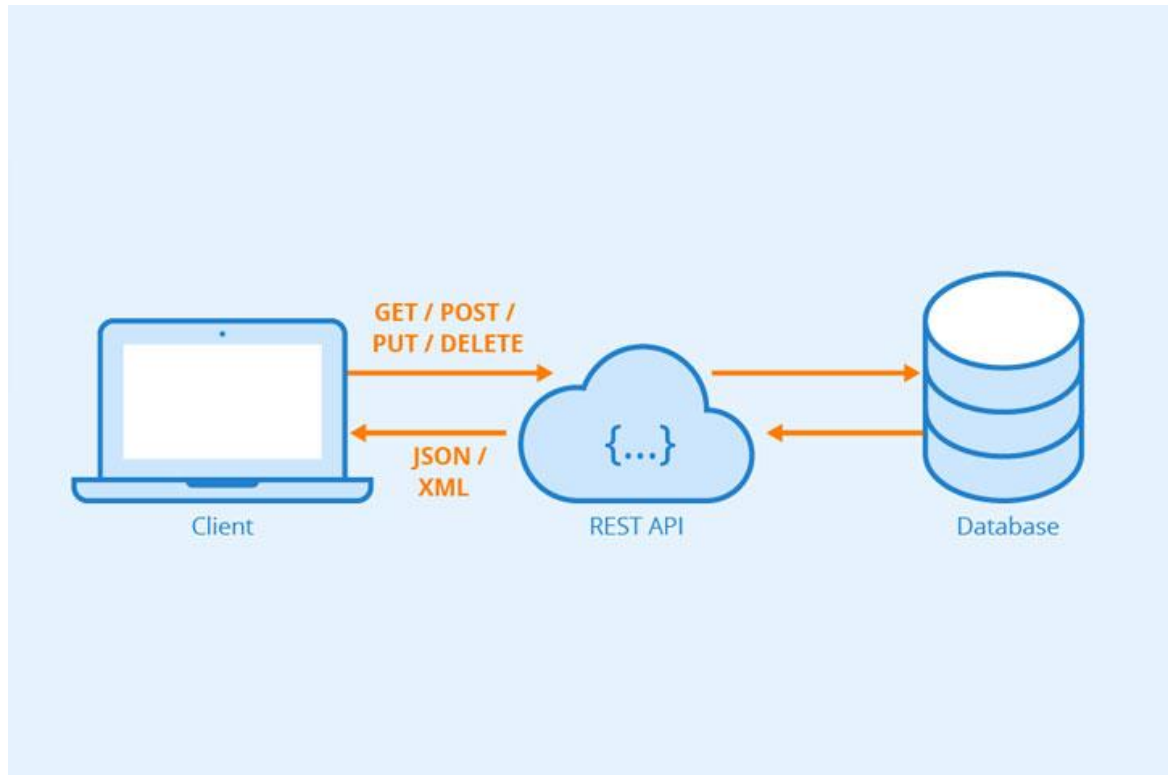


Рисунок 9 – Зв'язок між клієнтською частиною та сервером

API забезпечує кілька важливих функцій, таких як аутентифікація користувачів, отримання даних про товари, збереження інформації про відвідування сторінок, обробка рейтингів та багато іншого. Вся логіка взаємодії з базою даних зосереджена на сервері, що дозволяє клієнтському додатку в Unity бути легким і ефективним.

Нижче наведено файлову структуру проекту API (див. Рис.10).

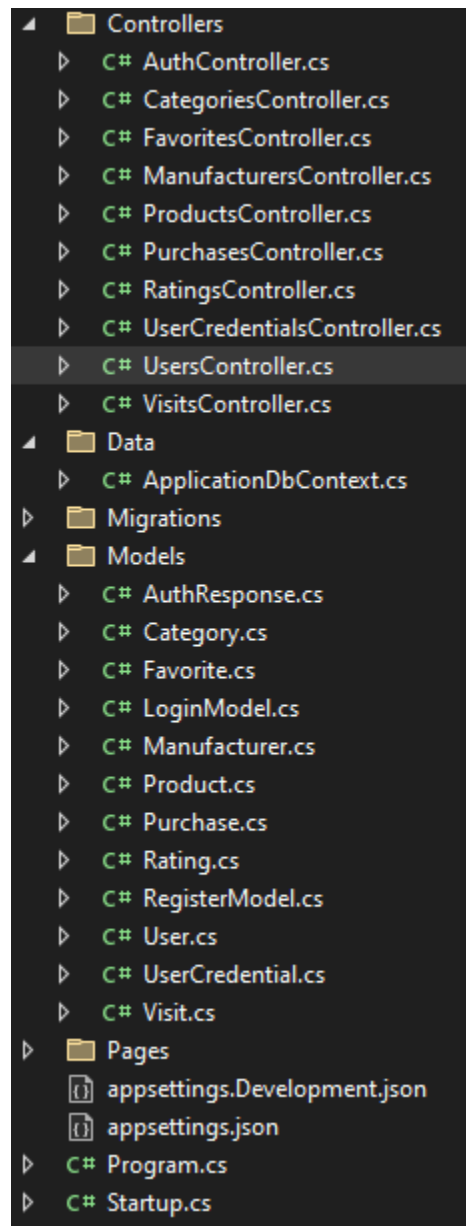


Рисунок 10 – Файлова структура проекту API

Опис API:

1. Models (Моделі)

Моделі визначають структуру даних, які використовуються в системі. Вони відповідають таблицям у базі даних та забезпечують типізований доступ до

даних. Наприклад, модель `User` (див. Лістинг 10) представляє користувача, а модель `Product` представляє товар.

Лістинг 10 Модель User

```
namespace ProductSalesApi.Models
{
    public class User
    {
        public int UserID { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public DateTime BirthDate { get; set; }
        public string Gender { get; set; }
        public int Age { get; set; }
    }
}
```

2. Controllers (Контролери)

Контролери обробляють HTTP-запити, виконують бізнес-логіку та взаємодіють з базою даних через контекст даних. Вони відповідають за виконання CRUD-операцій (Create, Read, Update, Delete). Наприклад, `UserController` (див. Лістинг 11) обробляє запити, пов'язані з користувачами, а `ProductsController` – з товарами.

Лістинг 11 Контролер UsersController

```
namespace ProductSalesApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class UsersController : ControllerBase
```

```

    {
        private readonly ApplicationDbContext _context;
        public UsersController(ApplicationDbContext
context) => _context = context;

        [HttpGet]
        public async
Task<ActionResult<IEnumerable<User>>> GetUsers()
        {
            return await _context.Users.ToListAsync();
        }

        [HttpGet("{id}")]
        public async Task<ActionResult<User>> GetUser(int
id)
        {
            var user = await
_context.Users.FindAsync(id);
            if (user == null)
                return NotFound();

            return user;
        }
    }
}

```

3. Data (Дані)

В цій папці зберігається контекст даних `ApplicationDbContext`, який є проміжним шаром між базою даних і кодом застосунку. Контекст даних

використовує Entity Framework Core для взаємодії з базою даних, забезпечуючи легкий доступ до даних та їх управління (див. Лістинг 12).

Лістинг 12 Налаштування зв'язків в класі ApplicationDbContext

```
// Визначення первинного ключа для UserCredential
modelBuilder.Entity<UserCredential>().HasKey(uc => uc.UserID);

// Налаштування зв'язку один до одного між User і
UserCredential

        modelBuilder.Entity<UserCredential>()
            .HasOne<User>()
            .WithOne()
            .HasForeignKey<UserCredential>(uc =>
uc.UserID);

// Налаштування первинного ключа для Rating
        modelBuilder.Entity<Rating>()
            .HasKey(r => new { r.UserID, r.ProductID
});

// Налаштування зв'язку багато-до-одного між Product і
Category

        modelBuilder.Entity<Product>()
            .HasOne<Category>()
            .WithMany()
            .HasForeignKey(p => p.CategoryID);
```

4. Startup (Запуск)

Клас Startup налаштовує сервіси та середовище додатку. Він визначає конфігурацію служб, таких як підключення до бази даних, аутентифікація за допомогою JWT (JSON Web Token), та налаштовує маршрутизацію запитів. Це забезпечує безпечний і організований запуск додатку.

Для тестування та перевірки роботи API використовувався інструмент Postman (див. Рис.11), який дозволяє відправляти HTTP-запити до API та отримувати відповіді. Це допомогло впевнитися в коректності роботи всіх ендпойнтів і забезпечити правильну взаємодію між клієнтським додатком і сервером.

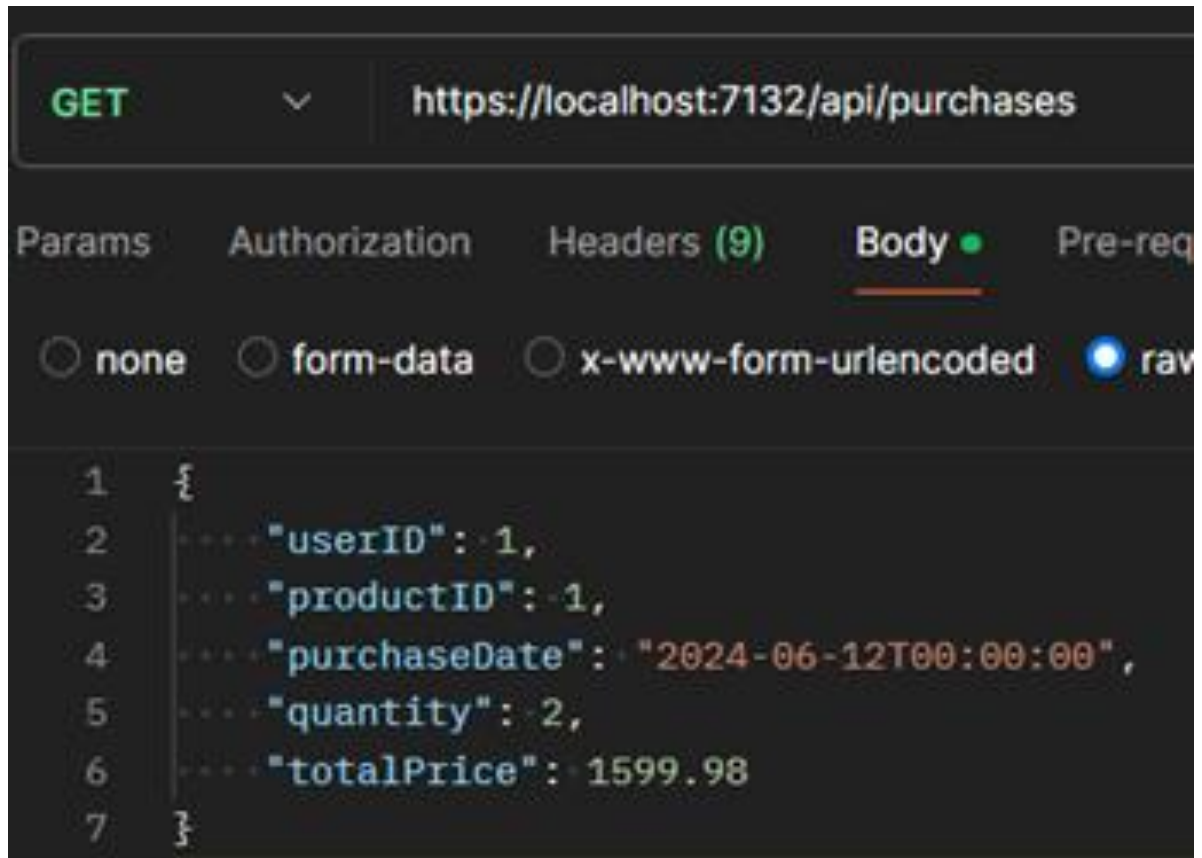


Рисунок 11 – Приклад запиту в Postman

API є критичним компонентом системи, що забезпечує безперерйну та безпечну взаємодію між клієнтом та сервером, дозволяючи рекомендаційній системі ефективно виконувати свої функції.

3.3.3 Клієнтська частина

Для створення клієнтської частини додатка був обраний Unity3D. Вибір цього інструменту обґрунтований його потужними можливостями для розробки інтерактивних та візуально привабливих додатків, а також його кросплатформеністю, що дозволяє розробляти додаток одразу для декількох операційних систем.

В додатку було створено декілька екранів, а також навігаційну панель (barNavigation). Основні екрани включають:

- **LoginScreen:** Екран для входу в акаунт, де користувач може ввести свої облікові дані для аутентифікації.
- **RegisterScreen:** Екран для реєстрації користувача, де перевіряється коректність введених даних. Це дозволяє новим користувачам створити акаунт у системі.
- **HomeScreen:** Головний екран додатку, де користувачу одразу показуються рекомендовані товари. На цьому екрані також є рядок для пошуку товарів.

Для представлення товару був створений префаб, який зберігає всю інформацію про товар. При натисканні на префаб, відкривається екран ProductScreen, який отримує інформацію з префаба та представляє сторінку для покупки цього товару.

Коли користувач переходить на екран ProductScreen, в таблицю з рейтингами (Ratings) одразу заноситься умовна оцінка, щоб можна було почати рекомендувати користувачу цікаві для нього товари. Також заноситься інформація в таблицю Visits (Відвідування сторінок). Якщо користувач додає товар в обране, оцінка товару змінюється на більш високу, щоб ще точніше підбирати товари. Дані заносяться в таблиці Favorites (Вибране) та Ratings (Рейтинги). Якщо користувач прибирає товар з обраного, він видаляється з таблиці Favorites.

Після покупки товару, користувач може виставити свою оцінку. Це дозволяє системі ще краще розуміти уподобання користувача і покращувати рекомендації.

За допомогою навігаційної панелі (barNavigation) користувач може переходити між основними екранами додатку:

- HomeScreen (Головний екран)
- CatalogScreen (Каталог товарів)
- BasketScreen (Кошик)
- FavoriteScreen (Обране)
- CabinetScreen (Особистий кабінет)

Таким чином, клієнтська частина додатку в Unity3D забезпечує зручний і інтерактивний інтерфейс для користувачів, дозволяючи їм легко взаємодіяти з системою та отримувати персоналізовані рекомендації товарів.

Опис екранів:

1. Екран LoginScreen

Екран LoginScreen (див. Рис.12) відповідає за аутентифікацію користувачів в системі. При вході на цей екран користувач вводить свої облікові дані (логін та пароль). Після натискання кнопки "Увійти" відбувається перевірка введених даних.

Інформація для аутентифікації користувачів береться з таблиці UserCredentials, де зберігаються логіни, хеші паролів та інші облікові дані користувачів. Якщо облікові дані користувача коректні, відбувається успішна авторизація, і система отримує UserID цього користувача.

Отримання UserID дозволяє системі негайно створювати персоналізовані рекомендації для користувача. Якщо в таблиці Ratings (Рейтинги) є хоча б одна оцінка від цього користувача, то система використовує ці дані для формування рекомендацій.

Якщо ж оцінок у таблиці Ratings немає, система рекомендує користувачу найбільш популярні товари з каталогу. Це вирішує проблему "холодного старту", коли система не має достатньо інформації про вподобання нового користувача для надання персоналізованих рекомендацій.

З екрану LoginScreen користувач також має можливість перейти на екран RegisterScreen для створення нового акаунта, якщо він ще не зареєстрований в системі.

AGS ALL GOODS STORE

Sign in to AGS

Login

Password

Sign In

Don't have an account ? / [Sign up](#)

Рисунок 12 – LoginScreen

2. Екран RegisterScreen

Екран RegisterScreen (див. Рис.13) призначений для реєстрації нових користувачів в системі. На цьому екрані користувач вводить свої персональні дані, такі як ім'я, прізвище, дату народження, стать, логін, пароль та електронну пошту.

При успішній реєстрації ці дані додаються до відповідних таблиць у базі даних:

- Таблиця Users зберігає основну інформацію про користувача, включаючи його ім'я, прізвище, дату народження та стать.
- Таблиця UserCredentials зберігає облікові дані користувача, включаючи логін, хеш пароля, сіль пароля та електронну пошту.

На екрані RegisterScreen присутня перевірка введених даних. Система перевіряє, чи заповнені всі поля і чи співпадають паролі. Якщо користувач не заповнив всі поля або паролі не співпадають, система видасть відповідне повідомлення про помилку.

Також система перевіряє, чи вже використовуються введені дані для реєстрації (логін або електронна пошта). Якщо такі дані вже існують у базі даних, користувач отримає повідомлення про помилку реєстрації.

Після успішної реєстрації користувача повертає на екран LoginScreen, щоб він міг ввести свої дані і увійти в систему.

Таким чином, екран RegisterScreen забезпечує зручний і безпечний процес реєстрації нових користувачів, надаючи необхідні перевірки введених даних для запобігання помилок і дублювань.

Sign up to AGS

Please enter your details

First Name

Last Name

Birth Date

1981

5

3

Gender

Login

Email

Password

Please make sure you passwords match.

Confirm Password

Рисунок 13 – RegisterScreen

3. Екран HomeScreen

Після входу в систему користувач одразу потрапляє на екран HomeScreen (див. Рис.14). Цей екран є головною сторінкою додатку, де користувач може знайти персоналізовані рекомендації товарів.

Внизу сторінки розташована навігаційна панель, яка допомагає користувачу швидко переміщуватися між різними екранами додатку, такими як HomeScreen, CatalogScreen, BasketScreen, FavoriteScreen та CabinetScreen.

На HomeScreen є рядок пошуку для швидкого пошуку товарів. Користувач може вводити запити для знаходження конкретних товарів у каталозі.

Головна частина екрану присвячена відображенню рекомендованих товарів. Ці товари розміщені в scroll rect, що дозволяє користувачу прокручувати список рекомендацій. Всередині scroll rect знаходяться об'єкти (префаби), які відображають зображення, назву та ціну товару.

Список рекомендованих товарів постійно оновлюється в залежності від взаємодії користувача з товарами, включаючи перегляди, додавання до обраного та оцінки.

При натисканні на обраний товар відкривається екран ProductScreen, який представляє детальну інформацію про товар і дозволяє здійснити покупку.

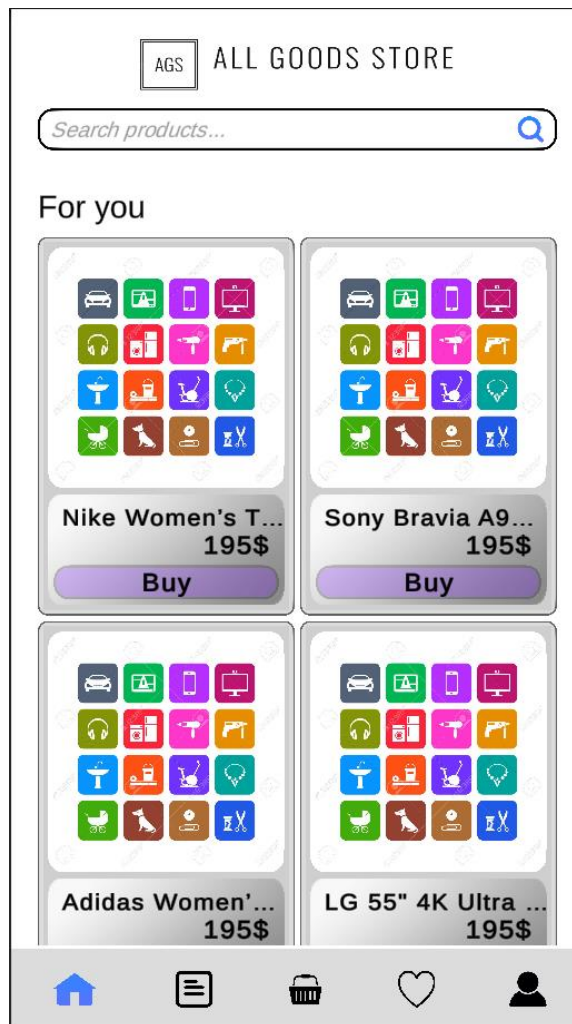


Рисунок 14– HomeScreen

4. Екран ProductScreen

Екран ProductScreen (див. Рис.15) надає всю необхідну інформацію про обраний товар. На цьому екрані користувач може детально ознайомитися з товаром, переглянути його характеристики та здійснити покупку.

Основні елементи ProductScreen включають:

На цьому екрані є scroll Rect, в якому користувач може переглядати зображення товару. Перегортання зображень здійснюється за допомогою прокручування, що дозволяє ознайомитися з різними ракурсами або варіантами товару.

Користувач має можливість додати товар в обране або видалити його з обраного. В залежності від дій користувача, дані про товар заносяться або видаляються з таблиці Favorites в базі даних. Це дозволяє зберігати вподобання користувача та враховувати їх при формуванні рекомендацій.

На екрані є кнопка для здійснення покупки товару. Після натискання на цю кнопку користувач може придбати товар.

Після покупки товару користувач має можливість залишити свою оцінку. Це дозволяє покращувати систему рекомендацій, враховуючи реальні вподобання та досвід користувачів. Дані про оцінку заносяться в таблицю Ratings в базі даних.

Таким чином, ProductScreen забезпечує користувача всією необхідною інформацією про товар і надає можливість легко додавати товари в обране, здійснювати покупки та залишати оцінки, що робить процес взаємодії з товаром зручним та інтуїтивно зрозумілим.

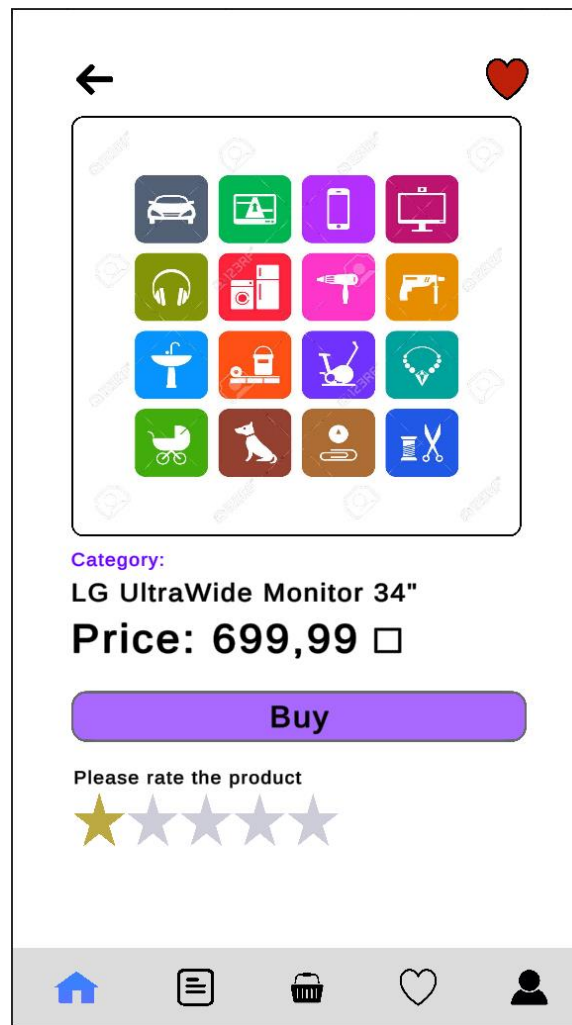


Рисунок 15 – *ProductScreen*

5. Екран CatalogScreen

Екран CatalogScreen (див. Рис.16) відповідає за перегляд каталогу товарів та здійснення пошуку товарів. Цей екран надає користувачам можливість швидко знаходити потрібні товари та переходити до детальної інформації про них.

У верхній частині екрану розташована строка пошуку, де користувач може вводити ключові слова для пошуку товарів у каталозі. Це дозволяє швидко знаходити товари за назвою або описом.

З лівого боку екрану розташовано список категорій. Користувач може вибрати будь-яку категорію, і або в центральній частині екрану будуть

відображені підкатегорії, якщо вони є, або відкріється екран з товарами, що належать до обраної категорії.

У центральній частині екрану відображаються товари, які відповідають обраній категорії або пошуковому запиту. Також користувач може переглядати всі доступні товари у каталозі.

При натисканні на товар користувач перенаправляється на екран ProductScreen, де він може ознайомитися з детальною інформацією про товар.

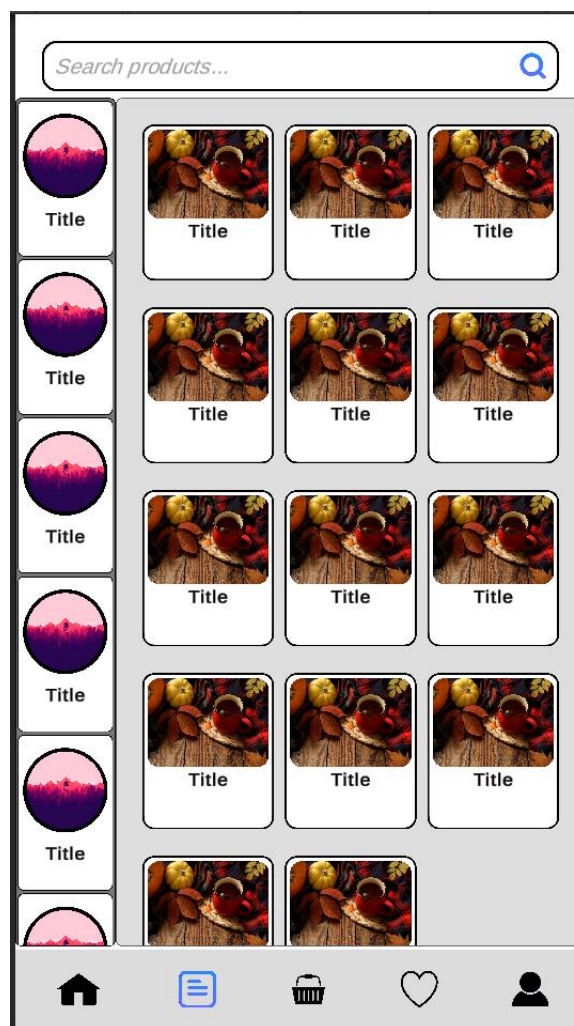


Рисунок 16 – *CatalogScreen*

6. Екран BasketScreen

Екран BasketScreen (див. Рис.17) призначений для перегляду товарів, доданих у кошик, та для здійснення покупок. Він забезпечує зручний інтерфейс для управління покупками та отримання рекомендацій.

В центральній частині екрану відображаються товари, які були додані користувачем у кошик. Користувач може переглядати деталі кожного товару, змінювати кількість або видаляти товари з кошика.

Якщо кошик пустий, тоді на екрані знаходиться горизонтальний скролл, що містить товари, які користувач переглядав раніше. Це дозволяє швидко повернутися до переглянутих товарів та додати їх до кошика, якщо потрібно.

Під списком переглянутих товарів розташований ще один скролл, як на HomeScreen, який містить персоналізовані рекомендації товарів. Ці рекомендації оновлюються в залежності від взаємодії користувача з товарами та допомагають знайти нові цікаві товари для покупки.

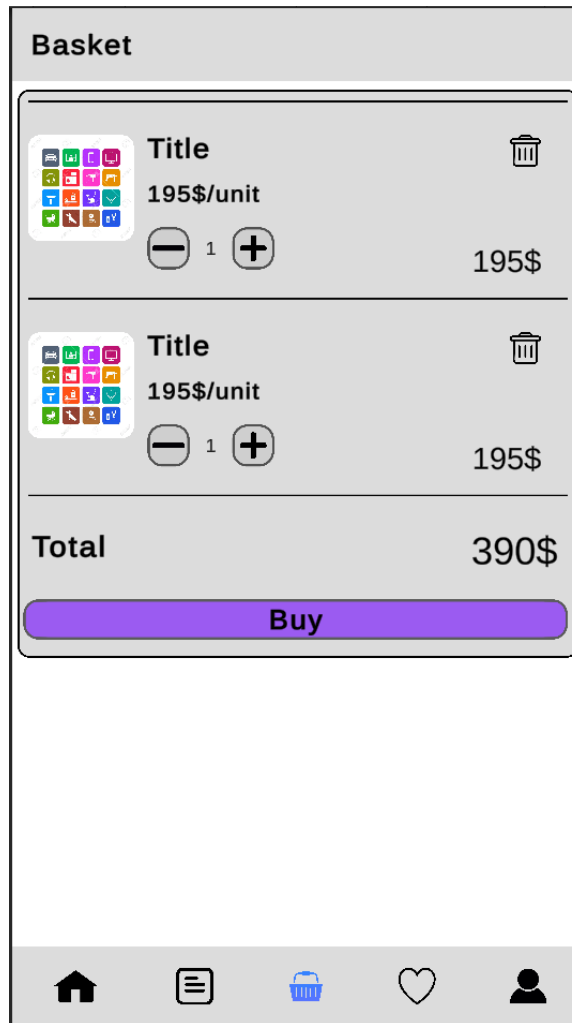


Рисунок 17 – *BasketScreen*

7. Екран FavoriteScreen

Екран FavoriteScreen (див. Рис.18) призначений для відображення товарів, доданих користувачем до обраного. Він забезпечує зручний інтерфейс для перегляду та управління вподобаними товарами.

В центральній частині екрану розташований скролл, який містить товари, додані користувачем до обраного. Кожен товар представлений префабом, що відображає його зображення, назву та ціну.

Користувач може переглядати всі товари, які він додав до обраного, і при бажанні видаляти їх з цього списку. Це дозволяє легко керувати вподобаними товарами і повертатися до них у зручний момент.

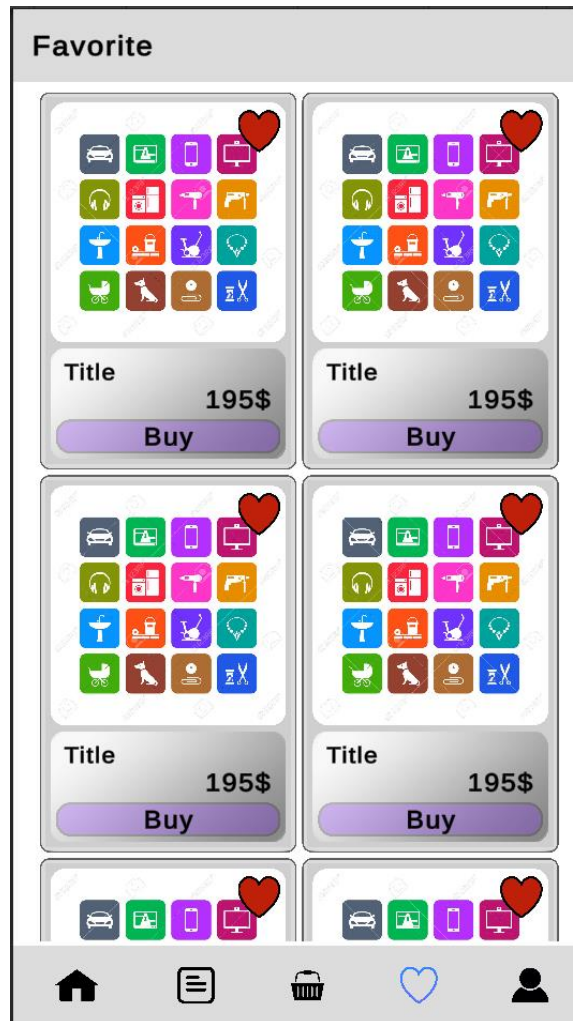


Рисунок 18 – *FavoriteScreen*

8. Екран CabinetScreen

Екран CabinetScreen (див. Рис.19) призначений для відображення інформації про користувача та надання можливості керувати своїм акаунтом.

На цьому екрані відображається особиста інформація користувача, така як ім'я, прізвище, електронна пошта та інші деталі профілю. Це дозволяє користувачу переглядати та за необхідності оновлювати свої особисті дані.

На екрані також розташована кнопка для виходу з системи. Натискання цієї кнопки виконує вихід користувача з його акаунта, повертаючи його на екран LoginScreen для повторної аутентифікації або входу іншого користувача.

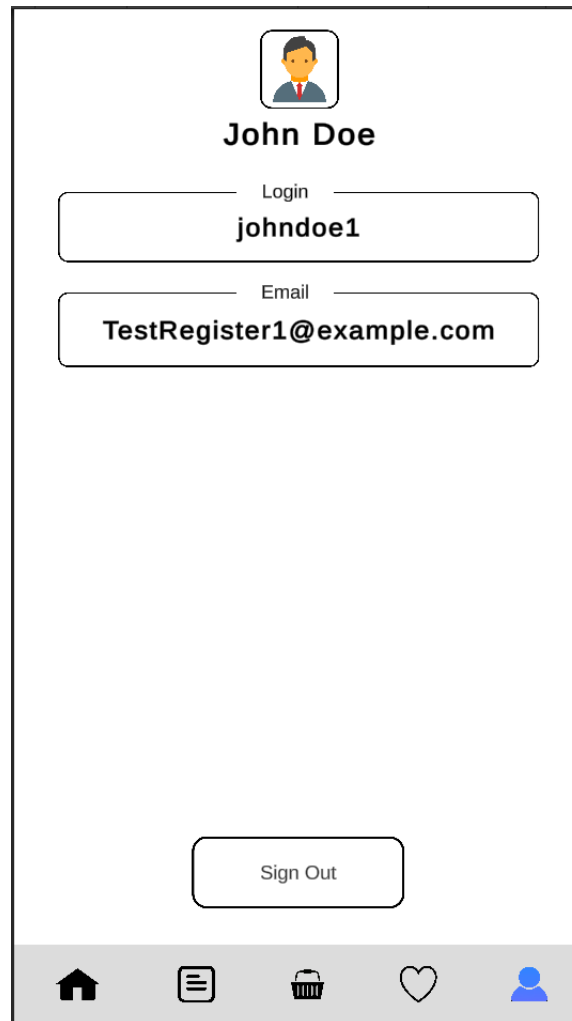


Рисунок 19 – *CabinetScreen*

3.3.4 Реалізація рекомендаційної системи на основі item-based

Рекомендаційна система на основі item-based використовує інформацію про взаємодію користувачів з товарами для створення рекомендацій. Основна ідея полягає в тому, щоб знаходити товари, які схожі на ті, що користувач вже оцінив, і рекомендувати їх. Цей підхід ефективний, оскільки він дозволяє

враховувати індивідуальні вподобання користувачів та забезпечувати персоналізовані рекомендації.

Система використовує дані з таблиці Ratings для побудови матриці взаємодій користувачів з товарами. Кожен раз, коли користувач оцінює товар, до цього товару знаходяться 10 рекомендацій. Для кожного оціненого товару знаходиться 10 рекомендацій, які додаються до загального списку. Чим більше товарів оцінено – тим більше загальний список і тим більш унікальні рекомендації, які видаються користувачеві. Потім загальний список перемішується та користувачу видаються перші 10 рекомендацій. Користувач завжди бачить лише 10 рекомендацій. Це забезпечує, що користувач бачить різноманітні набори рекомендацій при кожному взаємодії з системою.

Принцип роботи системи:

Спочатку будується матриця взаємодій. Матриця взаємодій є основою для роботи рекомендаційної системи на основі item-based. Вона представляє собою двовимірний масив, де рядки відповідають товарам, а стовпці – користувачам. Кожен елемент цієї матриці містить рейтинг, який конкретний користувач поставив певному товару.

Матриця взаємодій є важливою, оскільки вона дозволяє алгоритму визначати схожість між товарами на основі їх оцінок різними користувачами. Це, в свою чергу, дозволяє формувати персоналізовані рекомендації для кожного користувача. Завдяки цій матриці, система може виявити патерни в оцінках користувачів і використовувати їх для знаходження товарів, які ймовірно сподобаються користувачу.

Дані з таблиці Ratings використовуються для побудови матриці взаємодій між користувачами і товарами. Кожен елемент цієї матриці представляє рейтинг, який користувач поставив певному товару (див. Лістинг 13).

Лістинг 13 Побудова матриці взаємодій:

```
public (double[,] userItemMatrix, List<int> productIds,
List<int> userIds) PreprocessData(List<ApiManager.Product>
products, List<ApiManager.Rating> ratings)
{
    var userItemMatrixDict = new Dictionary<int, Dictionary<int,
double>>();

    /* Пропуск частини коду*/
    double[,] userItemMatrix = new double[productIds.Count,
userIds.Count];

    /* Пропуск частини коду*/

    return (userItemMatrix, productIds, userIds);
}
```

Система визначає товари, які користувач вже оцінив (див. Лістинг 14), та категорії цих товарів (див. Лістинг 15).

Лістинг 14 Отримання оцінених товарів

```
public List<int> GetUserRatings(List<ApiManager.Rating>
ratings, int userId)
{
    return ratings.Where(r => r.UserID == userId).Select(r =>
r.ProductID).ToList();
}
```

Лістинг 15 *Отримання категорій*

```
public List<int>
GetRatedCategories(List<ApiManager.Product> products,
List<int> ratedProducts)
{
    return products.Where(p =>
ratedProducts.Contains(p.ProductID)).Select(p =>
p.CategoryID).Distinct().ToList();
}
```

Для кожного оціненого товару знаходяться схожі товари з тієї ж категорії (див. Лістинг 16), які потім перемішуються (див. Лістинг 17) і видаються користувачу (див. Лістинг 18). Всі рекомендації перемішуються, і користувачу видаються перші 10 з них. Це забезпечує, що користувач бачить різні набори товарів при кожній взаємодії з системою.

Лістинг 16 *Знаходження схожих товарів з тієї ж категорії*

```
var recommendations = distances
.OrderBy(d => d.distance)
.Where(d => !ratedProducts.Contains(productIds[d.index])
&& products.First(p => p.ProductID ==
productIds[d.index]).CategoryID == categoryId)
.Take(numRecommendations)
.Select(d => new Recommendation { ProductId =
productIds[d.index], Title = products.First(p => p.ProductID
== productIds[d.index]).Title, Distance = d.distance })
.ToList();
```

Лістинг 17 Перемішування рекомендацій для унікальності

```
var random = new System.Random();
uniqueRecommendations = uniqueRecommendations.OrderBy(x =>
random.Next()).ToList();
```

Лістинг 18 Видача рекомендацій користувачу

```
Debug.Log("Top 10 Recommendations:");
for (int i = 0; i < 10 && i <
uniqueRecommendations.Count; i++)
{
    Debug.Log($"Product ID:
{uniqueRecommendations[i].ProductId}, Title:
{uniqueRecommendations[i].Title}, Distance:
{uniqueRecommendations[i].Distance:F6}");
}
```

Косинусна схожість (див. Лістинг 19) є метрикою, яка використовується для визначення схожості між двома товарами. Вона обчислюється як косинус кута між двома векторами в багатовимірному просторі. У даному проекті, ці вектори представляють рейтинги, які різні користувачі поставили двом товарам.

Косинусна схожість набуває значень від -1 до 1, де 1 означає повну схожість, 0 – відсутність схожості, а -1 – повну протилежність.

Лістинг 19 Метод для обчислення косинусної схожості

```
private double CosineDistance(double[,] matrix, int
index1, int index2)
{
    double dotProduct = 0;
    double normA = 0;
```

```

double normB = 0;
int length = matrix.GetLength(1);

for (int i = 0; i < length; i++)
{
    dotProduct += matrix[index1, i] *
matrix[index2, i];
    normA += Math.Pow(matrix[index1, i], 2);
    normB += Math.Pow(matrix[index2, i], 2);
}

return 1 - (dotProduct / (Math.Sqrt(normA) *
Math.Sqrt(normB)));
}

```

У системі передбачено, що при взаємодії користувача з товаром вносяться зміни в таблицю Ratings для покращення рекомендацій. Це дозволяє алгоритму надавати більш релевантні пропозиції на основі поведінки користувача.

Коли користувач відвідує сторінку з товаром, в таблицю Ratings додається запис з фіксованим значенням RatingValue (див. Лістинг 20). Це дозволяє алгоритму враховувати переглянуті товари при генерації рекомендацій.

Лістинг 20 Додавання рейтингу при відвідуванні сторінки товару

```

IEnumerator AddVisitCoroutine(int productId)
{
    string visitsUrl = baseUrl + "visits";
    Visit visit = new Visit
    {
        UserId = this.UserID,

```

```

        ProductId = productId,
        VisitDate = System.DateTime.UtcNow
    }; /*Пропуск частини коду*/
AddOrUpdateRating(productId, 5); /*Пропуск частини коду*/
}

```

Коли користувач додає товар в обране, в таблицю Ratings додається запис з вищим значенням RatingValue, оскільки цей товар є більш цікавим для користувача (див. Лістинг 21). Це дозволяє алгоритму надавати більш точні рекомендації, враховуючи вподобання користувача.

Лістинг 21 Додавання рейтингу при додаванні товару в обране

```

IEnumerator AddFavoriteCoroutine(int productId)
{
    string favoritesUrl = baseUrl + "favorites";
    Favorite favorite = new Favorite
    {
        UserId = this.UserID,
        ProductId = productId,
        FavoriteDate = System.DateTime.UtcNow
    };
    /*Пропуск частини коду*/
    AddOrUpdateRating(productId, 5);
    /*Пропуск частини коду*/
}

```

3.4 Висновки з 3 розділу

1. Реалізація рекомендаційної системи на основі item-based забезпечує персоналізовані та релевантні рекомендації для користувачів, враховуючи їхні вподобання та взаємодії з товарами.
2. Розроблена система демонструє значний потенціал для покращення користувацького досвіду та підвищення продажів, що робить її важливим інструментом для сучасних платформ.
3. Основні характеристики та достоїнства розробленої системи:
 - Система надає рекомендації на основі індивідуальних оцінок користувачів, що підвищує їх релевантність та точність.
 - При формуванні рекомендацій враховуються категорії оцінених товарів, що дозволяє надавати більш релевантні пропозиції.
 - Використання алгоритму item-based дозволяє легко масштабувати систему для великої кількості товарів та користувачів.
 - Клієнтський інтерфейс на Unity3D забезпечує зручну та інтуїтивно зрозумілу взаємодію користувача з системою.

ВИСНОВКИ

1. Вивчені основи рекомендаційних систем, зокрема підходи на основі колаборативної фільтрації.
2. Розроблена архітектура системи рекомендацій для продажу товарів, що включає клієнтську частину на базі Unity3D та серверну частину на базі .NET Core.
3. Спроектвана база даних в MS SQL Server, яка включає таблиці користувачів, товарів, категорій, виробників, рейтинги, відвідування сторінок та обране.
4. Реалізовані API для взаємодії з базою даних, що забезпечує доступ до даних про користувачів, товари, рейтинги, відвідування та обране.
5. Реалізований клієнтський інтерфейс на Unity3D з використанням різних екранів, що забезпечують зручну навігацію та взаємодію користувача з системою.
6. Реалізований алгоритм item-based рекомендаційної системи, що використовує косинусну схожість для визначення схожості між товарами та надає персоналізовані рекомендації на основі оцінок користувачів.
7. Всі API були ретельно протестовані за допомогою Postman, що забезпечує надійність та стабільність системи.
8. Подальший розвиток системи може включати: покращення алгоритмів рекомендацій, зокрема впровадження гібридних підходів, які комбінують колаборативну фільтрацію з контентними методами; оптимізацію продуктивності системи для обробки ще більшої кількості даних та користувачів; розширення функціональності клієнтського інтерфейсу, зокрема додавання нових можливостей для фільтрації та сортування товарів..

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безверхий А. І. Архітектура та проектування програмного забезпечення: навчально-методичний посібник для студентів спеціальності 121 "Інженерія програмного забезпечення". Запоріжжя: ЗДІА, 2017. 198 с.
2. Anil Maheshwari Data Analytics Made Accessible : Apress, 1 травня 2014 р. С. 448.
3. William Poundstone Priceless: The Myth of Fair Value (and How to Take Advantage of It) : Hill and Wang, 1 січня 2010 р. С. 336.
4. Martin Kleppmann Designing Data-Intensive Applications : O'Reilly Media, 16 березня 2017 р. С. 616.
5. Nathan Marz and James Warren Big Data: Principles and Best Practices of Scalable Realtime Data Systems : Manning Publications, 10 травня 2015 р. С. 328.
6. PostgreSQL Documentation 15: PostgreSQL 15.3 Documentation : веб-сайт. URL: <https://www.postgresql.org/docs/current/index.html> (дата звернення: 10.03.2024).
7. Index | Node.js v19.9.0 Documentation : веб-сайт. URL: <https://nodejs.org/docs/latest-v19.x/api/> (дата звернення: 10.03.2024).
8. Express - Node.js web application framework : веб-сайт. URL: <https://expressjs.com/> (дата звернення: 10.03.2024).
9. Introduction | Vue.js : веб-сайт. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 10.06.2023).
10. Angular vs React vs Vue: Core Differences | BrowserStack : веб-сайт. URL: <https://www.browserstack.com/guide/angular-vs-react-vs-vue> (дата звернення: 12.03.2023).
11. Figma: the collaborative interface design tool. : веб-сайт. URL: <https://www.figma.com/> (дата звернення: 12.03.2023).

12. Diagrams : веб-сайт. URL: <https://app.diagrams.net/> (дата звернення: 15.03.2023).
13. Google Scholar: Collaborative Filtering : веб-сайт. URL: <https://scholar.google.com/scholar?q=collaborative+filtering> (дата звернення: 15.03.2023).
14. Berkovich S., Zeleznikow J. A Recommender System Based on Collaborative Filtering for Educational Services : Springer, 2021. С. 234-245.
15. Zhang Y., Chowdhury M. H. A Personalized Recommendation System Based on User Behavior Analysis : IEEE, 2020. С. 112-118.
16. Liu B., Hu M., Xie T. A Recommender System for Online Shopping Based on User Preferences : Journal of Computer Science and Technology, 2019. С. 745-756.
17. Sarwar B., Karypis G., Konstan J., Riedl J. Item-based collaborative filtering recommendation algorithms : Proceedings of the 10th International Conference on World Wide Web, 2001. С. 285-295.
18. Su X., Khoshgoftaar T. M. A Survey of Collaborative Filtering Techniques : Advances in Artificial Intelligence, 2009. С. 421-434.
19. Юрьєв В. В., Безверхий А. І. Рекомендаційна система продажу товарів на основі колаборативної фільтрації. Збірник наукових праць студентів, аспірантів, докторантів і молодих вчених «Молода наука-2024» / Запорізький національний університет. Запоріжжя: ЗНУ, 2024. Т.5. С. 215-216.

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Юрьєв Володимир Вікторович, студент 4 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz19bd-18@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему **«Рекомендаційна система продажу товарів на основі колаборативної фільтрації»** відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;
- згоден на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 14.06.2024 Підпис _____ Юрьєв Володимир Вікторович
(студент)

Дата 14.06.2024 Підпис _____ Безверхий Анатолій Ігорович
(науковий керівник)