

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод формування оптимальних наборів пропозицій в системах електронної торгівлі

Галузь знань _____ 12 – Інформаційні технології _____
Шифр і назва галузі знань
Спеціальність _____ 122 – Комп'ютерні науки _____
Шифр і назва спеціальності
Освітня програма _____ Комп'ютерні науки _____
Назва освітньої програми

Виконав: _____ студент 4 курсу, група КН-18-1 _____ Власюк _____ М.І. Власюк _____
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: _____ к.т.н., доцент кафедри КН _____ Манзюк _____ Е.А. Манзюк _____
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: _____ к.т.н., доцент кафедри КН _____ Багрій _____ Р.О. Багрій _____
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

_____ Бармак _____ О.В. Бармак _____
Підпис Ініціали, прізвище

16 червня 2022 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

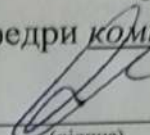
Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор О.В. Бармак

«16» березня 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод формування оптимальних наборів пропозицій в системах електронної торгівлі»

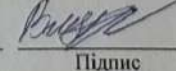
2. Завдання видано студенту Власюку Миколі Ігоровичу
(прізвище, ім'я, по батькові)

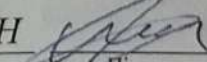
3. Керівник роботи доцент кафедри КН Манзюк Едуард Андрійович
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «1» березня 2022 р. № 18

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка методу формування оптимальних наборів пропозицій в системах електронної торгівлі. При формуванні оптимальних наборів пропозицій слід враховувати популярність товарних пропозицій в системі торгівлі, так і їхню ціну. Слід забезпечити виконання функцій внесення, редагування та перегляду інформації яка необхідна для роботи методу, й одержання наступних результатів роботи методу: автоматизований вивід найпопулярніших товарів та найвигідніших товарів, визначення умовного показника популярності товару у системі торгівлі

Виконавець: студент 4 курсу, група КН-18-1  М.І. Власюк
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КН  Е.А. Манзюк
Науковий ступінь, посада Підпис Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод формування оптимальних наборів пропозицій в системах електронної торгівлі»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-18-1 Власюк Микола Ігорович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Манзюк Едуард Андрійович

Кваліфікаційна робота бакалавра містить:

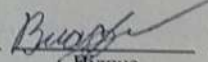
Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
61	31	19	12	4

Метою кваліфікаційної роботи бакалавра є розробка методу формування оптимальних наборів пропозицій в системах електронної торгівлі. Для реалізації цього методу було використано мову програмування PHP, мову стилів CSS, та мову розмітки HTML, а також систему керування базами даних MySQL Server.

Розроблений метод призначений обробки вхідних даних, які будуть обраховані та виданий кінцевий результат у вигляді рекомендаційного списку. Реалізоване автоматичне формування оптимальних наборів товарів, дозволяє підвищити ефективність магазину, підвищує ефективність реклами для заманювання нових клієнтів

Напрямами практичного використання розробленого методу формування оптимальних наборів пропозицій в системах електронної торгівлі, визначення умовного показника популярності товару та автоматизована рекомендація найкращого товару для клієнтів інтернет магазину .

Ключові слова: система електронної торгівлі, клієнт, товар, метод.

Виконавець: студент 4 курсу, група КН-18-1  М.І. Власюк
Курс, група виконавця Підпис Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
1.1 Аналіз інформаційних моделей	6
1.2 Огляд теоретичних підходів до розв’язку подібних задач	12
1.3 Аналіз існуючих програмних рішень.....	16
1.4 Аналіз сучасних засобів створення програмного забезпечування	18
1.5 Мета, задачі та вимоги до реалізації інформаційної системи	20
Розділ 2 Проєктування інформаційної системи	21
2.1 Аналіз та автоматизація обробки потоків інформації.....	21
2.1.1 Моделі, методи, інформаційна технологія системи	21
2.1.2 Функціональна структура інформаційної системи	22
2.2 Інформаційна структура системи	24
2.2.1 Проектна архітектура системи та взаємозв’язок компонентів.....	24
2.2.2 Інформаційна модель.....	26
2.3 Вибір засобів розробки інформаційної системи	36
2.3.1 Вибір мови програмування	36
2.3.2 Вибір фреймворку.....	37
2.3.3 Вибір редактора програмного коду.....	37
2.3.4 Вибір системи керування базами даних	37
Розділ 3 Програмна реалізація інформаційної системи	41
3.1 Структура та функціональне призначення програмних складових системи.....	41
3.2 Особливості реалізації програмних складових системи	44
3.3 Тестування інформаційної системи	53
3.4 Особливості розробки методу формування оптимальних наборів пропозицій	55
3.5 Вимоги до розгортання інформаційної системи.....	59
Висновки	60
Перелік посилань.....	61
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База інформації
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
ПЗ	Пояснювальна записка
ПП	Програмний продукт
СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ	Система керування базами інформації
ХНУ	Хмельницький національний університет.
MS	Microsoft
SCADA	Supervisory Control And Data Acquisition System
CIL	Common Intermediate Language
МОВА РОЗРОБКИ PHP	Hypertext Preprocessor
ООП	Об'єктно-орієнтоване програмування

Вступ

Стрімке зростання обсягу доступної цифрової інформації та кількості відвідувачів Інтернету створило потенційну проблему інформаційного навантаження, що перешкоджає своєчасному доступу до елементів, що цікавлять в Інтернеті. Інформаційно-пошукові системи, такі як Google, DevilFinder та Altavista, частково вирішили цю проблему, але пріоритизація та персоналізація (коли система зіставляє доступний контент з інтересами та уподобаннями користувача) інформації були відсутні. Це збільшило попит на рекомендаційні системи більше, ніж будь-коли раніше [1].

Рекомендаційні системи – це системи фільтрації інформації, що вирішують проблему інформаційного навантаження шляхом фільтрації фрагмента важливої інформації з великого обсягу динамічної-генерованої інформації відповідно до переваг, інтересів або поведінкою користувача, що спостерігається, щодо об'єкта. Рекомендаційна система може передбачити, чи віддасть перевагу конкретний користувач той чи інший елемент, на основі профілю користувача .

Рекомендаційні системи вигідні як постачальникам послуг, і користувачам. Вони знижують транзакційні витрати на пошук і вибір товарів серед онлайн-покупок. Системи рекомендацій також довели свою ефективність у покращенні процесу та якості прийняття рішень. В умовах електронної комерції рекомендаційні системи збільшують доходи, оскільки є ефективним засобом продажу більшої кількості продуктів .

В наукових бібліотеках рекомендаційні системи підтримують користувачів, дозволяючи їм вийти за рамки пошуку за каталогом. Отже, неможливо переоцінити необхідність використання ефективних та точних методів рекомендацій у рамках системи, яка надаватиме релевантні та надійні рекомендації для користувачів [2].

Мета: Створення спеціального методу для підбору оптимальних рішень в електронній комерції, для полегшення фільтрації та рекомендації потрібних товарів, для клієнтів інтернет магазинів чи пересічних осіб.

Об'єкт: Процес автоматизації підбору оптимальних рішень в електронній комерції.

Предмет: Методи обробки інформації у системах електронної торгівлі.

1.1 Аналіз інформаційних моделей

Системи обробки інформації та інформації - повний набір операцій (збір, введення, запис, перетворення, зчитування, зберігання, знищення, реєстрація), що виконуються за допомогою апаратних і програмних засобів, включаючи обмін каналами передачі інформації. [3].

З модернізацією програмного забезпечення з'явилося багато різноманітних програмних засобів обробки інформації, які написані різними мовами програмування на основі вищевказаних методів. Різноманітність ПП пов'язана з конкретними обставинами кожної переробної галузі. Наприклад, в обробці графічних зображень широко використовуються методи розпізнавання зображень і криптографічні методи, засновані на перетворенні Фур'є. [3].

Не кожна компанія може дозволити собі замовити програму у розробника програмного забезпечення, яка дозволить ефективно обробляти інформацію, що стосується сфери діяльності компанії. Такий підхід навіть бажаний, оскільки автоматизовані системи обробки базуються на конкретних базах даних, структура яких може сильно відрізнятися від одного бізнесу до іншого, не кажучи вже про від однієї галузі до іншої. Одним з найпоширеніших методів обробки інформації є пакет Microsoft Office, оскільки він встановлюється практично на кожному комп'ютері. Якщо користувач не може працювати безпосередньо в програмному середовищі конструкторського бюро, його можливості досить широкі, але примітивні. Інструменти, доступні широкому колу споживачів, включають організацію бази даних, запити та пошук інформації, фільтрацію інформації, графічне представлення тощо.

Сьогодні, у зв'язку зі світовою популярністю комп'ютерних систем у сфері автоматизації промислових процесів, все ширше використовуються системи збору та управління операцією та даними (SCADA-Supervisory Control And Data Acquisition System). SCADA – це лише один із компонентів системи керування автоматизацією, яка на даний момент являє собою складний набір програмного та апаратного забезпечення. Переважна більшість систем

автоматизації базується на промислових контролерах, які є основним засобом збору, обробки інформації, регулювання параметрів процесу, сигналізації, захисту та блокування (нижній шар системи).

Оброблена контролером інформація передається в комп'ютеризовану систему, яка є робочим місцем оператора-техніка, де інформація процесу надалі обробляється і представляється оператору в інтуїтивному вигляді (топ АСУ ТП). Системи SCADA знаходяться на найвищому рівні в ієрархії програмного та апаратного забезпечення промислової автоматизації. Якщо спробувати коротко описати основні функції, то можна сказати, що система SCADA збирає інформацію про процес, забезпечує інтерфейс з оператором, зберігає історію процесу та керує процесом у міру необхідності. Система SCADA — це система диспетчерського контролю та збору даних (Supervisory Control And Data Acquisition). Це набір контрольно-контрольних пристроїв і спосіб взаємодії з технічними об'єктами. Сьогодні цей термін позначає набір програмно-технічних засобів, що використовуються для виконання роботи оператора.

При створенні інформаційної системи велика кількість локальних підсистем об'єднується в функціональний вузол, і ці функціональні вузли часто мають різні програмні інтерфейси. Це значно ускладнює завдання узгодження таких підсистем і уповільнює роботу системи в цілому. Тому локальні функціональні блоки рекомендується підключати до однієї програмної платформи. SCADA – система реалізує цей підхід, тому її використання зараз широко поширене та актуальне. Апаратне забезпечення також розроблено безпосередньо під програмний комплекс, що дозволяє створювати інформаційні системи дешевше та за мінімальний час.

На даному етапі все більшого розвитку також набувають методи людино-орієнтованої комп'ютерної обробки інформації.

Обробка інформації може здійснюватися за допомогою набору комп'ютерів та інформаційних технологій, які обґрунтовано виконують чотири основних набори операцій: збирають інформацію та передають її на ПК, обробляють за попередньо розробленими алгоритмами та програмами,

публікують оброблену інформацію для користувача. , для зберігання та отримання інформації. Залежно від експлуатаційних можливостей і ступеня застосування зазначеного обладнання при виконанні цих операцій розрізняють часткову і комплексну механізацію операцій, їх часткову і повну автоматизацію. У практиці та літературі повну або комплексну автоматизацію інформаційних операцій називають машинними системами або автоматизованими системами обробки інформації.

Експлуатаційні можливості сучасних технологічних засобів, що використовуються при автоматичному зборі та обробці інформаційних систем, дозволяють автоматизувати деякі процедури в цих функціях. Стан науково-практичної розробки та технічний рівень вищезазначеного комплексу визначають можливість автоматизації реалізації таких процедур процесу управління:

- У прогнозуванні та плануванні - Багатоваріантні розрахунки виконуються під час розробки прогнозів, довгострокових і поточних економічних і соціальних планів розвитку бізнесу, а також операційних і виробничих планів і планів готовності до технології виробництва для подальшого визначення найкращого набору відповідних погодинних планів. З точки зору показників (години, зміни, тижні тощо) та об'єктів (робоче місце, поле тощо):;

- В організації - змодельуйте організаційну структуру, яка керує та імітує виробничий процес за різними критеріями та параметрами, щоб вибрати найкращий;

- Координація та нагляд - Подання наказів на робоче місце (поки що на низовому рівні управління виробництвом) на основі планів, процесів або інструкцій, встановлених для певних видів робіт або операцій;

- в контролі - стежити за станом всіх параметрів керованого об'єкта, а також за своєчасним і повним виконанням команд управління;

- Бухгалтерський облік - збір (у виробничій каденції) і систематична обробка всієї фактичної (у тому числі довідкової, планової, нормативної та

іншої) достовірної інформації про наявність і рух ресурсів, а також про поточні стани, процеси і явища у виробництві та економіці підприємства. в очікуванні;

– При аналізі - порівняння нормативних, планових і фактичних показників, що характеризують окремі методи та функції чи процеси виробничо-господарської та іншої діяльності, виявлення відхилень від інформаційних параметрів, що вказують на причини та винуватців (за кількістю, вартістю, відносними та іншими кількісними показниками) серед них. відхилення, оцінити виконання плану в різних аспектах і виявити фактори, що впливають на ці відхилення;

– З точки зору звітності – за допомогою спеціального масиву перекладів – каталогів – автоматично формують (на основі основної інформації) зведені показники у типовій формі для бухгалтерської, статистичної та іншої звітності, а водночас створюють машинні носії зі зведеними звітні показники для передачі за допомогою каналів зв'язку найвищого рівня для зовнішніх агенцій (установ).

Як відомо, в умовах роботи системи автоматичного збору та обробки економічної інформації головною основою наукового прогнозування та планування є обґрунтовані норми та стандарти, які застосовуються до всіх процесів і явищ, що відбуваються на цільовому об'єкті. керувати. Їхні процедури розробки, аналізу та синтезу також повністю автоматизовані [4].

Рекомендаційна система визначається як стратегія прийняття рішень користувачами у складних інформаційних середовищах. Крім того, рекомендаційна система була визначена з точки зору електронної комерції як інструмент, який допомагає користувачам шукати записи цікавості, пов'язаних з інтересами та уподобаннями користувачів. Рекомендаційна система була визначена як засіб допомоги та розширення соціального процесу використання рекомендацій, для іншого прийняття рішень, коли немає достатніх особистих знань чи досвіду альтернатив. Рекомендаційні системи вирішують проблему інформаційного навантаження, з яким зазвичай стикаються користувачі, надаючи їм персоналізований, ексклюзивний контент та рекомендації щодо послуг. Останнім часом були розроблені різні підходи до створення

рекомендаційних систем, які можуть використовувати спільну фільтрацію, фільтрацію на основі контенту або гібридну фільтрацію [2].

У сфері електронної комерції існує кілька систем. Такі системи можна розділити на такі категорії за об'єктами та суб'єктами, що є їх складовими елементами:

– Системи B2B (комерційний бізнес) охоплюють усі рівні та види взаємодії між суб'єктами - взаємодії юридичних осіб (виробників і споживачів, продавців і покупців) у розподілі, обміні, купівлі-продажу, споживанні. Система багато в чому типова для електронної комерції.

– Системи B2C (бізнес-споживачі) характеризуються тим, що учасниками бізнес-процесу є фізичні та юридичні особи (торговельні компанії). Взаємодія між ними заснована на електронних технологіях і покликана забезпечити купівлю товарів в інтернет-магазинах або інших формах електронної комерції. Ядром системи B2C, безсумнівно, є електронна роздрібна торгівля.

– Система G2B (адміністративно-ділова) є послугою державного замовлення. Система охоплює всі види операцій між компаніями та державними організаціями. Йдеться про торгові угоди на закупівлю товарів, продуктів, послуг для державних потреб та державних коштів.

– Система C2C (споживач-споживач) відрізняється тим, що споживачі — фізичні та юридичні особи — взаємодіють у них для обміну діловою інформацією та обміну товарами, досвідом та послугами. Система супроводжує аукціонні операції, консигнаційні операції та бартерні угоди між фізичними особами.

Формуються також інші системи, але вони менше розповсюджені і не такі масштабні, як перелічені вище.

Електронні вітрини — це проста організаційна форма електронної роздрібною торгівлі в системі B2C. Він може бути окремим або частиною інтернет-магазину. Електронна вітрина – це спеціалізований веб-сайт, який містить інформацію про перелік товарів, що пропонуються покупцям, і порядок

їх замовлення. У цьому випадку метою веб-присутності є збір в режимі реального часу попередніх замовлень на її реалізацію.

Інтернет-магазин – це спеціалізований веб-сайт, який за допомогою електронного каталогу в інтерактивному режимі забезпечує повний цикл купівлі-продажу товарів через Інтернет. Тому інтернет-магазин надає такі операції: розпізнавання, підбір товарів, замовлення, розрахунки, відстеження виконання замовлення.

Для функціонування Інтернет-магазину необхідно мати як мінімум такі програмно-апаратні компоненти:

- Інтернет-вітрину (фронт-офіс) на Веб -сервері;
- електронні каталоги;
- електронну платіжну систему;
- інформаційну систему, інтегровану у фронт-офіс, тобто бек-офіс (склад, бухгалтерія, відділ доставки та ін.).

Вітрина інтернет-магазину має активний контент. Він може бути статичним, заснованим на звичайному файлі HTML, або динамічним - відображати інформацію з бази даних.

Інтернет-присутність містить інформацію про назву, профіль, статус магазину, асортимент товарів, способи оплати, знижки, порядок доставки товару, гарантії, а також електронний каталог, посібник для пошуку товарів і послуг з наступними характеристиками: назва ; призначення; ціна ; виробник; конкурентоспроможність тощо. По суті, це систематичний веб-сайт продукту з доступною системою навігації.

Відрізняють кілька типів каталогів:

- загальний, що забезпечує широкий список різних сайтів;
- комерційний, який концентрує увагу на комерційних характеристиках ринку постачальників, конкурентному середовищі.
- спеціалізований, як правило, присвячений одній вузькій тематиці.

Електронні розрахункові системи є невід’ємною частиною інфраструктури електронної комерції. Вони являють собою сукупність

платіжних засобів, способів діяльності фінансових суб'єктів, договорів, нормативних актів. Особливе значення в цій системі мають платіжні інструменти, зокрема готівкові розрахунки за доставку товарів, банківські перекази, пластикові картки, смарт-картки, електронні гроші, електронні чеки тощо. [4].

1.2 Огляд теоретичних підходів до розв'язку подібних задач

Об'єктно-орієнтоване програмування (ООП, іноді об'єктно-орієнтоване програмування, раніше об'єктно-орієнтоване програмування; «Об'єктно-орієнтоване програмування» (ООП) — це парадигма програмування, яка розглядає програму як набір «об'єктів», які взаємодіють один з одним. Основа PLO складається з чотирьох основних концепцій: інкапсуляції, успадкування, поліморфізму та абстракції.

Основною перевагою ООП є оптимальна модульність програмного забезпечення (тисячу функцій мови програмування можна замінити в ООП десятками класів зі своїми методами). Хоча ця парадигма виникла в 1960-х роках, вона не знайшла широкого застосування до 1990-х років, коли розвиток комп'ютерів і комп'ютерних мереж зробив можливим писати надзвичайно велике та складне програмне забезпечення, що призвело до модифікованого підходу до написання програм. Сьогодні багато мов програмування або підтримують ООП (мова розробки PHP, Lua), або повністю об'єктно-орієнтовані (включаючи Java, C#, C++, Python, Ruby і Objective-C, ActionScript 3, Swift тощо).

Спадкування - клас може мати "підкласи" суперкласів, спеціальні, розширені версії. Можна навіть сформуванати ціле спадкове дерево. Наприклад, клас Dog може мати такі підкласи, як пекінес, вівчарка тощо. Так, Сірко може бути екземпляром класу Пастух. Підкласи успадковують властивості та поведінку своїх суперкласів і можуть вводити власні. Спадкування може бути одиничним (один безпосередній батьківський клас) і множинним (кілька

батьківських класів). Це залежить від вибору програміста, який реалізує клас і мову програмування. Наприклад, у Java дозволено лише одне успадкування, тоді як у C++ дозволено обидва.

Приховування інформації (інкапсуляція) – приховування деталей про роботу класів від об'єктів, що їх використовують або надсилають їм повідомлення.

Абстракція — це спрощення складної реальності шляхом моделювання відповідних задач класів і використання найбільш прийняттого рівня деталізації деяких аспектів проблеми..

Поліморфізм відноситься до залежності поведінки від класу, який викликав цю поведінку, тобто два або більше класів можуть по-різному реагувати на одне повідомлення. Залежно від того, що передається, виберіть відповідну підпрограму. [5].

У рамках архітектурного шаблону на основі ООП Метод-вигляд-контролер (MVC) програма поділена на три окремі, але взаємопов'язані частини відповідно до функціонального розподілу між компонентами. За зберігання інформації та її структуру відповідають моделі. Представлення відповідає за представлення цієї інформації користувачеві, інтерфейсу програми. Контролер керує компонентом, отримує сигнали у відповідь на дії користувача (зміна положення курсору миші, натискання кнопки, введення інформації в текстове поле) і передачу даних методу.

Метод є центральним компонентом шаблону MVC і відображає поведінку застосунка, незалежну від інтерфейсу користувача. Метод стосується прямого керування інформацією, логікою та правилами застосунку.

Активна модель – відстежує зміни в моделі та реагує на них.

Пасивна медель – оновлюється через контролер.

Подання може бути будь-яким поданням інформації, отриманої на виході, наприклад графіком або діаграмою. Декілька типів (уявлень) однієї і тієї ж інформації можуть існувати одночасно, наприклад, гістограми для управління компанією та електронні таблиці для бухгалтерського обліку.

Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Метод інкапсулює ядро інформації і основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу інформації.

Вигляд може мати декілька взаємопов'язаних областей, наприклад різні таблиці і поля форм, в яких відображаються дані.

Функція контролера – відстежувати певні події, які відбуваються внаслідок дій користувача. Контролери дозволяють структурувати код, групуючи пов'язані дії в окремі класи. Наприклад, типовий проект MVC може мати користувацький контролер, який містить набір методів для керування обліковими записами користувачів, таких як реєстрація, вхід, зміна профілів та зміна паролів.

Зареєстровані події транслуються в різних запитах, які надсилаються компонентам моделі або об'єктам, відповідальним за відображення інформації. Відокремлення моделей від інформаційних типів дозволяє використовувати різні компоненти незалежно для відображення інформації. Таким чином, якщо користувач вносить зміни в інформаційну модель через контролер, інформація, надана одним або кількома компонентами візуалізації, автоматично адаптується до змін, що відбуваються. [6].

Сучасні інформаційні системи використовують Систему управління базами даних для забезпечення використання баз даних. Система управління базами даних — це програмно-апаратна система, яка забезпечує визначення, створення, функціонування, контроль, керування та використання баз даних. Програми баз даних можуть бути частиною системи керування базами даних або окремими. Найпопулярнішими системами керування базами даних є MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird і IBM DB2..

Через зв'язок між базами даних і системами керування базами даних термін «база даних» іноді використовується необґрунтовано та неточно для позначення систем управління базами даних. Але необхідно розрізняти бази даних (сховища інформації) і системи управління базами даних (засоби роботи з

базами даних). Систему управління базою даних можна видалити з інформаційної системи, але база даних буде продовжувати існувати. І навпаки, система управління базами даних може працювати без будь-якої бази даних.

Загалом, ви не можете просто перемістити базу даних з однієї системи керування базами даних в іншу. Але системи керування базами даних використовують стандарти (SQL, ODBC, JDBC), які спрощують багато маніпуляцій з інформацією та дозволяють різним програмам використовувати бази даних із різних систем керування базами даних. Системи управління базами даних часто класифікують відповідно до моделей організації інформації. Здебільшого в системах управління базами даних використовується реляційний підхід, коли дані представлені у вигляді таблиці. Кінцеві користувачі (і програми) не можуть використовувати базу даних безпосередньо. Усі інформаційні операції здійснюються за допомогою спеціальних запитів, що надсилаються в систему управління базами даних. Система управління базами даних обробляє їх і повертає результати. Тільки системи керування базами даних використовують базу даних безпосередньо.

Сучасні системи керування базами даних забезпечують функції щодо керування інформацією, які можна поділити на такі групи:

- Оголошення інформації – відтворення, заміна та видалення визначень, які описують організацію інформації.
- Змінення інформації – додавання інформації, їх заміна та видалення.
- Одержання інформації – відтворення інформації за запитом застосунку у формі, яка дозволяє їх безпосереднє використання.
- Адміністрування інформації – реєстрування та відслідковування дій користувачів, дотримання безпеки роботи з інформацією, забезпечування надійності та цілісності інформації, моніторинг продуктивності, резервне копіювання та відновлення інформації тощо [7].

Отже, під час роботи над дипломним проектом буде використовуватись об'єктно-орієнтовані елементи, архітектурний шаблон проектування MVC, фреймворку Laravel, а також Система керування базами даних MySQL.

1.3 Аналіз існуючих програмних рішень

Спільні рекомендаційні системи були реалізовані у різних галузях застосування.

Spotify – це користувацька система CF, яка дає рекомендації музичних альбомів і виконавців. У Spotify, коли користувач спочатку входить в систему, користувачеві надається список із 125 виконавців, які він може оцінити відповідно до того, наскільки йому подобається їх слухати. Список складається з двох різних розділів. Перший сеанс складається з найбільш часто оцінюваних виконавців, і це дає активному користувачеві можливість оцінити виконавців, яких інші оцінили однаково, так що існує рівень подібності між профілями різних користувачів. Другий сеанс генерується на основі випадкового вибору елементів із всієї матриці елементів користувача, так що всі виконавці та альбоми в кінцевому підсумку оцінюються в певний момент на початкових етапах оцінювання [8].

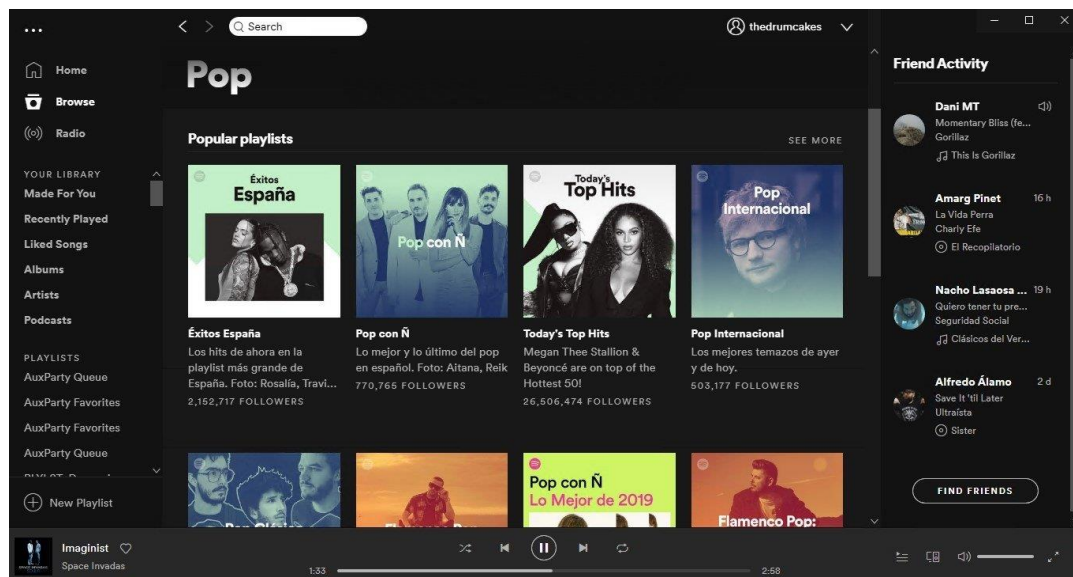


Рисунок 1.1 Зовнішній вигляд мобільного додатку Spotify

Amazon використовує алгоритми диверсифікації тем для покращення рекомендацій. У системі використовується метод спільної фільтрації для подолання проблеми масштабованості шляхом створення таблиці схожих

елементів в автономному режимі з використанням матриці елементів. Потім система рекомендує інші продукти, які схожі онлайн відповідно до історії покупок користувачів [9].

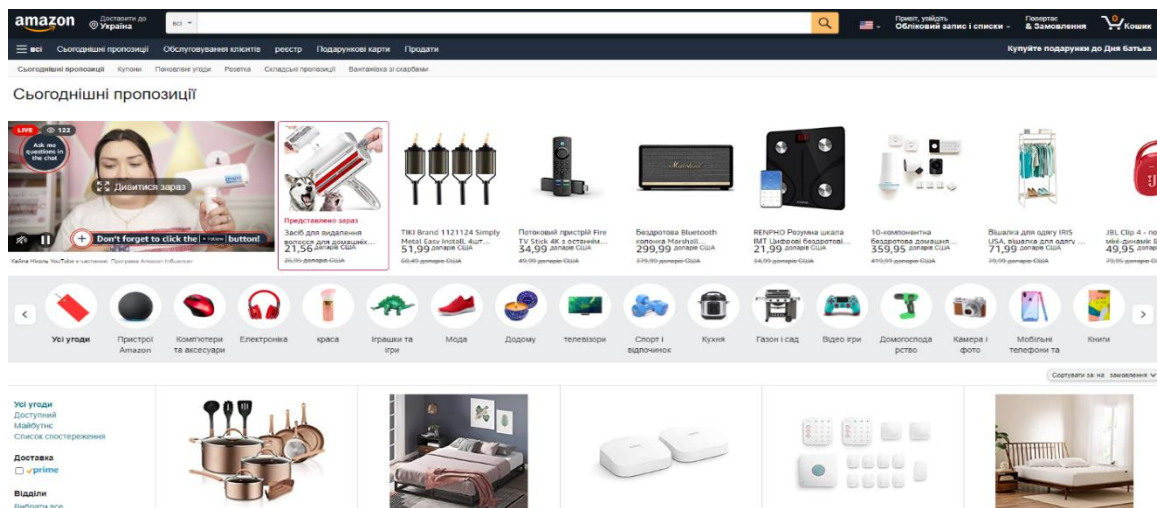


Рисунок 1.2 Користувацький інтерфейс системи електронної торгівлі Amazon

Таблиця 1.1 Порівняння застосунків з інтегрованими рекомендаційними системами

	Spotify	Amazon
Рекомендаційні списки	Присутні	Присутні
Кросплатформність	Тільки як мобільний додаток	Доступність на усіх можливих платформах
Панель адміністрування	Присутні	Присутні
Особистий кабінет	Присутні	Присутні
Система оцінювання	Присутні	Присутні
Пошук товарів з використанням елементів фільтрації	Присутні	Присутні

1.4 Аналіз сучасних засобів створення програмного забезпечення

Веб-продукт — розподілений веб-продукт, де користувач є браузером, а сервер — веб-сервером. Веб-браузер може бути реалізацією так званого тонкого клієнта - логіка програми централізована на сервері, а функція браузера в основному полягає у відображенні інформації, завантаженої з сервера мережею, і повернення інформації користувача. Однією з переваг такого підходу є те, що клієнт не прив'язаний до конкретної операційної системи користувача, тому веб-продукт є мультиплатформним сервісом. Незважаючи на всю універсальність і відносну простоту розробки, мережеві продукти стали широко популярними в кінці 1990-х і на початку 2000-х років.

Веб-продукт, додаток, додаток або додаток — це програма користувача, яка дозволяє вирішувати багато конкретних прикладів завдань користувача. Ця концепція була введена, щоб підкреслити відмінність від операційних систем, драйверів, бібліотек тощо (які забезпечують функціональність комп'ютерної системи та підтримують її повну функціональність), а також інструментів та середовищ розробки [10].

На даний момент для розробки веб застосунків можна вибрати наприклад із 2 існуючих платформ: .NET та Мова розробки PHP.

Microsoft .NET – програмна технологія, представлена фірмою Microsoft як платформа для створення, звичайних програм, та веб-продуктів. Багато у чому є продовженням ідей та базових принципів, вкладених в технологію Java. Однією з ідей .NET є кроссрозробність та сумісність з різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

.NET – крос-платформна технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проекті Mono (в рамках угоди між Microsoft з Novell), DotGNU.

Захист авторських прав відноситься до створення середовищ здійснювання (CLR Common Language Runtime) для програм .NET. Компілятори для .NET випускаються багатьма фірмами для різних мов вільно.

.NET поділяється на дві основні частини – середовище здійснювання (по суті віртуальна машина) та інструментарій розробки.

Середовища розробки .NET-програм: Visual Studio .NET (C++, C#, J#), Sharp Develop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse має додаток для розробки .NET-програм. Застосовні програми також можна розроблювати в текстовому редакторі та використовувати консольний компілятор.

Як і технологія Java, середовище розробки .NET створює байт-код, який призначений для реалізації віртуальною машиною. Рідна мова введення в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або скорочено IL. Сучасні методи динамічної компіляції дозволяють досягти таких же рівнів швидкості, як і традиційні «статичні» компілятори (наприклад, C++), і проблеми швидкості часто залежать від якості компілятора. [11].

Також до цієї платформи належить технологія ASP.NET, яку в подальшому можна використовувати для розширення системи обліку.

Мова розробки PHP (Hypertext Preprocessor – мова розробки PHP: гіпертекстовий препроцесор). мова розробки PHP підтримуються більшістю хостинг-провайдерів. мова розробки PHP – проект відкритого програмного забезпечення .

Мова розробки PHP інтерпретується веб-сервером як HTML-код, який передається клієнту. На відміну від мови сценаріїв JavaScript, мова розробки коду PHP невидима для користувача, оскільки веб-браузер отримує завершений html-код. Це перевага безпеки, але вона знижує інтерактивність сторінки. Але ніхто не забороняє використовувати мову розробки PHP для створення коду JavaScript, який уже працює на стороні клієнта.

На даній платформі Мова розробки PHP, на відміну від .NET в якості бази інформації необхідно використовувати систему керування базами даних , MySQL замість MS SQL Server у .NET [12].

Згідно цього аналізу, можна зробити висновок що МОВА РОЗРОБКИ PHP найбільше підходить для розробки даного методу, тому що він має змогу працювати з усіма найпопулярнішими мовами світу, і є одним із найлегших та доступніших для розуміння і розробки програмного забезпечення .

1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Основною метою кваліфікаційної роботи бакалавра є розробка методу формування оптимальних наборів пропозицій в системах електронної торгівлі, на платформі PHP. Для досягнення мети ставляться наступні завдання:

- Провести аналіз систем формування наборів пропозиції за критеріями максимальної відповідності та популярності.
- Розробити метод формування наборів пропозицій.
- Розробити метод формування пропозицій на основі товарів широкого вжитку.
- Реалізувати інформаційну технологію у вигляді програмного модуля пропозицій.
- Провести експериментальне дослідження ефективності на основі виборів з пропонованих наборів

Розділ 2 Проектування інформаційної системи

2.1 Аналіз та автоматизація обробки потоків інформації

2.1.1 Моделі, методи, інформаційна технологія системи

Метод формування оптимальних наборів пропозицій в системах електронної торгівлі призначений для полегшення навігації та фільтрації товару, зменшення часу пошуку потрібного рішення у виборі товару який може сподобатись клієнтові.



Рисунок 2.1 Використання методу підбору оптимальних наборів пропозицій у системі електронної торгівлі

Кроки здійснювання методу :

– Перший елемент : Зареєстрований користувач переглядає список товарів та їхній детальний перегляд з можливістю голосування, та виставлення рейтингу товару.

– Другий елемент: Після того як користувач у формі голосування виставив рейтинг заданому товару, данні з цієї форми передаються у базу інформації.

– Третій елемент:

$$P(H_i|A) = \frac{P(H_i)+P(A|H_i)}{P(A)} \quad (2.1)$$

$P(H_i|A)$ – це коефіцієнт рейтингу товару, який розраховується за допомогою голосування та кількості відвідування, клієнтами даного товару

$P(H_i)$ – це оцінка рейтингу товару яку обирають клієнти під час голосування.

$P(A|H_i)$ – це базовий коефіцієнт рейтингу товару який виставляє адміністратор/менеджер магазину для кожного товару.

$P(A)$ – це коефіцієнт кількості відвідування даного товару користувачами.

– Четвертий елемент : Після обробки отриманих інформації метод формує список оптимальних рішень(каталог із декількох найпопулярніших товарів), видає його користувачеві для ознайомлення та подальшого прийняття заключного рішення.

Отже створення методу формування оптимальних наборів пропозицій дозволить автоматизувати відповідні функціональні елементи системи електронної торгівлі та покращить взаємодію користувача з сайтом.

2.1.2 Функціональна структура інформаційної системи

Натрапивши в інтернеті на інтернет магазин, користувач може зайти та переглянути позицію яка йому сподобалась у цьому магазині, після чого він

може купити цей товар, додати у свій кошик, додати до улюблених товарів, та проголосувати за нього у наданні рейтингу популярності даного товару.

Після чого цей товар може бути розміщений на головній сторінці як один із рекомендованих найпопулярніших товарів серед усього асортименту, або рекомендації по особистому підбору клієнта за його вподобаннями.

Відповідно до даної проблематики були реалізовані наступні групи функцій.

Для отримання повного набору функцій користування інтернет магазину, звичайному пересічному клієнту потрібно пройти реєстрацію та авторизацію, після чого він зможе робити покупки в магазині редагувати особистий кабінет та багато іншого.

На діаграмі 2.2 схематично зображено роботу даного бізнес процесу.

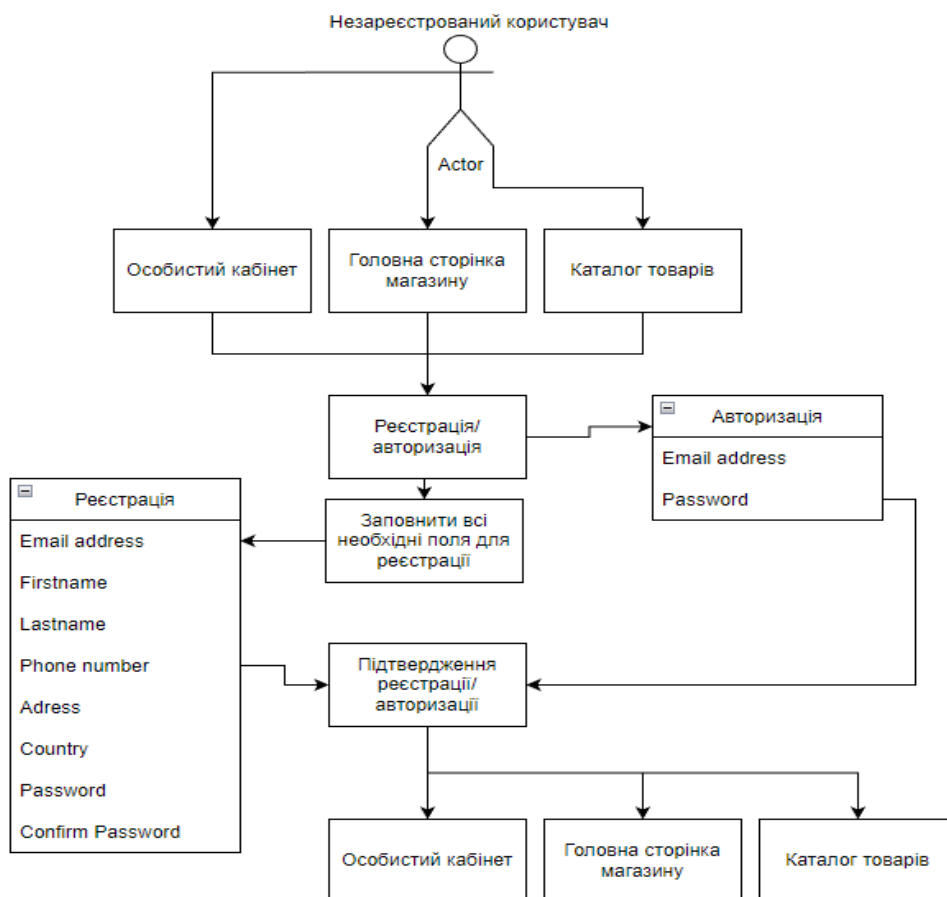


Рисунок 2.2 Діаграма активності реєстрації/аторизації та первинних можливостей клієнта.

Після авторизації клієнт може зайти на головну сторінку та переглянути рекомендації товарів, як особисті так і загальні, якщо користувач захоче змінити свої вподобання та оновити свій рекомендаційний список йому досить знайти потрібну позицію та проголосувати за нього в рейтинговому голосуванні.

На діаграмі 2.3 схематично зображено роботу даного бізнес процесу.

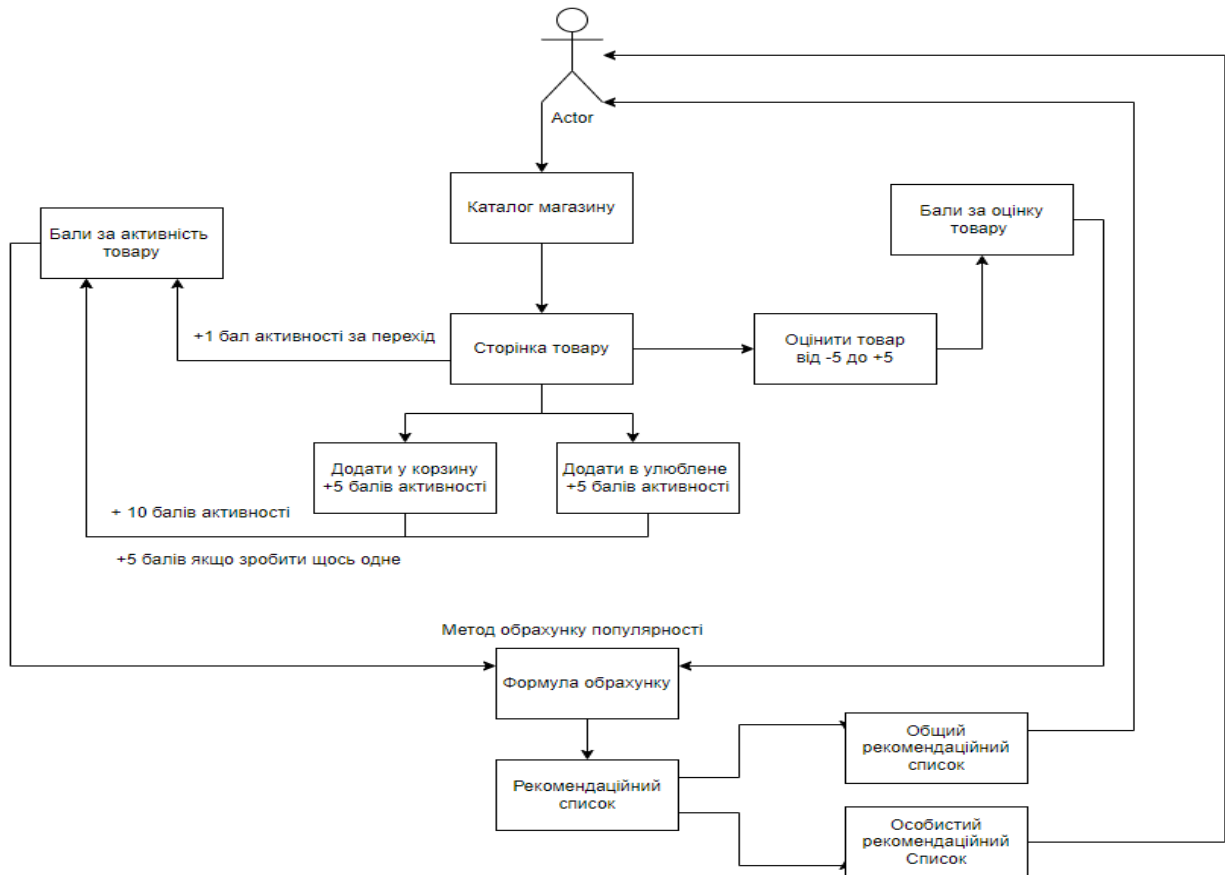


Рисунок 2.3 Діаграма активності обрахунку популярності та взаємодії з користувачем

2.2 Інформаційна структура системи

2.2.1 Проектна архітектура системи та взаємозв'язок компонентів

Для безперервної передачі інформації, роботи функцій та методу формування оптимальних наборів пропозицій в системах електронної торгівлі представлено наступну архітектуру(рис 2.4)

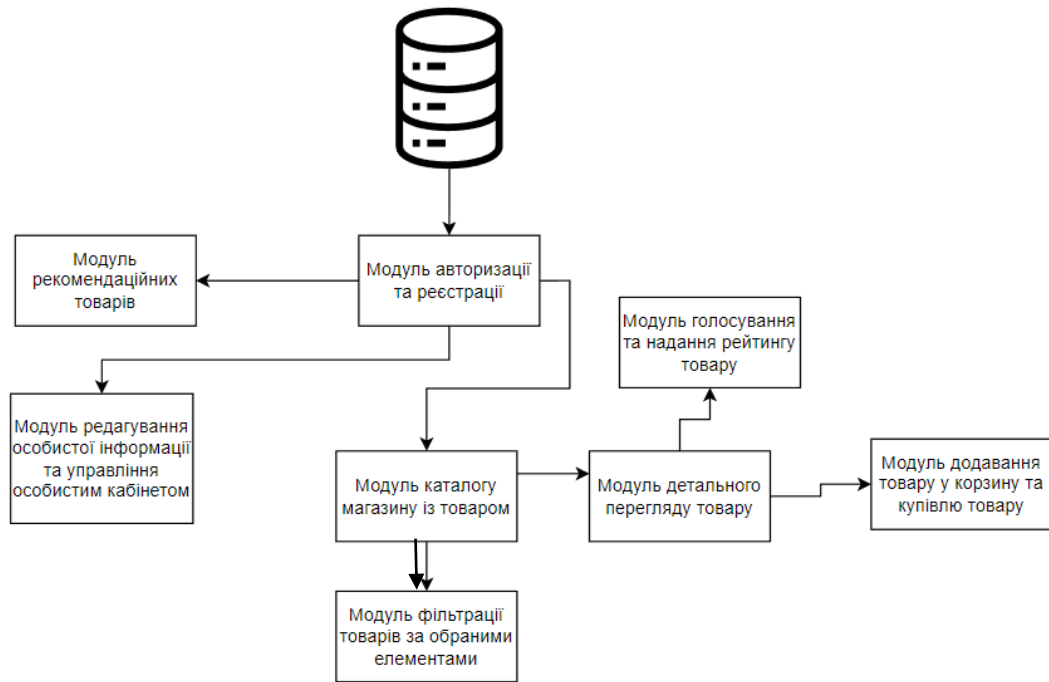


Рисунок 2.4 Архітектура управління системою електронної торгівлі та методу формування оптимальних наборів пропозицій

– Модуль авторизації та реєстрації дозволяє зареєструватися новому користувачу у системі, після чого авторизуватися та підтвердити свої данні для користування усіма користувацькими функціями інтернет магазину.

– Модуль рекомендаційних товарів дозволяє швидку фільтрацію контенту та представлення клієнту інформації про найпопулярніші позиції.

Таблиця 2.1 – Групи функцій

№ п/п	Групи функцій
1	Реєстрація та авторизація користувача
2	Обрахунок популярності та взаємодії з користувачем

– Модуль заміна особистої інформації та управління особистим кабінетом дозволяє змінювати особисту інформацію у вкладці особистий кабінет також переглянути історію своїх покупок та змінити свої улюблені позиції на інші за бажанням.

– Модуль каталогу магазину із товаром дає можливість переглянути весь наявний асортимент у магазині.

– Модуль фільтрації за обраними елементами дозволяє відсіяти товар по брендам, розмірам, та ціні позиції.

– Модуль детального перегляду товару дозволяє детальніше переглянути обраний товар, ближче подивитись на фото, прочитати його характеристику, обрати потрібний розмір та кількість.

– Модуль голосування та представлення рейтингу товару дозволяє поставити рейтингову оцінку обраному товару

– Модуль додавання товару у корзину та купівлю товару дозволяє додати товар у ваш товарний кошик або відразу купити обраний вами товар.

2.2.2 Інформаційна модель

Для розробки структури бази інформації для метода підбору оптимальних рішень в системах електронної торгівлі необхідно оприділити сутності поставленої предметної області та їх властивості.

На рисунку 2.5 зображена дата логічна Метод бази інформації інтернет-магазину, яка передбачає такі таблиці:

- Tags (категорія),
- Size (розміри одягу),
- Roles(Ролі користувачів),
- Product (Товари які є в магазині),
- Orders (Замовлення),
- Type_img (Тип картинок «Це табличка довідник»),
- Img_content (Всі картинки які не є товаром),

- History_orders (Історія замовлень),
- Favorites (Улюблені товари),
- Collections (Колекції),
- Checkouts (Оформлення замовлення),
- Brands (Бренди товарів),
- User(Користувачі),
- Popularity(Популярність),

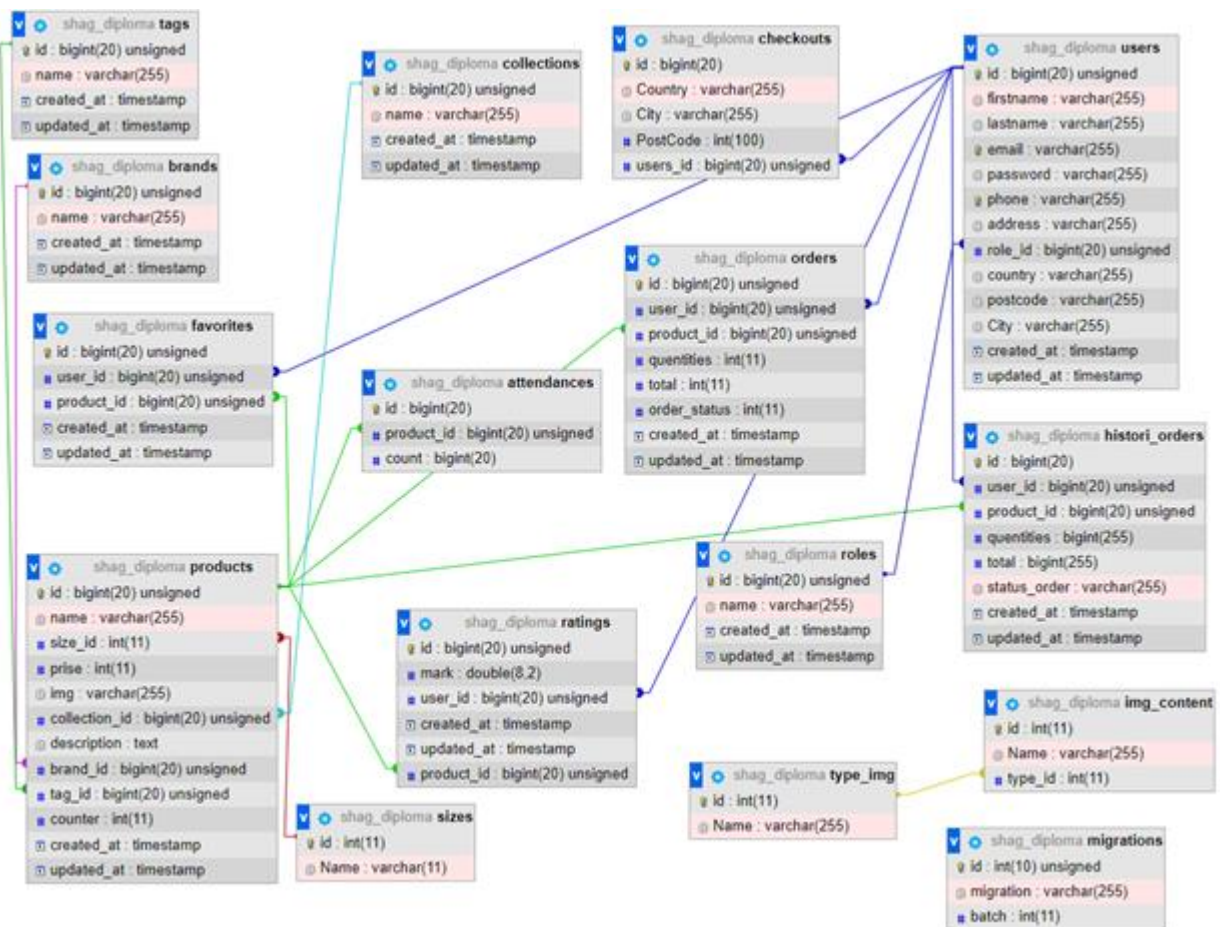


Рисунок 2.5 – Даталогічна модель бази інформації веб застосунку

Кожна з таблиць Бази інформації призначена для збереження інформації окремих сутностей. Властивості сутностей є атрибутами таблиць. Наприклад, сутність

«Користувачі» (рисунок 2.6) має властивості: прізвище, ім'я, дата та час реєстрації та видалення з бази, номер телефону, логін, пароль, адреса.

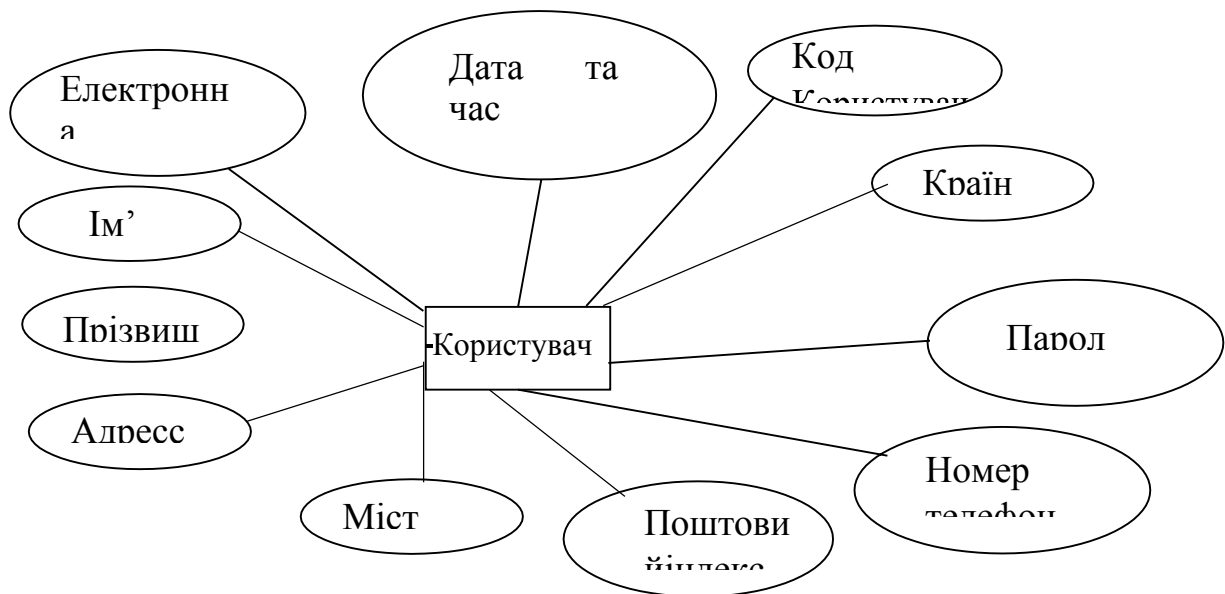


Рисунок 2.6 - Властивості сутності " Користувач"

Таблиця «Type_img» призначена для збереження інформації про картинка які не є товарами (таблиця 2.2).

Таблиця 2.2 – Атрибути таблиця «Type_img»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	name	VARCHAR	Назва типу картинки

Таблиця «Sizes» призначена для збереження інформації про розміри товарів (таблиця 2.3).

Таблиця 2.3 – Атрибути таблиця «Sizes»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	name	VARCHAR	Назва розміру

Таблиця «User» призначена для збереження інформації про користувачів.
(таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «User»

№ п/п	Назва	Тип інформації	Опис
1	ID	BIGINT	Унікальний ідентифікатор
2	email	VARCHAR	Електронна адреса/Також є логіном на сайті
3	parol	VARCHAR	Пароль
4	create_at	Timestamp	Дата реєстрації
5	Role_id	BIGINT	Код користувача/ (Адмін або звичайний користувач)
6	firstname	VARCHAR	Ім'я юзера
7	lastname	VARCHAR	Прізвище юзера
8	phone	VARCHAR	Номер телефона
9	adress	VARCHAR	Адреса
10	country	VARCHAR	Країна
11	postcode	VARCHAR	Поштовий індекс
12	City	VARCHAR	Місто
13	apdate_at	Timestamp	Дата коли було оновлено

Таблиця «Tags» призначена для збереження інформації про, теги товарів які є у магазині. (таблиця 2.5).

Таблиця 2.5 – Атрибути таблиця «Tags»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	name	VARCHAR	Назва тега
3	apdate_at	Timestamp	Дата оновлення
4	create_at	Timestamp	Дата створення

Таблиця «Collections» призначена для збереження інформації про колекції товарів (таблиця 2.6).

Таблиця 2.6 – Атрибути таблиця «Collections»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	Name	VARCHAR	Назва колекції
3	create_at	Timestamp	Дата створення категорії
4	apdate_at	Timestamp	Дата оновлення

Таблиця «Sizes» призначена для збереження інформації розмірів товарів (таблиця 2.7).

Таблиця 2.7 – Атрибути таблиця «Sizes»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	name	VARCHAR	Назва розміру

Таблиця «Roles» призначена для збереження інформації про ролі користувачів сайту (таблиця 2.8).

Таблиця 2.8 – Атрибути таблиця «Roles»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	name	VARCHAR	Назва ролі

Таблиця «Orders» призначена для збереження інформації про замовлення (таблиця 2.9).

Таблиця 2.9 – Атрибути таблиця «Orders»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	order_status	INT	Статус замовлення
3	create_at	Timestamp	Дата створення
4	apdate_at	Timestamp	Дата оновлення
5	user_id	BIGINT	Індекс користувача
6	product_id	BIGINT	Індекс продукту
7	quantities	INT	Кількість продукту
8	total	INT	Загальна сума замовлення

Таблиця «Products» призначена для збереження інформації про, товари які є у магазині (таблиця 2.10).

Таблиця 2.10 – Атрибути таблиця «Products»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	name	VARCHAR	Назва товару
3	create_at	Timestamp	Дата створення
4	update_at	Timestamp	Дата оновлення
5	size_id	INT	Індекс розміру
6	prise	INT	Ціна товару
7	Img	VARCHAR	Назва картинка
8	collection_id	BIGINT	Індекс колекції
9	description	VARCHAR	Опис товару
10	brand_id	BIGINT	Індекс бренду
11	tag_id	BIGINT	Індекс тегу
12	counter	INT	Рахунок відвідування певного товару для підрахунку популярності
13	popularity_id	BIGINT	Індекс популярності та оцінювання

Таблиця «Img_content» призначена для збереження інформації про те де будуть використані картинки (таблиця 2.11).

Таблиця 2.11 – Атрибути таблиця «Img_content»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	Name	VARCHAR	Назва для використання картинки
3	type_id	INT	Індекс типу картинки

Таблиця «Orders» призначена для збереження інформації про замовлення (таблиця 2.12).

Таблиця 2.12 – Атрибути таблиця «Orders»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	order_status	INT	Статус замовлення
3	create_at	Timestamp	Дата створення
4	apdate_at	Timestamp	Дата оновлення
5	user_id	BIGINT	Індекс користувача
6	product_id	BIGINT	Індекс продукту
7	quantities	INT	Кількість продукту
8	total	INT	Загальна сума замовлення

Таблиця «Img_content» призначена для збереження інформації про те де будуть використані картинки (таблиця 2.13).

Таблиця 2.13 – Атрибути таблиця «Img_content»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	Name	VARCHAR	Назва де буде використовуватись картинка
3	type_id	INT	Індекс типу картинки

Таблиця «Histori_order» призначена для збереження інформації про категорії постів (таблиця 2.14).

Таблиця 2.14 – Атрибути таблиця «Histori_order»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	order_status	INT	Статус замовлення
3	create_at	Timestamp	Дата створення
4	apdate_at	Timestamp	Дата оновлення
5	user_id	BIGINT	Індекс користувача
6	product_id	BIGINT	Індекс продукту
7	quantities	INT	Кількість продукту
8	total	INT	Загальна сума замовлення

Таблиця «Favorities» призначена для збереження інформації про товари які сподобались покупцеві і які він додав в улюблені(таблиця 2.15).

Таблиця 2.15 – Атрибути таблиця «Favorities»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	user_id	BIGINT	Індекс користувача
3	create_at	Timestamp	Дата створення категорії
4	apdate_at	Timestamp	Дата оновлення
5	product_id	BIGINT	Індекс продукту

Таблиця «Brands» призначена для збереження інформації про, бренди товарів (таблиця 2.16).

Таблиця 2.16 – Атрибути таблиця «Brands»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	Name	VARCHAR	Назва бренду
3	create_at	Timestamp	Дата створення
4	apdate_at	Timestamp	Дата оновлення

Таблиця «Ratings» призначена для збереження інформації про популярність товару і відгуки про нього (таблиця 2.17).

Таблиця 2.17 – Атрибути таблиця «Ratings»

№ п/п	Назва	Тип інформації	Опис
1	id	BIGINT	Унікальний ідентифікатор
2	mark	DOUBLE	Оцінка даного товару
3	product_id	BIGINT	Індекс продукту
4	user_id	BIGINT	Індекс користувача
5	create_at	Timestamp	Дата створення
6	apdate_at	Timestamp	Дата оновлення

За допомогою таблиці юзер буде опрацьовуватися бізнес процеси реєстрації та авторизації. Та виводу інформації про користувача де це потрібно по типу особистого кабінету і форми підтвердження замовлення.

Табличка типу картинка буде використовуватися як довідник для того щоб в табличці контенту картинок вибрати тип і потім виводити ці картинки на головній сторінці

Табличка з тегами буде використовуватись для того щоб надавати товарам теги і виводити їх разом із товарами, вона я також довідником.

Табличка колекції використовується як таблиця для фільтрації продукції, кожен продукт відноситься до відповідних колекції, які будуть записані у дану таблицю. Табличка розмірів відповідає за розміри товарів які присутні у магазині. Табличка ролей відповідає за ролі юзерів на сайті, на даний момент там є тільки Адмін і звичайний юзер, за допомогою бізнес процесів адміністратора, він може змінювати ролі для користувачів. Табличка продуктів відповідає за товар та його детальний опис, потім за допомогою методів виводу у відповідному контролері уся інформація виводиться з бази інформації на сторінку магазину. Також з цієї таблиці підтягується інформація через поле id у іншій таблиці для виводу на відповідних сторінках таких як: улюблені товари, замовлення.

Табличка замовлення відповідає за збереження інформації про той товар який знаходиться у корзині клієнта, і підтягує дані із таблиці продуктів. Табличка історії замовлень відповідно відповідає за замовлення які були підтвержені, інформація з цієї таблиці виводить у вкладку особистого кабінету в пункту історія замовлень. Табличка улюблені відповідає за збереження і вивід інформації у вкладці улюблене та в особистому кабінеті клієнта. Табличка бренди відповідає за назву брендів товарів які містяться у таблиці продукт, і є довідником. Табличка популярності відповідає за збереження відгуків та оцінки товарів які містяться у таблиці продукт, і є довідником.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

Після проведення дослідження, для реалізації даної дипломної роботи було обрано мову програмування **МОВА РОЗРОБКИ РНР**. Ця мова програмування є дуже зручною для вирішення задач, які в сукупності дадуть

рішення нашої задачі по створенню методу формування оптимальних наборів пропозицій в системах електронної торгівлі, також МОВА РОЗРОБКИ PHP піддержуються більшістю хостинг-провайдерів та є найбільш зрозумілим та легким в опановуванні серед всіх аналогів на ринку.

2.3.2 Вибір фреймворку

Після проведення дослідження було обрано фреймворк на базі шаблону MVC та основних принципах ООП а саме Laravel. Даний фреймворк є одним із популярніших на ринку, та є універсальним вирішенням для будь-якої проблеми чи завдання.

2.3.3 Вибір редактора програмного коду

Для розробки даного програмного веб застосування було обрано найпопулярнішу та найсучаснішу середу розробки Visual Code, даний редактор коду був розроблений компанією Microsoft як перша кросплатформова середа розробки у лінійці Visual Studio для веб застосунків. Даний редактор підтримує усі найпопулярніші мови розробки, найновіші оновлення у світі IT, та дуже зручну та гнучку кастомізацією додатковими допоміжними доповненнями які допомагають та полегшують розробку програмного продукту.

2.3.4 Вибір системи керування базами даних

Після проведення дослідження усіх наявних на теперішній час Система керування базами даних , для керування та збереження інформації на роботі з ними було обрано MySQL, Система керування базами даних забирає на себе всі зазначені методи та функції на кожному з етапів життєвого циклу БД - у створення, ведення та використання. Усі функції Система керування базами

даних можна розбити на групи: керування базами інформації; розробка, налагодження й здійснювання програм; здійснювання допоміжних операцій.

В СУБД передбачені три рівні управління:

1. керування файлами, здійснюється в процесі їх генерації та експлуатації. Головними операціями є відкриття і закриття, перейменування, організація, відновлення баз інформації, зняття звітів по базах інформації;
2. Управління записами, яке має читання, додавання, видалення і впорядкування усіх записів;
3. керування полями записів (атрибутами).

Для наглядного прикладу чому було обрано MySQL а не найближчу альтернативу MSSQL наведено відмінності роботи цих двох Система керування базами даних наступне порівняння:

– MySQL в основному використовується для проектів або програм MOBA РОЗРОБКИ PHP. SQL-сервер в основному використовується для .Net або проектів Windows.

– MySQL має кілька механізмів зберігання, які дають розробникам велику гнучкість у використанні механізму для таблиць залежно від продуктивності. Одним з найпопулярніших механізмів зберігання є InnoDB. SQL-сервер може використовувати лише один або один механізм зберігання.

– У MySQL після здійснювання запиту його не можна скасувати в середині шляху. На сервері SQL запит може бути скасований на півдорозі після виконання.

– MySQL не має жодного інструменту, який забезпечує безпеку. SQL-сервер має інструмент безпеки, який забезпечує безпеку інформації і т. д., який є базовим інструментом аналізу безпеки MS.

– MySQL безкоштовний та безкоштовний у використанні. SQL-сервер ліцензований і необхідно придбати ліцензії для використання та запуску кількох баз інформації. Це справді дорого купити кілька ліцензій.

Таблиця 2.18 Порівняння між MySQL та SQL Server

Порівняння між MySQL та SQL Server	MySQL	SQL Server
Визначення	MySQL – це система управління реляційними базами інформації з відкритим вихідним кодом.	SQL-сервер – це система управління реляційними базами інформації.
Запрограмований	MySQL в основному програмувався мовою програмування C та C++.	SQL Server був переважно запрограмований на C++, але деякі частини також і на C.
Розроблено	MySQL був розроблений Oracle.	SQL Server розробили Microsoft.
Багатомовного	MySQL доступний лише англійською мовою.	SQL Server доступний кількома мовами.
платформи	MySQL підтримує безліч платформ, порівняно з SQL-сервером.	SQL-сервер підтримує лише платформи Linux та Windows.
Ліцензія	Це лише з відкритим вихідним кодом.	Це комерційний
Синтаксис	Встановлено, що синтаксис MySQL трохи складний	Синтаксис SQL-сервера простіше та простіше у використанні.

– MySQL вимагає оплатити підтримку або допомогу. SQL-сервер має більшу підтримку спільноти або онлайн-форумів. Підтримка Microsoft також сплачується за підтримку сервера bSQL.

– Якщо MySQL використовується з .Net, спочатку необхідно завантажити сторонні інструменти. Якщо SQL-сервер використовується з .Net, вбудовані класи інфраструктури забезпечують сумісність та можуть також запускатися із запитом LINQ.

– MySQL використовує Enterprise Manager як інструмент IDE. SQL-сервер використовує керуючу студію як IDE для розробки запитів.

Розділ 3 Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення програмних складових системи

Під час розробки методу формування оптимальних наборів пропозицій в системах електронної торгівлі, був використаний фреймворк Laravel оснований на базі шаблону MVC. Головна ідея шаблону MVC полягає в розбитті програми на моделі, види та контролери. Метод представляє і реалізовує бізнес-логіку веб застосунку. Вид відображає зовнішній вигляд сторінок застосунку. Контролер реалізовує зв'язок між моделями, видами і приймає дані, які надходять від користувача.

В даному веб застосунку були реалізовані наступні моделі :

- User – це клас який відповідає за бізнес-логіку яка пов'язана із інформацією про користувача.
- ShopingCart – це клас який відповідає за бізнес-логіку яка пов'язана із корзиною замовлень.
- Role – це клас який відповідає за бізнес-логіку яка пов'язана із ролями користувачів.
- Rating – це клас який відповідає за бізнес-логіку яка пов'язана із методом формування оптимальних наборів пропозицій та рейтингу товарів.
- Product – це клас який відповідає за бізнес-логіку яка пов'язана із позиціями товарів у інтернет магазині.
- Order – це клас який відповідає за бізнес-логіку яка пов'язана замовленнями користувача.
- Favorite – це клас який відповідає за бізнес-логіку яка пов'язана із товарами які користувач обрав як улюблені
- Description – це клас який відповідає за бізнес-логіку яка пов'язана із детальний описом обраного користувачем товару.

В даному веб застосунку були реалізовані наступні види:

- Main – даний вид відповідає за відображення головної сторінки системи електронної торгівлі та методу підбору оптимальних пропозицій.
- Content_ShoppingCart – даний вид відповідає за відображення корзини замовлень.
- Content_Shop – даний вид відповідає за відображення за каталог позицій товарів магазину.
- Favorite_content – даний вид відповідає за відображення сторінки на якій відображаються улюблені товари.
- Descriptions_content – даний вид відповідає за відображення сторінки обраного товару та його детального опису.
- register_content – даний вид відповідає за відображення сторінки з формою для реєстрації користувача.
- login_content – даний вид відповідає за відображення сторінки з формою для авторизації користувача.
- Admin_content – даний вид відповідає за відображення панель адміністративних функцій для користувача який є адміністратором.

В даному веб застосунку були реалізовані наступні контролери:

- MainController – даний контролер відповідає за обробку інформації які пов'язані з головною сторінкою.
- ShoppingCartController – даний контролер відповідає за обробку інформації які пов'язані з корзиною замовлень.
- ShopController – даний контролер відповідає за обробку інформації які пов'язані з каталогом товарів.
- Favorite_Controller – даний контролер відповідає за обробку інформації які пов'язані з улюбленими товарами.
- Description_Controller – даний контролер відповідає за обробку інформації які пов'язані з детальним описам товарів.
- AdminController – даний контролер відповідає за обробку інформації які пов'язані з панеллю адміністратора.

– AccountController – даний контролер відповідає за обробку інформації які пов’язані з користувачем та особистим кабінетом.

– AutenteficationController – даний контролер відповідає за обробку інформації які пов’язані з авторизацією у системі електронної торгівлі.

Представлені моделі, види та контролери організують стабільне функціонування методу підбору оптимальний наборів пропозицій та системи електронної торгівлі. Подібне розбиття коду програмного продукту на три головних компоненти дозволяють полегшити розробку та оновлення програми.

Для прикладу на рисунку 3.1 зображено діаграму класів проекту

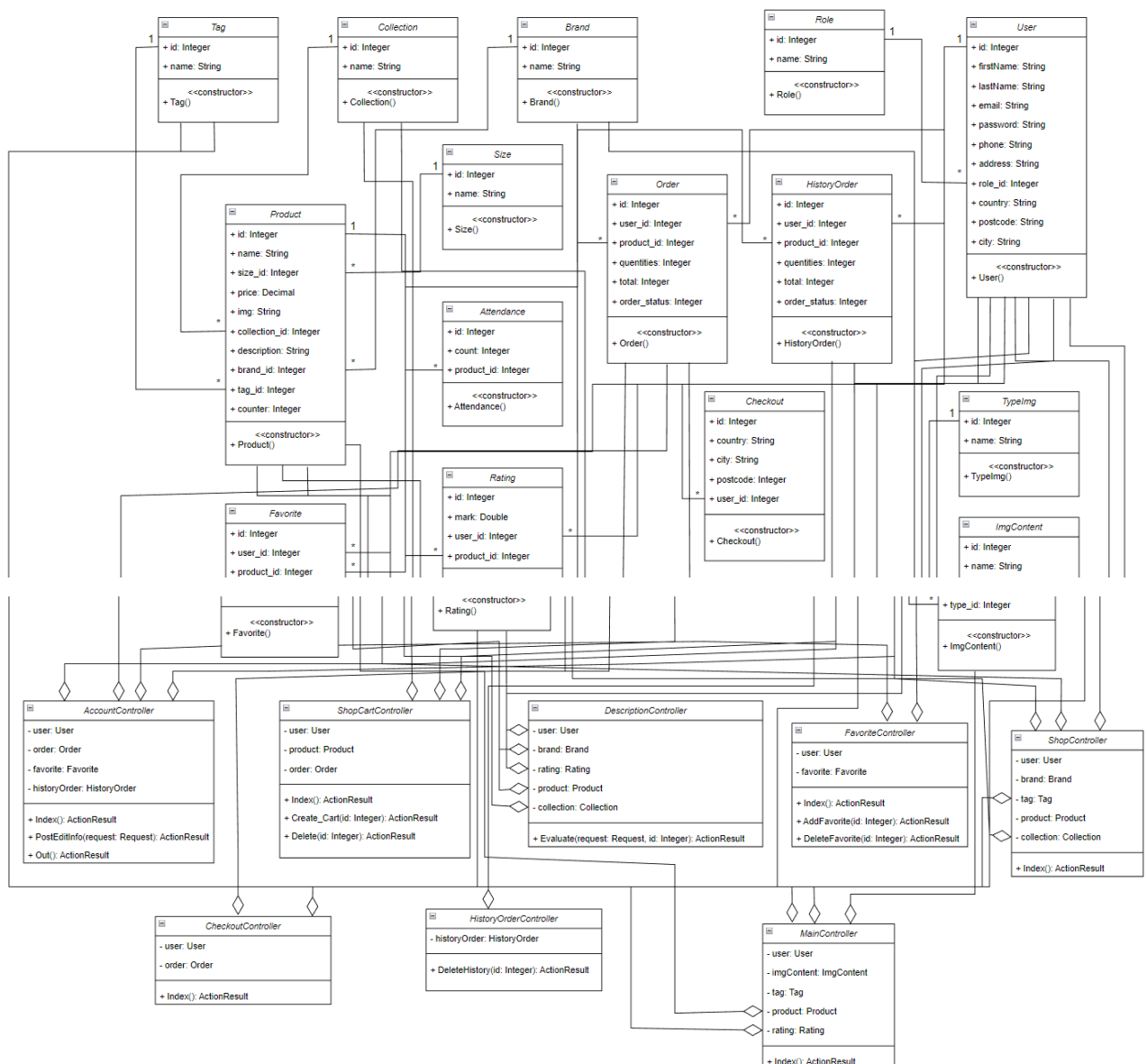


Рисунок 3.1 – Діаграма класів методу підбору наборів оптимальних рішень інтегрованих у систему електронної торгівлі.

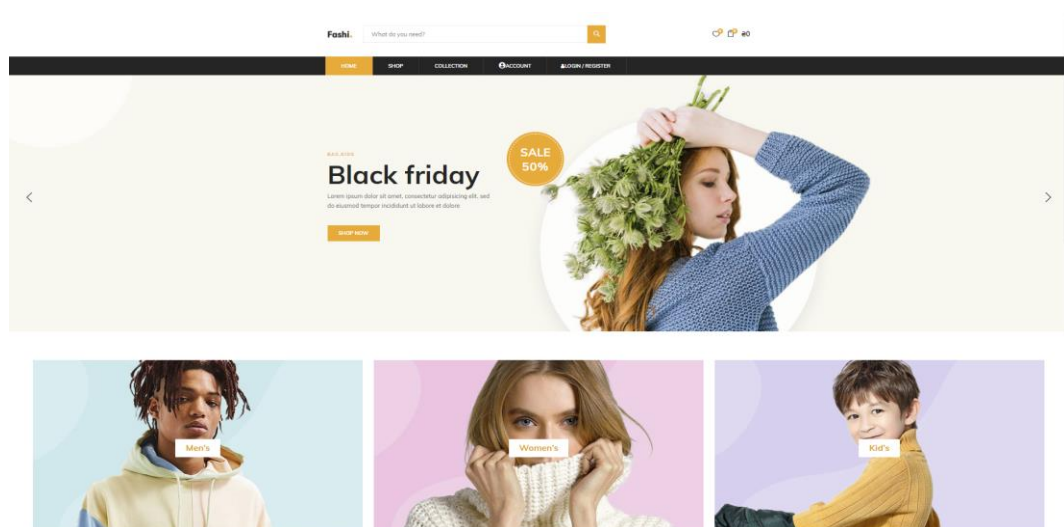
3.2 Особливості реалізації програмних складових системи

Основними модулями програмного продукту є :

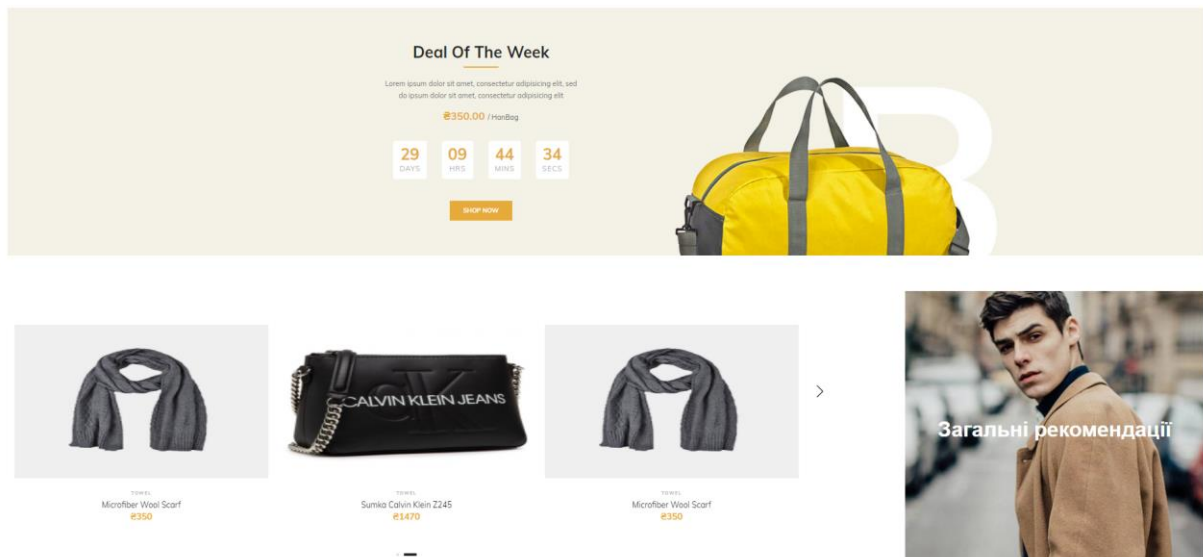
- Головна сторінка з методом формування оптимальних наборів пропозицій.
- Сторінки з формами реєстрації та авторизації.
- Сторінка з особистим кабінетом користувача.
- Модулі додавання та заміна інформації на інформації користувачем.
- Сторінка з каталогом позицій усіх товарів.
- Пошук товарів за допомогою фільтрів або пошукової строки.
- Перегляд детальної інформації про товари.
- Панель керування адміністратора.
- Додавання товару у кошик з покупками, або моментальне замовлення.
- Голосування за рейтинг товару.

Перше місце куди потрапляє користувач, являється головною сторінкою на ній завжди можна знайти актуальну інформацію, новинки, знижки, акції, та список рекомендованих популярних товарів у магазині.

На рисунку 3.2-3.4 зображений зовнішній вигляд головної сторінки



Рисунки 3.2– Візуальний аспект головної сторінки



Рисунки 3.3 – Візуальні аспекти головної сторінки системи електронної торгівлі.

Головна сторінка була реалізована за допомогою наступного коду:

```

<!-- Hero Section Begin -->
<section class="hero-section">
  <div class="hero-items owl-carousel">
    @foreach($sing as $singMainBanner)
      @if($singMainBanner->typeIn(['Name'] == 'MainBanner' )
        <div class="single-hero-items set-bg" data-setbg="assets/img/{{$singMainBanner['Name']}}">
          <div class="container">
            <div class="row">
              <div class="col-lg-5">
                <span>Bag, kids</span>
                <h1>Black Friday</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore</p>
                <a href="{{route('Shop')}}" class="primary-btn">Shop Now</a>
              </div>
            </div>
            <div class="off-card">
              <h2>Sale <span>50%</span></h2>
            </div>
          </div>
        </div>
      @endif
    @endforeach
  </div>
</section>
<!-- Hero Section End -->

<!-- Banner Section Begin -->
<div class="banner-section spad">
  <div class="container-fluid">
    <div class="row">
      @foreach($sing as $singMainBanner)
        @if($singMainBanner->typeIn(['Name'] == 'Men' )
          <div class="col-lg-4">
            <div class="single-banner">
              
              <div class="inner-text">
                <h3><a style="color:goldenrod" href="http://public/Shop?collection_id=1">Men's</a></h3>
              </div>
            </div>
          </div>
        @endif
      @endforeach
    </div>
  </div>

```

```

@Foreach($ing as $ingMainBanner)
  @if($ingMainBanner->Type[eg['Name']] == 'Women' )
    <div class="col-lg-4">
      <div class="single-banner">
        
        <div class="inner-text">
          <h3 style="color:goldenrod">Women</h3>
        </div>
      </div>
    </div>
  @endif
@endforeach
@foreach($ing as $ingMainBanner)
  @if($ingMainBanner->Type[eg['Name']] == 'Kids' )
    <div class="col-lg-4">
      <div class="single-banner">
        
        <div class="inner-text">
          <h3 style="color:goldenrod">Kids</h3>
        </div>
      </div>
    </div>
  @endif
@endforeach
</div>
</div>

```

Рисунок 3.4 – Фрагмент коду головної сторінки.

Авторизація є не менш важливим модулем як для звичайного користувача так і для адміністраторів. Для того щоб отримати усі базові функції потрібно спочатку зареєструватись якщо ви в перше на цьому сайті та авторизуватись, приклади форм реєстрації, авторизації та лістинг коду зображено на рисунках 3.5-3.8.

Fashi. What do you need?

HOME SHOP COLLECTION ACCOUNT LOGIN / REGISTER

Home > Login

Login

Username or email address *

Password *

Save Password [Forget your Password](#)

SIGN IN

[OR CREATE AN ACCOUNT](#)

Address: Khmelnitsky
Phone: +380980393886
Email: istauy1712@gmail.com

Information
Home
Shop
Login/Register

My Account
My Account
Favorites
Shopping Cart

Copyright ©2022 All rights reserved | This template is made with by Nikolai Vlasuk

Рисунок 3.5 – Візуальний вигляд форми авторизації.

За реалізацію форми авторизації відповідає наступний код:

```

public function getlogin()
{
    $helper= new User();
    return view('Auth.login', ['favorite_counter'=>$helper->favorite_counter(),
    'order_counter'=>$helper->order_counter(),
    'total'=>$helper->total(),
    'hover_cart'=>$helper->hover_cart()]);
}

public function postlogin(Request $request)
{
    $this->validate($request, [
        'email'=>'required|max:255',
        'password'=>'required|min:6'
    ]);

    if(!Auth::attempt($request->only(['email', 'password'])))
    {
        return redirect()->back()->with('info', "Неправильний email або пароль");
    }

    $user = User::where('email', $request->input('email'))->get();
    $userId = 0;
    foreach($user as $u)
    {
        $userId = $u->role_id;
    }

    if($userId == 1)
    {
        $request->session()->push('admin_session', $request->input('email'));
        return redirect()->route('admin.index');
    }
    else
    {
        $request->session()->push('name_session', $request->input('email'));
    }

    return redirect()->route('index_main')->with('info', 'Ви успішно авторизувались');
}
}

```

Рисунок 3.6 – Фрагмент коду який реалізує авторизацію.

The screenshot shows the registration page of the 'Fashi' website. At the top, there is a search bar with the text 'What do you need?' and a magnifying glass icon. To the right of the search bar are icons for a shopping cart and a user profile, with the number '00' next to the cart icon. Below the search bar is a dark navigation bar with the following menu items: HOME, SHOP, COLLECTION, ACCOUNT, and LOGIN / REGISTER. The main content area is titled 'Register' and contains the following form fields:

- Email address *
- Firstname *
- Lastname *
- Phone number *
- Adress *
- Country *
- Password *
- Confirm Password *

At the bottom of the form is a large orange button labeled 'REGISTER'. Below this button is a link that says 'OR LOGIN'.

Рисунок 3.7 – Візуальний вигляд форми реєстрації

За реалізацію форми реєстрації відповідає наступний код:

```

public function getregistration()
{
    $helper= new User();
    return view('Auth.register',
        [
            'favorite_counter'=>$helper->favorite_counter(),
            'order_counter'=>$helper->order_counter(),
            'hover_cart'=>$helper->hover_cart(),
            'total'=>$helper->total(),
        ]);
}

public function postregistration(Request $request)
{
    $this->validate($request,[
        'email'=>'required|unique:users|email|max:255',
        'password'=>'min:6|required_with:password_confirm|same:password_confirm',
        'password_confirm'=>'min:6',
        'firstname'=>'required|max:100',
        'lastname'=>'required|max:100',
        'phone'=>'required|:10',
        'address'=>'required|max:255',
        'country'=>'required|max:255',
    ]);

    User::create([
        'email'=>$request->input('email'),
        'password'=>Hash::make($request->input('password')),
        'role_id'=>2,
        'firstname'=>$request->input('firstname'),
        'lastname'=>$request->input('lastname'),
        'phone'=>$request->input('phone'),
        'address'=>$request->input('address'),
        'country'=>$request->input('country'),
    ]);

    return redirect()->route('index_main')->with('info','Ви успішно зареєструвалися');
}

```

Рисунок 3.8 – Фрагмент коду який реалізує реєстрацію.

Також однією із головних сторінок є особистий кабінет з всією інформацією яку надав користувач, з можливістю її зміни, також користувач може переглянути історію покупок, та свої улюблені товари. Зовнішній вигляд даного компоненту та його лістинг зображено на рисунку 3.9-3.10.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\User;
use App\Order;
use Illuminate\Contracts\Session\Session;
use Illuminate\Support\Facades\Auth;
use App\Favorite;
use App\History_Order;

class AccountController extends Controller
{
    public function index()
    {
        $H=new History_Order();
        $F=new Favorite();
        $session_Value=session('name_session');
        $user_in_session=User::where('email',$session_Value)->get();
        if(!session('name_session')){
            return redirect()->route('login')->with('info',"Для того чтобы зайти в кабинет нужно авторизироваться");
        }
        $helper= new User();

        return view('Account.ST_Account',
            ['user_in_session'*>$user_in_session,
            'favorite'*>$F->favorite_out(),
            'History_Order'*>$H->History_out(),
            ..

```

```

        'favorite_counter'=>$helper->favorite_counter(),
        'order_counter'=>$helper->order_counter(),
        'total'=>$helper->total(),
        'hover_cart'=>$helper->hover_cart()
    });
}

public function postEditInfo(Request $request )
{
    $session_Value=session('name_session');
    $user_in_session=User::where('email',$session_Value)->get();
    $temp_ID=0;
    foreach($user_in_session as $i)
    {
        $temp_ID=$i->id;
    }

    $tempInfo=User::find($temp_ID) ;
    $tempInfo->firstname=$request->get('firstname');
    $tempInfo->lastname=$request->get('lastname');
    $tempInfo->email=$request->get('email');
    $tempInfo->phone=$request->get('phone');
    $tempInfo->address=$request->get('address');
    $tempInfo->country=$request->get('country');
    $tempInfo->postcode=$request->get('postcode');
    $tempInfo->City=$request->get('city');
}

```

Рисунок 3.9 – Фрагмент коду який реалізує особистий кабінет

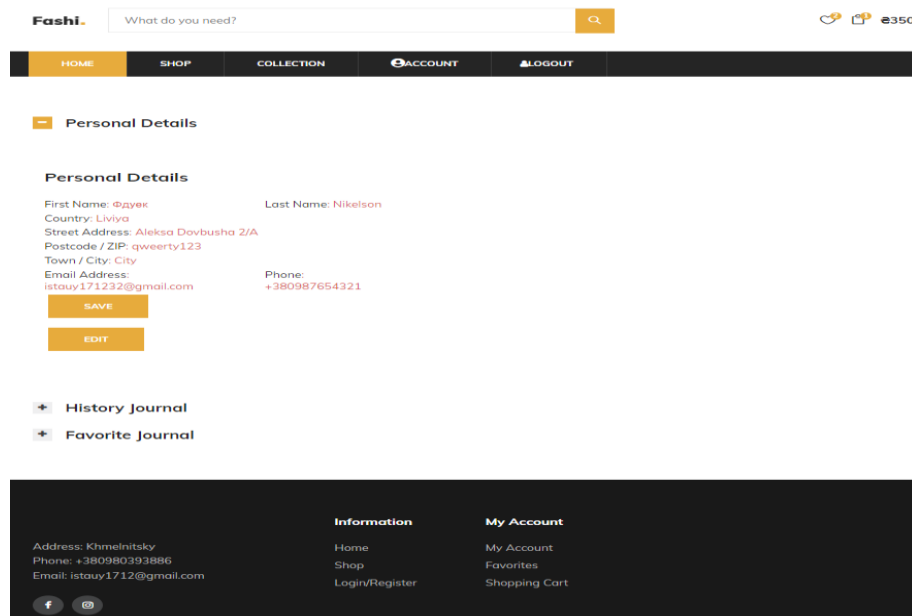


Рисунок 3.10 – Візуальний вигляд особистий кабінет




IMAGE	PRODUCT NAME	PRICE	QUANTITY	TOTAL	DELETE
	Microfiber Wool Scarf	€350	- 1 +	€350	×
	Guangzhou sweater	€670	- 1 +	€670	×
	Green Shirts	€900	- 1 +	€900	×

Рисунок 3.11 – Зовнішній вигляд історій замовлень.

Основною сторінкою для користувача є каталог з усіма позиціями товарів у магазині. Користуючись каталогом клієнт може шукати потрібний товар за допомогою фільтрів та пошукової строки, для більшої інформативності користувач може переглянути детальний опис товару. Приклади зовнішнього вигляду сторінки та його лістинг зображені на рисунках 3.12 – 3.13

```
public function index(ShopFilter $request)
{
    //$cloth=Product::all();
    $cloth = Product::filter($request)->paginate(6);
    $collection = Collection::all();
    $brand = Brand::all();
    $helper= new User();
    $tag = Tag::all();
    return view('Shop.ST_Shop',[
        'cloth'=>$cloth,
        'favorite_counter'=>$helper->favorite_counter(),
        'order_counter'=>$helper->order_counter(),
    ])
```

```

'total'=>$helper->total(),
'hover_cart'=>$helper->hower_cart(),
'brand'=>$brand,
'collection'=>$collection,
'tag'=>$tag]);
}
}

```

Рисунок 3.12 – Фрагмент коду який реалізує сторінку каталогу з товарами

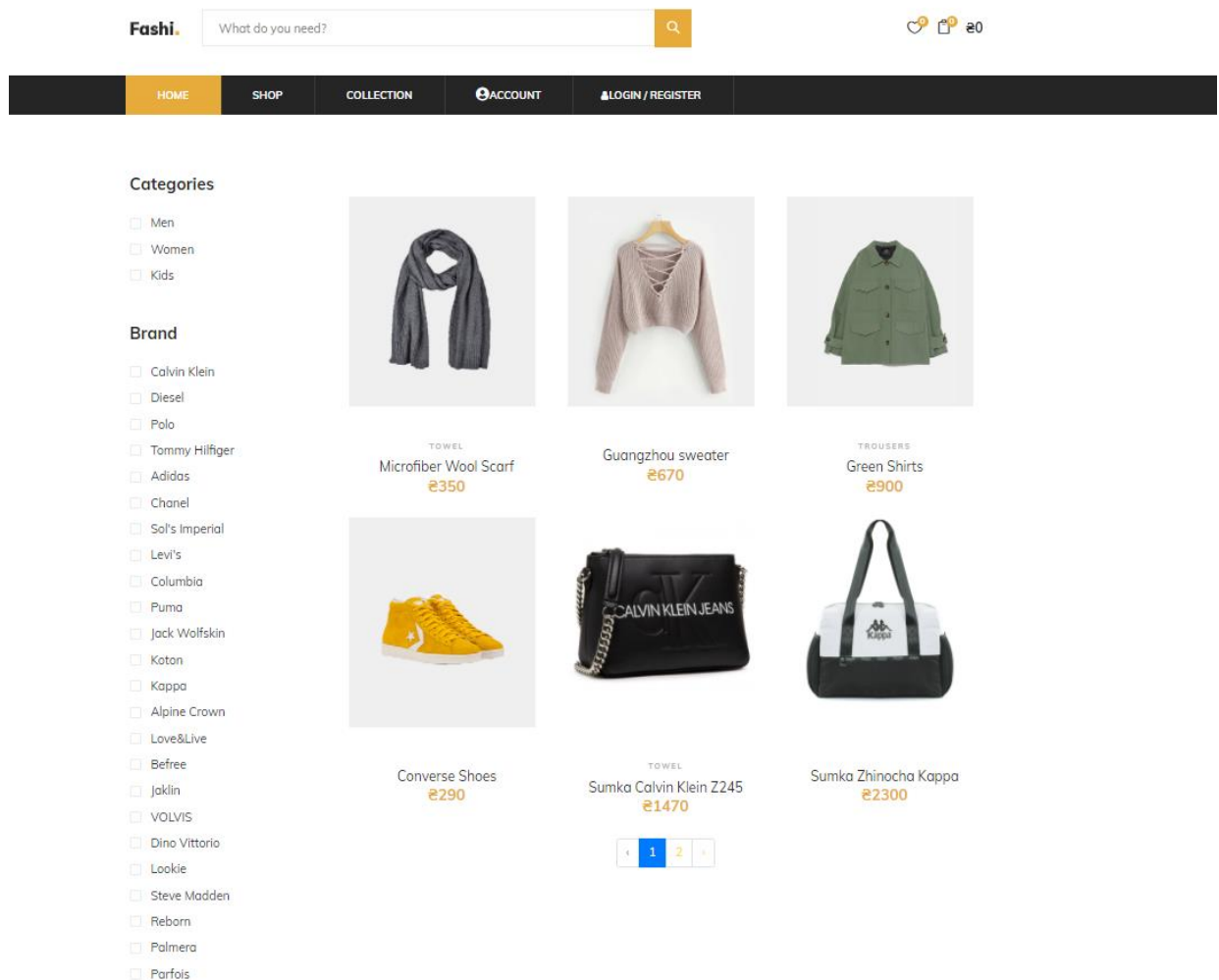


Рисунок 3.13 – Вигляд сторінки каталогу з товарами

Для більшої зручності до програмного продукту було додано корзину із замовленнями користувача. Візуальний вигляд та лістинг коду зображено на 3.14 – 3.15

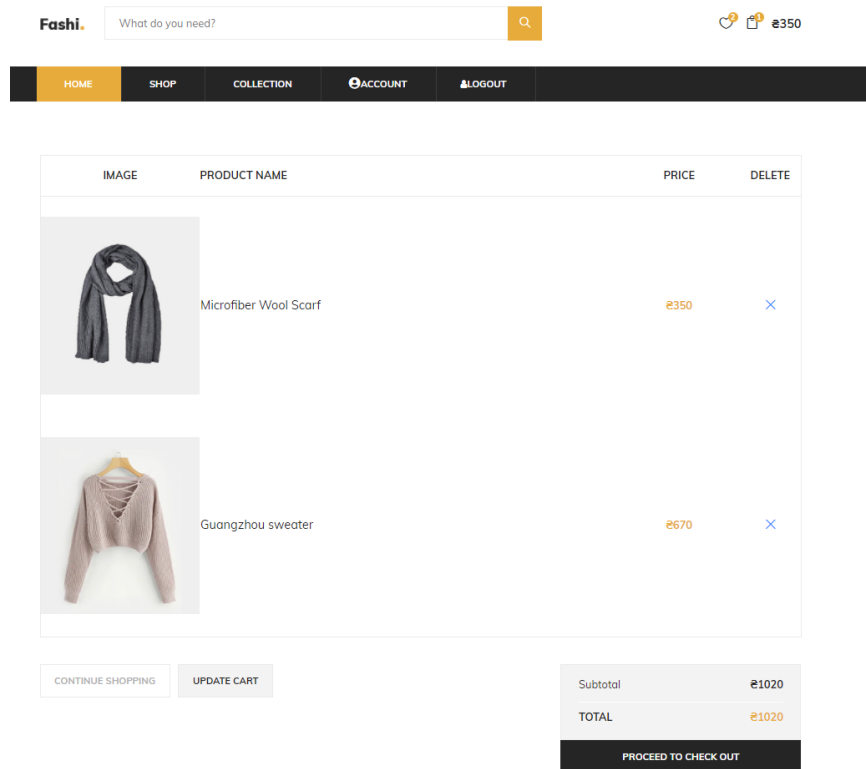


Рисунок 3.14 – Візуальний вигляд корзини замовлень.

```

public function create_cart($id)
{
    if(session('name_session'))
    {
        $session_value=session('name_session');
        $user_in_session=User::where('email',$session_value)->get();
        $temp_ID=0;
        foreach($user_in_session as $i)
        {
            $temp_ID=$i->id;
        }

        $product=Product::where('id',$id)->get();
        $temp_p_price=0;

        foreach($product as $i)
        {
            $temp_p_price=$i->price;
        }

        $userCheck = Order::where([
            'user_id' => $temp_ID,
            'product_id'=>$id
        ])->get();

        if($userCheck->count())
        {
            return redirect()->back()->with('info',"Товар вже існує");
        }
    }
}

```

```

        ShoppingCart::create([
            'product_id'=>$id,
            'user_id'=>$temp_ID,
            'quantities'=>1,
            'total'=>$temp_P_price,
        ]);

        return redirect()->back()->with('info', "Товар добавлено");
    }

    return redirect()->route('login')->with('info', "Авторизуйтесь");
}

public function delete($id)
{
    $order=Order::find($id);
    $order->delete();
    return redirect()->back()->with('info', "Видалено з корзини");
}
}

```

Рисунок 3.15 Лістинг корзини замовлень

3.3 Тестування інформаційної системи

Під час розробки будь-якої програми чи веб застосунка, обов'язковим етапом є тестування системи та коду, це забезпечує розробників інформацією про роботу програмного продукту, коректність роботи усіх модулів, функцій згідно з поставлених задач та вимог завдання. Для перевірки роботи даного веб застосунку було проведено два види тестування : unit-тестування та test-case тестування.

Для тестування було обрано модуль авторизації, ця функція забезпечує отримання усіх функцій доступу та функціоналу системи електронної торгівлі для користувача. Функція яка тестує даний метод, має назву testLogin.

Дана функція виконується поступово

1. Перехід через меню навігації у відповідну вкладку з формою авторизації.
2. Заповнення форми авторизації інформацією яку надав користувач.
3. Відправка та обробка інформації на інформації користувачем.
4. Здійснювання методу авторизації, який повертає результат в змінну «response», чи успішно був авторизований користувач;
5. Контрольна перевірка результату змінної «response» на статус коду 200 (успішне виконання).

Детальний лістинг коду та результат здійснювання зображено на рисунках 3.15 та 3.16

```
1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Illuminate\Foundation\Testing\RefreshDatabase;
8
9  class UserTest extends TestCase
10 {
11     /**
12      * A basic unit test example.
13      *
14      * @return void
15      */
16     public function testLogin()
17     {
18         $response = $this->post('/postlogin',[
19             'email'=>'istauy@gmail.com',
20             'password'=>'1234567890'
21         ]);
22         $this->assertEquals(200,$response->status());
23     }
24 }
```

Рисунок 3.16 – Лістинг коду testLogin.

PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

.

1 / 1 (100%)

Time: 182 ms, Memory: 14.00 MB

OK (1 test, 1 assertion)

Рисунок 3.17 – Результат здійснювання методу тестування testLogin.

Процес тестування модуля авторизації пройдено: успішно

Ще один видом тестування яким було проведено тестування програмного продукту, є test-case тестування.

Таблиця 3.1 – Результат тесту «Перевірка основних функцій користувача після авторизації у системі.»

Перевірка основних функцій користувача після авторизації у системі		
Дія	Результат який очікується	Результат тесту
Перейти на сторінку “Login”	<p>1)Сторінка “Login з формою авторизації відкрита.”</p> <p>2)На формі доступно два поля “Email” та “Password”.</p> <p>3)Панель навігації знаходиться зверху над назвою форми “Login”</p> <p>4)Гіперпосилання на проходження реєстрації знаходиться нижче кнопки “SIGN IN”</p> <p>5)Кнопка авторизації знаходиться вище гіперпосилання та нижче поля “Password”.</p>	<p>– Після авторизації користувача було переміщено на головну сторінку, де знаходиться актуальна інформація про новинки та популярні позиції товарів.</p> <p>– Користувач може обрати товар який йому сподобався до улюблених, у корзину покупок чи зразу здійснити покупку товару.</p> <p>– Також клієнт має змогу користуватися фільтрами пошуку на пошуковою стрічкою.</p> <p>– Клієнт може проголосувати за рейтинг будь якого товару який йому до вподоби.</p>

3.4 Особливості розробки методу формування оптимальних наборів пропозицій

Метод формування оптимальних наборів пропозицій є основним елементом існуючої системи електронної торгівлі. Для того щоб користуватися даним методом користувачеві спочатку потрібно зареєструватися у системі, після чого йому будуть доступні загальні рекомендації (рисунок 3.18) по усім товарам серед магазину. Для того щоб з’явився особистий список рекомендацій (рисунок 3.19) користувачеві потрібно проголосувати за товар який йому

подобається та надати рейтинг пріоритету для себе (рисунок 3.20), після чого система автоматично згенерує список рекомендацій під користувача, для більш детальної інформації лістинги коду зображені на (Рисунках 3.21 – 3.22).

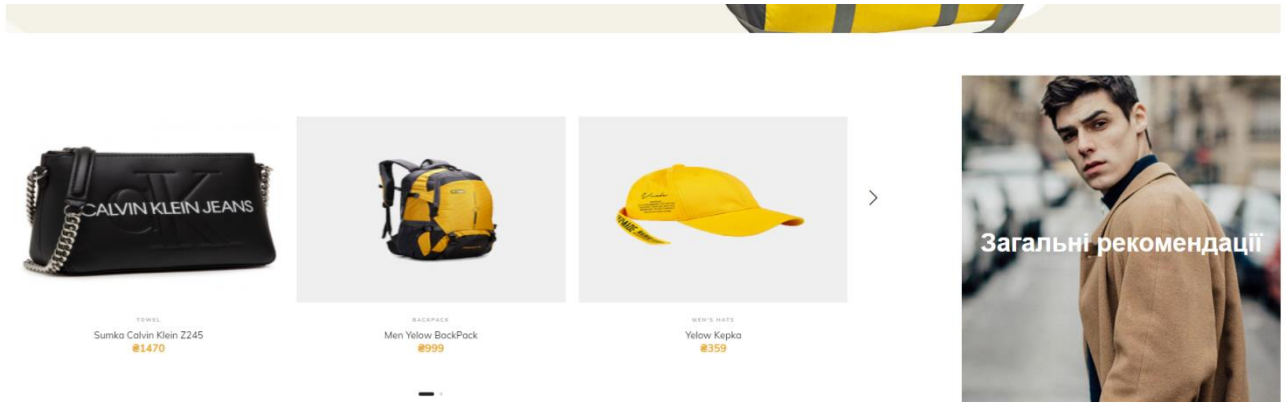


Рисунок 3.18 – Загальні рекомендації товарів.

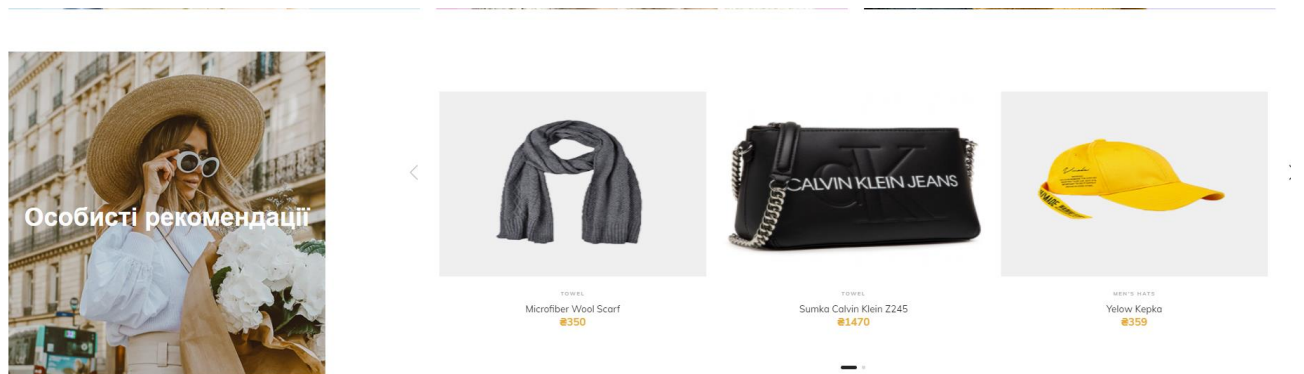


Рисунок 3.19 – Особистий список рекомендацій товарів.

```

$rating=Rating::where('user_id',$userId)->orderBy('mark','desc')->get();

$=Rating::find($rating[0]->id);
//dd($r->products->Tags->name);
$typeProduct=$r->products->Tags->name;
$typeProducts[]=$typeProduct;
$count=1;
for($i=1;$i<count($rating); $i++)
{
    $product = Rating::find($rating[$i]->id);

    if($typeProduct == $product->products->Tags->name)
    {
        continue;
    }
    else
    {
        array_push($typeProducts, $product->products->Tags->name);
        $count++;
        $typeProduct = $product->products->Tags->name;
    }

    if($count == 4)
    {
        break;
    }
}

$typeProductsId = [];

foreach($typeProducts as $name)
{
    $tagId = Tag::where('name',$name)->get();
    foreach($tagId as $tag)
    {
        $typeProductsId[] = $tag->id;
    }
}

```

Рисунок 3.20 – Лістинг методу формування оптимальних наборів пропозицій



Рисунок 3.21 – Форма голосування та представлення рейтингу товару користувачем.

```

    public function evaluate(Request $request, $id)
    {
        $userID=0;
        $user=User::where('email',session('name_session'))->get();
        foreach($user as $p)
        {
            $userID=$p->id;
        }
        $userproduct=Rating::where(['user_id'=>$userID,'product_id'=>$id])->get();
        if(count($userproduct)!=0)
        {
            return redirect()->back();
        }
        else
        {
            Rating::create([
                'product_id'=>$id,
                'user_id'=>$userID,
            ]);
        }

        $counter=1;
        $product=Product::where('id',$id)->get();
        foreach($product as $p)
        {
            $counter=$p->counter;
        }
        $mark=$request->input('rating1')/$counter;
        Rating::where(['user_id'=>$userID,'product_id'=>$id])->update([
            'mark'=>$mark
        ]);
    }

```

Рисунок 3.22 – Лістинг коду голосування та представлення рейтингу товару користувачем.

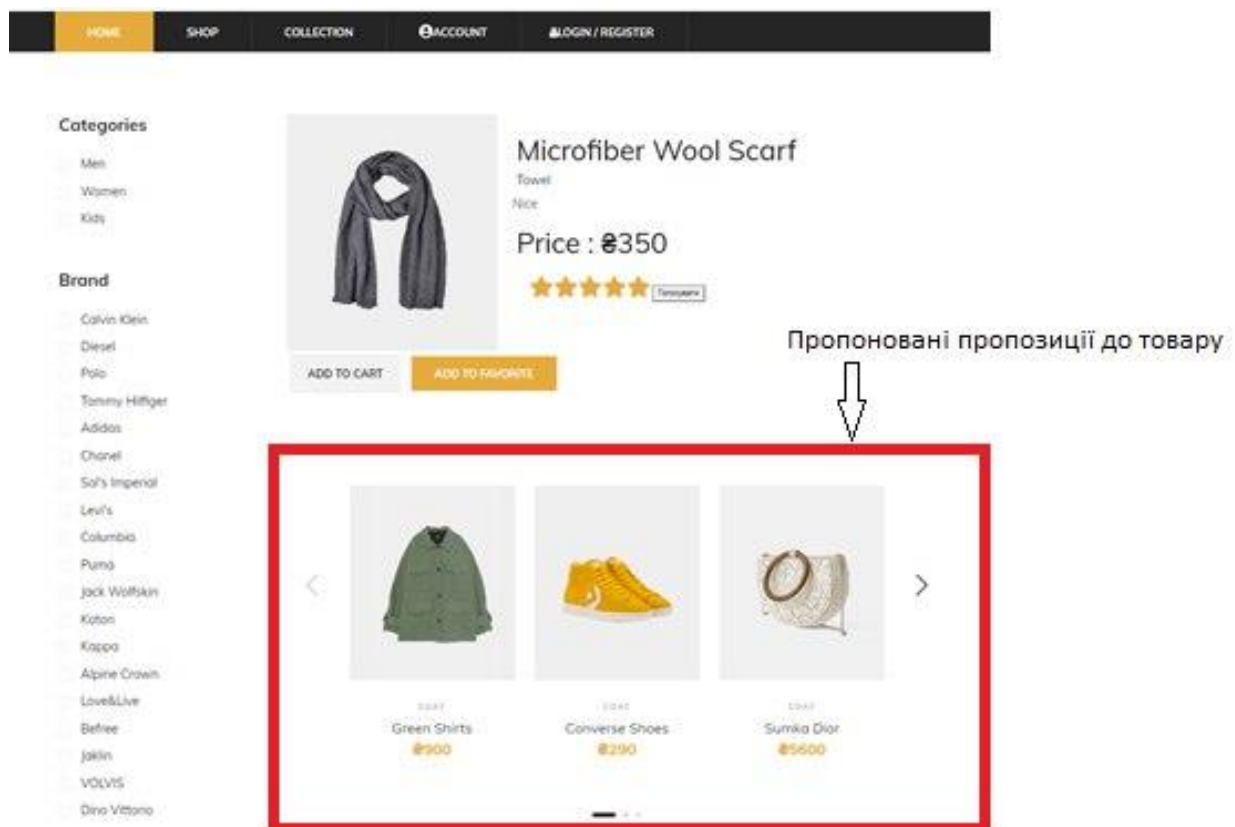


Рисунок 3.23 – список рекомендацій до обраного користувачем товару.

3.5 Вимоги до розгортання інформаційної системи

Для розгортання та функціонування автоматизованої системи необхідно:

- Веб-браузер Google Chrome, з версії 95 до поточно поточної версія;
- Веб-браузер Opera, з версії 78 – до поточно поточної версія;
- Веб-браузер Microsoft Edge, з версії 92 – до поточно поточної версія;
- Веб-браузер Mozilla Firefox, з версії 91 – до поточно поточної версія;
- Веб-браузер Safari, з версії 15.1 – до поточно поточної версія.
- вимоги до провайдера:
- MySQL Server, з версії 2.0 – до поточно поточної версія;
- Дисковий простір – більше 1000 Мб;
- Підтримання Laravel 8.

Висновки

Під час роботи над кваліфікаційною роботою бакалавра було проведено аналіз поставленої предметної області, визначено основну проблематику даної задачі, були складені задачі які потрібно було вирішити. Було обрано та описано технології, методи, сервіси, які використовуються у створенні методу формування оптимальних наборів пропозицій та інтегрованого у систему електронної торгівлі.

Створено веб-технологію з можливістю продажу товарів через онлайн сервіс, та метод який допомагає швидше орієнтуватись та робити своє рішення про покупки пересічному користувачеві. Було розроблено функціональну структуру додатку та представлено її у вигляді груп функцій.

Розроблено структуру бази інформації, її реляційну проекцію. Було описано таблиці, поля бази інформації та реалізовано взаємодію між таблицями та групами функцій. Проведено тестування основних груп функцій та програмних модулів системи.

Згідно з отриманим результатом, можна зробити висновок, що розроблена веб-технологія методу формування оптимальних наборів пропозицій інтегрована в систему електронної торгівлі працює, вимоги технічного завдання виконанні повністю.

Перелік посиланьч

1. **Інформаційно пошукові системи** [Електронний ресурс]. – Режим доступу:

<https://sites.google.com/site/socialnafnformatika/home/informacijno-posukovi-sistemi>

2. **Рекомендаційна система** [Електронний ресурс]. – Режим доступу:
<https://www.sciencedirect.com/science/article/pii/S1110866515000341#:~:text>

3. **Обробка інформації та інформації** [Електронний ресурс]. – Режим доступу

https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0_%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97

4. **Електронна комерція** [Електронний ресурс]. – Режим доступу
https://pidru4niki.com/13710106/marketing/sistemi_formi_elektronnoyi_torgivli

5. **ООП** [Електронний ресурс]. – Режим доступу
<https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE->

[%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)

6. **MVC** [Електронний ресурс]. – Режим доступу
<https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80>

7. **Бази інформації та Система керування базами даних** [Електронний ресурс]. – Режим доступу

https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85

8. **Spotify** [Електронний ресурс]. – Режим доступу
<https://www.spotify.com/ua-en/>

9. **Amazon** [Електронний ресурс]. – Режим доступу
<https://www.amazon.com/>

10. **Веб-Веб-продукт** [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/Веб-Веб-продукт>

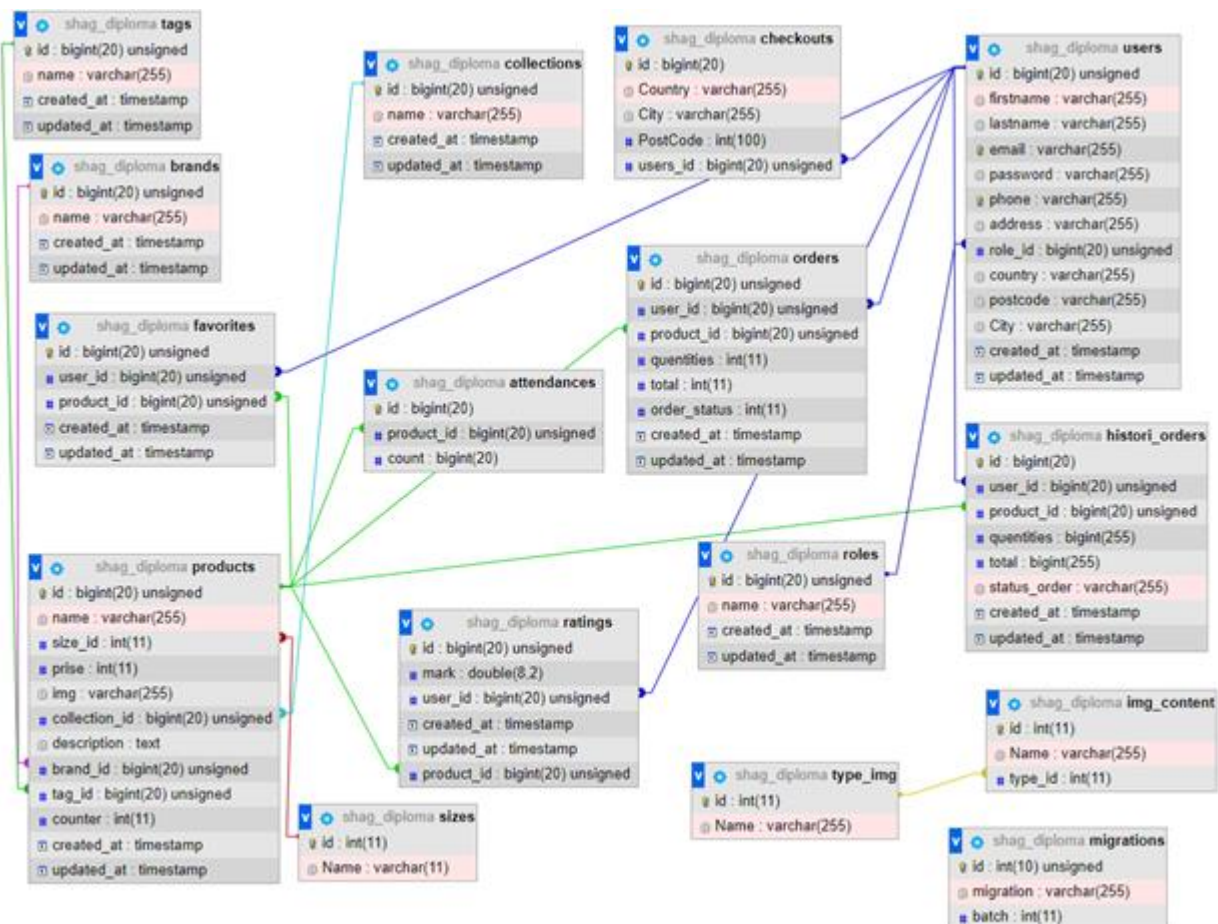
11. **Microsoft. .NET** [Електронний ресурс]. – Режим доступу: <https://www.microsoft.com/net>.

12. **МОВА РОЗРОБКИ PHP** [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/мова_розробки_PHP

ДОДАТКИ

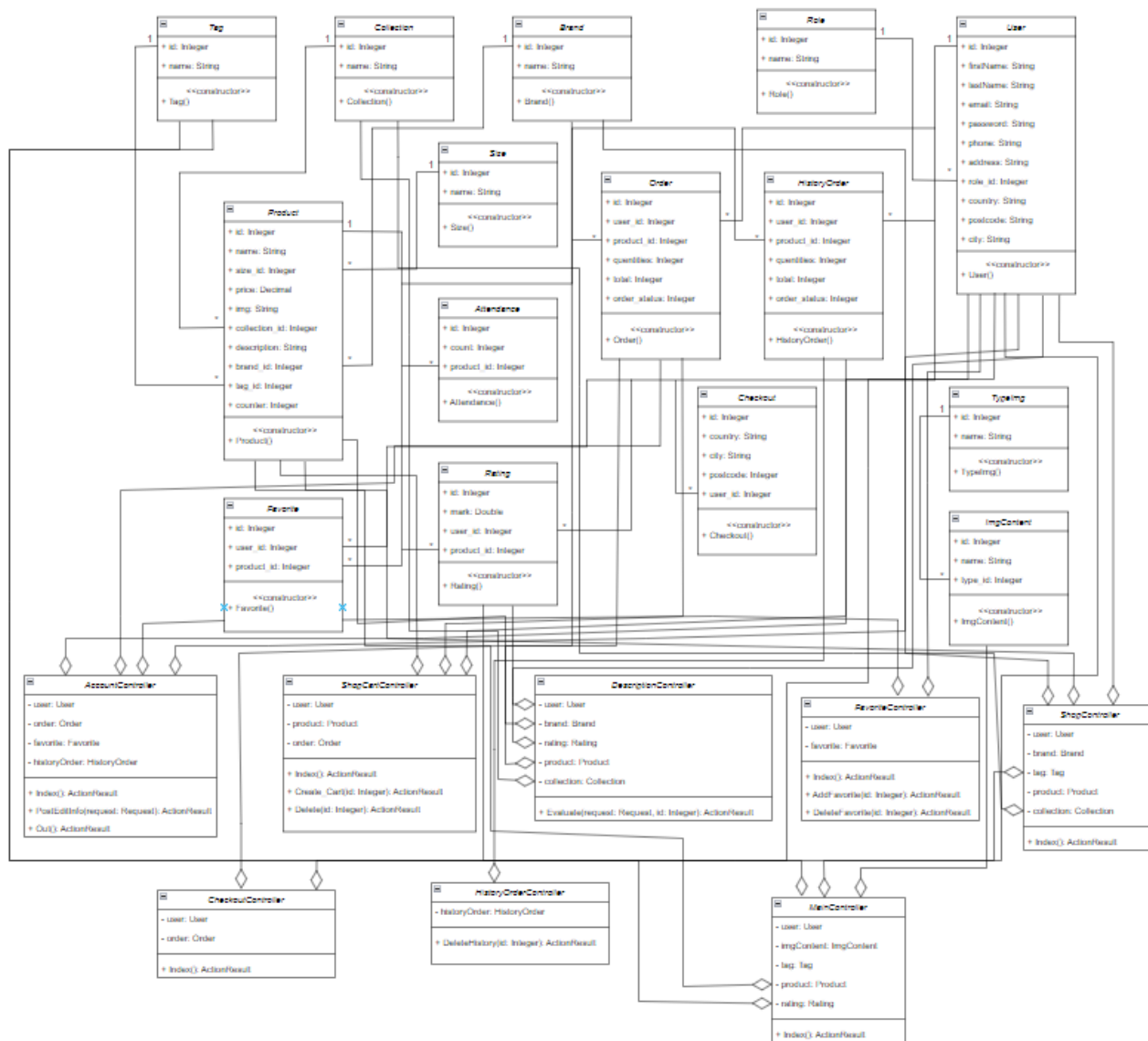
Додаток А

Структура бази інформації веб застосунку з методом формування оптимальних наборів пропозицій інтегрованого в систему електронної торгівлі.



Додаток Б

Діаграма класів методу підбору наборів оптимальних рішень інтегрованих у систему електронної торгівлі



Додаток В

Програмні коди

```

<!-- Header Section Begin -->
<div class="container">
  <div class="inner-header">
    <div class="row">
      <div class="col-lg-1 col-md-1">
        <div class="logo">
          <a href="{{route('index_main')}}">
            
          </a>
        </div>
      </div>
      <div class="col-lg-7 col-md-6">
        <div class="advanced-search">
          <!-- <button type="button" class="category-btn">All Categories</button> -->

          <div class="input-group">
            <form action="{{route('Shop')}}" method="GET">
              <input name="search_field" type="text" placeholder="What do you need?">
              <button type="submit"><i class="ti-search"></i></button>
            </form>
          </div>

        </div>
      </div>
      <div class="col-lg-1 text-right col-md-1">

    </div>

    <div class="col-lg-3 text-right col-md-3">

      <ul class="nav-right">
        <li class="heart-icon">
          <a href="{{route('Favorite')}}">
            <i class="icon_heart_alt"></i>

            <span>{{ $favorite_counter }}</span>

          </a>
        </li>
        <li class="cart-icon">
          <a href="{{route('ShopingCart')}}">
            <i class="icon_bag_alt"></i>

```

```

        <span>{{ $order_counter }}</span>

    </a>

    <div class="cart-hover">
        <?мова розробки PHP
        $totalPrise=0;
        ?>
        @foreach( $hover_cart as $product )
        <?мова розробки PHP
        $totalPrise+=$product->Products['prise'];
        ?>

        <div class="select-items">
            <table>
                <tbody>
                    <tr>
                        <td class="si-pic"></td>

                        <td class="si-text">
                            <div class="product-selected">
                                <p>{{ $product->Products['prise'] }}x {{ $product->quantities }}</p>
                                <h6>Kabino Bedside Table</h6>
                            </div>
                        </td>
                        <td class="si-close">
                            <i class="ti-close"></i>
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
        @endforeach
        <div class="select-total">
            <span>total:</span>
            <h5>{{ $totalPrise }}</h5>
        </div>

        <div class="select-button">
            <a href="{{ route('ShopingCart') }}" class="primary-btn view-card">VIEW CARD</a>
            <a href="{{ route('Chekout') }}" class="primary-btn checkout-btn">CHECK OUT</a>
        </div>
    </div>

</li>
<li class="cart-price">{{ $total }} </li>

</ul>
</div>
</div>
</div>

```

```

</div>
<div class="nav-item">
  <div class="container">
    <div class="nav-depart">

</div>
<nav class="nav-menu mobile-menu">
  <ul>
    <li class="active"><a href="{{route('index_main')}}">Home</a></li>
    <li><a href="{{route('Shop')}}">Shop</a></li>
    <li><a href="#">Collection</a>
      <ul class="dropdown">
        <li><a href="http://public/Shop?collection_id=1">Men's</a></li>
        <li><a href="http://public/Shop?collection_id=2">Women's</a></li>
        <li><a href="http://public/Shop?collection_id=3">Kid's</a></li>
      </ul>
    </li>
    @if(session('admin_session'))
    <li><a href="{{route('admin.index')}}">Admin</a></li>

    @endif
    <li>
      @if(session('admin_session'))
      <a class="logo" href="{{route('admin.index')}}" class="login-panel"><i style="font-size:18px ;"
class="fas fa-user-circle"> </i>Account</a>
      @else
      <a class="logo" href="{{route('Account')}}" class="login-panel"><i style="font-size:18px ;"
class="fas fa-user-circle"> </i>Account</a>
      @endif

    </li>
    @if(Auth::check())
    <li><li> <a href="{{route('out')}}" class="login-panel"><i class="fa fa-user"></i>Logout</a></li>
    @else
    <li><li> <a href="{{route('login')}}" class="login-panel"><i class="fa fa-user"></i>Login /
Register</a></li>

    @endif

    </li>

  </ul>
</nav>
<div id="mobile-menu-wrap"></div>
</div>
</div>
</header>
<!-- Header End -->

<?мова розробки PHP

```

```

<!-- Hero Section Begin -->
<section class="hero-section">
  <div class="hero-items owl-carousel">
    @foreach($img as $ImgMainBaner)
      @if($ImgMainBaner->TypeImg['Name'] == 'MainBaner' )
        <div class="single-hero-items set-bg" data-setbg="assets/img/{{$ImgMainBaner['Name']}}">
          <div class="container">
            <div class="row">

              <div class="col-lg-5">
                <span>Bag,kids</span>
                <h1>Black friday</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
                  incididunt ut labore et dolore</p>
                <a href="{{route('Shop')}}" class="primary-btn">Shop Now</a>
              </div>

            </div>
            <div class="off-card">
              <h2>Sale <span>50%</span></h2>
            </div>

          </div>
        </div>
      @endif
    @endforeach
  </div>
</section>
<!-- Hero Section End -->

<!-- Banner Section Begin -->
<div class="banner-section spad">
  <div class="container-fluid">
    <div class="row">
      @foreach($img as $ImgMainBaner)
        @if($ImgMainBaner->TypeImg['Name'] == 'Men' )
          <div class="col-lg-4">
            <div class="single-banner">
              
              <div class="inner-text">
                <h4><a style="color:goldenrod" href="http://public/Shop?collection_id=1">Men's</a></h4>
              </div>
            </div>
          </div>
        @endif
      @endforeach
      @foreach($img as $ImgMainBaner)
        @if($ImgMainBaner->TypeImg['Name'] == 'Women' )
          <div class="col-lg-4">
            <div class="single-banner">
              

```

```

        <div class="inner-text">
            <h4><a style="color:goldenrod" href="http://public/Shop?collection_id=2">Women's</a></h4>
        </div>
    </div>
</div>
@endif
@endforeach
@foreach($img as $ImgMainBaner)
    @foreach($img as $ImgMainBaner)
        @if($ImgMainBaner->TypeImg['Name'] == 'Kids' )
            <div class="col-lg-4">
                <div class="single-banner">
                    
                    <div class="inner-text">
                        <h4><a style="color:goldenrod" href="http://public/Shop?collection_id=3">Kid's</a></h4>
                    </div>
                </div>
            </div>
        @endif
    @endforeach
</div>
</div>
<!-- Banner Section End -->

<!-- Women Banner Section Begin -->
@if(count($personalRecommendedProducts) > 0)
<section class="women-banner spad">
    <div class="container-fluid">
        <div class="row">
            @foreach($img as $ImgMainBaner)
                @if($ImgMainBaner->TypeImg['Name'] == 'WomenScrol' )
                    <div class="col-lg-3">
                        <div class="product-large set-bg" data-setbg="assets/img/products/{{$ImgMainBaner['Name']}}">
                            <h2>Особисті рекомендації</h2>
                        </div>
                    </div>
                @endif
            @endforeach
            <div class="col-lg-8 offset-lg-1">
                <div class="filter-control">
                    </div>
                <div class="product-slider owl-carousel">
                    @foreach($personalRecommendedProducts as $item)
                        <div class="product-item">
                            <div class="pi-pic">
                                

```



```

                {{item->prise}}
            </div>
        </div>
    </div>

    @endforeach

</div>
</div>

@foreach($img as $ImgMainBaner)
    @if($ImgMainBaner->TypeImg['Name'] == 'MenScrol' )
    <div class="col-lg-3 offset-lg-1">
        <div class="product-large set-bg m-large" data-setbg="assets/img/products/{{$ImgMainBaner['Name']}}">
            <h2>Загальні рекомендації</h2>

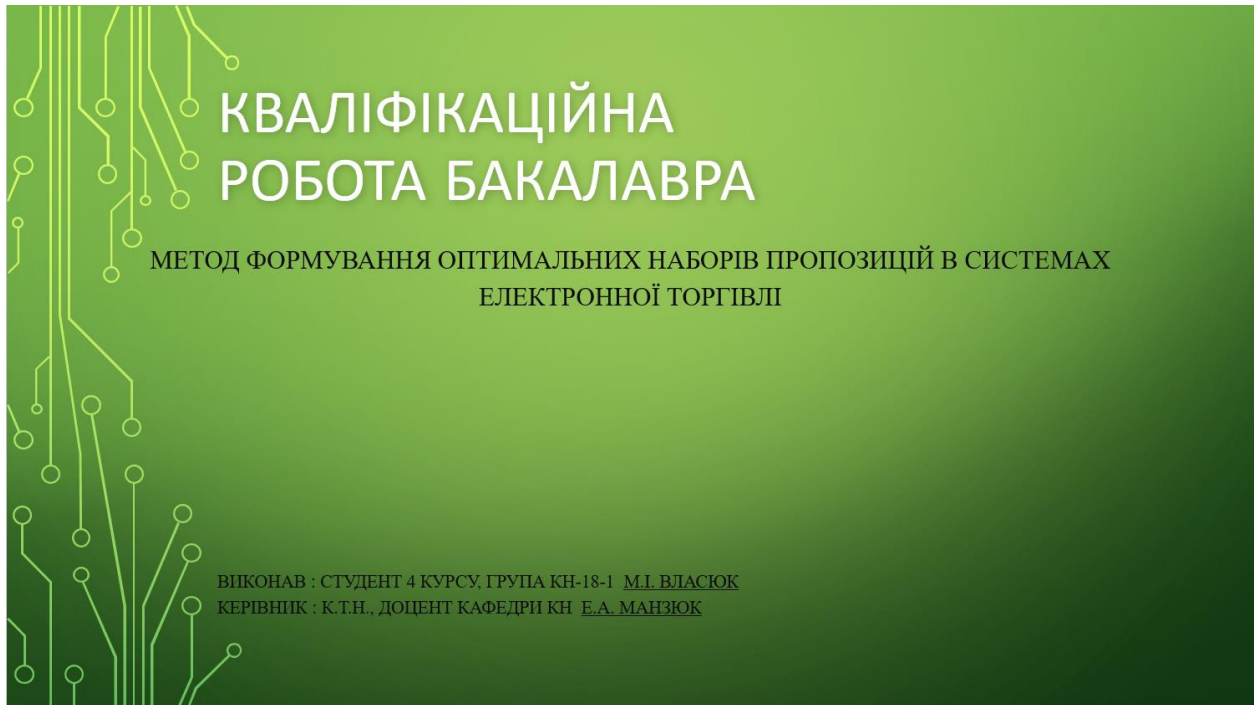
            </div>
        </div>
    @endif
@endforeach
</div>
</section>
<!-- Man Banner Section End -->

<!-- Footer Section Begin -->
<footer class="footer-section">
    <div class="container">
        <div class="row">
            <div class="col-lg-3">
                <div class="footer-left">
                    <div class="footer-logo">
                        <a href="#"></a>
                    </div>
                    <ul>
                        <li>Address: Khmelnitsky</li>
                        <li>Phone: +380980393886</li>
                        <li>Email: istauy1712@gmail.com</li>
                    </ul>
                    <div class="footer-social">
                        <a href="https://www.facebook.com/kolia.vlasiuk.3/"><i class="fa fa-facebook"></i></a>
                        <a href="https://www.instagram.com/eternal_sin_asmodeus/"><i class="fa fa-instagram"></i></a>
                    </div>
                </div>
            </div>
            <div class="col-lg-2 offset-lg-1">
                <div class="footer-widget">
                    <h5>Information</h5>
                    <ul>
                        <li><a href="{route('index_main')}">Home</a></li>

```


Додаток Г

Презентаційний матеріал



КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

МЕТОД ФОРМУВАННЯ ОПТИМАЛЬНИХ НАБОРІВ ПРОПОЗИЦІЙ В СИСТЕМАХ
ЕЛЕКТРОННОЇ ТОРГІВЛІ

ВИКОНАВ : СТУДЕНТ 4 КУРСУ, ГРУПА КН-18-1 М.І. ВЛАСЮК
КЕРІВНИК : К.Т.Н., ДОЦЕНТ КАФЕДРИ КН Е.А. МАНЗЮК



АКТУАЛЬНІСТЬ

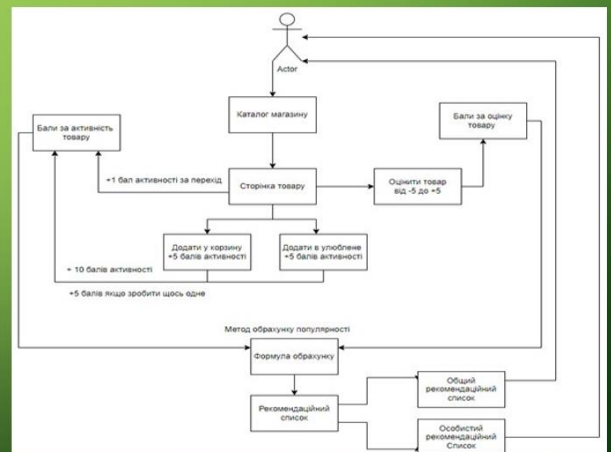
- У наш час є дуже багато систем електронної торгівлі (інтернет магазинів) і не у кожній системі присутній рекомендаційний список товарів який пришвидшує прийняття рішення користувача про покупку того чи іншого товару. Створення методу дозволить пришвидшити прийняття рішення, надання рейтингу обраному товару через голосування.

ЗАВДАННЯ

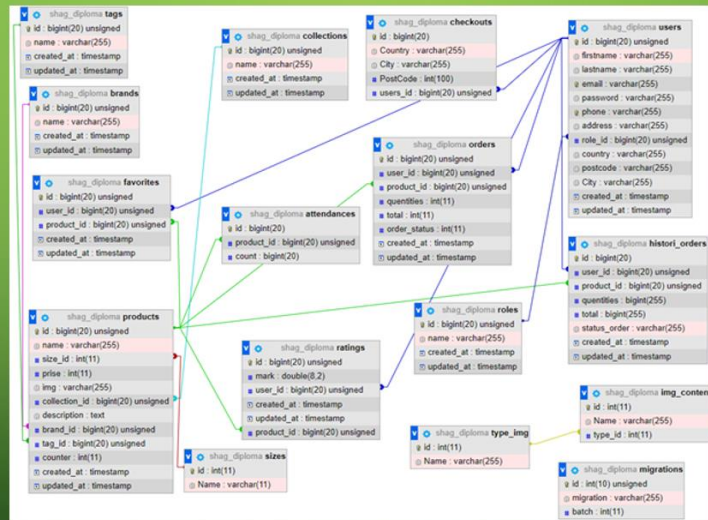
- Метою кваліфікаційної роботи бакалавра є розробка методу формування оптимальних наборів пропозицій в системах електронної торгівлі, на платформі PHP.
- 1. Провести аналіз систем формування наборів пропозиції за критеріями максимальної відповідності та популярності.
- 2. Розробити модель формування наборів пропозицій.
- 3. Розробити метод формування пропозицій на основі товарів широкого вжитку.
- 4. Реалізувати інформаційну технологію у вигляді програмного модуля пропозицій.
- 5. Провести експериментальне дослідження ефективності на основі виборів з пропонуванних наборів

КРОКИ ВИКОНАННЯ МЕТОДУ МЕТОДУ ПІДБОРУ ОПТИМАЛЬНИХ НАБОРІВ ПРОПОЗИЦІЙ У СИСТЕМІ ЕЛЕКТРОННОЇ ТОРГІВЛІ

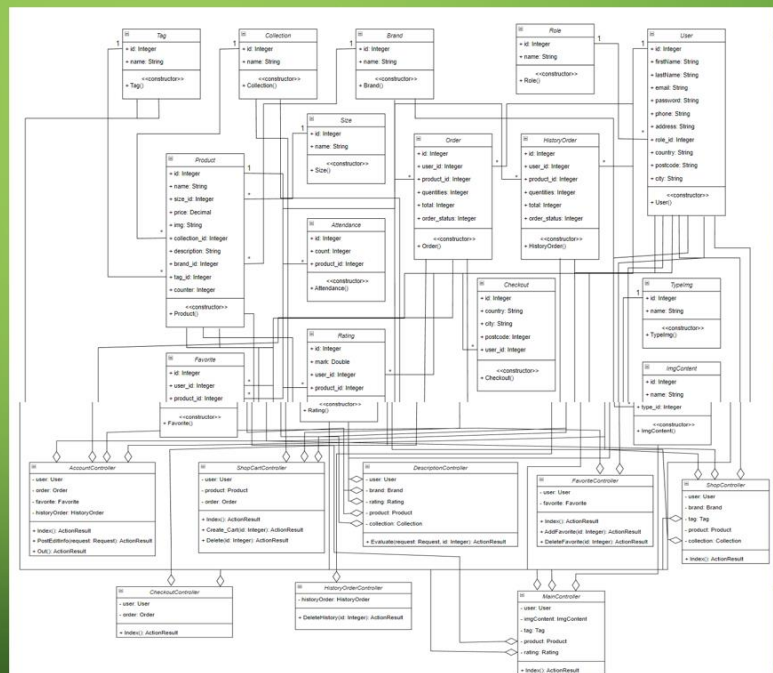
- Перший елемент : Зареєстрований користувач переглядає список товарів та їхній детальний перегляд з можливістю голосування, та виставлення рейтингу товару.
- Другий елемент: Після того як користувач у формі голосування виставив рейтинг заданому товару, данні з цієї форми передаються у базу даних.
- Третій елемент:
 - $P(H_i|A) = (P(H_i) + P(\Delta_i|H_i)) / (P(A))$ (2.1)
 - $P(H_i|A)$ – це коефіцієнт рейтингу товару, який розраховується за допомогою голосування та кількості відвідування, клієнтами даного товару
 - $P(H_i)$ – це оцінка рейтингу товару яку обирають клієнти під час голосування.
 - $P(\Delta_i|H_i)$ – це базовий коефіцієнт рейтингу товару який виставляє адміністратор/менеджер магазину для кожного товару.
 - $P(A)$ – це коефіцієнт кількості відвідування даного товару користувачами.
- Четвертий елемент : Після обробки отриманих даних метод формує список оптимальних рішень(каталог із декількох найпопулярніших товарів) да видає його користувачеві для ознайомлення та подальшого прийняття заключного рішення.



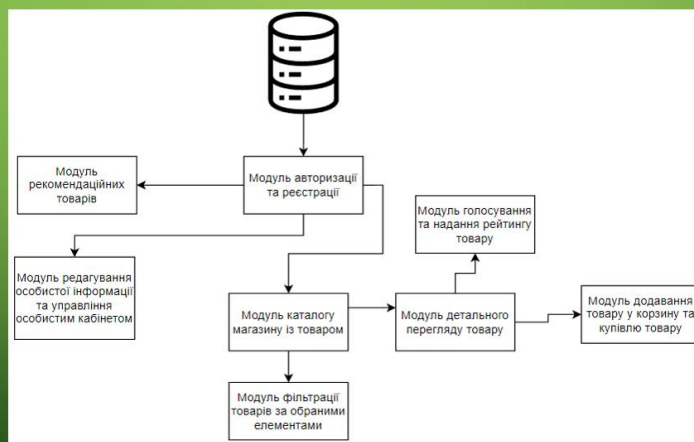
ДАТАЛОГІЧНА МОДЕЛЬ БАЗИ ДАНИХ



ДІАГРАМА КЛАСІВ МЕТОДУ ПІДБОРУ НАБОРІВ ОПТИМАЛЬНИХ РІШЕНЬ ІНТЕГРОВАНІХ У СИСТЕМУ ЕЛЕКТРОННОЇ ТОРГІВЛІ



АРХІТЕКТУРА УПРАВЛІННЯ СИСТЕМОЮ ЕЛЕКТРОННОЇ ТОРГІВЛІ ТА МЕТОДУ ФОРМУВАННЯ ОПТИМАЛЬНИХ НАБОРІВ ПРОПОЗИЦІЙ



ДЕМОНСТРАЦІЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ

Feah! What do you need?

Home Shop Collections Account My Account

Home > Login

Login

Username or email address *

Password *

Show Password Hide Password

[OR CREATE AN ACCOUNT](#)

Information **My Account**

Address: Kiev, Ukraine
Phone: +380963333333
Email: feah@feah.com.ua

Home My Account
Shop Profile
Login/Register Shipping Cart

Copyright ©2022. All rights reserved. This website is made with [Feah!](#) Webshop.

Feah! What do you need?

Home Shop Collections Account My Account

Home > Register

Register

Email address *

Firstname *

Lastname *

Phone number *

Address *

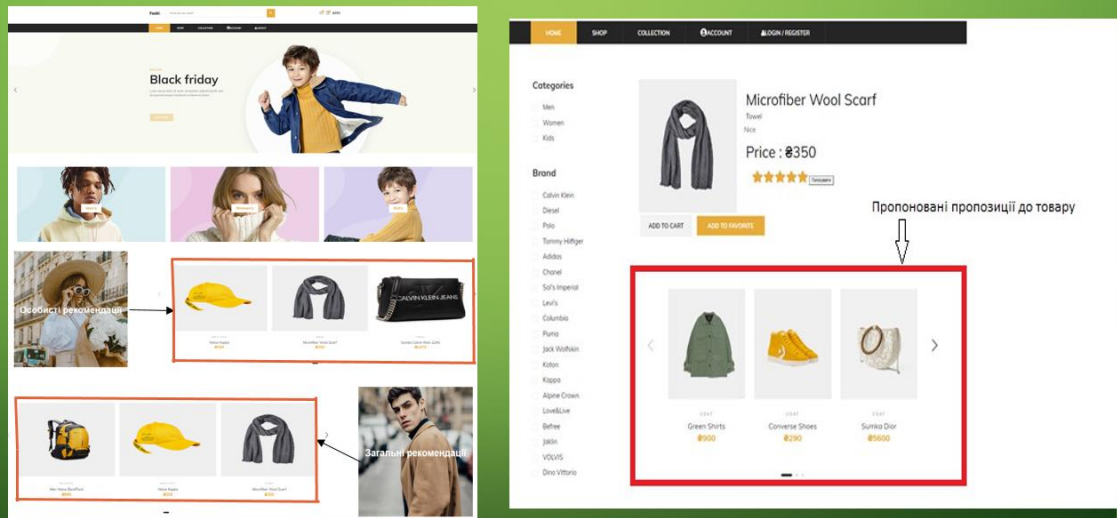
Country *

Password *

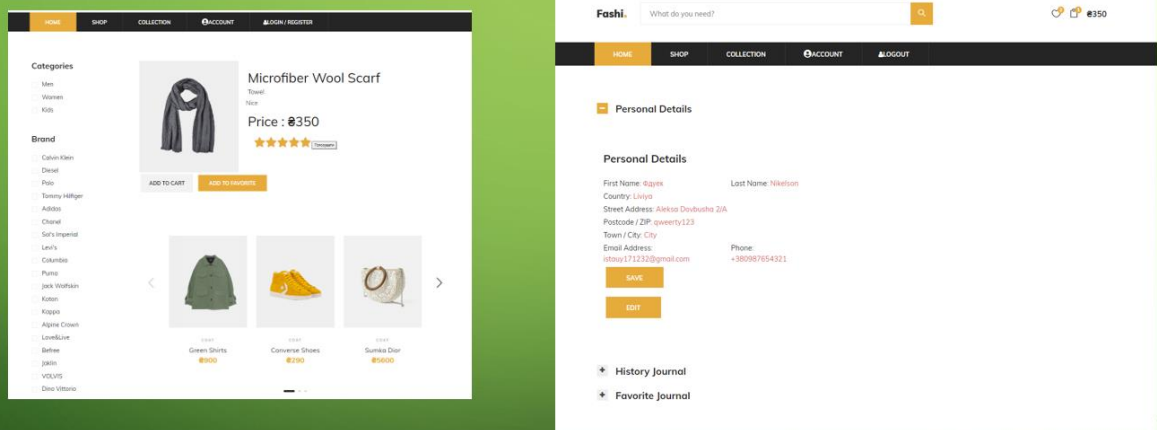
Confirm Password *

[OR LOGIN](#)

ДЕМОНСТРАЦІЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ



ДЕМОНСТРАЦІЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ



ДЕМОНСТРАЦІЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ

Fashi. What do you need? 🛒 #350

HOME SHOP COLLECTION ACCOUNT LOGOUT

IMAGE	PRODUCT NAME	PRICE	DELETE
	Microfiber Wool Scarf	€250	<input type="button" value="X"/>
	Guangzhou sweater	€570	<input type="button" value="X"/>

Subtotal **€820**
TOTAL **€1020**

Personal Details
History Journal

IMAGE	PRODUCT NAME	PRICE	QUANTITY	TOTAL	DELETE
	Microfiber Wool Scarf	€250	<input type="text" value="1"/>	€250	<input type="button" value="X"/>
	Guangzhou sweater	€570	<input type="text" value="1"/>	€570	<input type="button" value="X"/>
	Green Shirt	€200	<input type="text" value="1"/>	€200	<input type="button" value="X"/>

ДЕМОНСТРАЦІЯ РОБОТИ ВЕБ-ЗАСТОСУНКУ

Users

NAME	LAST NAME	PHONE	EMAIL	ROLE	COUNTRY	POSTCODE	CITY	OPERATIONS
Alexs	Brikssss	1234567899	adgbasfhsxfq@gmail.com	user	Costa	qweerty123	Boston	<input type="button" value="edit"/> <input type="button" value="delete"/> <input type="button" value="add"/>
Фадєєв	Nikelson	+380987654321	lstaury171232@gmail.com	user	Livya	qweerty123	City	<input type="button" value="edit"/> <input type="button" value="delete"/> <input type="button" value="add"/>
Kolia	Temp	1234567878	refo@gmail.ru	Admin	Ukraine	47779seve	Khmel	<input type="button" value="edit"/> <input type="button" value="delete"/> <input type="button" value="add"/>
Nick	Vlasuk	+380980393887	lstaury1712@gmail.com	user	Ukraine			<input type="button" value="edit"/> <input type="button" value="delete"/> <input type="button" value="add"/>
Alexs	Briks	+380987654329	l@gmail.com	user	Livya			<input type="button" value="edit"/> <input type="button" value="delete"/> <input type="button" value="add"/>

Name Price

Name Price

Description

Description

Example file input

Brand

Collection

Size

Tag

ВИСНОВОК

- Під час роботи над кваліфікаційною роботою бакалавра було проведено аналіз предметної області, визначено основну проблематику даної задачі, були складені задачі які потрібно було вирішити. Було обрано та описано технології, методи, сервіси, які використовуються у створенні методу формування оптимальних наборів пропозицій та інтегрованого у систему електронної торгівлі.
- Створено веб-технологію з можливістю продажу товарів через онлайн сервіс, та метод який допомагає швидше орієнтуватись та робити своє рішення про покупці пересічному користувачеві. Було розроблено функціональну структуру додатку та представлено її у вигляді груп функцій.
- Розроблено Структуру бази даних, її реляційну проекцію. Було описано таблиці, поля бази даних та реалізовано взаємодію між таблицями та групами функцій. Проведено тестування основних груп функцій та програмних модулів системи.
- Згідно з отриманим результатом, можна зробити висновок, що розроблена веб-технологія методу формування оптимальних наборів пропозицій інтегрована в систему електронної торгівлі працює справно вимоги технічного завдання виконанні повністю.

ДЯКУЮ ЗА УВАГУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 3.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 8%**

ID: 105483 Название: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод формування оптимальних наборів пропозицій в системах електронної торгівлі Добавлено в БД: 2022-06-15 Авторы: М.І. Власюк Руководители: Е.А. Манзюк Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	51117	792	7247 (14%)	125 (16%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Кафедра КН

ID перевірки:
1011587290

Дата перевірки:
15.06.2022 14:24:28 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.06.2022 14:25:39 EEST

ID користувача:
100005671

Назва документа: КРБ_Власюк_Виправлений варіант

Кількість сторінок: 61 Кількість слів: 8237 Кількість символів: 62248 Розмір файлу: 2.25 MB ID файлу: 1011456509

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

12.6% Схожість

Найбільша схожість: 3.25% з Інтернет-джерелом (http://elartu.tntu.edu.ua/bitstream/123456789/18982/2/2017_Opir_spe..)

10.3% Джерела з Інтернету

282

Сторінка 63

6.54% Джерела з Бібліотеки

138

Сторінка 67

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

54.8% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

54.8% Вилученого тексту з Бібліотеки

1

Сторінка 67

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

36

Підозріле форматування

19
сторінок

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод формування оптимальних наборів пропозицій в системах електронної торгівлі

Автор: студент групи КН-18-1 Власюка Миколи Ігоровича

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент Манзюк Е.А.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Власюка М.І. не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; серед запозичень знаходяться загальновідомі терміни, скорочення та матеріали оглядових джерел.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 16%;

- за системою Unichек: 12.6 %.

Керівник роботи



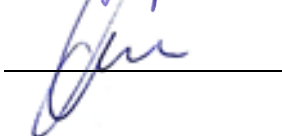
Едуард МАНЗЮК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-18-1 Власюка Миколи Ігоровича*

за темою: Метод формування оптимальних наборів пропозицій в системах електронної торгівлі

1. Актуальність обраної теми

Стрімке зростання обсягу доступної цифрової інформації та кількості відвідувачів Інтернету створило потенційну проблему інформаційного навантаження, що перешкоджає своєчасному доступу до елементів, що цікавлять в Інтернеті. Інформаційно-пошукові системи, такі як Google, DevilFinder та Altavista, частково вирішили цю проблему, але пріоритизація та персоналізація (коли система зіставляє доступний контент з інтересами та уподобаннями користувача) інформації були відсутні. Це збільшило попит на рекомендаційні системи більше, ніж будь-коли раніше. Рекомендаційні системи вигідні як постачальникам послуг, і користувачам. Вони знижують транзакційні витрати на пошук і вибір товарів серед онлайн-покупок. Системи рекомендацій також довели свою ефективність у покращенні процесу та якості прийняття рішень. В умовах електронної комерції рекомендаційні системи збільшують доходи, оскільки є ефективним засобом продажу більшої кількості продуктів

Відповідно, метод формування оптимальних наборів пропозицій в системах електронної торгівлі є актуальною задачею комп'ютерних наук.

2. Повнота розкриття мети та завдань роботи

Метою кваліфікаційної роботи бакалавра є розробка методу формування оптимальних наборів пропозицій в системах електронної торгівлі, на платформі PHP. Для досягнення мети ставляться наступні завдання: провести аналіз систем формування наборів пропозицій за критеріями максимальної відповідності та популярності, розробити модель формування наборів пропозицій, розробити метод формування пропозицій на основі товарів широкого вжитку, реалізувати інформаційну технологію у вигляді програмного модуля пропозицій, провести експериментальне дослідження ефективності на основі виборів з пропонованих наборів.

3. Зміст кожного розділу роботи

Перший розділ присвячений проведенню аналізу предметної області та визначенню основних параметрів для розв'язку поставленої задачі. Другий розділ присвячений проєктуванню функціональної структури інформаційної системи. Третій розділ присвячений програмній реалізації спроектованої функціональної структури інформаційної системи.

4. Оцінка розробленої інформаційної системи, її практична цінність

Реалізований програмний застосунок на базі методу формування оптимальних наборів пропозицій в системах електронної торгівлі може бути інтегровано в існуючі інтернет-магазини та системи торгівлі задля забезпечення оптимального набору пропозицій під час роботи із товарами.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота виконана на належному науково-методичному рівні та відповідає встановленим вимогам щодо оформлення такого роду праць.

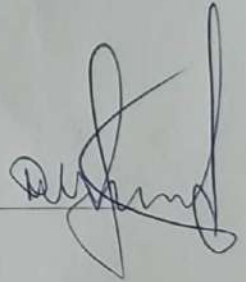
6. Недоліки кваліфікаційної роботи бакалавра

Кваліфікаційна робота бакалавра виглядала б привабливіше, якби у ній було розглянуто більше математичних підходів щодо вирішення задачі формування оптимальних наборів пропозицій.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Рецензент Макарушкін Д. А. доцент кафедри АКТТ





ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра

студента гр. КН-18-1 Власюка Миколи Ігоровича

за темою Метод формування оптимальних наборів пропозицій в системах електронної торгівлі

1. Актуальність теми

Стрімке зростання обсягу доступної цифрової інформації та кількості відвідувачів Інтернету створило потенційну проблему інформаційного навантаження, що перешкоджає своєчасному доступу до елементів, що цікавлять в Інтернеті. Інформаційно-пошукові системи, такі як Google, DevilFinder та Altavista, частково вирішили цю проблему, але пріоритизація та персоналізація (коли система зіставляє доступний контент з інтересами та уподобаннями користувача) інформації були відсутні. Це збільшило попит на рекомендаційні системи більше, ніж будь-коли раніше. Рекомендаційні системи вигідні як постачальникам послуг, і користувачам. Вони знижують транзакційні витрати на пошук і вибір товарів серед онлайн-покупок. Системи рекомендацій також довели свою ефективність у покращенні процесу та якості прийняття рішень. В умовах електронної комерції рекомендаційні системи збільшують доходи, оскільки є ефективним засобом продажу більшої кількості продуктів

Відповідно, метод формування оптимальних наборів пропозицій в системах електронної торгівлі є актуальною задачею комп'ютерних наук.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Тема кваліфікаційної роботи відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра, адже метою роботи є створення методу формування оптимальних наборів пропозицій в системах електронної торгівлі. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій.

3. Професійні та особистісні якості бакалавра

Студент Власюк М.І. під час роботи над кваліфікаційною роботою бакалавра та в процесі навчання проявив себе як старанний та дисциплінований студент,

продемонстрував достатній рівень знань та умінь за спеціальністю "Комп'ютерні науки", проявив себе відповідальним студентом. Опанував необхідні професійні навички за напрямком «Комп'ютерні науки».

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи студент Власюк М.І. показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Усі поставлені вимоги до роботи виконані в повному обсязі, проведено аналіз актуальності та відомих досліджень в межах обраної теми, також реалізоване відповідне програмне забезпечення.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

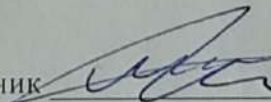
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Реалізований програмний застосунок на базі методу формування оптимальних наборів пропозицій в системах електронної торгівлі може бути інтегровано в існуючі інтернет-магазини та системи торгівлі задля забезпечення оптимального набору пропозицій під час роботи із товарами.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи достатній рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Керівник



к.т.н., доц. кафедри КН МАНЗІУК Е.А.