

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ

(назва теми)

РЕКОМЕНДУВАННЯ КОМП'ЮТЕРНИХ ІГОР

RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
RECOMMENDING COMPUTER GAMES

Виконав(ла): студент(ка) 2 курсу, групи КНТ-142м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЛОГВИНЕНКО Д.С.

(ПРИЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРИЗВИЩЕ та ініціали)

2023_____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ ПЗ, д.т.н, проф. _____
Сергій СУББОТІН
« _____ » _____ 2023 року

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

ЛОГВИНЕНКА Даниїла Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) _____ Дослідження та програмна реалізація методів
рекомендування комп'ютерних ігор. Research and Software Implementation of
Methods for Recommending Computer Games _____

керівник проєкту (роботи) _____ к.т.н., доцент СКРУПСЬКИЙ Степан Юрійович _____,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 14 » листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) _____ 01 грудня 2023 року _____
3. Вихідні дані до проєкту (роботи) _____ рекомендована література, технічне
завдання _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити) _____ 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали
і методи. 3. Розробка архітектури програми. 4. Реалізація програми.
5. Експлуатація, тестування та експериментальне дослідження програми. _____

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	СКРУПСЬКИЙ С.Ю., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання « 04 » вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

_____ Даниїл ЛОГВИНЕНКО
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Степан СКРУПСЬКИЙ
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
101 с., 24 рис., 3 табл., 3 дод., 11 джерел.

КОМП'ЮТЕРНА ГРА, РЕКОМЕНДУВАННЯ, ПРОГНОЗУВАННЯ,
RANDOM FOREST, ВЕБЗАСТОСУНОК.

Об'єкт роботи – процес надання інформації про комп'ютерні ігри.
Предмет роботи – методи рекомендування комп'ютерних ігор. Мета роботи – розробити програмне забезпечення надання інформації про комп'ютерні ігри з можливістю рекомендування ігор користувачам на основі виявлення майбутніх трендів.

Матеріали, методи та технічні засоби: мова Python як основна мова розробки, вебфреймворк Django, система MySQL.

Результати. Програма, яка розроблена для любителів комп'ютерних ігор і надає їм можливість знаходити нові для них комп'ютерні ігри, ділитися з іншими гравцями думками щодо комп'ютерних ігор з власного досвіду, впорядкувати інформацію про персонажів комп'ютерних ігор. До складу програми інтегровано рекомендування комп'ютерних ігор. Проведено експериментальне дослідження, яке підтвердило доцільність використання ансамблів дерев рішень порівняно з окремими деревами рішень щодо прогнозування оцінки комп'ютерної гри, на основі чого встановлюються майбутні тренди, що визначають рекомендації, надані користувачам.

Висновки. Наукова новизна роботи полягає в запропонованому методі рекомендування комп'ютерних ігор, який базується на створенні рекомендацій з комп'ютерних ігор, які мають найкращі очікувані оцінки та найбільше зростання, використовуючи для прогнозування оцінок ансамблі дерев рішень на основі методу Random Forest.

Галузь використання. Платформа для любителів комп'ютерних ігор.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 101 pages, 24 figures, 3 tables, 3 appendixes, 11 sources.

COMPUTER GAME, RECOMMENDATION, PREDICTION, RANDOM FOREST, WEB APPLICATION.

The object of the work is the process of providing information about computer games. The subject of the work is methods of recommending computer games. The purpose of the work is to develop software for providing information about computer games with the possibility of recommending games to users based on the detection of future trends.

Materials, methods and technical means: Python language as the main development language, Django web framework, MySQL system.

The results. The program, which is designed for fans of computer games and gives them the opportunity to find new computer games, to share with other players their thoughts about computer games from their own experience, to organize information about the characters of computer games. Computer game recommendations are integrated into the program. An experimental study was conducted. It confirmed the feasibility of using ensembles of decision trees compared to individual decision trees for predicting the evaluation of a computer game, based on which future trends are established determining the recommendations provided to users.

Conclusions. The scientific novelty of the work is in the proposed method for recommending computer games, which is based on the creation of computer game recommendations that have the best expected ratings and the greatest growth, using ensembles of decision trees based on the Random Forest method to predict ratings.

Field of use. A platform for fans of computer games.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Програмні інструменти надання інформації про комп'ютерні ігри.....	10
1.2 Порівняння розроблюваної програми з існуючими	13
1.3 Висновки за розділом 1	15
2 Матеріали і методи.....	16
2.1 Рекомендування комп'ютерних ігор у розрізі визначення трендів	16
2.2 Метод рекомендування комп'ютерних ігор	18
2.3 Висновки за розділом 2	20
3 Розробка архітектури програми.....	22
3.1 Визначення сценаріїв використання програми користувачами	22
3.2 Рішення щодо вибору засобів розробки	25
3.3 Визначення структури даних у базі даних	26
3.4 Висновки за розділом 3	38
4 Реалізація програми	39
4.1 Визначення послідовності роботи з програмою	39
4.2 Особливості реалізації програми.....	40
4.3 Висновки за розділом 4	57
5 Експлуатація, тестування та експериментальне дослідження програми	58
5.1 Призначення програми	58
5.2 Виконання програми.....	58
5.3 Експериментальне дослідження	62
5.4 Висновки за розділом 5	64
Висновки	66
Перелік джерел посилання	67
Додаток А Технічне завдання	68

Додаток Б Текст програми	73
Додаток В Слайди презентації.....	94

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД – база даних;

ДР – дерево рішень.

ВСТУП

Комп'ютерні ігри є сферою життя людей, яка значно впливає на сучасне життя, культуру, взаємовідношення. На даний момент відбувається стрімке зростання кількості людей, які грають в комп'ютерні ігри, враховуючи таким чином тенденції пандемії, в підсумку очікується, що кількість людей, які грають у комп'ютерні ігри в світі досягне 3,38 мільярдів людей у 2023 році [1]. Дана цифра виражається не тільки в привертанні уваги, але і в тому, що світовий ринок комп'ютерних ігор у економічному вираженні має досягти отримання 187,7 мільярдів доларів доходу вже цього року [1].

Відповідно і кількість можливих комп'ютерних ігор на ринку є величезною, а тому знайти те, що може зацікавити користувача непросто. Виявлення тенденцій на ринку може в цьому допомогти. Врахування тенденцій ринку в рекомендаціях для користувачів програми може бути корисним інструментом, який разом з іншими особливостями може привернути увагу потенційних користувачів. Додатковими можливостями є акцент на персонажах комп'ютерних ігор, приділення уваги на основі централізованої платформи, якою і має бути програма, що розробляється, взаємодії різних користувачів стосовно окремих персонажів, які в іграх існують. У той же час майданчики для продажу ігор є достатньо централізованими, входження нових гравців є достатньо ускладненим, тому фактично рекламування, надання доступу до таких майданчиків є достатньо важливими і практично затребуваним рішенням.

Метою роботи є розробити програмне забезпечення надання інформації про комп'ютерні ігри з можливістю рекомендування ігор користувачам на основі виявлення майбутніх трендів.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Програмні інструменти надання інформації про комп'ютерні ігри

Зважаючи на те, що програми не створюються для того, щоб виконувати виключно рекомендації комп'ютерних ігор, було розглянуто програмне забезпечення, що надає інформацію про комп'ютерні ігри з можливістю або відсутністю можливості рекомендації як однією з основних властивостей програм, що розглядаються. Спершу до огляду представлено одну з основних платформ, яка окрім інформування про комп'ютерні ігри виконує ще й розповсюдження ігор, – Steam (рис. 1.1) [2].

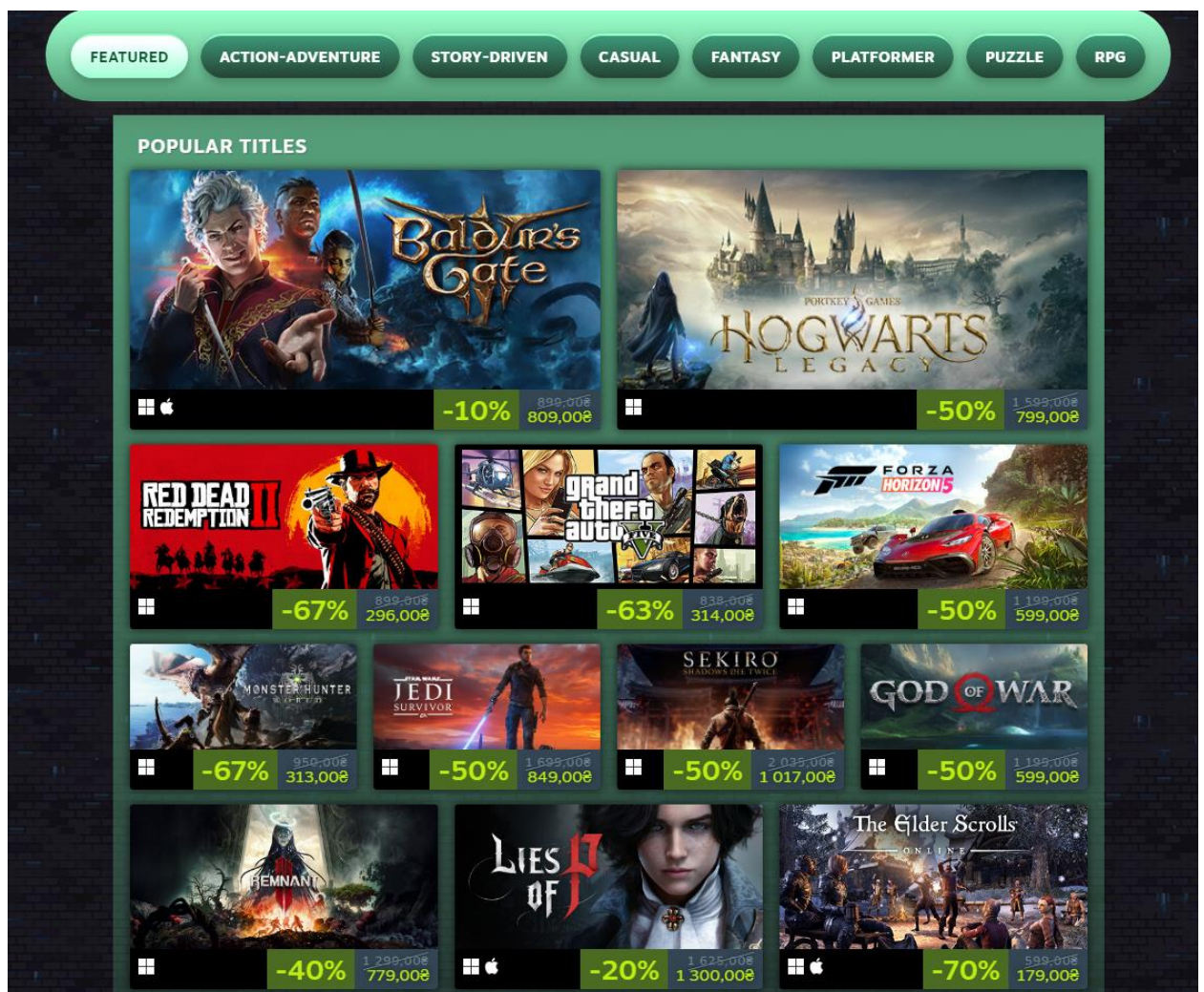
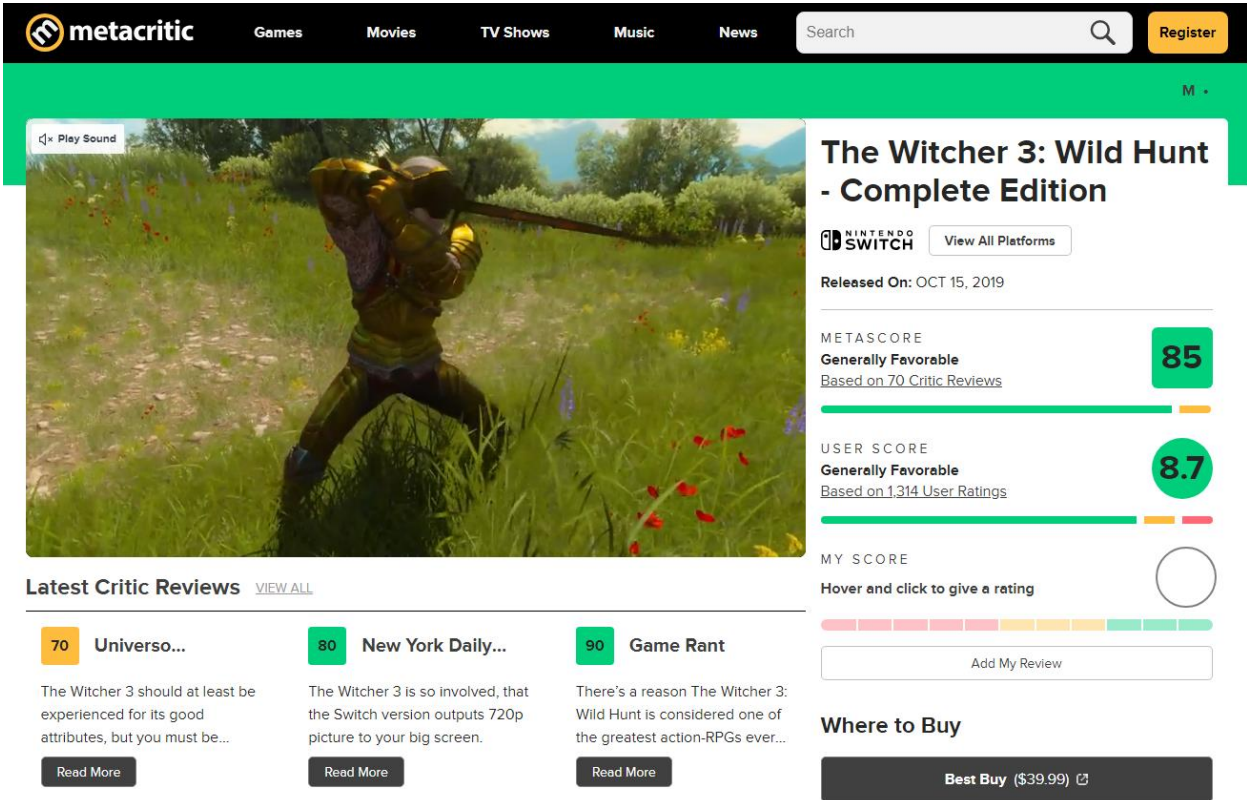


Рисунок 1.1 – Сторінка каталогу комп'ютерних ігор за жанром на платформі Steam

Враховуючи направленість платформи на продаж комп'ютерних ігор, великим мінусом інтерфейсу можна вважати наявність великої кількості реклами. Фактично реклама займає велику кількість простору як на сторінках окремих комп'ютерних ігор, так і на сторінці пошуку, де немає фактично банерів, але знижки домінують при відображенні.

Комп'ютерні ігри можна переглядати як за рейтингом, так і найновіші або за жанром, який необхідно обрати з меню. За кожною комп'ютерною грою надаються дані стосовно відгуків за грою, дати публікації, розробника, видавця, доступних мов, жанрів, посилань на вебсайт, вартості гри, трендів щодо зміни вартості, опису гри, фотографій гри, відео, системних вимог.

Для визначення того, чи варто грати в комп'ютерну гру, важливими є відгуки, а одним із найвідоміших вебсайтів, де представляються відгуки, є Metacritic (рис. 1.2) [3].



metacritic Games Movies TV Shows Music News Search Register

The Witcher 3: Wild Hunt - Complete Edition

NINTENDO SWITCH View All Platforms

Released On: OCT 15, 2019

METAScore
Generally Favorable
Based on 70 Critic Reviews
85

USER SCORE
Generally Favorable
Based on 1,314 User Ratings
8.7

MY SCORE
Hover and click to give a rating
Add My Review

Latest Critic Reviews [VIEW ALL](#)

Score	Publication	Review Snippet	Read More
70	Universo...	The Witcher 3 should at least be experienced for its good attributes, but you must be...	Read More
80	New York Daily...	The Witcher 3 is so involved, that the Switch version outputs 720p picture to your big screen.	Read More
90	Game Rant	There's a reason The Witcher 3: Wild Hunt is considered one of the greatest action-RPGs ever...	Read More

Where to Buy
Best Buy (\$39.99) ↗

Рисунок 1.2 – Сторінка комп'ютерної гри на вебсайті Metacritic

Metacritic серед відгуків на фільми, музичні альбоми містить також і відгуки на комп'ютерні ігри. Відгуки розділяються на відгуки критиків та на відгуки звичайних користувачів, відповідно і середня оцінка рахується окремо. Але окрім відгуків надається також і базова інформація про гру, що зокрема включає опис, сертифікат щодо вікових обмежень, доступних платформ для комп'ютерної гри, дати видання, розробника, жанрів, посилання для придбання гри, що не у всіх країнах працює. При цьому переважаючий простір займають відгуки. Однак стосовно функціональності вся вона пов'язана зі створенням відгуків, а інших функцій окрім відображення рейтингів, певних переліків комп'ютерних ігор, пошуку комп'ютерних ігор немає. Пошук відбувається одразу і за іграми, і за фільмами, музикою, що не сприяє зручності, до того ж єдиним критерієм є назва.

На платформі Steam пошук реалізується через визначення ряду критеріїв, включаючи спеціальні пропозиції за назвою, жанри ігор, кількість гравців, підтримувані пристрої, платформи, доступні мови, а також діапазон вартості.

Третім розглянутим вебзастосунком є IMDb (рис. 1.3) [4]. IMDb відомий як онлайн база даних для надання інформації про фільми, але також і відеоігри. Зважаючи на певні аналогії між цими категоріями, фільмами і відеоіграми, це є достатньо логічним. Пошук загалом відбувається за всім контентом, однак в критеріях пошуку можна встановити пошук саме відеоігор. Особливостями IMDb є наявність всесвітньо відомого рейтингу. За кожною комп'ютерною грою надається інформація про режисера, авторів сценарію, відомих зірок, отримані винагороди, фотографії, відгуки користувачів, дату випуску, країну походження, офіційний вебсайт, основну мову. На кожній сторінці відеогри надається перелік подібних ігор, який можна вважати певною системою рекомендацій.

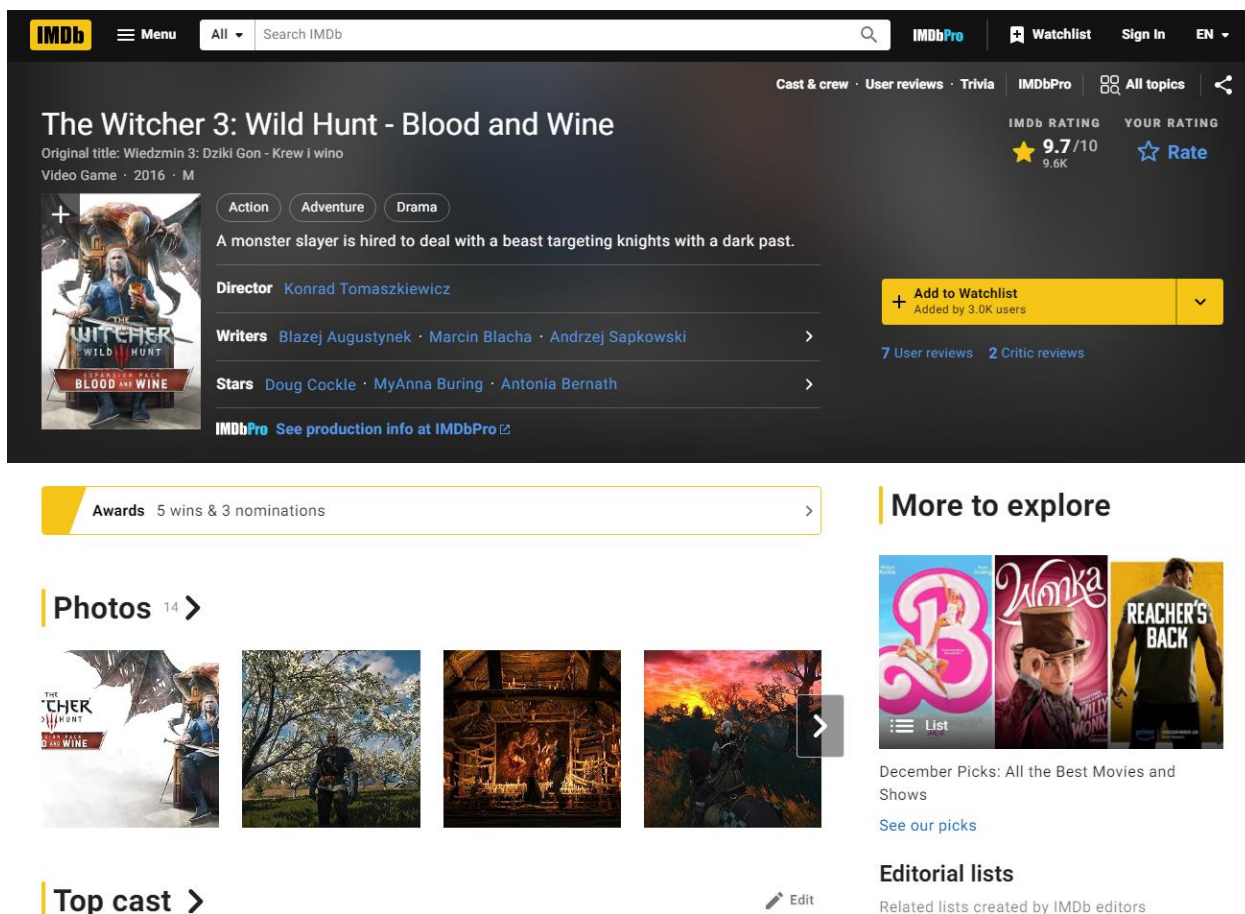


Рисунок 1.3 – Сторінка відеогри на вебсайт IMDB

1.2 Порівняння розроблюваної програми з існуючими

Зведене порівняння виконувалось за програмами Steam, Metacritic, IMDB Video Games з визначенням відмінностей порівняно з Gamerecsoft – програмою, яка розробляється (табл. 1.1).

Зі всіх аналогів Metacritic відзначається дуже обмеженим пошуком, який включає не тільки комп'ютерні ігри, а виконується тільки за назвою, тому щодо пошуку нових комп'ютерних ігор такий варіант є незручним. Тільки IMDB з аналогів дозволяє зберігати ігри та повертатися до них потім. Жоден з аналогів не виділяє окремо персонажі комп'ютерних ігор, тобто вся взаємодія відбувається лише стосовно комп'ютерних ігор цілком: шукати персонажів, переглядати інформацію лише за ними, а не за всією грою, ділитися враженнями неможливо в аналогах. Використання такої можливості у розробці має додатково привернути увагу до вебзастосунку.

Таблиця 1.1 – Аналіз розроблюваної та існуючих програм

Показник	Steam	Metacritic	IMDB Video Games	Gamerecsoft
Пошук комп'ютерних ігор	Багато-критеріальний	За назвою, суміщений з іншими типами контенту	Багато-критеріальний, суміщений з іншими типами контенту	Багато-критеріальний
Придбання комп'ютерних ігор у програмі	+	—	—	—
Збереження комп'ютерних ігор	—	—	+	+
Виділення персонажів комп'ютерних ігор для пошуку і інформування	—	—	—	+
Визначення трендів	Зміна вартості	—	—	Зміна оцінки
Рекомендування комп'ютерних ігор	—	—	За подібністю на задану гру	За трендами

Також надання рекомендацій комп'ютерних ігор, що має бути виконано у розробці, має сприяти зручності користувачів під час пошуку

нових ігор. Надання рекомендацій доступно тільки в IMDB Video Games. Логіка передбачає надання рекомендацій на сторінці відеогри, відповідно рекомендації стосуються подібності інших ігор до неї.

1.3 Висновки за розділом 1

Визначені завдання на роботу формують наступний перелік:

- аналіз існуючих програм, які виконують надання інформації про комп'ютерні ігри;
- дослідження методів рекомендування комп'ютерних ігор;
- проєктування програми надання інформації про комп'ютерні ігри;
- реалізація програми надання інформації про комп'ютерні ігри.

У першому розділі проаналізовано програми, які виконують надання інформації про комп'ютерні ігри. До них віднесено Steam, Metacritic, IMDB Video Games, визначено особливості розробки порівняно з цими рішеннями.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Рекомендування комп'ютерних ігор у розрізі визначення трендів

Виходячи з мети роботи, де передбачено першочергово приділити увагу саме трендовим комп'ютерним іграм, потрібно визначити фактично задачу рекомендування комп'ютерних ігор за цією концепцією. Тренди можна визначати різними способами, тому спершу має бути обраний такий спосіб.

У даному випадку вирішено розглядати тренди щодо змін оцінки комп'ютерних ігор. Під цим мається на увазі виявлення в першу чергу позитивної зміни оцінки комп'ютерної гри протягом тривалого часу. Таким тривалим часом може бути або послідовність місяців, протягом якої відбувається і підтримується зміна, або рік. Такі зміни можна вважати стійкими, а відповідно отримання такої інформації вказує на те, що така комп'ютерна гра починає привертати нову увагу користувачів.

Зміна оцінки може бути обумовлена змінами в самій грі, а вираження це може мати, наприклад, у змінах сценарію. При представленні комп'ютерних ігор сценарій найчастіше передбачає певну постановку концепції, а не повний опис сценарію, тобто фактично його можна розглядати як опис комп'ютерної гри.

Тоді задача рекомендування комп'ютерних ігор полягає в тому, щоб визначити певну підмножину комп'ютерних ігор, які або мають високу загальну оцінку, при цьому ця оцінка не знижується, або мають тренд щодо збільшення оцінки протягом тривалого періоду часу. Для того щоб такий тренд виявити, потрібно виконати прогнозування значення оцінки комп'ютерної гри.

Кожна комп'ютерна гра характеризується значеннями своїх властивостей, при цьому встановлюється час, який пройшов для гри з періоду її випуску. На основі значень всіх властивостей комп'ютерної гри

потрібно визначити тренд щодо зміни оцінки, яку грі виставили користувачі в середньому, та саму таку оцінку, після чого визначити комп'ютерні ігри, які можуть відноситися до рекомендованих. Ці і є задача рекомендування комп'ютерних ігор у тому розрізі, в якому вона розглядається в даній роботі.

Оскільки центральною проблемою розв'язання цієї задачі є прогнозування оцінки комп'ютерної гри, то необхідно визначити спосіб створення такої моделі прогнозування. Така модель прогнозування може бути створена на основі різних методів, зокрема лінійної регресії, дерев рішень (ДР), нейронних мереж, методу опорних векторів та інших методів прогнозування. У даній роботі акцент зроблений на ДР, хоча окремо перевірялись результати як базових методів і лінійної регресії, і методу опорних векторів, однак вони призвели до отримання гірших результатів прогнозування, а лінійна регресія не завжди призводила до отримання практично застосовуваних результатів, тобто в деяких випадках відхилення результатів були значними (випадки, коли відхилення отриманої оцінки в багатьох випадках становить декілька балів точно не можна розглядати як практично придатний варіант рішення).

ДР – це непараметричний метод навчання, за якого навчання відбувається з учителем, а застосовуватися він може для завдань класифікації і регресії (прогнозування), при цьому метою є створення моделі, яка прогнозує значення цільової змінної через створення простих правил прийняття рішень, виведених із характеристик даних: логічним з цього є визначення роботи ДР через рекурсивне розділення вхідних даних на підмножини на основі значення одного атрибута кожен раз, відповідно потрібно підібрати атрибут, щодо якого буде відбуватися перевірка, а також встановити порогове значення, в чому загалом і полягає навчання цієї моделі [5].

Одним з варіантів моделей ДР є метод Random Forest, який заснований на концепції «bagging», ідея якої полягає в тому, щоб для прогнозування використати групу незалежних моделей і використати

усереднений прогноз, отриманий від них, для обчислення підсумкового результату, а оскільки моделі незалежні, то і помилки не корельовані, тоді якщо моделі не мають систематичних помилок, то середнє значення багатьох помилок має бути близьким до нуля [6].

2.2 Метод рекомендування комп'ютерних ігор

Модель прогнозування оцінки комп'ютерної гри має бути побудована на основі наступних вхідних ознак:

- сценарій комп'ютерної гри;
- рік видання комп'ютерної гри;
- кількість років, яка пройшла до моменту визначення середньої оцінки (роки до оцінювання);
- жанри комп'ютерної гри;
- сертифікат щодо вікових обмежень.

Вихідною ознакою є середня оцінка, яку було встановлено комп'ютерній грі на основі відгуків користувачів. При цьому має бути встановлено поріг щодо кількості таких відгуків, тобто мінімальна кількість відгуків, за якої ці дані про таку комп'ютерну гру можуть використовуватися як екземпляр вибірки даних. Якщо нижній бар'єр не подолано, то такий екземпляр має бути відкинутий, адже на основі декількох відгуків не можна виконати якісне оцінювання.

Жанри комп'ютерної гри мають бути представлені вектором з нулів та одиниць, що відповідають попередньо сформованому вектору жанрів. Тобто має бути заданий перелік жанрів, який розглядається. Після цього для кожного жанру має бути визначено, чи входить він до кожної комп'ютерної гри. Якщо не входить, то має задаватися нульове значення, якщо входить – 1. Таким чином формується вектор, що представляє жанри конкретної комп'ютерної гри як один з екземплярів вибірки даних.

Сертифікат щодо вікових обмежень має бути перетворено на конкретне вікове обмеження, тобто потрібно використовувати кількість років, з якої дозволяється гра, перегляд цієї комп'ютерної гри.

На другому етапі має бути виконане перетворення тексту сценарію шляхом організації вкладання слів, для чого запропоновано використовувати метод word2vec [7]. Для кожного слова даний метод повинен обчислити певну оцінку, сформовану як вектор дійсних чисел.

Створення моделі word2vec має відбуватися на наборі всіх доступних описів сценаріїв комп'ютерних ігор, при чому з використанням тільки однієї природної мови. Для створення і навчання моделі тексти мають бути оброблені, відкидаючи частини тексту, що не є словами, прибираючи стоп-слова, в підсумку виділяючи тільки інформативну частину та на ній навчаючи модель. Після того, як модель буде навчена, вона має бути застосована для оброблення кожного сценарію. За кожним сценарієм має бути відділена початкова фраза, яка складається не більше ніж з 300 символів разом з пробілами. Після цього така фраза має бути оброблена через відкидання нетекстових даних, стоп-слів, а тоді подана на вхід моделі, що має обчислити вектор для кожного слова з фрази. Тоді значення кожного елемента у векторі має бути усереднено на основі значень, що відповідають кожному слову. Після цього отриманий вектор і є вхідними даними майбутньої моделі прогнозування і представляє опис комп'ютерної гри.

На третьому етапі має бути виконано навчання ансамблю ДР, застосовуючи для цього сформовану вибірку на перших двох етапах. Вибірka має складатися зі значень вхідних ознак та значення вихідної ознаки.

На четвертому етапі реалізується практичне застосування навченої моделі. Для цього має бути визначено комп'ютерні ігри, які розглядаються в якості можливих рекомендацій. Тобто таких, які взагалі можна розглядати такими. Для кожної такої комп'ютерної гри потрібно визначити її поточну оцінку та встановити період прогнозування, тобто кількість років, яка має

пройти на той момент, коли має бути обчислена середня оцінка, після видання комп'ютерної гри.

Окремо варто зазначити, що прогнозування може виконуватися на місяць вперед або на інший строк. Однак тоді потрібно використовувати відповідні дані, для яких визначено конкретне число і є відомим різні моменти виконання обчислень середньої оцінки. У експериментальному дослідженні, яке проводилось в розділі 5, виконувалось прогнозування за річним принципом. Тобто в такому випадку потрібно визначити конкретний період прогнозування і задати потрібну кількість років (або місяців у відповідному випадку).

Після того, як значення для кожної комп'ютерної гри, для якої виконується прогнозування, обчислені, необхідно виконати прогнозування на основі навченої моделі прогнозування за допомогою ансамблю ДР.

На п'ятому етапі для кожної комп'ютерної гри з четвертого етапу необхідно визначити зміну оцінки. При цьому потрібно встановити граничну зміну та граничну оцінку. Тоді всі комп'ютерні ігри, які мають прогнозовану оцінку вище граничної, а також всі комп'ютерні ігри, які мають зміну граничної оцінки протягом періоду (року чи місяця) не меншу ніж задана, мають бути виділені у підмножину результатів. Ця підмножина і є остаточним результатом роботи метода рекомендування комп'ютерних ігор, що запропоновано в роботі.

Таким чином визначена підмножина комп'ютерних ігор в результаті складається з ігор, які мають найкращий загальний рівень та тих, для яких передбачається тенденція до значної зміни оцінки.

2.3 Висновки за розділом 2

У розділі 2 розглянуто проблему рекомендування комп'ютерних ігор, визначено постановку задачі, в якій вона розглядалась у розділі, запропоновано метод її розв'язання.

Наукова новизна роботи полягає в запропонованому методі рекомендування комп'ютерних ігор, який базується на створенні рекомендацій з комп'ютерних ігор, які мають найкращі очікувані оцінки та найбільше зростання, використовуючи для прогнозування оцінок ансамблі ДР на основі методу Random Forest.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

3.1 Визначення сценаріїв використання програми користувачами

Користувачі програми без особливих прав доступу повинні мати можливість використовувати наступні сценарії:

- пошук комп'ютерних ігор;
- рекомендування комп'ютерних ігор;
- перегляд деталізації розробника комп'ютерних ігор;
- пошук компаній-розробників комп'ютерних ігор;
- пошук осіб, задіяних у розробці комп'ютерних ігор;
- перегляд деталізації особи, задіяної в створенні комп'ютерних ігор;
- перегляд деталізації комп'ютерної гри;
- перегляд деталізації персонажа комп'ютерної гри;
- пошук персонажів комп'ютерних ігор;
- перегляд рейтингу всіх комп'ютерних ігор;
- перегляд рейтингу комп'ютерних ігор за жанром;
- перегляд рейтингу комп'ютерних ігор за доступною мовою в інтерфейсі;
- перегляд рейтингу комп'ютерних ігор за платформою;
- перегляд рейтингу нових комп'ютерних ігор;
- перегляд рейтингу персонажів комп'ютерних ігор;
- надання зворотного зв'язку відносно комп'ютерної гри;
- рекомендування зворотного зв'язку відносно комп'ютерної гри;
- надання зворотного зв'язку відносно персонажа комп'ютерної гри;
- рекомендування зворотного зв'язку відносно персонажа комп'ютерної гри;
- публікація емоції стосовно комп'ютерної гри;
- публікація відповіді на емоцію стосовно комп'ютерної гри;
- перегляд всіх власних оформлених зворотних зв'язків;
- перегляд всіх власних реакцій на дії інших гравців;

- створення закладки на комп'ютерній грі;
- прибирання закладки з комп'ютерної гри;
- перегляд комп'ютерних ігор, віднесених користувачем до своїх закладок;
- перегляд деталізації даних користувача;
- оформлення підписки на розробника комп'ютерних ігор;
- відключення підписки на розробника комп'ютерних ігор;
- оформлення підписки на гравця;
- відключення підписки на гравця;
- перегляд нових комп'ютерних ігор від розробників, на яких є підписка;
- перегляд стрічки дій гравців, на яких є підписка;
- авторизація.

Не всі з цих сценаріїв можуть використовуватися одразу в певній мірі. Певні з них, а точніше ті, які передбачають збереження деяких даних у базі даних, потребують авторизації, тому під час роботи вони обов'язково включають сценарій авторизації. Це відповідним чином враховано на діаграмі варіантів використання для користувача (рис. 3.1).

Менеджер програми є користувачем з особливими правами доступу, а права ці полягають в тому, що він може вносити ті дані, з якими працюють користувачі без таких прав, включаючи комп'ютерні ігри, персонажі цих ігор, розробників, учасників розробки. Сценарії менеджера програми (рис. 3.1):

- реєстрація нової компанії-розробника комп'ютерних ігор;
- редагування даних розробника комп'ютерних ігор;
- редагування даних комп'ютерної гри;
- реєстрація нової комп'ютерної гри;
- реєстрація персонажа комп'ютерної гри;
- редагування даних персонажа комп'ютерної гри.

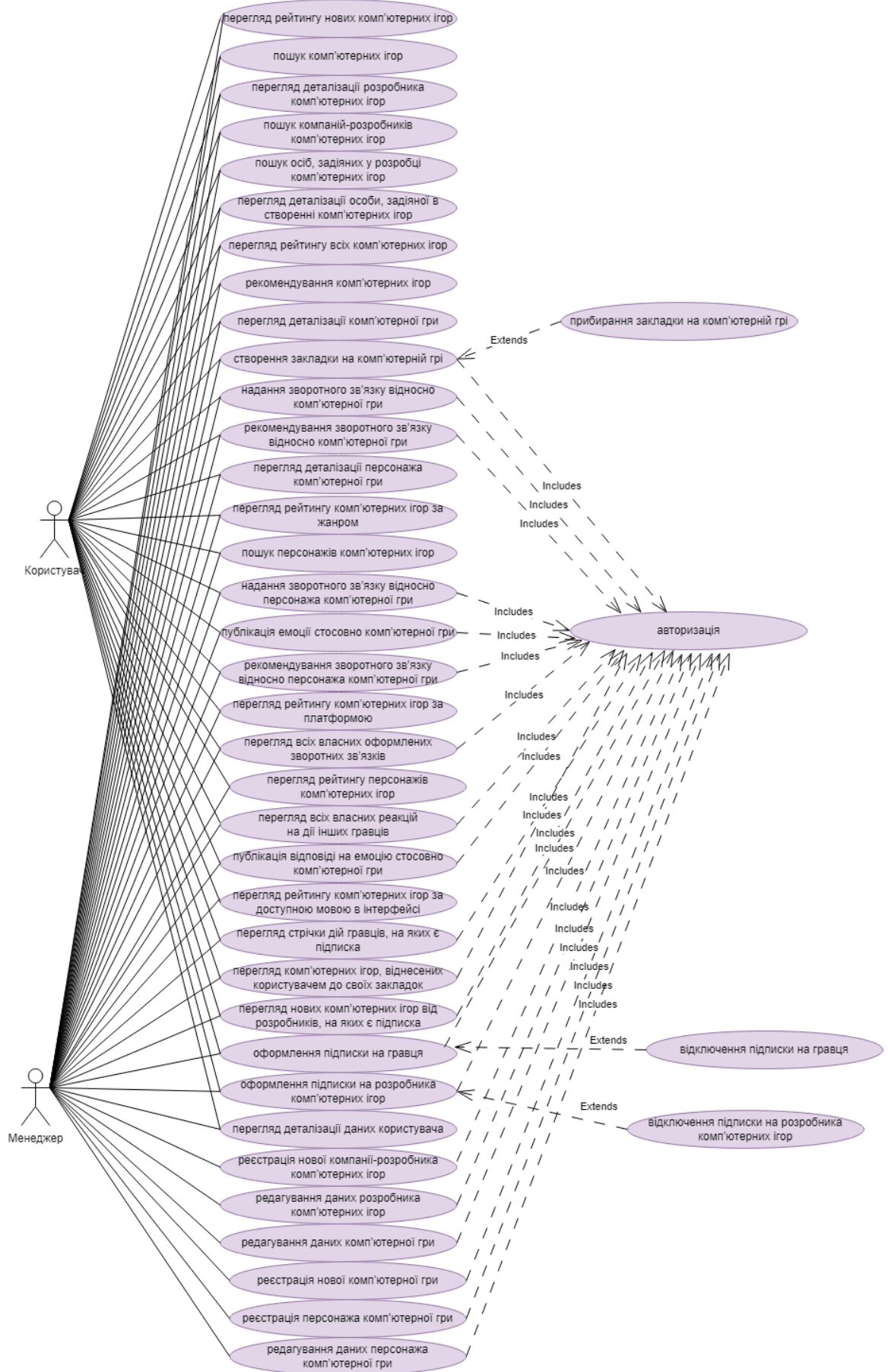


Рисунок 3.1 – Діаграма варіантів використання програми надання інформації про комп'ютерні ігри

3.2 Рішення щодо вибору засобів розробки

Мова Python має значні переваги стосовно саме завдань роботи з даними, реалізації зокрема моделей на основі ДР, тому її вибір став очевидним як основною мови створення проєкту, зокрема враховуючи розробку саме вебзастосунку, а Python є однією з мов, які дозволяють створювати такі застосунки доволі якісно. Проте фреймворки для Python розділяються на декілька груп, серед яких мікрофреймворки, до яких зокрема відноситься Bottle [8], і фреймворки для повноцінної розробки зокрема і масштабних вебзастосунків, серед яких виділяється Django [9] (табл. 3.1). Для реалізації та проєктування обрано Django, адже він має широкий набір вбудованих засобів (об'єктно-реляційне відображення, засоби захисту від атак, аутентифікація, авторизація, керування обліковими записами), що на відміну від Bottle дозволяє не використовувати додатковий час на знаходження, встановлення, налаштування відповідних можливостей, ускладнене ще й нижчим рівнем підтримки.

Таблиця 3.1 – Порівняння засобів для розробки програми надання інформації про комп'ютерні ігри

Критерій	Django	Bottle
Рівень фреймворку	Високий рівень	Мікрофреймворк
Масштаб проєктів, для яких призначений	Середній, великий	Малий
Рівень підтримки	Вищий	Нижчий
Вбудовані засоби	Об'єктно-реляційне відображення, засоби захисту від атак, аутентифікація, авторизація, керування обліковими записами	Перелічені засоби відсутні, але переважно можуть бути під'єднані за вибором

3.3 Визначення структури даних у базі даних

База даних (БД) у програмі виконує роль сховища, що зберігає дані комп'ютерних ігор, відгуків про ці комп'ютерні ігри.

Комп'ютерна гра в БД характеризується різною інформацією, однак базова інформація знаходиться в таблиці Computergame, до якої входять поля:

- id – первинний ключ, ідентифікатор комп'ютерної гри;
- title – поле назви комп'ютерної гри;
- description – поле презентаційного опису комп'ютерної гри, який представляє основну інформацію про гру, її ідею;
- storyline – поле опису сценарію комп'ютерної гри, який вже є деталізованим представленням;
- mainimage – поле основного зображення комп'ютерної гри, представляє собою вміст самого зображення;
- release – поле дати випуску комп'ютерної гри;
- rate – поле середньої оцінки комп'ютерної гри за виставленими оцінками з відгуків гравців;
- certificate – поле сертифікату щодо вікових обмежень для комп'ютерної гри;
- forparents – поле опису питань, які пов'язані з віковими обмеженнями щодо того, які користувачі можуть грати в цю гру;
- groupplay – поле з визначенням можливості грати в комп'ютерну гру разом з іншими користувачами;
- developer – зовнішній ключ до таблиці БД Developer, поле розробника, який створив (розробив) цю комп'ютерну гру і з яким вона асоціюється;
- gamepage – поле вебадреси, за якою заходиться опис цієї гри або вебсайт комп'ютерної гри.

Дані розробників комп'ютерних ігор знаходяться в таблиці Developer, до якої входять поля:

- id – первинний ключ, ідентифікатор розробника комп'ютерної гри;
- companypname – поле назви компанії, яка є розробником комп'ютерних ігор;
- webaddress – поле веб адреси, за якою можна отримати додаткову інформацію про компанію-розробника комп'ютерних ігор;
- aboutcompany – поле презентаційного опису компанії-розробника комп'ютерних ігор;
- photo – поле фотографії або зображення, яке презентує компанію-розробника комп'ютерних ігор, представляє собою вміст самого зображення;
- country – поле країни, в якій розташовано головний офіс компанії-розробника комп'ютерних ігор або з якою асоціюється компанія-розробник.

Окрім розробників, що створюють комп'ютерні ігри, ігри також характеризуються персонажами, які в них представлені, що дозволяє створювати певну спільноту щодо того, яка інформація про них відома, які враження вони викликають тощо.

Дані персонажів комп'ютерних ігор знаходяться в таблиці Computergamecharacter, до якої входять поля:

- id – первинний ключ, ідентифікатор персонажа комп'ютерної гри;
- name – поле імені персонажа комп'ютерної гри;
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, в якій цей персонаж з'являється;
- mainimage – поле основного зображення персонажа, представляє собою вміст самого зображення;
- description – поле презентаційного опису цього персонажу;
- storyline – поле з детальним описом персонажа комп'ютерної гри, його участі в сценарії гри;
- rate – поле середньої оцінки персонажа комп'ютерної гри за виставленими оцінками з відгуків гравців.

Зворотній зв'язок є сутністю предметної області, яка визначає загальну оцінку стосовно комп'ютерної гри або персонажа комп'ютерної гри, виставлену користувачем.

Безпосередньо зворотній зв'язок стосовно персонажа комп'ютерної гри визначений у таблиці Computergamecharacterfeedback, до якої входять поля:

- id – первинний ключ, ідентифікатор зворотного зв'язку стосовно персонажа комп'ютерної гри;
- computergamecharacter – зовнішній ключ до таблиці БД Computergamecharacter, поле персонажа комп'ютерної гри, відносно якого цей зворотній зв'язок створено;
- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, тобто користувача, який надав цей зворотній зв'язок стосовно персонажа комп'ютерної гри;
- feedback – текст зворотного зв'язку стосовно персонажа комп'ютерної гри;
- rate – оцінка, в яку своє загальне враження за цим персонажем комп'ютерної гри оцінив гравець;
- release – час створення цього зворотного зв'язку стосовно персонажа комп'ютерної гри.

Інші гравці можуть визначати, що зворотній зв'язок гравця стосовно персонажа комп'ютерної гри виявився для них корисним, створюючи рекомендації. Рекомендації зворотного зв'язку стосовно персонажів комп'ютерної гри знаходяться в таблиці Computergamecharacterfeedbackrecom, до якої входять поля:

- id – первинний ключ, ідентифікатор рекомендації зворотного зв'язку стосовно персонажа комп'ютерної гри;
- computergamecharacterfeedback – зовнішній ключ до таблиці БД Computergamecharacterfeedback, зворотній зв'язок стосовно персонажа комп'ютерної гри, до якого визначено цю рекомендацію;

– gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, який створив цю рекомендацію зворотного зв'язку стосовно персонажа комп'ютерної гри;

– rescom – поле типу рекомендації: рекомендації не обов'язково є позитивним враженням, вони можуть бути і позитивними, і негативними, і визначати неінформативність загалом зворотного зв'язку.

Зображення, скріншоти комп'ютерних ігор представлені в таблиці БД Computergameimage, до якої входять поля:

– id – ідентифікатор скріншота комп'ютерної гри;

– computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, яку показано на цьому скріншоті (зображенні);

– image – поле зображення комп'ютерної гри, представляє собою вміст самого зображення;

– detail – поле опису того, що на зображенні комп'ютерної гри показано.

Відеороліки з презентаціями комп'ютерних ігор представлені в таблиці БД Computergamevideo, до якої входять поля:

– id – ідентифікатор відеороліка комп'ютерної гри;

– computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, яку показано на цьому відеороліку;

– video – поле відеороліку, представляє собою посилання на файл відеороліка;

– detail – поле опису того, що на відеороліку цієї комп'ютерної гри показано.

Зворотній зв'язок стосовно комп'ютерної гри визначений у таблиці Computergamefeedback, до якої входять поля:

– id – первинний ключ, ідентифікатор зворотного зв'язку стосовно комп'ютерної гри;

– computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, відносно якої цей зворотній зв'язок створено;

- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, тобто користувача, який надав цей зворотній зв'язок стосовно комп'ютерної гри;
- feedback – текст зворотного зв'язку стосовно комп'ютерної гри;
- rate – оцінка, в яку своє загальне враження за цією комп'ютерною грою оцінив гравець;
- release – поле часу створення цього зворотного зв'язку стосовно комп'ютерної гри.

Рекомендації зворотного зв'язку стосовно комп'ютерної гри знаходяться в таблиці Computergamefeedbackrecom, до якої входять поля:

- id – первинний ключ, ідентифікатор рекомендації зворотного зв'язку стосовно комп'ютерної гри;
- computergamefeedback – зовнішній ключ до таблиці БД Computergamefeedback, зворотній зв'язок стосовно комп'ютерної гри, до якої визначено цю рекомендацію;
- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, який створив цю рекомендацію зворотного зв'язку стосовно комп'ютерної гри;
- recom – поле типу рекомендації: рекомендації не обов'язково є позитивним враженням, вони можуть бути і позитивними, і негативними, і визначати неінформативність загалом зворотного зв'язку.

Окрім зворотного зв'язку гравці можуть також ділитися емоціями стосовно комп'ютерних ігор. Емоції відрізняються від зворотного зв'язку тим, що вони не мають оцінки. Тобто емоціями може бути фактично будь-яка інформація стосовно комп'ютерної гри. Таким чином може бути визначено певний факт проходження гри, враження від гри, коментування якогось епізоду тощо.

Емоції гравця стосовно комп'ютерної гри знаходяться в таблиці Computergameemotion, до якої входять поля:

- id – первинний ключ, ідентифікатор емоції гравця стосовно комп'ютерної гри;

- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, відносно якої цю емоцію створено;

- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, який створив цю емоцію зворотного зв'язку стосовно комп'ютерної гри;

- emotion – поле текстового опису емоції (або фактично інформації стосовно загалом комп'ютерної гри або участі конкретного гравця у комп'ютерній грі);

- release – поле часу створення цієї емоції стосовно комп'ютерної гри.

На емоції інші гравці можуть відповідати. Це є їхньою відмінністю також від зворотного зв'язку. Тобто на відміну від зворотного зв'язку тут можна висловити не тільки рекомендації, але і висловити фактично свою емоцію на емоцію.

Відповіді на емоції від комп'ютерної гри знаходяться в таблиці Computergameemotionanswer, до якої входять поля:

- id – первинний ключ, ідентифікатор відповіді на емоцію гравця стосовно комп'ютерної гри;

- computergameemotion – зовнішній ключ до таблиці БД Computergameemotion, поле емоції гравця стосовно комп'ютерної гри, відповідь на яку створено;

- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, який створив цю відповідь;

- answer – текст відповіді на емоцію гравця стосовно комп'ютерної гри;

- release – поле часу створення цієї відповіді на емоцію стосовно комп'ютерної гри.

Оскільки в інтерфейсах комп'ютерних ігор можуть бути різні мови, то важливо зазначати щодо комп'ютерної гри, які саме мови в ній використовуються. Загальний перелік мов визначається таблицею Language, до якої входять поля:

- id – ідентифікатор мови (природної мови);

– name – назва природної мови.

Безпосередньо мови інтерфейсу в комп'ютерній грі знаходяться в таблиці БД Computergamelanguage, до якої входять поля:

- id – первинний ключ, ідентифікатор мови інтерфейсу комп'ютерної гри;
- interface – поле типу інтерфейсу, для якого ця мова забезпечується (інтерфейс, аудіо озвучення, субтитри);
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, в якій ця мова доступна;
- language – зовнішній ключ до таблиці БД Language, поле мови, яка доступна в цьому інтерфейсі.

Комп'ютерні ігри зазвичай класифікуються за жанрами, тому перелік жанрів було виділено в таблицю Genre, до якої входять поля:

- id – первинний ключ, ідентифікатор можливого жанру комп'ютерної гри;
- name – поле назви можливого жанру комп'ютерної гри.

Безпосередньо жанри вже конкретних комп'ютерних ігор знаходяться в таблиці Computergamegenre, до якої входять поля:

- id – первинний ключ, ідентифікатор жанру конкретної комп'ютерної гри;
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, для якої характерний цей жанр;
- genre – зовнішній ключ до таблиці БД Genre, поле жанру комп'ютерної гри, яким вона характеризується.

Комп'ютерні ігри, представлені у форматі відеоігор, на даний момент нерідко мають склад учасників, який фактично нагадує склад учасників фільмів. Це режисер, автор сценарію, різні інші учасники (фактично акторів). Усі ці ролі представляють так звану особу – базову в цьому сенсі сутність БД, яку винесено в таблицю Person, до якої входять поля:

- id – первинний ключ, ідентифікатор особи, яка може відігравати різні ролі при створенні комп'ютерних ігор;

- name – поле імені особи, яка може відігравати різні ролі при створенні комп'ютерних ігор;

- photo – поле фотографії особи, яка може відігравати різні ролі при створенні комп'ютерних ігор.

Інформація про режисерів комп'ютерних ігор представлена в таблиці Director, до якої входять поля:

- id – первинний ключ, ідентифікатор режисера комп'ютерної гри;

- person – зовнішній ключ до таблиці БД Person, поле особи, яка в цій комп'ютерній грі відіграє роль режисера;

- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, в якій ця особа була режисером;

- description – поле опису особливостей участі цієї особи як режисера при створенні цієї комп'ютерної гри.

Інформація про авторів сценаріїв комп'ютерних ігор представлена в таблиці Writer, до якої входять поля:

- id – первинний ключ, ідентифікатор автора сценарію комп'ютерної гри;

- person – зовнішній ключ до таблиці БД Person, поле особи, яка в цій комп'ютерній грі відіграє роль автора сценарію;

- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, в якій ця особа була автором сценарію;

- description – поле опису особливостей участі цієї особи як автора сценарію при створенні цієї комп'ютерної гри.

Інформація про інших учасників створення комп'ютерних ігор (акторів, помічників) представлена в таблиці Participant, до якої входять поля:

- id – первинний ключ, ідентифікатор учасника створення комп'ютерної гри;

- person – зовнішній ключ до таблиці БД Person, поле особи, яка в цій комп'ютерній грі відіграє певну роль учасника (актора, помічника тощо);
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, в якій ця особа була учасником створення;
- role – поле з описом конкретної ролі, що описує участь цієї особи в створенні цієї комп'ютерної гри;
- description – поле опису особливостей участі цієї особи при створенні цієї комп'ютерної гри.

Закладки, які використовуються для збереження комп'ютерних ігор для зручного повернення до них, знаходяться в таблиці Gamebookmark, до якої входять поля:

- id – первинний ключ, ідентифікатор закладки гравця на комп'ютерній грі;
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, на якій створено закладку;
- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, який створив цю закладку на комп'ютерній грі.

Інформація про операційні системи (платформи), на яких можна грати в комп'ютерну гру, є важливою, адже дозволяє кожному гравцю зорієнтуватися в новій грі та визначити, чи підтримується в ній ця платформа.

Перелік платформ, на яких в принципі можуть запускатися комп'ютерні ігри на даний момент, знаходиться в таблиці БД OSplatform, до якої входять поля:

- id – первинний ключ, ідентифікатор платформи, на якій в принципі може запускатися комп'ютерна гра;
- name – назва платформи, на якій в принципі може запускатися комп'ютерна гра.

Платформи, на яких можуть запускатися конкретні комп'ютерні ігри, знаходяться в таблиці Gameplatform, до якої входять поля:

- id – первинний ключ, ідентифікатор платформи, на якій може запускатися певна комп'ютерна гра;
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, яка на цій платформі може запускатися;
- osplatform – зовнішній ключ до таблиці БД OSplatform, поле платформи, на якій може запускатися ця комп'ютерна гра;
- minversion – поле мінімальної версії платформи, на якій ця комп'ютерна гра може запускатися.

Вимоги до комп'ютерної гри дозволяють зрозуміти гравцю, чи зможе він на своєму конкретному пристрої запустити таку гру. Вимоги визначаються спершу самими формальними позиціями, наприклад, обсягом оперативної пам'яті, яка необхідна.

Перелік цих формальних вимог, які можуть визначатися вже далі для самої комп'ютерної гри, знаходиться в таблиці Platformrequirement, до якої входять поля:

- id – первинний ключ, ідентифікатор можливої вимоги до комп'ютерної гри;
- name – назва можливої вимоги (тобто сама вимога) до комп'ютерної гри.

Конкретні вимоги до конкретної гри визначаються таблицею Gameplatformrequirement, до якої входять поля:

- id – первинний ключ, ідентифікатор конкретної вимоги до комп'ютерної гри;
- platformrequirement – зовнішній ключ до таблиці БД Platformrequirement, поле можливої вимоги до комп'ютерної гри, за якою конкретизовано значення;
- gameplatform – зовнішній ключ до таблиці Gameplatform, поле платформи, на якій може запускатися певна комп'ютерна гра і для якої визначено дане значення вимоги;

- minvalue – поле мінімального значення, яке необхідно забезпечити для того, щоб грати в цю комп'ютерну гру;

- recomvalue – поле рекомендованого значення, яке необхідно забезпечити для того, щоб грати в цю комп'ютерну гру.

Визначення підписки на комп'ютерні ігри розробника виконується через таблицю БД Developerread, до якої входять поля:

- id – первинний ключ, ідентифікатор підписки на комп'ютерні ігри розробника;

- developer – зовнішній ключ до таблиці Developer, поле розробника, на комп'ютерні ігри якого оформлено підписку;

- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, який визначив необхідною для себе цю підписку.

Гравці у комп'ютерні ігри є фактично користувачами програми. До них віднесено і адміністраторський обліковий запис.

Таблиця гравців Gameuser має поля:

- id – первинний ключ, ідентифікатор гравця;

- username – поле логіну гравця, що він використовує для доступу до програми;

- password – поле паролю гравця, що він використовує для доступу до програми;

- active – поле активності облікового запису (можна авторизуватись за ним чи ні);

- staff – поле участі гравця в програмі як менеджера (або відсутності такого доступу, якщо це звичайний гравець);

- photo – поле фотографії користувача або певного аватару;

- description – поле презентаційного опису гравця.

Типи пристроїв, для яких доступні версії комп'ютерної гри, знаходяться в таблиці Devicetype, до якої входять поля:

- id – первинний ключ, ідентифікатор типу пристроїв;

- name – назва типу пристроїв (рис. 3.2).

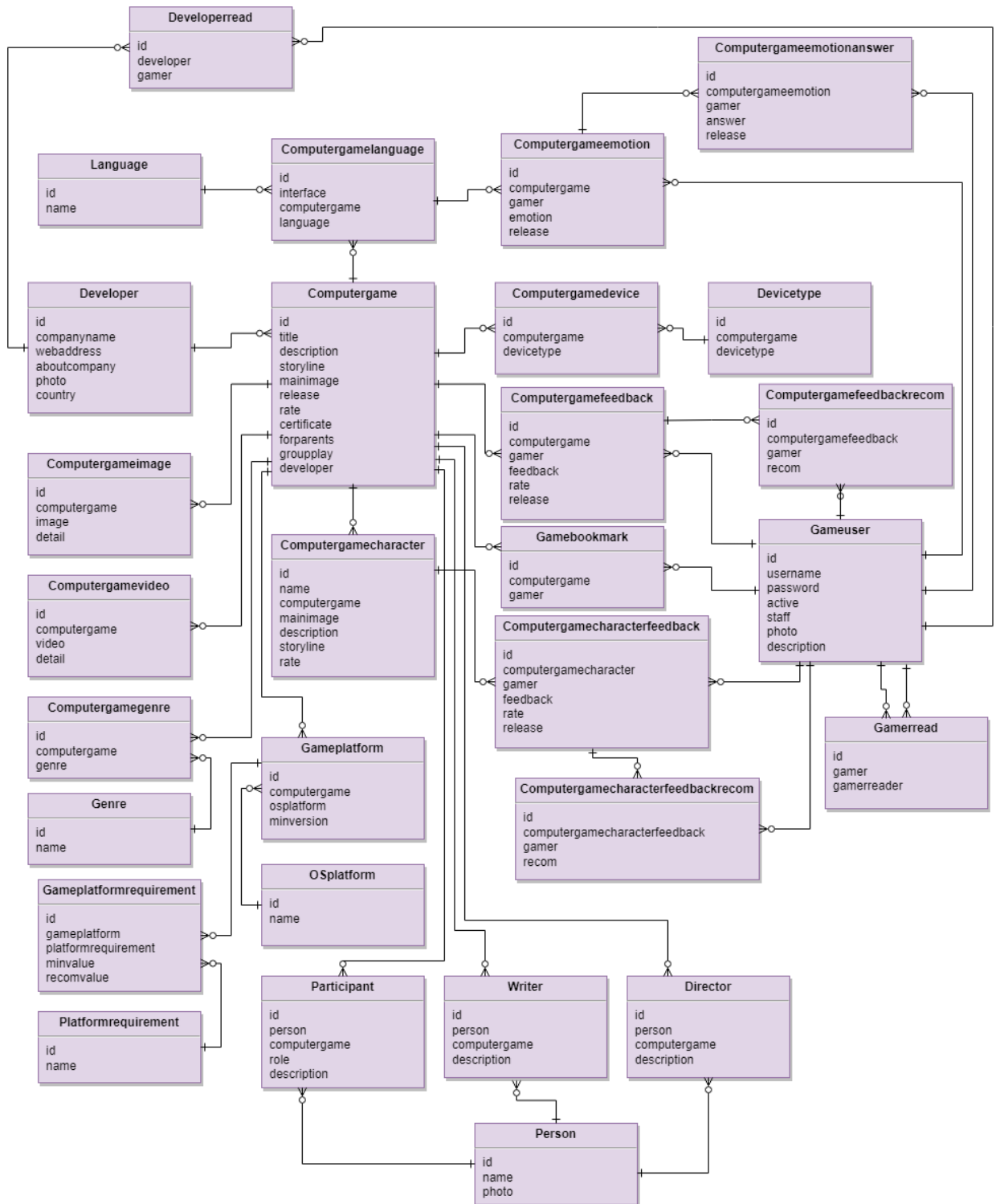


Рисунок 3.2 – Схема БД

Визначення підписки на гравців виконується через таблицю БД Gamerread, до якої входять поля:

– id – первинний ключ, ідентифікатор підписки на гравця;

- gamerreader – зовнішній ключ до таблиці БД Gamer, поле гравця, який підписався на іншого гравця;
- gamer – зовнішній ключ до таблиці БД Gameuser, поле гравця, на якого було створено підписку.

Конкретні типи пристроїв, на яких доступна можливість грати в певну комп'ютерну гру, знаходяться в таблиці Computergamedevice, до якої входять поля:

- id – первинний ключ, ідентифікатор запису про конкретний тип пристрою, на якому можна запустити цю комп'ютерну гру;
- computergame – зовнішній ключ до таблиці БД Computergame, поле комп'ютерної гри, яка може запускатися на цьому типі пристрою;
- devicetype – зовнішній ключ до таблиці БД Devicetype, поле типу пристрою, на якому може запускатися ця комп'ютерна гра.

БД організовано на основі системи керування MySQL.

3.4 Висновки за розділом 3

У розділі 3 виконано детальне моделювання сценаріїв роботи користувачів, якими можуть бути звичайні користувачі або менеджери, з програмою, забезпечено вибір засобів розробки, включаючи мову Python, фреймворку Django, систему MySQL, виконано проєктування і деталізований опис структури роботи з даними у БД.

4 РЕАЛІЗАЦІЯ ПРОГРАМИ

4.1 Визначення послідовності роботи з програмою

Базовою точкою входу в програму є відображення рейтингу всіх комп'ютерних ігор (рис. 4.1). Оскільки всі подальші дії залежать від меню, що надається користувачу, то далі відбувається формування меню за рівнем доступу. Далі з меню можна обрати один з 3 основних розділів, що охоплюють роботу з засобами стосовно комп'ютерних ігор, розробників або комунікації. Після виконання авторизації права доступу можуть змінитися, тому знов відбувається формування меню за рівнем доступу.

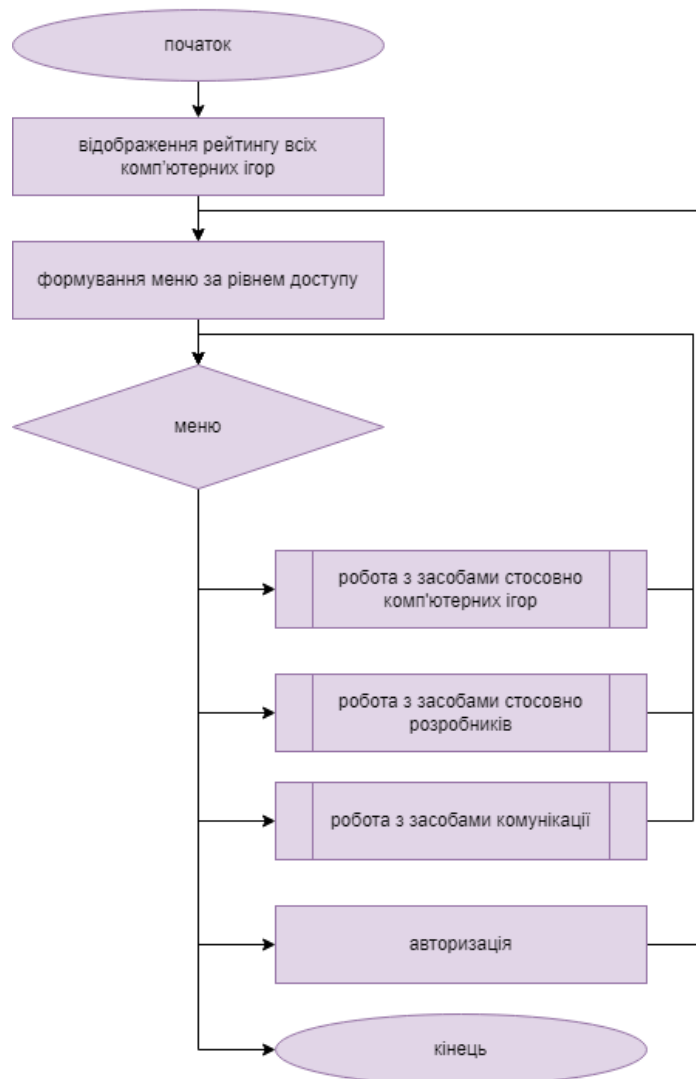


Рисунок 4.1 – Схема функціонування програми надання інформації про комп'ютерні ігри

4.2 Особливості реалізації програми

Підтримку перегляду деталізації розробника комп'ютерних ігор реалізовано через функцію `developer_games_view`, що окрім параметра `request` має ще параметр `developer_id`, який відповідає ідентифікатору розробника комп'ютерних ігор. Результатом роботи функції є повернутий шаблон вебсторінки `developer.html`, який побудовано на основі безкоштовного шаблону [10] і який заповнено даними про нові комп'ютерні ігри (опубліковані не раніше ніж за місяць до поточної дати), які випущені цим розробником, через ключ `newgames`, найкращі ігри через ключ `bestgames` та дані про самого розробника через ключ `developer`. За іграми повертаються тільки 5 з них, а також за кожною комп'ютерною грою перевіряється її внесення до закладок, а підписка на розробника перевіряється окремо і вноситься в дані розробника (наявна чи ні).

Підтримку перегляду рейтингу всіх комп'ютерних ігор реалізовано через функцію `best_computer_games_view`. Дана функція наповнює шаблон вебсторінки `bestoverall.html` даними про 500 найкращих комп'ютерних ігор за середньою оцінкою, також визначаючи, чи внесена кожна комп'ютерна гра в закладки у авторизованого користувача (якщо він авторизований).

Підтримку перегляду рейтингу комп'ютерних ігор за жанром реалізовано через функцію `best_genre_computer_games_view`. Функція окрім параметра `request` має також параметр `genre_id`, що є ідентифікатором жанру комп'ютерних ігор. Дана функція наповнює шаблон вебсторінки `bestgenre.html` даними про 500 найкращих комп'ютерних ігор, які мають заданий жанр, за середньою оцінкою, також визначаючи, чи внесена кожна комп'ютерна гра в закладки у авторизованого користувача.

Підтримку перегляду рейтингу комп'ютерних ігор за доступною мовою в інтерфейсі реалізовано через функцію `best_language_computer_games_view`. Функція окрім параметра `request` має також параметр `language_id`, що є ідентифікатором мови, за якою маж бути

доступний інтерфейс, аудіо супроводження або субтитри в грі. Дана функція наповнює шаблон вебсторінки `bestlang.html` даними про 500 найкращих комп'ютерних ігор, які відповідають умові наявності засобів цієї мовою, за середньою оцінкою, також визначаючи, чи внесена кожна комп'ютерна гра в закладки у авторизованого користувача.

Підтримку перегляду рейтингу комп'ютерних ігор за платформою реалізовано через функцію `best_platform_computer_games_view`. Функція окрім параметра `request` має також параметр `osplatform_id`, що є ідентифікатором платформи, під якою мають працювати комп'ютерні ігри з рейтингу. Дана функція наповнює шаблон вебсторінки `bestlang.html` даними про 500 найкращих комп'ютерних ігор, які відповідають умові можливості роботи з заданою платформою, за середньою оцінкою, також визначаючи, чи внесена кожна комп'ютерна гра в закладки у авторизованого користувача.

Підтримку перегляду рейтингу нових комп'ютерних ігор реалізовано через функцію `best_new_computer_games_view`. Дана функція наповнює шаблон вебсторінки `bestnew.html` даними про 500 найкращих комп'ютерних ігор за середньою оцінкою, які були опубліковані протягом місяця з поточної дати, також визначаючи, чи внесена кожна комп'ютерна гра в закладки у авторизованого користувача.

Підтримку перегляду рейтингу персонажів комп'ютерних ігор реалізовано через функцію `best_computer_game_characters_view`. Дана функція наповнює шаблон вебсторінки `bestcharacters.html` даними про 500 найкращих персонажів комп'ютерних ігор за середньою оцінкою, також визначаючи, чи внесена комп'ютерна гра, до якої належить цей персонаж, в закладки у авторизованого користувача.

Підтримку перегляду всіх власних оформлених зворотних зв'язків реалізовано через функцію `own_gamer_feedbacks_view`. Визначаються всі записи про зворотні зв'язки користувача, який авторизувався, стосовно як комп'ютерних ігор, так і персонажів комп'ютерних ігор. За кожним

зворотнім зв'язком обчислюється кількість рекомендацій кожного типу за цим записом.

Підтримку перегляду всіх власних реакцій на дії інших гравців реалізовано через функцію `own_recommendations_view`. До результатів у шаблон вебсторінки вносяться рекомендації авторизованого користувача, надані зворотним зв'язком на комп'ютерні ігри, за ключем `games`, рекомендації авторизованого користувача, надані зворотнім зв'язком стосовно персонажів комп'ютерних ігор, за ключем `characters`, відповіді на емоції стосовно комп'ютерних ігор, надані за ключем `emotions`. Відповідно за кожною рекомендацією визначається також кількість рекомендацій, яку сумарно за кожним типом рекомендацій надали користувачі даному зворотному зв'язку, а також відповідно кількість відповідей на емоції користувача щодо комп'ютерної гри.

Підтримку перегляду комп'ютерних ігор, віднесених користувачем до своїх закладок реалізовано через функцію `bookmark_games_view`. Функція працює тільки для авторизованих користувачів, інакше шаблон вебсторінки `bookmarks.html` повернеться порожнім.

Підтримку перегляду нових комп'ютерних ігор від розробників, на яких є підписка, реалізовано через функцію `read_developer_games_view`. Шаблон вебсторінки `readdeveloper.html` заповнено даними комп'ютерних ігор, які були випущені за місяць до поточного моменту, впорядковуючи ігри за датою публікації.

Підтримку перегляду стрічки дій гравців, на яких є підписка у авторизованого користувача, реалізовано через функцію `read_gamers_view`. Відбираються всі користувачі, на які підписався авторизований користувач, а тоді виділяються всі емоції, зворотні зв'язки відносно комп'ютерних ігор, персонажів комп'ютерних ігор, створені користувачами, які були відібрані. Дані передаються в шаблон вебсторінки `readgamers.html`.

Підтримку перегляду деталізації особи, задіяної в створенні комп'ютерних ігор реалізовано через функцію `person_view`, в якій є параметр

person_id, що є ідентифікатором такої особи. Тоді окрім даних особи виділяються комп'ютерні ігри, в створенні яких особа прийняла участь як режисер, як автор сценарію, як учасник створення в різних інших ролях. Також виділяються 5 найновіших ігор, в створенні яких взяла участь ця особа. Всі дані передаються в шаблон вебсторінки person.html.

Підтримку перегляду деталізації персонажа комп'ютерної гри реалізовано через функцію game_character_view, яка заповнює шаблон вебсторінки character.html даними стосовно самого персонажа та його комп'ютерної гри через ключ game_character, зворотніми зв'язками, наданими стосовно цього персонажа комп'ютерної гри, через ключ feedbacks.

Підтримку перегляду деталізації комп'ютерної гри реалізовано через функцію game_view, в якій параметром game_id задається ідентифікатор комп'ютерної гри. За цією комп'ютерною грою в шаблон вебсторінки game.html передаються за ключем game дані самої комп'ютерної гри, включаючи дані розробника, типів пристроїв, на яких підтримується гра, жанрів цієї комп'ютерної гри, за ключем characters передаються персонажі цієї комп'ютерної гри, за ключем images – фотографії, за ключем videos – відеороліки, за ключем feedbacks – зворотні зв'язки відносно цієї комп'ютерної гри, за ключем emotions – емоції відносно цієї комп'ютерної гри, за ключем languages – мови, які доступні за цією комп'ютерною грою, за ключем directors – режисери комп'ютерної гри, за ключем writers – автори сценарію, participants – інші учасники створення комп'ютерної гри, за ключем platforms – платформи, на яких підтримується комп'ютерна гра, за ключем requirements – вимоги, які мають бути забезпечені стосовно комп'ютерної гри. Також за всіма зворотними зв'язками, емоціями обчислюється відповідно кількість рекомендацій або відповідей.

Підтримку оформлення підписки на гравця та підтримку відключення підписки на гравця реалізовано через функцію read_gamer_view, що має параметр gamer_username з логіном гравця, на якого потрібно оформити підписку. При цьому перевіряється, чи користувач, який авторизувався, має

підписку на цього користувача. Якщо так, то підписка вилучається, а якщо ні, то підписка створюється. Це дозволило опрацювати одразу два сценарії роботи.

Підтримку оформлення підписки на розробника комп'ютерних ігор та підтримку відключення підписки на розробника комп'ютерних ігор реалізовано через функцію `read_developer_view`, що має параметр `developer_id`, який визначає ідентифікатор розробника. При цьому перевіряється, чи користувач, який авторизувався, має підписку на цього розробника. Якщо так, то підписка вилучається, а якщо ні, то підписка створюється. Це дозволило опрацювати одразу два сценарії роботи.

Підтримку створення закладки на комп'ютерній грі та підтримку прибирання закладки з комп'ютерної гри реалізовано через функцію `bookmark_game_view`, яка має параметр `game_id`, що є ідентифікатором комп'ютерної гри. При цьому перевіряється, чи користувач, який авторизувався, створював вже закладку на цій комп'ютерній грі. Якщо так, то закладка вилучається, а якщо ні, то закладка створюється. Це дозволило опрацювати одразу два сценарії роботи.

Підтримку рекомендування зворотного зв'язку відносно комп'ютерної гри реалізовано через функцію `game_feedback_recom_view`, яка має параметри `gamefeed_id`, `recomtype`, перший з яких визначає номер зворотного зв'язку відносно комп'ютерної гри, який необхідно рекомендувати, а другий – тип рекомендації, дані власника беруться з авторизації.

Підтримку рекомендування зворотного зв'язку відносно персонажа комп'ютерної гри реалізовано через функцію `character_feedback_recom_view`, яка має параметри `characterfeed_id`, `recomtype`, перший з яких визначає номер зворотного зв'язку відносно персонажа комп'ютерної гри, який необхідно рекомендувати, а другий – тип рекомендації, дані власника беруться з авторизації.

Підтримку пошуку комп'ютерних ігор реалізовано через функцію `computer_games_search_view`, визначаючи параметри пошуку з запиту до

сторінки, виділяючи з них розробника (developer), назву (title), рік, в який найраніше мала бути виконана публікація гри (releaseyear1), рік, в який найпізніше мала бути виконана публікація гри (releaseyear2), країни компанії-розробника, сертифікат щодо перегляду або участі дітей (certificate), найнижчий допустимий середній бал (rate1), найвищий допустимий середній бал (rate2) комп'ютерної гри, можливість гри командно (group), підтримуваний тип пристрою (device), ім'я персонажа гри (character), мінімальна кількість зворотних зв'язків (back1), максимальна кількість зворотних зв'язків (back2), мінімальна кількість емоцій (emo1), максимальна кількість емоцій (emo2), мова, яка має підтримуватися в комп'ютерній грі (lang), жанри комп'ютерної гри (genres) – комбінація з нулів та одиниць за кожним жанром, ім'я особи, яка мала приймати участь у створенні комп'ютерної гри (person), підтримувана платформа (platform). Усі результати відбираються та заповнюють шаблон вебсторінки `gamesearch.html`.

Підтримку рекомендації комп'ютерних ігор реалізовано через функцію `recommended_computer_game_view`, яка відбирає комп'ютерні ігри за методом рекомендації комп'ютерних ігор, який запропоновано, дозволяючи користувачу обрати жанр комп'ютерних ігор, і які він бажає грати. Жанр задається ідентифікатором через параметр `genre_id`. Усі результати відбираються та заповнюють шаблон вебсторінки `gamerecommend.html`.

Підтримку пошуку компаній-розробників комп'ютерних ігор реалізовано через функцію `developer_serach_view`, параметри пошуку в яку задаються через параметри запиту, тобто через базовий параметр `request`, присутній у всіх функціях. До параметрів пошуку належать назва розробника (name), країни, в якій знаходиться розробник (country), кількість комп'ютерних ігор, створених розробником (number), найменший допустимий середній бал за всіма іграми (rate). Усі результати відбираються та заповнюють шаблон вебсторінки `developersearch.html`.

Підтримку пошуку осіб, задіяних у розробці комп'ютерних ігор? реалізовано через функцію `person_search_view`, в яку параметри пошуку передаються через `request` з виділенням самих цих значень, включаючи ім'я особи (`name`), роль, яку під час створення комп'ютерних ігор хоча би раз мала виконувати ця особа (`role`), назва комп'ютерної гри, в створенні якої приймала участь особа (`game`). Усі результати відбираються та заповнюють шаблон вебсторінки `personsearch.html`.

Підтримку пошуку персонажів комп'ютерних ігор реалізовано через функцію `character_search_view`, в яку параметри пошуку передаються через `request` з виділенням самих цих значень, включаючи ім'я персонажа (`name`), назву комп'ютерної гри (`game`), до якої належить персонаж), найнижчий допустимий середній бал персонажа (`rate1`), найвищий допустимий середній бал (`rate2`) персонажа. Усі результати відбираються та заповнюють шаблон вебсторінки `charactersearch.html`.

Підтримку надання зворотного зв'язку відносно комп'ютерної гри реалізовано через функцію `game_feedback_add_view`, безпосередня робота якої підтримується формою класу `ComputergamefeedbackForm` на основі шаблону вебсторінки `addfeedback.html`. Ідентифікатор комп'ютерної гри задається параметром `game_id`.

Підтримку надання зворотного зв'язку відносно персонажа комп'ютерної гри реалізовано через функцію `character_feedback_add_view`, безпосередня робота якої підтримується формою класу `ComputergamecharacterfeedbackForm` на основі шаблону вебсторінки `addcharacterfeedback.html`. Ідентифікатор персонажа комп'ютерної гри задається параметром `character_id`.

Підтримку публікації емоції стосовно комп'ютерної гри реалізовано через функцію `emotion_add_view`, безпосередня робота якої підтримується формою класу `EmotionForm` на основі шаблону вебсторінки `addemotion.html`. Ідентифікатор комп'ютерної гри задається параметром `game_id`.

Підтримку публікації відповіді на емоцію стосовно комп'ютерної гри реалізовано через функцію `emotion_answer_add_view`, безпосередня робота якої підтримується формою класу `EmotionanswerForm` на основі шаблону вебсторінки `addemotionanswer.html`. Ідентифікатор емоції, на яку створюється відповідь, задається параметром `emotion_id`.

Підтримку перегляду деталізації даних користувача реалізовано через функцію `gamer_view`. Усі дані передаються в шаблон вебсторінки `gamer.html`, включаючи зворотні зв'язки користувача відносно комп'ютерних ігор через ключ `game_feedbacks`, зворотні зв'язки користувача відносно персонажів комп'ютерних ігор через ключ `character_feedbacks`, дані про користувача через ключ `user`. Логін користувача, для якого ці дані збираються, задається параметром `gamer_username`.

Підтримку реєстрація нової компанії-розробника комп'ютерних ігор реалізовано через функцію `developer_add_view`, безпосередня робота якої підтримується формою класу `DeveloperForm` на основі шаблону вебсторінки `adddeveloper.html`. Використовувати цю функцією можуть тільки авторизовані користувачі, які є менеджерами.

Підтримку редагування даних розробника комп'ютерних ігор реалізовано через функцію `developer_edit_view`, безпосередня робота якої підтримується формою класу `DeveloperForm` на основі шаблону вебсторінки `editdeveloper.html`. Використовувати цю функцією можуть тільки авторизовані користувачі, які є менеджерами.

Підтримку редагування даних комп'ютерної гри реалізовано через функцію `game_edit_view`, безпосередня робота якої підтримується формою класу `ComputergameForm` на основі шаблону вебсторінки `editgame.html`. Використовувати цю функцією можуть тільки авторизовані користувачі, які є менеджерами.

Підтримку реєстрації нової комп'ютерної гри реалізовано через функцію `game_add_view`, безпосередня робота якої підтримується формою класу `ComputergameForm` на основі шаблону вебсторінки `addgame.html`.

Використовувати цю функцією можуть тільки авторизовані користувачі, які є менеджерами.

Підтримку реєстрації персонажа комп'ютерної гри реалізовано через функцію `character_add_view`, безпосередня робота якої підтримується формою класу `CharacterForm` на основі шаблону вебсторінки `addcharacter.html`. Використовувати цю функцією можуть тільки авторизовані користувачі, які є менеджерами.

Підтримку редагування даних персонажа комп'ютерної гри реалізовано через функцію `character_edit_view`, безпосередня робота якої підтримується формою класу `CharacterForm` на основі шаблону вебсторінки `editcharacter.html`. Використовувати цю функцією можуть тільки авторизовані користувачі, які є менеджерами.

Модель комп'ютерної гри представлена класом `Computergame` з атрибутами:

- поле `title` з типом `CharField` – поле назви комп'ютерної гри;
- поле `description` з типом `CharField` – поле презентаційного опису комп'ютерної гри, який представляє основну інформацію про гру, її ідею;
- поле `storyline` з типом `CharField` – поле опису сценарію комп'ютерної гри, який вже є деталізованим представленням;
- поле `mainimage` з типом `ImageField` – поле основного зображення комп'ютерної гри, представляє собою вміст самого зображення;
- поле `release` з типом `DateField` – поле дати випуску комп'ютерної гри;
- поле `rate` з типом `FloatField` – поле середньої оцінки комп'ютерної гри за виставленими оцінками з відгуків гравців;
- поле `certificate` з типом `CharField` – поле сертифікату щодо вікових обмежень для комп'ютерної гри;
- поле `forparents` з типом `CharField` – поле опису питань, які пов'язані з віковими обмеженнями щодо того, які користувачі можуть грати в цю гру;

- поле `groupplay` з типом `BooleanField` – поле з визначенням можливості грати в комп'ютерну гру разом з іншими користувачами;
- поле `developer` з типом `ForeignKey` – зовнішній ключ до моделі `Developer`, поле розробника, який створив (розробив) цю комп'ютерну гру і з яким вона асоціюється;
- поле `gamepage` з типом `URLField` – поле веб адреси, за якою заходиться опис цієї гри або вебсайт комп'ютерної гри;
- поле `genres` з типом `ManyToManyField` – поле жанрів комп'ютерної гри, що фактично реалізує зв'язуючу таблицю в реляційній структурі, яка була описана для БД в розділі 3;
- поле `devicetypes` з типом `ManyToManyField` – поле типів пристроїв, на яких можна грати в цю комп'ютерну гру, що фактично реалізує зв'язуючу таблицю в реляційній структурі.

Модель розробників комп'ютерних ігор представлена класом `Developer` з атрибутами:

- `companyname` з типом `CharField` – поле назви компанії, яка є розробником комп'ютерних ігор;
- `webaddress` з типом `URLField` – поле веб адреси, за якою можна отримати додаткову інформацію про компанію-розробника комп'ютерних ігор;
- `aboutcompany` з типом `CharField` – поле презентаційного опису компанії-розробника комп'ютерних ігор;
- `photo` з типом `ImageField` – поле фотографії або зображення, яке презентує компанію-розробника комп'ютерних ігор, представляє собою вміст самого зображення;
- `country` з типом `CharField` – поле країни, в якій розташовано головний офіс компанії-розробника комп'ютерних ігор або з якою асоціюється компанія-розробник.

Модель персонажів комп'ютерних ігор представлена класом `Computergamecharacter` з атрибутами:

- name з типом CharField – поле імені персонажа комп'ютерної гри;
- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, в якій цей персонаж з'являється;
- mainimage з типом ImageField – поле основного зображення персонажа, представляє собою вміст самого зображення;
- description з типом CharField – поле презентаційного опису цього персонажу;
- storyline з типом CharField – поле з детальним описом персонажа комп'ютерної гри, його участі в сценарії гри;
- rate з типом FloatField – поле середньої оцінки персонажа комп'ютерної гри за виставленими оцінками з відгуків гравців.

Модель зворотного зв'язку стосовно персонажа комп'ютерної гри представлена класом Computergamecharacterfeedback з атрибутами:

- computergamecharacter з типом ForeignKey – зовнішній ключ до моделі Computergamecharacter, поле персонажа комп'ютерної гри, відносно якого цей зворотній зв'язок створено;
- gamer з типом ForeignKey – зовнішній ключ до моделі Gameuser, поле гравця, тобто користувача, який надав цей зворотній зв'язок стосовно персонажа комп'ютерної гри;
- feedback з типом CharField – текст зворотного зв'язку стосовно персонажа комп'ютерної гри;
- rate з типом SmallIntegerField – оцінка, в яку своє загальне враження за цим персонажем комп'ютерної гри оцінив гравець;
- release з типом DateTimeField – час створення цього зворотного зв'язку стосовно персонажа комп'ютерної гри.

Модель рекомендації зворотного зв'язку стосовно персонажів комп'ютерної гри представлена класом Computergamecharacterfeedbackrecom з атрибутами:

- computergamecharacterfeedback з типом ForeignKey – зовнішній ключ до моделі Computergamecharacterfeedback, зворотній зв'язок стосовно персонажа комп'ютерної гри, до якого визначено цю рекомендацію;

- gamer з типом ForeignKey – зовнішній ключ до моделі Gameuser, поле гравця, який створив цю рекомендацію зворотного зв'язку стосовно персонажа комп'ютерної гри;

- rescom з типом CharField – поле типу рекомендації: рекомендації не обов'язково є позитивним враженням, вони можуть бути і позитивними, і негативними, і визначати неінформативність загалом зворотного зв'язку.

Модель зображень, скріншотів комп'ютерних ігор представлена класом Computergameimage з атрибутами:

- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, яку показано на цьому скріншоті (зображенні);

- image з типом ImageField – поле зображення комп'ютерної гри, представляє собою вміст самого зображення;

- detail з типом CharField – поле опису того, що на зображенні комп'ютерної гри показано.

Модель відеороликів з презентаціями комп'ютерних ігор представлена класом Computergamevideo з атрибутами:

- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, яку показано на цьому відеороліку;

- video з типом URLField – поле відеороліку, представляє собою посилання на файл відеороліка;

- detail з типом CharField – поле опису того, що на відеороліку цієї комп'ютерної гри показано.

Модель зворотного зв'язку стосовно комп'ютерної гри визначений у таблиці Computergamefeedback з атрибутами:

- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, відносно якої цей зворотній зв'язок створено;

- gamer з типом ForeignKey – зовнішній ключ до моделі Gameuser, поле гравця, тобто користувача, який надав цей зворотній зв'язок стосовно комп'ютерної гри;

- feedback з типом CharField – текст зворотного зв'язку стосовно комп'ютерної гри;

- rate з типом SmallIntegerField – поле оцінки, в яку своє загальне враження за цією комп'ютерною грою оцінив гравець;

- release з типом DateTimeField – поле часу створення цього зворотного зв'язку стосовно комп'ютерної гри.

Модель рекомендації зворотного зв'язку стосовно комп'ютерної гри представлена класом Computergamefeedbackrecom з атрибутами:

- computergamefeedback з типом ForeignKey – зовнішній ключ до моделі Computergamefeedback, зворотній зв'язок стосовно комп'ютерної гри, до якої визначено цю рекомендацію;

- gamer з типом ForeignKey – зовнішній ключ до моделі Gameuser, поле гравця, який створив цю рекомендацію зворотного зв'язку стосовно комп'ютерної гри;

- recom з типом SmallIntegerField – поле типу рекомендації: рекомендації не обов'язково є позитивним враженням, вони можуть бути і позитивними, і негативними, і визначати неінформативність загалом зворотного зв'язку.

Модель емоцій гравця стосовно комп'ютерної гри представлена класом Computergameemotion з атрибутами:

- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, відносно якої цю емоцію створено;

- `gamer` з типом `ForeignKey` – зовнішній ключ до моделі `Gameuser`, поле гравця, який створив цю емоцію зворотного зв'язку стосовно комп'ютерної гри;

- `emotion` з типом `CharField` – поле текстового опису емоції (або фактично інформації стосовно загалом комп'ютерної гри або участі конкретного гравця у комп'ютерній грі);

- `release` з типом `DateTimeField` – поле часу створення цієї емоції стосовно комп'ютерної гри.

Модель відповідей на емоції від комп'ютерної гри представлена класом `Computergameemotionanswer` з атрибутами:

- `computergameemotion` з типом `ForeignKey` – зовнішній ключ до моделі `Computergameemotion`, поле емоції гравця стосовно комп'ютерної гри, відповідь на яку створено;

- `gamer` з типом `ForeignKey` – зовнішній ключ до моделі `Gameuser`, поле гравця, який створив цю відповідь;

- `answer` з типом `CharField` – текст відповіді на емоцію гравця стосовно комп'ютерної гри;

- `release` з типом `DateTimeField` – поле часу створення цієї відповіді на емоцію стосовно комп'ютерної гри.

Модель мови представлена класом `Language` з єдиним атрибутом `name` з типом `CharField`, що є назвою природної мови.

Модель мов інтерфейсу в комп'ютерній грі представлена класом `Computergamelanguage` з атрибутами:

- `interface` з типом `CharField` – поле типу інтерфейсу, для якого ця мова забезпечується (інтерфейс, аудіо озвучення, субтитри);

- `computergame` з типом `ForeignKey` – зовнішній ключ до моделі `Computergame`, поле комп'ютерної гри, в якій ця мова доступна;

- `language` з типом `ForeignKey` – зовнішній ключ до моделі `Language`, поле мови, яка доступна в цьому інтерфейсі.

Модель жанрів комп'ютерних ігор представлена класом Genre з єдиним атрибутом name з типом CharField, що є назвою жанру комп'ютерної гри.

Модель особи, яка може відігравати різні ролі при створенні комп'ютерних ігор, представлена класом Person з атрибутами:

- name з типом CharField – поле імені особи, яка може відігравати різні ролі при створенні комп'ютерних ігор;
- photo з типом CharField – поле фотографії особи, яка може відігравати різні ролі при створенні комп'ютерних ігор.

Модель режисерів комп'ютерних ігор представлена класом Director з атрибутами:

- person з типом ForeignKey – зовнішній ключ до моделі Person, поле особи, яка в цій комп'ютерній грі відіграє роль режисера;
- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, в якій ця особа була режисером;
- description з типом CharField – поле опису особливостей участі цієї особи як режисера при створенні цієї комп'ютерної гри.

Модель авторів сценаріїв комп'ютерних ігор представлена класом Writer з атрибутами:

- person з типом ForeignKey – зовнішній ключ до моделі Person, поле особи, яка в цій комп'ютерній грі відіграє роль автора сценарію;
- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, в якій ця особа була автором сценарію;
- description з типом CharField – поле опису особливостей участі цієї особи як автора сценарію при створенні цієї комп'ютерної гри.

Модель учасників створення комп'ютерних ігор представлена класом Participant з атрибутами:

- person з типом ForeignKey – зовнішній ключ до моделі Person, поле особи, яка в цій комп'ютерній грі відіграє певну роль учасника;

- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, в якій ця особа була учасником створення;

- role з типом CharField – поле з описом конкретної ролі, що описує участь цієї особи в створенні цієї комп'ютерної гри;

- description з типом CharField – поле опису особливостей участі цієї особи при створенні цієї комп'ютерної гри.

Модель платформи комп'ютерних ігор представлена класом OSplatform з атрибутом name з типом CharField – назвою платформи, на якій в принципі може запускатися комп'ютерна гра.

Модель платформ, на яких можуть запускатися конкретні комп'ютерні ігри, представлена класом Gameplatform з атрибутами:

- computergame з типом ForeignKey – зовнішній ключ до моделі Computergame, поле комп'ютерної гри, яка на цій платформі може запускатися;

- osplatform з типом ForeignKey – зовнішній ключ до моделі OSplatform, поле платформи, на якій може запускатися ця комп'ютерна гра;

- minversion з типом CharField – поле мінімальної версії платформи, на якій ця комп'ютерна гра може запускатися.

Модель вимоги до комп'ютерних ігор представлена класом Platformrequirement з атрибутом name з типом CharField – назвою можливої вимоги (тобто самою вимогою) до комп'ютерної гри.

Модель конкретних вимоги до конкретної гри представлена класом Gameplatformrequirement з атрибутами:

- platformrequirement з типом ForeignKey – зовнішній ключ до моделі Platformrequirement, поле можливої вимоги до комп'ютерної гри, за якою конкретизовано значення;

- gameplatform з типом ForeignKey – зовнішній ключ до моделі Gameplatform, поле платформи, на якій може запускатися певна комп'ютерна гра і для якої визначено дане значення вимоги;

- minvalue з типом CharField – поле мінімального значення, яке необхідно забезпечити для того, щоб грати в цю комп'ютерну гру;

- recomvalue з типом CharField – поле рекомендованого значення, яке необхідно забезпечити для того, щоб грати в цю комп'ютерну гру.

Модель гравців представлена класом Gameuser з атрибутами:

- username з типом CharField – поле логіну гравця, що він використовує для доступу до програми;

- password з типом CharField – поле пароллю гравця, що він використовує для доступу до програми;

- active з типом BooleanField – поле активності облікового запису (можна авторизуватись за ним чи ні);

- staff з типом BooleanField – поле участі гравця в програмі як менеджера (або відсутності такого доступу, якщо це звичайний гравець);

- photo з типом ImageField – поле фотографії користувача або певного аватару;

- description з типом CharField – поле презентаційного опису гравця;

- поле developers з типом ManyToManyField – поле розробників комп'ютерних ігор, на яких підписався гравець, фактично реалізує зв'язуючу таблицю в реляційній структурі БД;

- поле reads з типом ManyToManyField – поле гравців, на яких підписався цей гравець, що фактично реалізує зв'язуючу таблицю в реляційній структурі БД;

- поле bookmarks з типом ManyToManyField – поле закладок на комп'ютерних іграх, що фактично реалізує зв'язуючу таблицю в реляційній структурі БД.

Модель типів пристроїв, для яких доступні версії комп'ютерної гри, представлена класом Devicetype з атрибутом name з типом CharField – назвою типу пристроїв.

4.3 Висновки за розділом 4

Виконані створення схеми функціонування програми, реалізація та опис цієї реалізації стосовно розроблених функцій і класів представлені як результати роботи над реалізацією програми.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Призначення програми

Програма призначена для любителів комп'ютерних ігор і надає їм можливість знаходити нові для них комп'ютерні ігри, ділитися з іншими гравцями думками щодо комп'ютерних ігор з власного досвіду, впорядкувати інформацію про персонажів комп'ютерних ігор.

5.2 Виконання програми

Виконання програми розпочинається з відображення рейтингу всіх комп'ютерних ігор. З цієї сторінки можна перемкнутися на інші варіанти рейтингів. Окрім того користувачу доступне меню для вибору інших функцій для роботи. У випадку спроби виконати дії, які потребують авторизації, користувач перенаправляється на сторінку авторизації.

Розділ меню «Ігри» включає засоби пошуку комп'ютерних ігор, відображення переліку рекомендацій (рис. 5.1), рейтингів, що влаштовано візуально аналогічно рекомендаціям, пошуку персонажів з комп'ютерних ігор, відображення комп'ютерних ігор, внесених у закладки, нових ігор розробників, на яких підписався користувач. Сторінка рекомендацій містить перелік з тих комп'ютерних ігор, які за запропонованим методом рекомендовано користувачу. Варто зазначити, що для коректної роботи моделі на даний момент необхідно, щоб сценарій був описаний англійською мовою. За кожною грою показується основна фотографія, назва та компанія-розробник. Біля компанії-розробника показується іконка з зображенням хрестика, що означає, що користувач не підписувався на цього розробника, або галочки, що означає, що користувач підписаний. Для зміни стану потрібно натиснути на іконку. При наведенні на фотографію курсора відображається середня оцінка цієї гри (рис. 5.2).

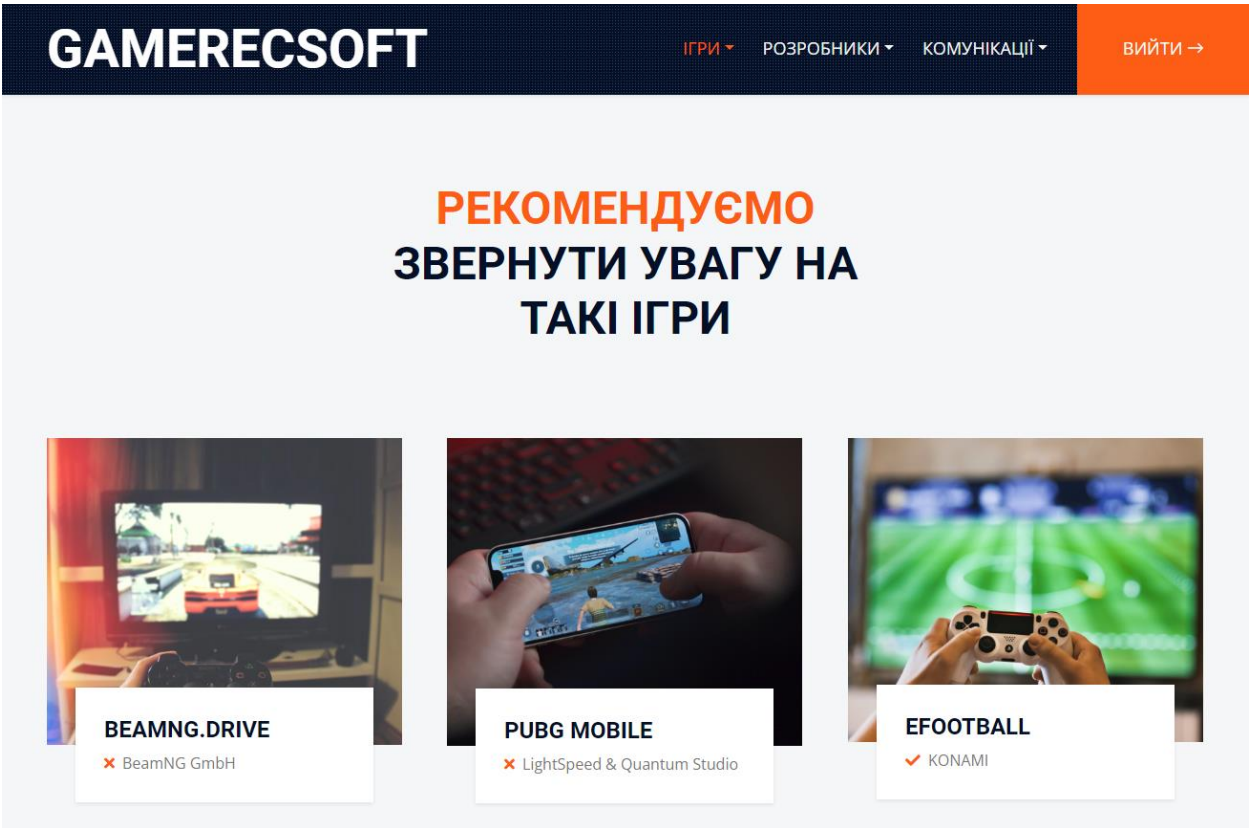


Рисунок 5.1 – Рекомендації комп’ютерних ігор

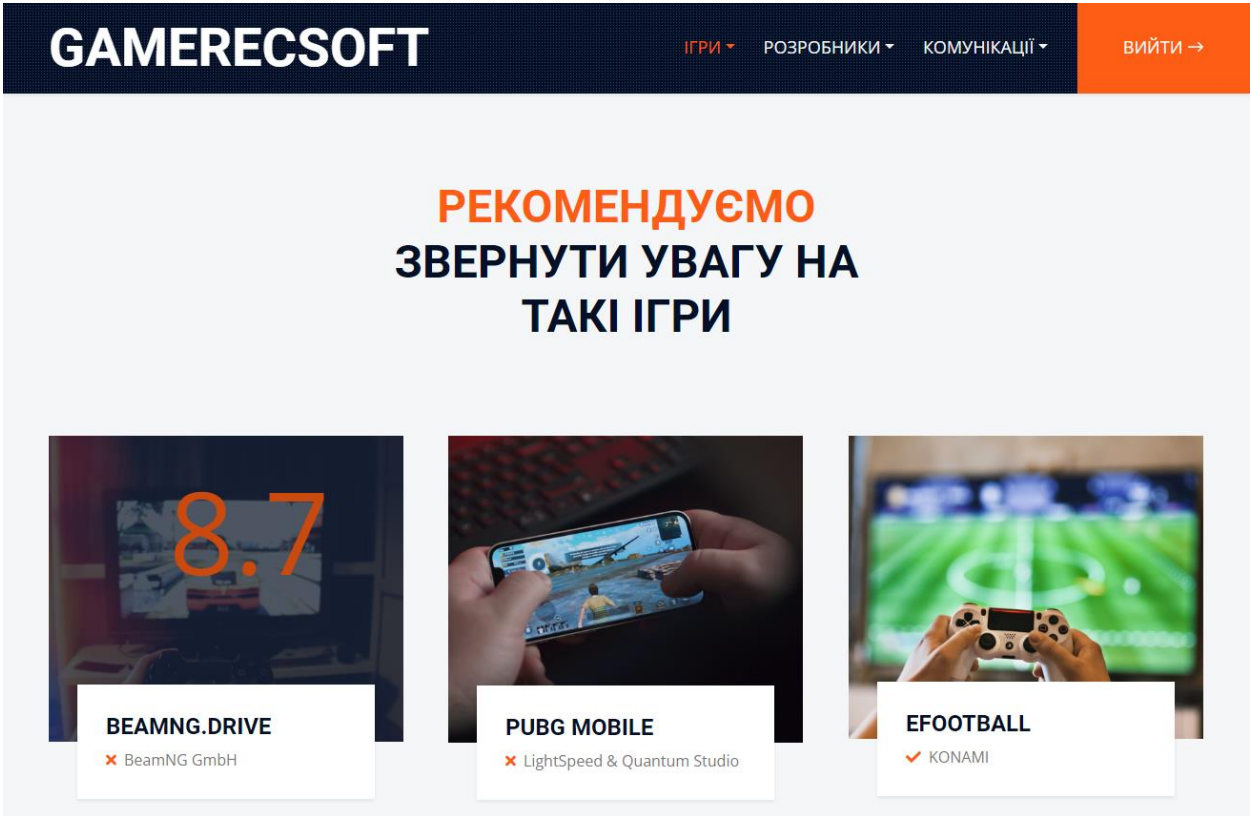


Рисунок 5.2 – Відображення оцінки комп’ютерної гри

Якщо натиснути на назву гри або на фотографію, то можна перейти до сторінки з даними комп'ютерної гри (рис. 5.3). На цій сторінці містяться всі дані про гру, надається можливість створити власний зворотній зв'язок (кнопка «Оцінити»), переглянути збережені зворотні зв'язки інших користувачів, створені емоції користувачів стосовно цієї гри, перелік персонажів комп'ютерної гри, дані про мови, платформи, режисерів, авторів сценарію, інших учасників.



Рисунок 5.3 – Сторінка комп'ютерної гри

Відповідно персонажа комп'ютерної гри можна відкрити на своїй окремій сторінці та переглянути про нього інформацію (рис. 5.4). Перейти до цієї сторінки можна також з пошуку персонажів. На сторінці окрім даних персонажа також містяться зворотні зв'язки користувача відносно персонажа (позначено враженнями). За кожним зворотнім зв'язком показано кількість

рекомендацій різних типів, а натискаючи на таку кнопку, можна додати власну рекомендацію.

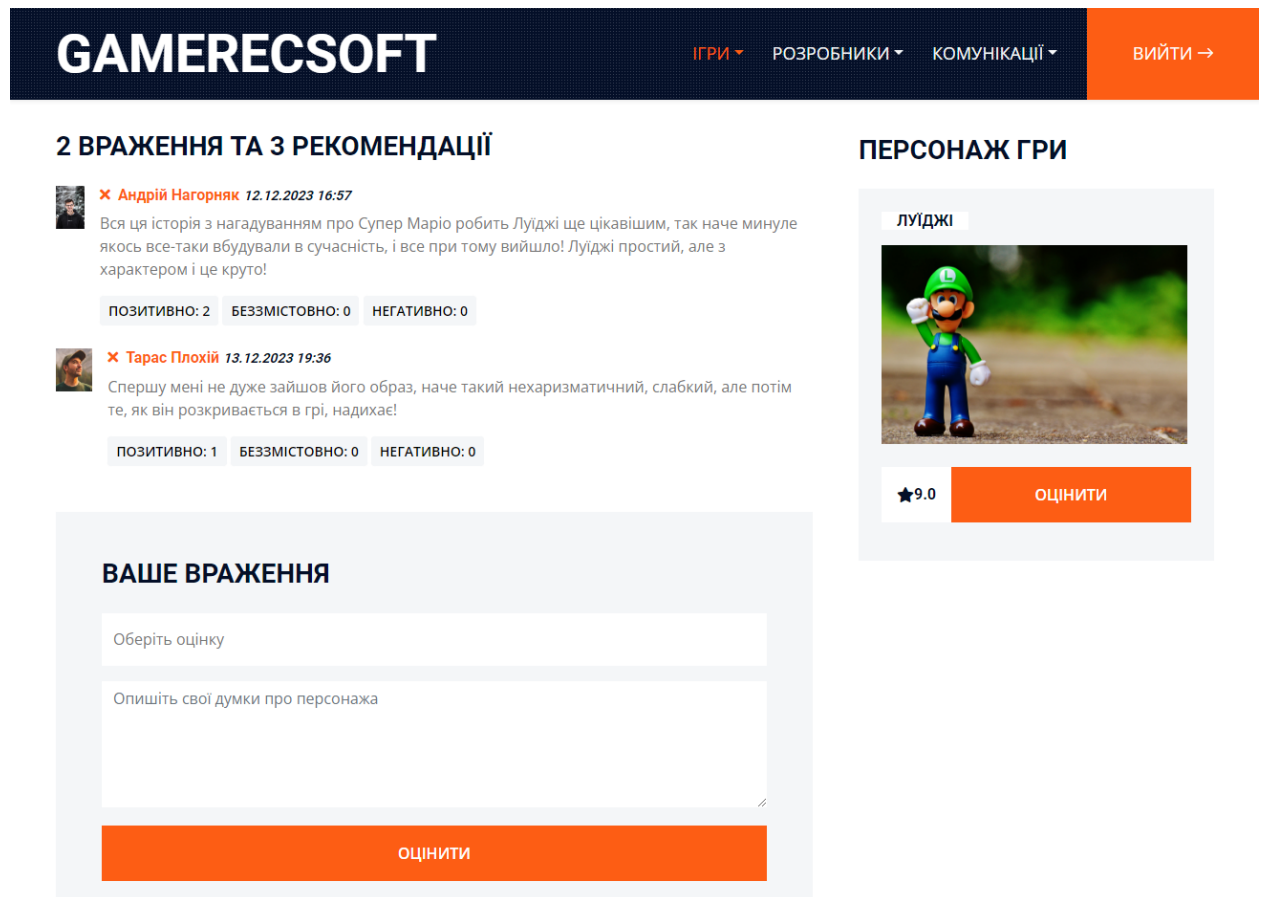


Рисунок 5.4 – Сторінка персонажа комп'ютерної гри

Розділ «Розробники» містить функції пошуку розробників, пошуку осіб, які беруть участь у створенні комп'ютерних ігор.

Розділ «Комунікації» включає перехід до створених власних зворотних зв'язків користувача, висловлених емоцій, переліку реакцій на інформацію від інших користувачів, відображення стрічки стосовно дій користувачів, на яких підписався користувач.

Для менеджера після авторизації доданий окремий розділ «Керування», що дозволяє обрати необхідну дію з керування, що є функціями, доступними тільки менеджеру. Також менеджеру з відповідних сторінок доступні кнопки редагування існуючих даних.

5.3 Експериментальне дослідження

Експериментальне дослідження проводилось стосовно прогнозування оцінки комп'ютерних ігор. Оцінка фактично є рейтингом комп'ютерної гри, який їй виставили користувачі. Для проведення дослідження використано вибірку [11], де представлені дані, зібрані з сервісу IMDb.

При обробці даних було визначено, що сертифікати вікових обмежень для різних комп'ютерних ігор представлені в різних системах. Тому було виконано в результаті перетворення наявних значень наступним чином:

- значення сертифіката «E» переведено в значення 0, адже це відсутність вікових обмежень;
- значення сертифіката «K-A» переведено в значення 0, адже це відсутність вікових обмежень;
- значення сертифіката «G» переведено в значення 0, адже це відсутність вікових обмежень;
- значення сертифіката «ЕС» переведено в значення 3, оскільки ці ігри доступні для користувачів від 3 років;
- значення сертифіката «E10+» переведено в значення 10, оскільки ці ігри доступні для користувачів від 10 років;
- значення сертифіката «12» переведено в значення 12, оскільки ці ігри доступні для користувачів від 12 років;
- значення сертифіката «PG» переведено в значення 13, оскільки цей варіант передбачає певне обмеження, коли користувач може грати, але з наглядом батьків;
- значення сертифіката «MA-13» переведено в значення 13, оскільки ці ігри доступні для користувачів від 13 років;
- значення сертифіката «PG-13» переведено в значення 13, оскільки ці ігри доступні для користувачів від 13 років;
- значення сертифіката «T» переведено в значення 13, оскільки ці ігри доступні для користувачів від 13 років;

- значення сертифіката «TV-14» переведено в значення 14, оскільки ці ігри доступні для користувачів від 14 років;
- значення сертифіката «M» переведено в значення 15, оскільки ці ігри доступні для користувачів від 15 років;
- значення сертифіката «MA-17» переведено в значення 17, оскільки ці ігри доступні для користувачів від 17 років;
- значення сертифіката «TV-MA» переведено в значення 17, оскільки ці ігри доступні для користувачів від 17 років;
- значення сертифіката «GA» переведено в значення 17, оскільки ці ігри доступні для користувачів від 17 років;
- значення сертифіката «AO» переведено в значення 18, оскільки ці ігри доступні для користувачів від 18 років.

Сценарій було перетворено до числової форми описаним у методі чином. Усі інші дані були початково числовими.

Результати дослідження (табл. 5.1) вказують, що за порівняння за середньоквадратичною похибкою ДР відзначились достатньо неточними значеннями порівняно з Random Forest. Найкращий результат ДР отримали при вхідних ознаках, складених з року видання, кількості років до оцінювання, сертифікату та жанрів, у 0,15347. За використання сценарію результати погіршилися.

У той же час Random Forest дозволив отримати найменшу похибку в 0,11783, що відбулось за подання на вхід року видання, кількості років до оцінювання, сертифікату, жанрів, сценарію. Як помітно з результатів, застосування сценарію дозволило для Random Forest зменшити похибку, відповідно вплив є позитивним.

Тож якщо порівнювати між собою результати ДР та Random Forest, то в найкращих для обох альтернатив випадках Random Forest дозволив покращити результат ДР на 23 %. Тож саме використання методу рекомендації комп'ютерних ігор на основі використання методу Random Forest є доцільним для подальшого застосування.

Таблиця 5.1 – Прогнозування оцінки комп'ютерних ігор

Метод або модель	Ознаки	Середньоквадратична похибка
ДР	Рік видання, роки до оцінювання, сертифікат, жанри	0,15347
ДР	Рік видання, роки до оцінювання, сертифікат, жанри, сценарій	0,16834
ДР	Рік видання, роки до оцінювання, сертифікат, сценарій	0,17001
ДР	Рік видання, роки до оцінювання, сценарій	0,17261
Random Forest	Рік видання, роки до оцінювання, сертифікат, сценарій	0,11903
Random Forest	Рік видання, роки до оцінювання, сценарій	0,11919
Random Forest	Рік видання, роки до оцінювання, сертифікат, жанри	0,14017
Random Forest	Рік видання, роки до оцінювання, сертифікат, жанри, сценарій	0,11783

5.4 Висновки за розділом 5

Усі сценарії роботи з програмою реалізовані, для користувачів у стандартній ролі і у ролі менеджера забезпечено роботу, всі сценарію

виконуються коректно. Експериментальне дослідження дозволило підтвердити доцільність використання запропонованого методу на основі виконання прогнозування оцінки комп'ютерної гри з порівнянням між собою методу Random Forest та ДР. Використання ансамблів ДР замість окремих ДР дозволило зменшити рівень середньоквадратичної похибки, що було отримано в результаті.

ВИСНОВКИ

У роботі розроблено програмне забезпечення надання інформації про комп'ютерні ігри з можливістю рекомендування ігор користувачам на основі виявлення майбутніх трендів.

Проаналізовано програми, які виконують надання інформації про комп'ютерні ігри. До них віднесено Steam, Metacritic, IMDB Video Games, визначено особливості виконаної розробки порівняно з цими рішеннями.

Програма розроблена для любителів комп'ютерних ігор і надає їм можливість знаходити нові для них комп'ютерні ігри, ділитися з іншими гравцями думками щодо комп'ютерних ігор з власного досвіду, впорядкувати інформацію про персонажів комп'ютерних ігор. Користувачі можуть залишати різними способами зворотній зв'язок щодо комп'ютерних ігор, персонажів ігор, рекомендувати інформацію від інших користувачів, зберігати ігри, підписуватися на розробників ігор, інших користувачів, виконувати пошук відповідної інформації про комп'ютерні ігри, персонажів, розробників, осіб, задіяних у створенні ігор. До складу програми інтегровано рекомендування комп'ютерних ігор.

Наукова новизна роботи полягає в запропонованому методі рекомендування комп'ютерних ігор, який базується на створенні рекомендацій з комп'ютерних ігор, які мають найкращі очікувані оцінки та найбільше зростання, використовуючи для прогнозування оцінок ансамблі дерев рішень на основі методу Random Forest.

Проведено експериментальне дослідження, яке підтвердило доцільність використання ансамблів дерев рішень порівняно з окремими деревами рішень щодо прогнозування оцінки комп'ютерної гри, на основі чого встановлюються майбутні тренди, що визначають рекомендації, надані користувачам. Значення середньоквадратичної похибки було зменшено на 23 %.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Morikawa, Y. The Gaming Industry Sees a Staggering Surge in Popularity [Electronic resource] / Y. Morikawa. – Access mode : <https://globaledge.msu.edu/blog/post/57295/the-gaming-industry-sees-a-staggering-surge-in-popularity>.
2. Welcome to Steam [Electronic resource]. – Access mode : <https://store.steampowered.com/>.
3. Video Game Reviews, Articles, Trailers and more – Metacritic [Electronic resource]. – Access mode : <https://www.metacritic.com/game/>.
4. IMDb Top 250 Rated Video Games – IMDb [Electronic resource]. – Access mode : <https://www.imdb.com/list/ls097840768/>.
5. Understanding Decision Trees in Machine Learning [Electronic resource]. – Access mode : <https://deepai.org/machine-learning-glossary-and-terms/decision-tree>.
6. Mansurova, M. Interpreting Random Forests [Electronic resource] / M. Mansurova. – Access mode : <https://towardsdatascience.com/interpreting-random-forests-638bca8b49ea>.
7. Word2Vec Explained [Electronic resource]. – Access mode : <https://towardsdatascience.com/word2vec-explained-49c52b4ccb71>.
8. Bottle [Electronic resource] : Python Web Framework – Bottle 0.13-dev documentation. – Access mode : <https://bottlepy.org/docs/dev/>.
9. Getting started with Django. Django [Electronic resource]. – Access mode : <https://www.djangoproject.com/start/>.
10. WEBUILD Free Website Template. Free CSS Templates. Free CSS [Electronic resource]. – Access mode : <https://www.free-css.com/free-css-templates/page283/webuild>.
11. IMDB Video Games [Electronic resource]. – Access mode : <https://www.kaggle.com/datasets/muhammadadiltalay/imdb-video-games>.

ДОДАТОК А
Технічне завдання

Вступ

Комп'ютерні ігри є сферою життя людей, яка значно впливає на сучасне життя, культуру, взаємовідношення. Відповідно і кількість можливих комп'ютерних ігор на ринку є величезною, а тому знайти те, що може зацікавити користувача непросто. Виявлення тенденцій на ринку може в цьому допомогти. Врахування тенденцій ринку в рекомендаціях для користувачів програми може бути корисним інструментом, який разом з іншими особливостями може привернути увагу потенційних користувачів. Додатковими можливостями є акцент на персонажах комп'ютерних ігор, приділення уваги на основі централізованої платформи, якою і має бути ця програма, взаємодії різних користувачів стосовно окремих персонажів, які в іграх існують.

A.1 Підстави для розробки

Завдання на роботу визначено самим виданим завданням від 4 вересня 2023 року, що має входити до складу пояснювальної записки, та наказом № 437 від 14 листопада 2023 р. за Національним університетом «Запорізька політехніка», де затверджено тему роботи: «Дослідження та програмна реалізація методів рекомендування комп'ютерних ігор».

A.2 Призначення розробки

Програма має бути розроблена для любителів комп'ютерних ігор і надавати їм можливість знаходження нових для них комп'ютерних ігор, ділитися з іншими гравцями думками щодо комп'ютерних ігор з власного досвіду.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Для того щоб досягти сформульовано призначення розробки в підсумку, потрібно виконати такі функціональні вимоги:

- пошук комп'ютерних ігор;
- рекомендування комп'ютерних ігор;
- перегляд деталізації розробника комп'ютерних ігор;
- реєстрація нової компанії-розробника комп'ютерних ігор;
- редагування даних розробника комп'ютерних ігор;
- пошук компаній-розробників комп'ютерних ігор;
- пошук осіб, задіяних у розробці комп'ютерних ігор;
- перегляд деталізації особи, задіяної в створенні комп'ютерних ігор;
- перегляд деталізації комп'ютерної гри;
- редагування даних комп'ютерної гри;
- реєстрація нової комп'ютерної гри;
- реєстрація персонажа комп'ютерної гри;
- редагування даних персонажа комп'ютерної гри;
- перегляд деталізації персонажа комп'ютерної гри;
- пошук персонажів комп'ютерних ігор;
- перегляд рейтингу всіх комп'ютерних ігор;
- перегляд рейтингу комп'ютерних ігор за жанром;
- перегляд рейтингу комп'ютерних ігор за доступною мовою в інтерфейсі;
- перегляд рейтингу комп'ютерних ігор за платформою;
- перегляд рейтингу нових комп'ютерних ігор;
- перегляд рейтингу персонажів комп'ютерних ігор;
- надання зворотного зв'язку відносно комп'ютерної гри;
- рекомендування зворотного зв'язку відносно комп'ютерної гри;
- надання зворотного зв'язку відносно персонажа комп'ютерної гри;

- рекомендування зворотного зв'язку відносно персонажа комп'ютерної гри;
- публікація емоції стосовно комп'ютерної гри;
- публікація відповіді на емоцію стосовно комп'ютерної гри;
- перегляд всіх власних оформлених зворотних зв'язків;
- перегляд всіх власних реакцій на дії інших гравців;
- створення закладки на комп'ютерній грі;
- прибирання закладки з комп'ютерної гри;
- перегляд комп'ютерних ігор, віднесених користувачем до своїх закладок;
- оформлення підписки на розробника комп'ютерних ігор;
- відключення підписки на розробника комп'ютерних ігор;
- оформлення підписки на гравця;
- відключення підписки на гравця;
- перегляд нових комп'ютерних ігор від розробників, на яких є підписка;
- перегляд стрічки дій гравців, на яких є підписка;
- авторизація;
- перегляд деталізації даних користувача.

A.3.2 Умови експлуатації та вимоги

Умови експлуатації програми визначаються тими засобами, які мають бути обрані під час проектування програми, включаючи зокрема підтримку обраної мови програмування, організацію роботи серверу визначеним чином, забезпечення на ньому роботи системи керування базами даних, яку має бути обрано, а також будь-яких додаткових засобів, які обрані та використані під час розробки програми та потребують інсталяції.

A.4 Порядок контролю та приймання

Контроль виконання заявлених складових роботи має відбуватися згідно з завданням, виданим на роботу, що є однією з підстав для розробки. Виконання роботи має бути завершено протягом 12 тижнів з проведенням захисту перед екзаменаційною комісією.

ДОДАТОК Б
Текст програми

```

from django.shortcuts import render
from .models import Developer
from .models import Gamebookmark
from .models import Gameuser
from .models import Developerread

def developer_games_view(request, developer_id):
    try:
        game_developer = Developer.objects.get(id =
developer_id)
    except Developer.DoesNotExist:
        return render(request,
"gamerecsoft/developer.html", {})
        developer_games = Computergame.objects.filter(developer
= game_developer)
        new_developer_games = developer_games.order_by("-
release")
        new_developer_games = new_developer_games.values()
        new_developer_games = new_developer_games[:5]
        best_developer_games = developer_games.order_by("-
rate")
        best_developer_games = best_developer_games.values()
        best_developer_games = best_developer_games[:5]
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            reads = Developerread.objects.filter(developer =
game_developer, gamer = user_gamer)
            if reads and reads.count() > 0:
                game_developer.reader = True
            else:
                game_developer.reader = False
            user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
            user_computer_games =
user_computer_games.select_related("computergame")
            user_computer_games =
user_computer_games.values_list("computergame")
            for computer_game in new_developer_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
            for computer_game in best_developer_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False

```

```

        return render(request, "gamerecsoft/developer.html", {
            "newgames": new_developer_games, "bestgames":
            best_developer_games, "developer": game_developer })

    def best_computer_games_view(request):
        best_games =
        Computergame.objects.select_related("developer")
        best_games = best_games.order_by("-rate")
        best_games = best_games.values()
        best_games = best_games[:500]
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
            request.user.username)
            user_computer_games =
            Gamebookmark.objects.filter(gamer = user_gamer)
            user_computer_games =
            user_computer_games.select_related("computergame")
            user_computer_games =
            user_computer_games.values_list("computergame")
            for computer_game in best_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
        return render(request, "gamerecsoft/bestoverall.html",
        { "games": best_games })

    from .models import Genre

    def best_genre_computer_games_view(request, genre_id):
        try:
            game_genre = Genre.objects.get(id = genre_id)
        except Genre.DoesNotExist:
            return render(request,
            "gamerecsoft/bestgenre.html", {})
        best_games =
        Computergame.objects.select_related("developer")
        best_games = Computergame.objects.filter(genres =
        game_genre)
        best_games = best_games.order_by("-rate")
        best_games = best_games.values()
        best_games = best_games[:500]
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
            request.user.username)
            user_computer_games =
            Gamebookmark.objects.filter(gamer = user_gamer)
            user_computer_games =
            user_computer_games.select_related("computergame")

```

```

        user_computer_games =
user_computer_games.values_list("computergame")
        for computer_game in best_games:
            if computer_game in user_computer_games:
                computer_game.reader = True
            else:
                computer_game.reader = False
        computer_genres = Genre.objects.all()
        return render(request, "gamerecsoft/bestgenre.html", {
"games": best_games, "genre": game_genre, "genres":
computer_genres })

    from .models import Language

    def best_language_computer_games_view(request,
language_id):
        try:
            game_language = Language.objects.get(id =
language_id)
        except Language.DoesNotExist:
            return render(request, "gamerecsoft/bestlang.html",
{})

            language_games =
Computergamelanguage.objects.filter(language = game_language)
            language_games =
language_games.values_list("computergame__id")
            language_games = language_games.distinct()
            best_games =
Computergame.objects.select_related("developer")
            best_games = best_games.filter(id__in = language_games)
            best_games = best_games.order_by("-rate")
            best_games = best_games.values()
            best_games = best_games[:500]
            if request.user.is_authenticated:
                user_gamer = Gameuser.objects.filter(username =
request.user.username)
                user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
                user_computer_games =
user_computer_games.select_related("computergame")
                user_computer_games =
user_computer_games.values_list("computergame")
                for computer_game in best_games:
                    if computer_game in user_computer_games:
                        computer_game.reader = True
                    else:
                        computer_game.reader = False
            computer_languages = Language.objects.all()

```

```

        return render(request, "gamerecsoft/bestlang.html", {
"games": best_games, "language": game_language, "languages":
computer_languages })

    from .models import OSplatform
    from .models import Gameplatform

    def best_platform_computer_games_view(request,
osplatform_id):
        try:
            os_platform = OSplatform.objects.get(id =
osplatform_id)
        except OSplatform.DoesNotExist:
            return render(request,
"gamerecsoft/bestplatform.html", {})
            platform_games = Gameplatform.objects.filter(osplatform
= os_platform)
            platform_games =
platform_games.values_list("computergame__id")
            platform_games = platform_games.distinct()
            best_games =
Computergame.objects.select_related("developer")
            best_games = best_games.filter(id__in = platform_games)
            best_games = best_games.order_by("-rate")
            best_games = best_games.values()
            best_games = best_games[:500]
            if request.user.is_authenticated:
                user_gamer = Gameuser.objects.filter(username =
request.user.username)
                user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
                user_computer_games =
user_computer_games.select_related("computergame")
                user_computer_games =
user_computer_games.values_list("computergame")
                for computer_game in best_games:
                    if computer_game in user_computer_games:
                        computer_game.reader = True
                    else:
                        computer_game.reader = False
                os_platforms = OSplatform.objects.all()
                return render(request, "gamerecsoft/bestlang.html", {
"games": best_games, "platform": os_platform, "platforms":
os_platforms })

    import datetime
    from dateutil.relativedelta import relativedelta

    def best_new_computer_games_view(request):

```

```

        best_games =
Computergame.objects.select_related("developer")
        curr_moment = datetime.datetime.now().date()
        new_date = curr_moment - relativedelta(months = 1)
        best_games = best_games.filter(release__gte = new_date)
        best_games = best_games.order_by("-rate")
        best_games = best_games.values()
        best_games = best_games[:500]
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
            user_computer_games =
user_computer_games.select_related("computergame")
            user_computer_games =
user_computer_games.values_list("computergame")
            for computer_game in best_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
        return render(request, "gamerecsoft/bestnew.html", {
"games": best_games })

```

```

from .models import Computergamecharacter

def best_computer_game_characters_view(request):
    game_characters =
Computergamecharacter.objects.filter(osplatform = os_platform)
    game_characters =
game_characters.select_related("computergame")
    game_characters = game_characters.order_by("-rate")
    game_characters = game_characters.values()
    game_characters = game_characters[:500]
    if request.user.is_authenticated:
        user_gamer = Gameuser.objects.filter(username =
request.user.username)
        user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
        user_computer_games =
user_computer_games.select_related("computergame")
        user_computer_games =
user_computer_games.values_list("computergame")
        for game_character in game_characters:
            if game_character.computergame in
user_computer_games:
                game_character.reader = True
            else:
                game_character.reader = False

```

```

        return render(request,
"gamerecsoft/bestcharacters.html", { "characters":
game_characters })

    from .models import Computergamecharacterfeedback
    from .models import Computergamefeedback
    from .models import Computergamecharacterfeedbackrecom
    from .models import Computergamefeedbackrecom

    def own_gamer_feedbacks_view(request):
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_game_feedbacks =
Computergamefeedback.objects.filter(gamer = user_gamer)
            user_game_feedbacks =
user_game_feedbacks.select_related("computergame")
            user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
            user_computer_games =
user_computer_games.select_related("computergame")
            user_computer_games =
user_computer_games.order_by("-release")
            for feedback in user_game_feedbacks:
                if feedback.computergame in
user_computer_games:
                    feedback.reader = True
                else:
                    feedback.reader = False
                recommendations =
Computergamecharacterfeedbackrecom.objects.filter(computergamech
aracterfeedback = feedback)
                recoms =
Computergamefeedbackrecom.objects.all()
                recoms = recoms.values_list("recom")
                recoms = recoms.distinct()
                recs = {}
                for recom in recoms:
                    rec_results = recommendations.filter(recom
= recom)
                    rec_results = rec_results.count()
                    recs[recom] = rec_results
                feedback.recoms = recs
            user_character_feedbacks =
Computergamecharacterfeedback.objects.filter(gamer = user_gamer)
            user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacter")
            user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacter__
computergame")

```

```

        user_character_feedbacks =
user_character_feedbacks.order_by("-release")
        for feedback in user_character_feedbacks:
            if feedback.computergamecharacter.computergame
in user_computer_games:
                feedback.reader = True
            else:
                feedback.reader = False
            recommendations =
Computergamefeedbackrecom.objects.filter(computergamefeedback =
feedback)

            recoms =
Computergamecharacterfeedbackrecom.objects.all()
            recoms = recoms.values_list("recom")
            recoms = recoms.distinct()
            recs = {}
            for recom in recoms:
                rec_results = recommendations.filter(recom
= recom)

                rec_results = rec_results.count()
                recs[recom] = rec_results
            feedback.recoms = recs
        return render(request,
"gamerecsoft/bestcharacters.html", { "game_feedbacks":
user_game_feedbacks, "character_feedbacks":
user_character_feedbacks })

def own_recommendations_view(request):
    if request.user.is_authenticated:
        user_gamer = Gameuser.objects.filter(username =
request.user.username)
        user_game_feedbacks =
Computergamefeedbackrecom.objects.filter(gamer = user_gamer)
        user_game_feedbacks =
user_game_feedbacks.select_related("computergamefeedback")
        user_game_feedbacks =
user_game_feedbacks.select_related("computergamefeedback__comput
ergame")

        user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
        user_computer_games =
user_computer_games.select_related("computergame")
        user_computer_games =
user_computer_games.order_by("-id")
        for feedback in user_game_feedbacks:
            if feedback.computergame in
user_computer_games:
                feedback.reader = True
            else:
                feedback.reader = False

```

```

        recommendations =
Computergamecharacterfeedbackrecom.objects.filter(computergamech
aracterfeedback = feedback)
        recoms =
Computergamefeedbackrecom.objects.all()
        recoms = recoms.values_list("recom")
        recoms = recoms.distinct()
        recs = {}
        for recom in recoms:
            rec_results = recommendations.filter(recom
= recom)

            rec_results = rec_results.count()
            recs[recom] = rec_results
            feedback.recoms = recs
        user_character_feedbacks =
Computergamecharacterfeedbackrecom.objects.filter(gamer =
user_gamer)

        user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacterfe
edback")

        user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacterfe
edback__computergamecharacter")

        user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacterfe
edback__computergamecharacter__computergame")

        user_character_feedbacks =
user_character_feedbacks.order_by("-id")
        for feedback in user_character_feedbacks:
            if feedback.computergamecharacter.computergame
in user_computer_games:
                feedback.reader = True
            else:
                feedback.reader = False
            recommendations =
Computergamefeedbackrecom.objects.filter(computergamefeedback =
feedback)

            recoms =
Computergamecharacterfeedbackrecom.objects.all()
            recoms = recoms.values_list("recom")
            recoms = recoms.distinct()
            recs = {}
            for recom in recoms:
                rec_results = recommendations.filter(recom
= recom)

                rec_results = rec_results.count()
                recs[recom] = rec_results
                feedback.recoms = recs
            user_emotion_answers =
Computergameemotionanswer.objects.filter(gamer = user_gamer)

```

```

        user_emotion_answers =
user_emotion_answers.select_related("computergameemotion")
        user_emotion_answers =
user_emotion_answers.select_related("computergameemotion__comput
ergame")
        user_emotion_answers =
user_emotion_answers.order_by("-release")
        for emotion in user_emotion_answers:
            if emotion.computergameemotion.computergame in
user_computer_games:
                emotion.reader = True
            else:
                emotion.reader = False
            answers =
Computergameemotionanswer.objects.filter(computergameemotion =
emotion)
            feedback.answers = answers.count()
            return render(request, "gamerecsoft/ownrecom.html",
{ "games": user_game_feedbacks, "emotions":
user_emotion_answers, "characters": user_character_feedbacks })

def bookmark_games_view(request):
    if request.user.is_authenticated:
        user_gamer = Gameuser.objects.filter(username =
request.user.username)
        bookmarks = Gamebookmark.objects.filter(gamer =
user_gamer)
        bookmarks =
bookmarks.select_related("computergame")
        bookmarks =
bookmarks.values_list("computergame__id")
        computer_games = Computergame.objects.filter(id__in
= bookmarks)
        computer_games =
computer_games.select_related("developer")
        return render(request,
"gamerecsoft/bookmarks.html", { "games": computer_games })
    else:
        return render(request,
"gamerecsoft/bookmarks.html", { "games": [] })

def read_developer_games_view(request):
    if request.user.is_authenticated:
        user_gamer = Gameuser.objects.filter(username =
request.user.username)
        user_gamer =
user_gamer.prefetch_related("developers")
        computer_games =
Computergame.objects.filter(developer__in =
user_gamer.developers)

```

```

        computer_games =
computer_games.select_related("developer")
        curr_moment = datetime.datetime.now().date()
        new_date = curr_moment - relativedelta(months = 1)
        computer_games = computer_games.filter(release__gte
= new_date)
        computer_games = computer_games.order_by("-
release")
        user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
        user_computer_games =
user_computer_games.select_related("computergame")
        user_computer_games =
user_computer_games.values_list("computergame")
        for computer_game in best_games:
            if computer_game in user_computer_games:
                computer_game.reader = True
            else:
                computer_game.reader = False
        return render(request,
"gamerecsoft/readdeveloper.html", { "games": computer_games })
    else:
        return render(request,
"gamerecsoft/readdeveloper.html", { "games": [] })

def read_gamers_view(request):
    if request.user.is_authenticated:
        user_gamer = Gameuser.objects.filter(username =
request.user.username)
        user_gamer = user_gamer.prefetch_related("reads")
        user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
        user_computer_games =
user_computer_games.select_related("computergame")
        user_computer_games =
user_computer_games.values_list("computergame")
        emotions =
Computergameemotion.objects.filter(gamer__in = user_gamer.reads)
        emotions = emotions.order_by("-release")
        emotions = emotions.values()
        emotions = emotions[:1000]
        for emotion in emotions:
            if emotion.computergameemotion.computergame in
user_computer_games:
                emotion.reader = True
            else:
                emotion.reader = False
        answers =
Computergameemotionanswer.objects.filter(computergameemotion =
emotion)

```

```

        feedback.answers = answers.count()
        game_feedbacks =
Computergamefeedback.objects.filter(gamer__in =
user_gamer.reads)
        game_feedbacks = game_feedbacks.order_by("-
release")
        game_feedbacks = game_feedbacks.values()
        game_feedbacks = game_feedbacks [:1000]
        for feedback in game_feedbacks:
            if feedback.computergame in
user_computer_games:
                feedback.reader = True
            else:
                feedback.reader = False
            recommendations =
Computergamefeedbackrecom.objects.filter(computergamefeedback =
feedback)
                reoms =
Computergamefeedbackrecom.objects.all()
                reoms = reoms.values_list("recom")
                reoms = reoms.distinct()
                recs = {}
                for recom in reoms:
                    rec_results = recommendations.filter(recom
= recom)
                        rec_results = rec_results.count()
                        recs[recom] = rec_results
                    feedback.reoms = recs
                character_feedbacks =
Computergamecharacterfeedback.objects.filter(gamer__in =
user_gamer.reads)
                character_feedbacks =
character_feedbacks.order_by("-release")
                character_feedbacks = character_feedbacks.values()
                character_feedbacks = character_feedbacks [:1000]
                for feedback in character_feedbacks:
                    if
feedback.computergamecharacterfeedback.computergamecharacter.com
putergame in user_computer_games:
                        feedback.reader = True
                    else:
                        feedback.reader = False
                    recommendations =
Computergamecharacterfeedbackrecom.objects.filter(computergamech
aracter = feedback)
                        reoms =
Computergamecharacterfeedbackrecom.objects.all()
                        reoms = reoms.values_list("recom")
                        reoms = reoms.distinct()
                        recs = {}

```

```

        for recom in recoms:
            rec_results = recommendations.filter(recom
= recom)
            rec_results = rec_results.count()
            recs[recom] = rec_results
            feedback.recoms = recs
        return render(request,
"gamerecsoft/readdeveloper.html", { "games": game_feedbacks,
"emotions": emotions, "characters": character_feedbacks })
    else:
        return render(request,
"gamerecsoft/readgamers.html", { "games": [], "emotions": [],
"characters": []})

from .models import Person
from .models import Director
from .models import Writer
from .models import Participant

def person_view(request, person_id):
    try:
        game_person = Person.objects.get(id = person_id)
    except Person.DoesNotExist:
        return render(request, "gamerecsoft/person.html",
{})

        director_games = Director.objects.filter(person =
game_person)
        director_games =
director_games.select_related("computergame")
        director_game_ids =
director_games.values_list("computergame__id")
        director_games = Computergame.objects.filter(id__in =
director_game_ids)
        director_games = director_games.order_by("-rate")
        director_games = director_games.values()
        director_games = director_games[:5]
        writer_games = Writer.objects.filter(person =
game_person)
        writer_games =
writer_games.select_related("computergame")
        writer_game_ids =
writer_games.values_list("computergame__id")
        writer_games = Computergame.objects.filter(id__in =
writer_game_ids)
        writer_games = writer_games.order_by("-rate")
        writer_games = writer_games.values()
        writer_games = writer_games[:5]
        participant_games = Participant.objects.filter(person =
game_person)

```

```

        participant_games =
participant_games.select_related("computergame")
        participant_game_ids =
participant_games.values_list("computergame__id")
        participant_games = Computergame.objects.filter(id__in
= writer_game_ids)
        participant_games = participant_games.order_by("-rate")
        participant_games = participant_games.values()
        participant_games = participant_games[:5]
        ids = [director_game_ids, writer_game_ids,
participant_game_ids]
        ids = list(dict.fromkeys(ids))
        new_games = Computergame.objects.filter(id__in = ids)
        new_games = new_games.order_by("-release")
        new_games = new_games.values()
        new_games = new_games[:5]
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)
            user_computer_games =
user_computer_games.select_related("computergame")
            user_computer_games =
user_computer_games.values_list("computergame")
            for computer_game in director_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
            for computer_game in writer_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
            for computer_game in participant_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
            for computer_game in new_games:
                if computer_game in user_computer_games:
                    computer_game.reader = True
                else:
                    computer_game.reader = False
        return render(request, "gamerecsoft/person.html", {
"dirgames": director_games, "wgames": writer_games, "partgames":
participant_games, "newgames": new_games })
    from .models import Computergamecharacter

```

```

def game_character_view(request, character_id):
    try:
        game_character =
Computergamecharacter.objects.select_related("computergame")
        game_character =
game_character.select_related("computergame__developer")
        game_character = game_character.get(id =
character_id)
    except Computergamecharacter.DoesNotExist:
        return render(request,
"gamerecsoft/character.html", {})
        character_feedbacks =
Computergamecharacterfeedback.objects.filter(computergamecharact
er = game_character)
        character_feedbacks =
character_feedbacks.select_related("gamer")
        for character_feedback in character_feedbacks:
            recommendations =
Computergamecharacterfeedbackrecom.objects.filter(computergamech
aracter = character_feedback)
            recoms =
Computergamecharacterfeedbackrecom.objects.all()
            recoms = recoms.values_list("recom")
            recoms = recoms.distinct()
            recs = {}
            for recom in recoms:
                rec_results = recommendations.filter(recom
= recom)

                rec_results = rec_results.count()
                recs[recom] = rec_results
            character_feedback.recoms = recs
        return render(request, "gamerecsoft/character.html", {
"character": game_character, "feedbacks": character_feedbacks })

from .models import Computergameimage
from .models import Computergamevideo
from .models import Gameplatform
from .models import Gameplatformrequirement

def game_view(request, game_id):
    try:
        computer_game =
Computergame.objects.select_related("developer")
        computer_game =
Computergame.objects.prefetch_related("devicetypes")
        computer_game =
Computergame.objects.prefetch_related("genres")
        computer_game = computer_game.get(id = game_id)
    except Computergamecharacter.DoesNotExist:
        return render(request, "gamerecsoft/game.html", {})

```

```

        characters =
Computergamecharacter.objects.filter(computergame =
computer_game)
        characters = characters.order_by("-rate")
        images = Computergameimage.objects.filter(computergame
= computer_game)
        videos = Computergamevideo.objects.filter(computergame
= computer_game)
        feedbacks =
Computergamefeedback.objects.filter(computergame =
computer_game)
        feedbacks = feedbacks.select_related("gamer")
        feedbacks = feedbacks.prefetch_related("gamer__reads")
        feedbacks = feedbacks.order_by("-release")
        emotions =
Computergameemotion.objects.filter(computergame = computer_game)
        emotions = emotions.order_by("-release")
        languages =
Computergamelanguage.objects.filter(computergame =
computer_game)
        languages = languages.select_related("language")
        languages = languages.values()
        languages = languages.distinct()
        directors = Director.objects.filter(computergame =
computer_game)
        directors = directors.select_related("person")
        writers = Writer.objects.filter(computergame =
computer_game)
        writers = writers.select_related("person")
        participants = Participant.objects.filter(computergame
= computer_game)
        participants = participants.select_related("person")
        platforms = Gameplatform.objects.filter(computergame =
computer_game)
        platforms = platforms.select_related("osplatform")
        requirements =
Gameplatformrequirement.objects.filter(gameplatform__computergam
e = computer_game)
        requirements =
requirements.select_related("gameplatform")
        requirements =
requirements.select_related("gameplatform__osplatform")
        requirements =
requirements.select_related("platformrequirement")
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_computer_games =
Gamebookmark.objects.filter(gamer = user_gamer)

```

```

        user_computer_games =
user_computer_games.select_related("computergame")
        user_computer_games =
user_computer_games.values_list("computergame")
        reads = Developerread.objects.filter(developer =
computer_game.developer, gamer = user_gamer)
        if reads and reads.count() > 0:
            computer_game.reader = True
        else:
            computer_game.reader = False
        for feedback in feedbacks:
            if user_gamer in feedback.gamer.reads:
                feedback.gamer.read = True
            else:
                feedback.gamer.read = False
            if feedback.computergame in
user_computer_games:
                feedback.reader = True
            else:
                feedback.reader = False
            recommendations =
Computergamefeedbackrecom.objects.filter(computergamefeedback =
feedback)

            recoms =
Computergamefeedbackrecom.objects.all()
            recoms = recoms.values_list("recom")
            recoms = recoms.distinct()
            recs = {}
            for recom in recoms:
                rec_results = recommendations.filter(recom
= recom)

                rec_results = rec_results.count()
                recs[recom] = rec_results
            feedback.recoms = recs
        for emotion in emotions:
            if user_gamer in feedback.gamer.reads:
                emotion.gamer.read = True
            else:
                emotion.gamer.read = False
            if emotion.computergameemotion.computergame in
user_computer_games:
                emotion.reader = True
            else:
                emotion.reader = False
            answers =
Computergameemotionanswer.objects.filter(computergameemotion =
emotion)

            feedback.answers_count = answers.count()
            feedback.answers = answers

```

```

        return render(request, "gamerecsoft/game.html", {
            "game": computer_game, "characters": characters, "images":
            images, "videos": videos, "feedbacks": feedbacks, "emotions":
            emotions, "languages": languages, "directors": directors,
            "writers": writers, "participants": participants, "platforms":
            platforms, "requirements": requirements })

    from django.shortcuts import get_object_or_404
    from django.http import HttpResponseRedirect

    def read_gamer_view(request, gamer_username):
        user_to_read = get_object_or_404(Gameuser, username =
gamer_username)
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_gamer = user_gamer.prefetch_related("reads")
            if not user_to_read in user_gamer.reads:
                user_gamer.reads.add(user_to_read)
                return HttpResponseRedirect (reverse("gamer-
profile", args=(gamer_username,)))
            else:
                user_gamer.reads.remove(user_to_read)
                return HttpResponseRedirect (reverse("gamer-
profile", args=(gamer_username,)))

    def read_developer_view(request, developer_id):
        game_developer = get_object_or_404(Developer, id =
developer_id)
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_gamer =
user_gamer.prefetch_related("developers")
            if not game_developer in user_gamer.developers:
                user_gamer.developers.add(game_developer)
                return HttpResponseRedirect
(reverse("developer-profile", args=(developer_id,)))
            else:
                user_gamer.developers.remove(game_developer)
                return HttpResponseRedirect
(reverse("developer-profile", args=(developer_id,)))

    def bookmark_game_view(request, game_id):
        computer_game = get_object_or_404(Computergame, id =
game_id)
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)

```

```

        user_gamer =
user_gamer.prefetch_related("bookmarks")
        if not computer_game in user_gamer.bookmarks:
            user_gamer.bookmarks.add(computer_game)
            return HttpResponseRedirect (reverse("game-
profile", args=(game_id,)))
        else:
            user_gamer.developers.remove(game_developer)
            return HttpResponseRedirect (reverse("game-
profile", args=(game_id,)))

    def gamer_view(request, gamer_username):
        user_profile = get_object_or_404(Gameuser, username =
gamer_username)
        user_game_feedbacks =
Computergamefeedback.objects.filter(gamer = user_profile)
        user_game_feedbacks =
user_game_feedbacks.select_related("computergame")
        user_game_feedbacks = user_game_feedbacks.order_by("-
release")
        for feedback in user_game_feedbacks:
            recommendations =
Computergamefeedbackrecom.objects.filter(computergamefeedback =
feedback)
            recomb =
Computergamecharacterfeedbackrecom.objects.all()
            recomb = recomb.values_list("recom")
            recomb = recomb.distinct()
            recomb = {}
            for recom in recomb:
                rec_results = recommendations.filter(recom =
recom)
                rec_results = rec_results.count()
                recomb[recom] = rec_results
            feedback.recomb = recomb
        user_character_feedbacks =
Computergamecharacterfeedback.objects.filter(gamer =
user_profile)
        user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacter")
        user_character_feedbacks =
user_character_feedbacks.select_related("computergamecharacter__
computergame")
        user_character_feedbacks =
user_character_feedbacks.order_by("-release")
        for feedback in user_character_feedbacks:
            recommendations =
Computergamecharacterfeedbackrecom.objects.filter(computergamech
aracterfeedback = feedback)
            recomb = Computergamefeedbackrecom.objects.all()

```

```

        recoms = recoms.values_list("recom")
        recoms = recoms.distinct()
        recs = {}
        for recom in recoms:
            rec_results = recommendations.filter(recom =
recom)

            rec_results = rec_results.count()
            recs[recom] = rec_results
        feedback.recoms = recs
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            user_gamer = user_gamer.prefetch_related("reads")
            if user_profile in user_gamer.reads:
                user_profile.read = True
            else:
                user_profile.read = False
            for feedback in user_game_feedbacks:
                if feedback.computergame in
user_computer_games:
                    feedback.reader = True
                else:
                    feedback.reader = False
            for feedback in user_character_feedbacks:
                if feedback.computergamecharacter.computergame
in user_computer_games:
                    feedback.reader = True
                else:
                    feedback.reader = False
        return render(request, "gamerecsoft/gamer.html", {
"game_feedbacks": user_game_feedbacks, "character_feedbacks":
user_character_feedbacks, "user": user_profile })

    def character_feedback_recom_view(request,
characterfeed_id, recomtype):
        character_feedback =
get_object_or_404(Computergamecharacterfeedback, id =
characterfeed_id)
        character_feedback =
character_feedback.select_related("computergamecharacter")
        if request.user.is_authenticated:
            user_gamer = Gameuser.objects.filter(username =
request.user.username)
            feedbacks =
Computergamecharacterfeedbackrecom.objects.filter(computergamech
aracterfeedback = character_feedback, gamer = user_gamer)
            if not feedbacks or feedbacks.count() <= 0:
                recom =
Computergamecharacterfeedbackrecom(computergamecharacterfeedback
= character_feedback, gamer = user_gamer, recom = recomtype)

```

```

        return HttpResponseRedirect
(reverse("character-profile",
args=(character_feedback.computergamecharacter,)))

def game_feedback_recom_view(request, gamefeed_id,
recomtype):
    game_feedback = get_object_or_404(Computergamefeedback,
id = gamefeed_id)
    game_feedback =
game_feedback.select_related("computergame")
    if request.user.is_authenticated:
        user_gamer = Gameuser.objects.filter(username =
request.user.username)
        feedbacks =
Computergamefeedbackrecom.objects.filter(computergamefeedback =
game_feedback, gamer = user_gamer)
        if not feedbacks or feedbacks.count() <= 0:
            recom =
Computergamefeedbackrecom(computergamefeedback = game_feedback,
gamer = user_gamer, recom = recomtype)
            return HttpResponseRedirect (reverse("game-
profile", args=( game_feedback.computergame,)))

```

ДОДАТОК В
Слайди презентації

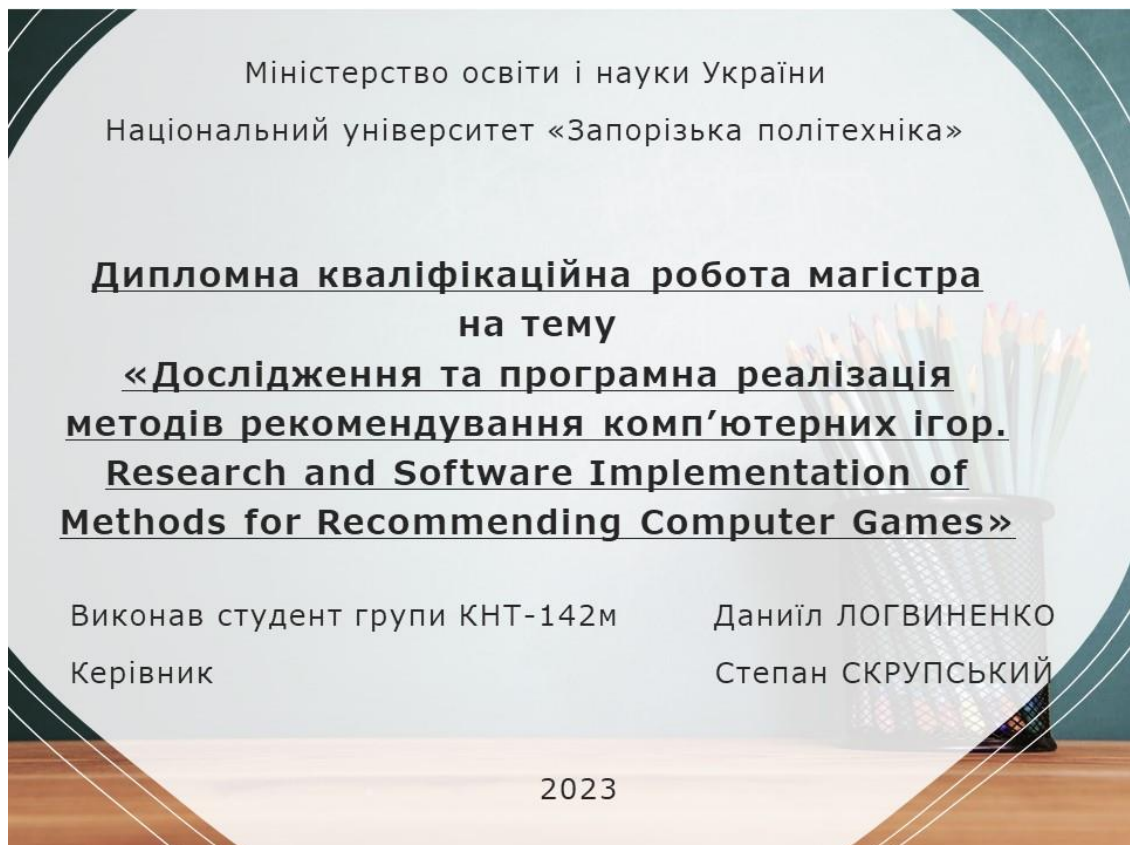


Рисунок В.1 – Титульний слайд



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

Завдання роботи

- аналіз існуючих програм, які виконують надання інформації про комп'ютерні ігри;
- дослідження методів рекомендування комп'ютерних ігор;
- проектування програми надання інформації про комп'ютерні ігри;
- реалізація програми надання інформації про комп'ютерні ігри.

3

Рисунок В.3 – Завдання роботи

Аналіз розроблюваної та існуючих програм

Показник	Steam	Metacritic	IMDB Video Games	Gamerecsoft
Пошук комп'ютерних ігор	Багато-критеріальний	За назвою, суміщений з іншими типами контенту	Багатокритеріальний, суміщений з іншими типами контенту	Багато-критеріальний
Придбання комп'ютерних ігор у програмі	+	–	–	–
Збереження комп'ютерних ігор	–	–	+	+
Виділення персонажів комп'ютерних ігор для пошуку і інформування	–	–	–	+
Визначення трендів	Зміна вартості	–	–	Зміна оцінки
Рекомендування комп'ютерних ігор	–	–	За подібністю на задану гру	За трендами

4

Рисунок В.4 – Аналіз розроблюваної та існуючих програм

Метод рекомендування комп'ютерних ігор

- визначення вибірки даних про комп'ютерні ігри з виконанням оброблення даних;
- перетворення тексту сценарію шляхом організації вкладання слів;
- навчання ансамблю дерев рішень, застосовуючи для цього сформовану вибірку за методом Random Forest;
- прогнозування оцінки для кожної заданої гри;
- визначення трендів зміни оцінки ігор, оцінок ігор та формування підмножини рекомендованих комп'ютерних ігор.

5

Рисунок В.5 – Метод рекомендування комп'ютерних ігор



Рисунок В.6 – Діаграма варіантів використання програми надання інформації
про комп'ютерні ігри

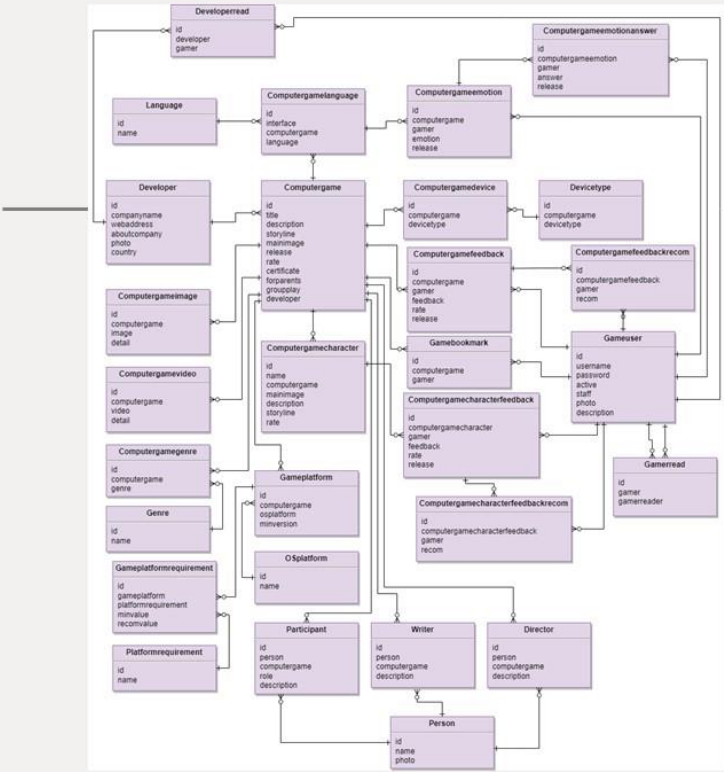
Порівняння засобів для розробки програми надання інформації про комп'ютерні ігри

Критерій	Django	Bottle
Рівень фреймворку	Високий рівень	Мікрофреймворк
Масштаб проєктів, для яких призначений	Середній, великий	Малий
Рівень підтримки	Вищий	Нижчий
Вбудовані засоби	Об'єктно-реляційне відображення, засоби захисту від атак, аутентифікація, авторизація, керування обліковими записами	Перелічені засоби відсутні, але переважно можуть бути під'єднані за вибором

7

Рисунок В.7 – Порівняння засобів для розробки програми надання інформації про комп'ютерні ігри

Схема бази даних



8

Рисунок В.8 – Схема бази даних

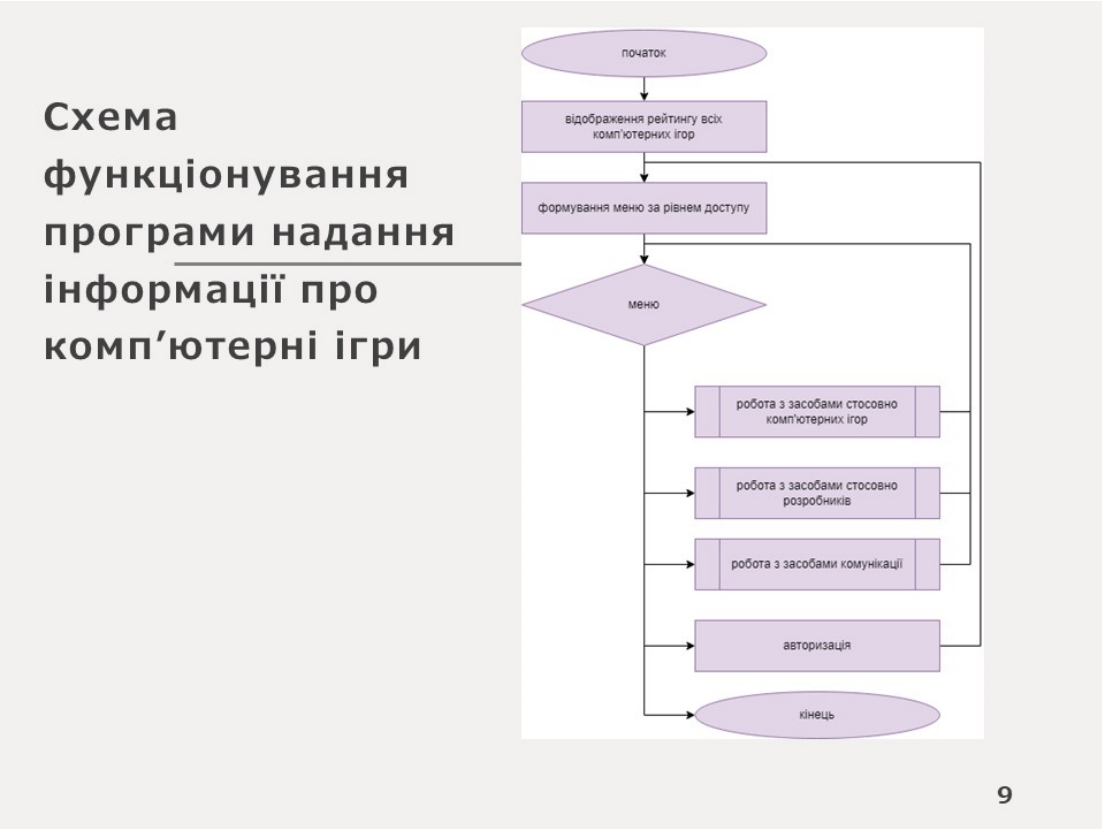


Рисунок В.9 – Схема функціонування програми надання інформації про комп'ютерні ігри

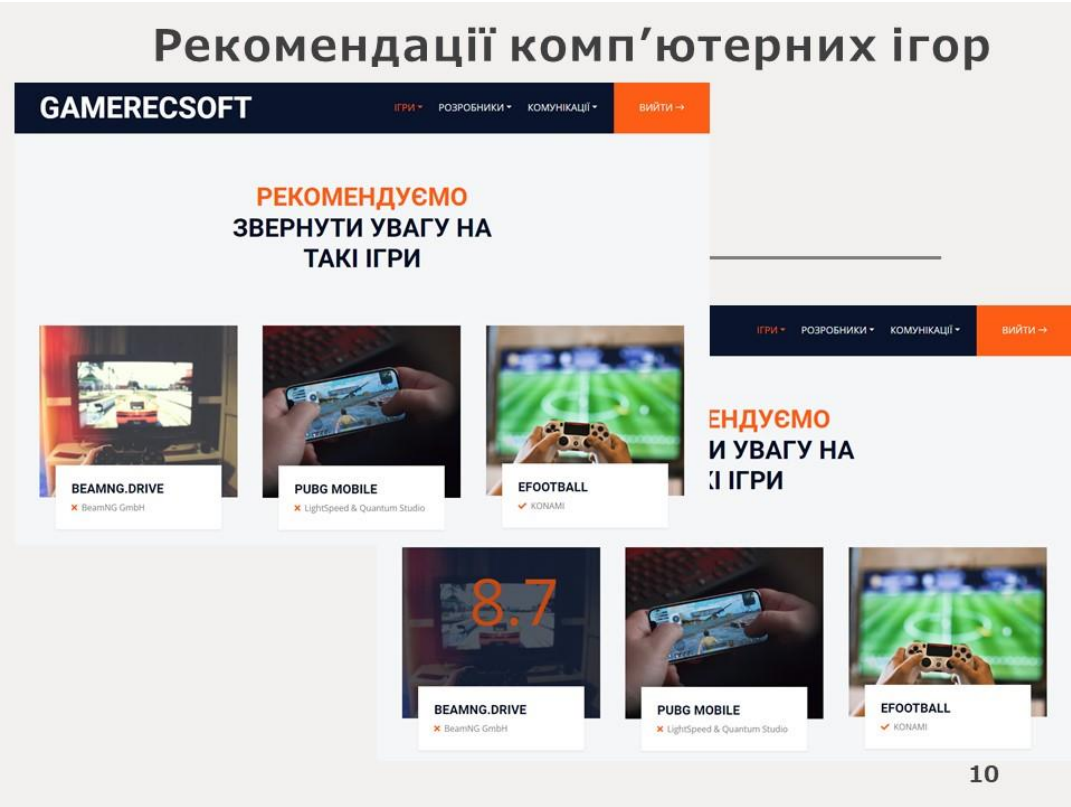


Рисунок В.10 – Рекомендації комп'ютерних ігор

Сторінка комп'ютерної гри

GAMERECSOFT
ІГРИ ▾
РОЗРОБНИКИ ▾
КОМУНІКАЦІЇ ▾
ВИЙТИ →

LEAGUE OF LEGENDS

★ 8.9



Багатокористувацька гра, в якій дві команди пропистоять одна одній, складючись рівно з 5 гравців кожна. Кожен гравець керує так званім чемпіоном у грі. Чемпіони мають різні характеристики, що відрізняють їх одного від іншого, при цьому надаючи різні характеристики щодо роботи в команді. За цією грою відбуваються кіберзмагання.

ОЦІНИТИ

ЖАНРИ

> МОВА

ПЛАТФОРМИ

> БАГАТОКОРИСТУВАЦЬКА
> WINDOWS
> MACOS

ПРИСТРОЇ

> ПЕРСОНАЛЬНИЙ КОМП'ЮТЕР

11

Рисунок В.11 – Сторінка комп'ютерної гри

Сторінка персонажа комп'ютерної гри

GAMERECSOFT
ІГРИ ▾
РОЗРОБНИКИ ▾
КОМУНІКАЦІЇ ▾
ВИЙТИ →

2 ВРАЖЕННЯ ТА 3 РЕКОМЕНДАЦІЇ


Андрій Нагорняк 12.12.2023 16:57
 Вся ця історія з нагадуванням про Супер Маріо робить Луджі ще цікавішим, так наче минуле якось все-таки вбудували в сучасність, і все при тому вийшло! Луджі простий, але з характером і це круто!
 ПОЗИТИВНО: 2 БЕЗЗМІСТОВНО: 0 НЕГАТИВНО: 0


Тарас Плохий 13.12.2023 19:36
 Спершу мені не дуже зайшов його образ, наче такий нехаризматичний, слабкий, але потім те, як він розкривається в грі, надихає!
 ПОЗИТИВНО: 1 БЕЗЗМІСТОВНО: 0 НЕГАТИВНО: 0

ПЕРСОНАЖ ГРИ

ЛУДЖІ

 ★ 9.0 ОЦІНИТИ

ВАШЕ ВРАЖЕННЯ

Оберіть оцінку
 Опишіть свої думки про персонажа
 ОЦІНИТИ

12

Рисунок В.12 – Сторінка персонажа комп'ютерної гри

Результати експериментального дослідження

Метод або модель	Ознаки	Середньоквадратична похибка
Дерево рішень	Рік видання, роки до оцінювання, сертифікат, жанри	0,15347
Дерево рішень	Рік видання, роки до оцінювання, сертифікат, жанри, сценарій	0,16834
Дерево рішень	Рік видання, роки до оцінювання, сертифікат, сценарій	0,17001
Дерево рішень	Рік видання, роки до оцінювання, сценарій	0,17261
Random Forest	Рік видання, роки до оцінювання, сертифікат, сценарій	0,11903
Random Forest	Рік видання, роки до оцінювання, сценарій	0,11919
Random Forest	Рік видання, роки до оцінювання, сертифікат, жанри	0,14017
Random Forest	Рік видання, роки до оцінювання, сертифікат, жанри, сценарій	0,11783

13

Рисунок В.13 – Результати експериментального дослідження

Висновки

У роботі розроблено програмне забезпечення надання інформації про комп'ютерні ігри з можливістю рекомендування ігор користувачам на основі виявлення майбутніх трендів.

Програма розроблена для любителів комп'ютерних ігор і надає їм можливість знаходити нові для них комп'ютерні ігри, ділитися з іншими гравцями думками щодо комп'ютерних ігор з власного досвіду, впорядкувати інформацію про персонажів комп'ютерних ігор. Користувачі можуть залишати різними способами зворотній зв'язок щодо комп'ютерних ігор, персонажів ігор, рекомендувати інформацію від інших користувачів, зберігати ігри, підписуватися на розробників ігор, інших користувачів, виконувати пошук відповідної інформації про комп'ютерні ігри, персонажів, розробників, осіб, задіяних у створенні ігор. До складу програми інтегровано рекомендації комп'ютерних ігор.

Наукова новизна роботи полягає в запропонованому методі рекомендування комп'ютерних ігор, який базується на створенні рекомендацій з комп'ютерних ігор, які мають найкращі очікувані оцінки та найбільше зростання, використовуючи для прогнозування оцінок ансамблі дерев рішень на основі методу Random Forest.

14

Рисунок В.14 – Висновки